DEEP LEARNING FOR OPERATIONAL STREAMFLOW FORECASTS:

A LONG SHORT-TERM MEMORY NETWORK

RAINFALL-RUNOFF MODULE FOR THE

NATIONAL WATER MODEL

by

JONATHAN M. FRAME

YONG ZHANG, COMMITTEE CHAIR
GREY S. NEARING
GEOFF TICK
BO ZHANG
HAMID MORADKHANI
CRAIG PELISSIER

A DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Geological Sciences
in the Graduate School of
The University of Alabama

TUSCALOOSA, ALABAMA

2022

ABSTRACT

This dissertation investigates deep learning (DL) and combining hydrologic process-based (PB) models with DL for a hybrid (HB) modeling approach (often referred to as “physics-informed machine learning” or “theory-guided learning”) for improving the predictive performance of streamflow in the U.S. National Water Model. An in-depth analysis is made of the benefits of DL and the potential drawbacks of the HB models. No evidence is found supporting the use HB models over the “pure” DL models in the use cases analyzed. The performance of the HB models is found to degrade in ungauged basins, whereas the DL models do not. The DL models are the best performing models for predicting extremely high runoff events, even when such events are not included in the training set. Adding physics inspired constraints to data-driven models causes a loss of system information relative to the DL models. As such, a “pure” DL model, specifically the Long Short-Term Memory (LSTM), is chosen as one of the core modules for the Next Generation (Nextgen) U.S. National Water Model. The LSTM (via Nextgen) is applied to simulate streamflow for a three-year period across the 191,020 km^2 New England region.

DEDICATION

To my beautiful wife, Tanya $ Frame, your love and support made this possible.

To Dorothy Frame, who taught me to follow my nose, which led me to Alabama.

LIST OF ABBREVIATIONS AND SYMBOLS

CAMELS	Catchment Attributes and Meteorology for Large-sample Studies
CDF	Cumulative Density Function
CONUS	CONtiguous United States
CUAHSI	Consortium of Universities for the Advancement of Hydrologic Science, Inc.
Daymet	Daily Surface Weather and Climatological Summaries
DL	Deep learning model
HB	Hybrid modeling; Some combination of PB and ML/DL models
HUC	Hydrology Unit Code
JAWR	Journal of the American Water Resources Association
KGE	Kling–Gupta efficiency
LSTM	Long Short-Term Memory.
LSTM_{A}	LSTM with atmospheric forcings as the only dynamic input
LSTM_{chrt}	LSTM post-processor with NWM channel router as the only dynamic input
LSTM_{ldas}	LSTM post-processor with NWM land surface model as the only dynamic input
LSTM_{PP}	LSTM post-processor with NWM outputs
LSTM_{PPnoQ}	LSTM post-processor with all NWM outputs EXCEPT streamflow
LSTM_{PPA}	LSTM post-processor with NWM outputs and atmospheric forcings
LSTM_{Qonly}	LSTM post-processor with NWM streamflow as the only dynamic input
MC-LSTM	Mass Conserving LSTM
ML	Machine learning model
NCAR	U.S. National Center for Atmospheric Research
Nextgen	Next generation U.S. National Water Model
NLDAS	North American Land Data Assimilation System

NOAA	U.S. National Oceanic and Atmospheric Administration
Noah-MP	Noah-Multiparameterization Land Surface Model
NSE	Nash-Sutcliffe Efficiency
NWM	U.S. National Water Model
NWM-Rv2	NWM version 2.0 retrospecive run
NWS	National Weather Service
PB	Process-based model
PRCP	Precipitation as a variable of atmospheric forcing
QLateral	NWM-predicted runoff into channel reach
ReLU	Rectified Linear Unit
RT	NWM Terrain Router
SAC-SMA	Sacramento Soil Moisture Accounting Model
SNOW-17	NWS' Snow Hydrology and Snowmelt Runoff
TRAD	Total radiation as a variable of atmospheric forcing
UA	University of Alabama
USGS	U.S. Geological Survey
WIS	USGS National Water Information System
WRF-Hydro	Weather Research and Forecasting hydrological modeling system

ACKNOWLEDGMENTS

Grey Nearing provided an immense amount of advisement on the research presented in this dissertation. He also made for a very thought provoking four years of my life. Thank you. All of the experiments presented here utilize NeuralHydrology, and the success of those experiments can be directly attributed to the remarkably high quality of the NeuralHydrology codebase. Thank you to the whole NeuralHydrology team, especially Frederick Kratzert. Craig Pelissier's collaboration on the first project I undertook as a PhD student was extremely educational. Even though my dissertation research took another path, the skills I gained from Craig's advisement helped me pull this together. Ed Clark, Trey Flowers, Fred Ogden and the entire National Water Center are doing the most exciting work in applied hydrology, and helped immensely with this dissertation. This dissertation includes contributions from over two dozen collaborators, listed at the beginning of each technical chapter. Financial support provided by: A.S. Johnson Travel Fund Scholarship – UA Department of Geological Science, NASA Terrestrial Hydrology Program (80NSSC18K0982), NOAA Cooperative Agreement (NA19NES4320002), CUAHSI National Water Center Innovators Program: Summer Institute, University of Alabama Department of Geological Sciences, and the U.S. National Water Center.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

This dissertation evaluates hydrologic models and their ability to predict streamflow in a highly complex and dynamic system (watershed). The experiments performed in this dissertation, making up the bulk of the scientific content (shown in Chapters 2, 3 and 4), were designed and performed with the aim of improving the predictive performance of the U.S. National Water Model (NWM) with deep learning (DL). The last technical chapter (5) of this dissertation presents an applied project, in which a DL model was designed, built, and tested for the Next Generation NWM Prototype Framework (Nextgen).

1.1 Background

1.1.1 The age of machine learning

Environmental data science is emerging rapidly and promises to solve many environmental-related problems at the global scale (Gibert et al., 2018). Many of these problems require continental and global scale hydrologic predictions. Simulations of the water cycle at these large scales, however, simply cannot account for the range of spatiotemporally heterogeneous physical processes needed to make accurate hydrological predictions (Tijerina et al., 2021). DL-based modeling is currently our best tool for making hydrologic predictions at these large spatial scales.

Deep learning (DL) has been used in hydrology for over 30 years (Hsu et al., 1995). There has been a consistent record in the hydrologic literature praising neural networks and calling for their expanded use for many hydrological applications (Abrahart, 1999; Govindaraju and Rao, 2000; Govindaraju, 2000; Piotrowski and Napiorkowski, 2011). A recent surge in DL research for hydrology has followed a series of papers showing undeniable superiority of the Long Short-Term Memory Network (LSTM), a DL architecture that was developed by Hochreiter (1991) in his dissertation and then formally published in a journal by Hochreiter and Schmidhuber (1997). DL (usually an LSTM architecture) is making better hydrological predictions in many sub-disciplines of hydrology, and for watershed hydrology, in particular, may actually be fundamentally changing this discipline Nearing et al. (2020).

LSTM was proposed as a hydrologic model by Krauße (2007):

> "[LSTM] is very promising because it provides an internal state which represents short and long term processes. Exactly these both classes of processes are necessary to represent both the state of the catchment and the dynamics of the rainfall-runoff process as a whole in fast reacting catchments"

and then again by Remesan and Mathew (2015) in the book "Hydrological data driven modelling". The first papers using LSTM for hydrology started appearing a few years later. Mhammedi et al. (2016) compared the LSTM to other models, but got poor results, which they attributed to over fitting. Gauch et al. (2021) showed that there is a minimum amount of training data needed, in order to get satisfactory results. As I will explain below, the data do not need to come from the basin which we are trying to make a prediction.

The major leap forward came from Kratzert et al. (2018), who took advantage of a large sample hydrologic dataset (Addor et al., 2017), which included static catchment attributes that were used train a single LSTM to make predictions in any basin, including ungauged basins (Kratzert et al., 2019). The recent LSTM-based hydrological research includes a thoughtfully developed, open-source, DL software library (NeauralHydrology: https://neuralhydrology.github.io/, accessed February 2022) that makes for relatively easy to use, reproducible, and adaptable environment for data-driven hydrologic studies

(Kratzert et al., 2022). The combination of NueralHydrology and the availability of a comprehensive large sample hydrological data set, has allowed for robust and rapid model comparison studies.

There has been a push to blend what is sometimes referred as ”process understanding” with machine learning (Reichstein et al., 2019). The reasoning behind this push is usually something along the lines of 1) model interpretability, and/or 2) to ensure that the model results do not violate physical principals. This dissertation challenges that reasoning. Two methods of theory-guided machine learning models, referred to as ”hybrid” (HB) models, are extensively tested for interpretability and physical realism, as compared to a ”pure” DL model.

It is often suggested to use DL models to diagnose problems with the process-based (PB) models and fix the PB models such that they make predictions as good as the DL models. I include this analysis in Chapter 2, but this goal is inherently limited and represents a disconnect between the theoretical foundations of DL and the desired use of a hydrological model. The very reason that DL is able to make better predictions than PB models is the lack of prior assumptions of the appropriate mathematical equations governing the main physical processes constraining the model architecture. If those constraints were representative of the function processing the inputs to predict the outputs, DL would be able to learn them (Hornik et al., 1989).

1.1.2 U.S. National Water Model

The U.S. National Water Model (NWM) is a tool used by the U.S. National Weather Service to forecast the distribution of water across the U.S. and its territories. The NWM is a major advancement for hydrological modeling, in the sense that continuous, real-time, forecasting at such a large scale was practically inconceivable a decade ago (Salas et al., 2018). The scale of the NWM presents a scientific challenge in modeling, as trade-offs need be made between the scale of deployment and predictive accuracy. It is simply impractical to represent all the hydrologic processes necessary to make accurate forecasts at four million miles of river reach with one single PB model. For instance, the hydrologic processes that dominate a catchment with intensive mono-crop farming practices in the American Mid-West are different than a barren desert of the American South-West. Although, location and climate are not always the best indices for clustering catchments by drivers of catchment behavior (Jehn et al., 2020).

The National Water Center is currently developing the ”Next Generation” NWM Prototype Framework (Nextgen), a modeling framework with the strategy of being model agnostic and scale independent (Ogden et al., 2021). This will allow specific models to be developed for specific catchments, which is a potential solution for applying the ”traditional” hydrological models. DL models, however, do not require different computational architectures because of their ability to distinguish catchment response based on hydrologic attributes of specific catchments (Kratzert et al., 2019). The majority of the research done for this dissertation was with the intention of including a DL model option for Nextgen.

The next section will outline several experiments that I performed in order to determine the best DL or HB model for operational forecasting with Nextgen. I compared the performance of ”pure” DL against different types of HB models, and considered many scenarios including prediction in ungauged basins, extremely large runoff events and the potential for long term mass biases. I then developed the Nextgen module for rainfall-runoff using the best performing data-driven model.

1.2 Research projects overview

1.2.1 Combining the U.S. National Water Model with Long Short-Term Memory networks for streamflow predictions and model diagnostics

I developed a DL-based hydrologic post-processor, which uses the outputs of the NWM as inputs to the LSTM model. This is an extremely simple architecture for HB modeling, and is effective for testing the informative content of PB models for machine learning. This project includes an in-depth analysis of the NWM, DL and HB model performance for a large sample of basins throughout the CONtiguous United States (CONUS), an analysis of performance in ungauged vs. gauged basins and a sensitivity analysis on all of the components of the NWM as inputs to the DL-based post-processor. The sensitivity analysis was designed to identify at which components of the NWM modeling chain information is lost (indicating a priority for improvement). This experiment is presented in Chapter 2, and published in the Journal of the American Water Resources Association (Frame et al., 2021b).

1.2.2 Extrapolation of DL models to extremely high runoff events not seen in training data

A persistent need in forecasting, in hydrology and beyond, is the need to develop, calibrate and/or train a model on some limited data set, then use that model to predict an event not captured in the data record. This is an exercise of extrapolation and/or interpolation. It is often assumed that PB models, and thus the related HB models, would out-perform a DL model during these kind of out-of-sample extreme events (Eagleson, 1991). But in my literature review, I found no experiments demonstrating, or even testing this assumption. The idea that hydrological models, based on physics-type equations, might be more reliable than DL when applied to out-of-sample conditions was drawn from early experiments on simpler (single layer) DL models. These immature and out-dated results, however, are still frequently cited (Beven, 2020a; Rasheed et al., 2022), which has likely hindered scientific progress.

I developed an experiment to use the return period of annual peak runoff events to split training and testing periods, which is presented in Chapter 3, and is accepted for Hydrology and Earth Systems Science (Frame et al., 2022b). The return period was chosen for this split because it provided a basis for categorizing target data in a hydrologically meaningful way that is both 1) consistent across basins and 2) maintains basin specific diversity. This is the first test of this hypothesis in the context of modern DL, and I believe the results are robust enough to begin a refute of that criticism.

1.2.3 Mass-conservation as a fundamental assumption of hydrologic modeling

Another criticism of DL based hydrology models is that they do not strictly enforce mass-conservation, which is a fundamental physical law of closed hydrologic systems. It is commonly assumed that rainfall-runoff process within a watershed follow this law. In essence, the assumption is that the difference between water input to and output from a watershed remains in the watershed. It has been argued that this type of system is best represented with a series of conservation equations. However, the only confident measurements we have of water mass at the watershed scale comes in the form of precipitation (in) and streamflow (out). Further, our measurements of precipitation and streamflow have some degree of uncertainty, and it has been suggested that these flawed measurements are the reason DL models make better predictions of observed streamflow than PB models (Beven, 2020b).

I designed an experiment to compare the long-term cumulative watershed runoff mass between observed and predicted streamflow. In this experiment, presented here in Chapter 4, I compared DL, PB and HB models. The HB model is specifically designed to obey a strict constraint of mass conservation. This allowed me to compare the role of mass-conservation itself, and the role of uncertain measurements, in predicting the long-term and event-based mass balance of the system. This work is under review for publication in the journal Hydrologic Processes (Frame et al., 2022a).

1.2.4 Application of deep learning for large-scale operational hydrologic forecasting with the Next Generation U.S. National Water Model Prototype Framework

Finally, I designed, built, and implemented an LSTM module for Nextgen (Frame et al., 2021a). This is presented in Chapter 5. This DL module is one of the first core features of Nextgen, which is scheduled to replace the NWM as the main hydrologic forecasting system throughout the U.S. by 2024. I trained the LSTM using NeuralHydrology on the CAMELS catchments, and performed a large scale, three-year, simulation of surface water runoff throughout New England.

REFERENCES

Abrahart, R. J. (1999). Neurohydrology: implementation options and a research agenda. *Area*, 31.2(1999):141–149.

Addor, N., Newman, A. J., Mizukami, N., and Clark, M. P. (2017). The CAMELS data set: catchment attributes and meteorology for large-sample studies. *Earth Syst. Sci*, 21:5293–5313.

Beven, K. (2020a). Deep learning, hydrological processes and the uniqueness of place. *Hydrological Processes*, 34(16):3608–3613.

Beven, K. (2020b). Deep learning, hydrological processes and the uniqueness of place. *Hydrological Processes*, 34(16):3608–3613.

Eagleson, P. S. (1991). Hydrologic science: A distinct geoscience. *Reviews of Geophysics*, 29(2):237–248.

Frame, J., Flowers, T., Ogden, F. L., Peckham, S. D., Bartel, R., Johnson, D. W., Frazier, N. J., Halgren, J. S., Mattern, D., Cui, S., Kratzert, F., and Nearing, G. S. (2021a). Deep learning for the next generation u.s. national water model. American Geophysical Union, Fall Meeting 2021.

Frame, J. M., Kratzert, F., Gupta, H. V., Ullrich, P., and Nearing, G. S. (2022a). On strictly enforced mass conservation constraints for modeling the rainfall-runoff process. *Hydrological Processes*, in review.

Frame, J. M., Kratzert, F., Klotz, D., Gauch, M., Shelev, G., Gilon, O., Qualls, L. M., Gupta, H. V., and Nearing, G. S. (2022b). 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., and Nearing, G. S. (2021b). Post-Processing the National Water Model with Long Short-Term Memory Networks for Streamflow Predictions and Model Diagnostics. *Journal of the American Water Resources Association*, pages 1–21.

Gauch, M., Mai, J., and Lin, J. (2021). The proper care and feeding of CAMELS: How limited training data affects streamflow prediction. *Environmental Modelling and Software*, 135(November 2020):104926.

Gibert, K., Horsburgh, J. S., Athanasiadis, I. N., and Holmes, G. (2018). Environmental Data Science. *Environmental Modelling and Software*, 106:4–12.

Govindaraju, R. and Rao, A. (2000). Task Committee on Application of Artificial Neural Networks in Hydrology. I: Preliminary Concepts. *Journal of Hydrologic Engineering*, 5(2):124–136.

Govindaraju, R. S. (2000). Artificial neural networks in hydrology. II: Hydrologic applications. *Journal of Hydrologic Engineering*, 5(2):124–137.

Hochreiter, S. (1991). Untersuchungen zu dynamischen neuronalen Netzen. *Unpublished doctoral dissertation, Institut für Informatik, Technische Universität, Munchen*, pages 1–71.

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

Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366.

Hsu, K.-l., Gupta, H. V., and Sorooshian, S. (1995). Artificial neural network modeling of the rainfall-runoff process. 31(10):2517–2530.

Jehn, F. U., Bestian, K., Breuer, L., Kraft, P., and Houska, T. (2020). Using hydrological and climatic catchment clusters to explore drivers of catchment behavior. *Hydrology and Earth System Sciences*, 24(3):1081–1100.

Kratzert, F., Gauch, M., Nearing, G., and Klotz, D. (2022). NeuralHydrology — A Python library for Deep Learning research in hydrology. *Journal of Open Source Software*, 7(71):4050.

Kratzert, F., Klotz, D., Herrnegger, M., and Hochreiter, S. (2018). A glimpse into the Unobserved : Runoff simulation for ungauged catchments with LSTMs. (Nips).

Kratzert, F., Klotz, D., Herrnegger, M., Sampson, A. K., Hochreiter, S., and Nearing, G. S. (2019). Towards Improved Predictions in Ungauged Basins: Exploiting the Power of Machine Learning. *Water Resources Research*, page 2019WR026065.

Krauße, T. (2007). *Fakultät Informatik*. PhD thesis, Technische Universität Dresden.

Mhammedi, Z., Hellicar, A., Rahman, A., Kasfi, K., and Smethurst, P. (2016). Recurrent neural networks for one day ahead prediction of stream flow. *ACM International Conference Proceeding Series*, pages 25–31.

Nearing, G. S., Kratzert, F., Sampson, A. K., Pelissier, C. S., Klotz, D., Frame, J. M., Prieto, C., and Gupta, H. V. (2020). What Role Does Hydrological Science Play in the Age of Machine Learning? *Water Resources Research*.

Ogden, F. L., Avant, B., Blodgett, D., Clark, E., Coon, E., Cosgrove, B., Cui, S., Kindl da Cunha, L., Farthing, M., Flowers, T., Frame, J. M., Frazier, N. J., Graziano, T., Gutenson, J., Johnson, D. W., Loney, D., Mattern, D., McDaniel, R., Moulton, J., Peckham, S. D., Jennings, K., Savant, G., Tubbs, C., Williamson, M., Garrett, J., Wood, A., and Johnson, J. M. (2021). The next generation water resources modeling framework: Open source, standards based, community accessible, model interoperability for large scale water prediction. american geophysical union, fall meeting 2021.

Piotrowski, A. P. and Napiorkowski, J. J. (2011). Optimizing neural networks for river flow forecasting - Evolutionary Computation methods versus the Levenberg-Marquardt approach. *Journal of Hydrology*, 407(1-4):12–27.

Rasheed, Z., Aravamudan, A., Sefidmazgi, A. G., Anagnostopoulos, G. C., and Nikolopoulos, E. I. (2022). Advancing flood warning procedures in ungauged basins with machine learning. *Journal of Hydrology*, page 127736.

Reichstein, M., Camps-valls, G., Stevens, B., Jung, M., Denzler, J., and Carvalhais, N. (2019). Deep learning and process understanding for data-driven Earth system science. *Nature*, 566:195 – 204.

Remesan, R. and Mathew, J. (2015). *Hydrological data driven modelling: A case study approach*. Springer.

Salas, F. R., Somos-Valenzuela, M. A., Dugger, A., Maidment, D. R., Gochis, D. J., David, C. H., Yu, W., Ding, D., Clark, E. P., and Noman, N. (2018). Towards Real-Time Continental Scale Streamflow Simulation in Continuous and Discrete Space. *Journal of the American Water Resources Association*, 54(1):7–27.

Tijerina, D., Condon, L., FitzGerald, K., Dugger, A., O'Neill, M. M., Sampson, K., Gochis, D., and Maxwell, R. (2021). Continental Hydrologic Intercomparison Project, Phase 1: A Large-Scale Hydrologic Model Comparison Over the Continental United States. *Water Resources Research*, 57(7):1–27.

CHAPTER 2

POST-PROCESSING THE NATIONAL WATER MODEL WITH LONG SHORT-TERM MEMORY NETWORKS FOR STREAMFLOW PREDICTIONS AND MODEL DIAGNOSTICS

Jonathan M. Frame, Frederik Kratzert, Austin Raney II, Mashrekur Rahmen, Fernando R. Salas and Grey S.Nearing

Published as Paper No. JAWR-20-0099-P of the Journal of the American Water Resources Association (JAWR), https://doi.org/10.1111/1752-1688.12964.

2.1 Abstract

We build three long short-term memory (LSTM) daily streamflow prediction models (deep learning networks) for 531 basins across the contiguous United States (CONUS), and compare their performance: (1) a LSTM post-processor trained on the United States National Water Model (NWM) outputs (LSTM_PP), (2) a LSTM post-processor trained on the NWM outputs and atmospheric forcings (LSTM_PPA), and (3) a LSTM model trained only on atmospheric forcing (LSTM_A). We trained the LSTMs for the period 2004–2014 and evaluated on 1994–2002, and compared several performance metrics to the NWM reanalysis. Overall performance ofthe three LSTMs is similar, with median NSE scores of 0.73 (LSTM_PP), 0.75 (LSTM_PPA), and 0.74(LSTM_A), and all three LSTMs outperform the NWM validation scores of 0.62. Additionally, LSTM_A outperforms LSTM_PP and LSTM_PPA in ungauged basins. While LSTM as a post-processor improves NWM predictions substantially, we achieved comparable performance with the LSTM trained without the NWM outputs(LSTM_A). Finally, we

performed a sensitivity analysis to diagnose the land surface component of the NWM as the source of mass bias error and the channel router as a source of simulation timing error. This indicates that the NWM channel routing scheme should be considered a priority for NWM improvement.

2.2 Introduction

The United States (U.S.) National Water Model (NWM), based on WRF-Hydro (Cosgrove et al., 2015), is an emerging large-scale hydrology simulator. Some specific details of the NWM advancements in large-scale hydrology are described by Elmer (2019), including increased resolution and number of stream reaches (2.7 million) for a model covering the contiguous United States (CONUS). A purported strength of WRF-Hydro is simulating hydrologic dynamics, and specifically the timing of hydrologic response (Salas et al., 2018). The predictive performance of the NWM (ability to match streamflow observations) has been shown to vary widely. (Hansen et al., 2019) evaluated the performance of the NWM in the Colorado River Basin in terms of drought and low flows; they found better performance in the Upper Colorado River Basin than in the Lower Colorado River Basin, and attributed this discrepancy to the NWM's ability to simulate snowpack. WRF-Hydro has generally poor performance in the Southwest and Northern Plains (Salas et al., 2018). Salas et al. (2018) hypothesized that error in WRF-hydro might come from lakes, reservoirs, floodplain dynamics, and soil parameter calibration.

NOAA personnel calibrated the NWM (version 2.0) at 1,457 gauged basins within the CONUS domain. As a point of comparison, the U.S. Geological Survey (USGS) records daily streamflow at 28,529 basins (https://nwis.waterdata.usgs.gov/nwis, accessed June 2020). Calibrating the model at each stream gauge within the NWM domain (which include all of CONUS and many U.S. territories) is a large computational expense, and while regionalization strategies can be used to improve real-time forecast accuracy without having to calibrate each individual basin, accuracy typically suffers compared to direct calibration. Due to these reasons and others, making accurate hydrologic predictions over large scales is a challenging problem, however, there are promising results in the machine learning (ML) and data science communities that may be directly applicable to improving the NWM.

ML is a powerful tool for hydrologic modeling, and there has been a call to merge ML with traditional hydrologic modeling (Reichstein et al. 2019; Nearing et al. 2020). One example of an ML approach that has been effective for hydrologic prediction is the "long short-term memory" network (LSTM) (Hochreiter, 1991; Hochreiter and Schmidhuber, 1997). The LSTM is a time-series deep learning method that is particularly well suited to model hydrologic processes because it mimics in certain ways the Markovian input-state-ouput structure of a dynamical system (Kratzert et al., 2018). LSTMs have been effective at simulating predictions of surface runoff at the daily time scale (Kratzert et al., 2019a), including in ungauged catchments where traditional methods of calibration do not work (Kratzert et al., 2019b), and also at sub-daily (hourly) time scales (Gauch et al., 2019). One potential problem with ML, however, is that it lacks a physical basis. While there are emerging efforts in hydrology to merge physical understanding with ML (Karpatne et al., 2017; Pelissier et al., 2019; Read et al., 2019; Chadalawada et al., 2020; Daw et al., 2020; Nearing et al., 2020; Tartakovsky et al., 2020; Hoedt et al., 2021), the field of theory-guided ML (Karpatne et al., 2017) is still relatively immature in hydrology.

The NWM informs forecasts of many hydrologic conditions, including river ice, snowpack, soil moisture, and inundation, which are used for management applications such as transportation, recreation, agriculture, and fisheries (NOAA, 2019). When ML is to be used in the NWM it should not disrupt the delivery of these hydrologic forecasts, therefore an ML prediction for streamflow that does not also include predictions of the other hydrologic states and variables must be run in parallel with the existing process-based hydrologic model. A natural question arises: does the existing NWM formulation benefit the already highly accurate LSTM predictions of streamflow?

Hydrologic post-processing can remove systematic errors in the model prediction, and has been shown to improve real-time forecast accuracy of both calibrated and uncalibrated basins, particularly in wet basins (Ye et al., 2014). The general methodology of post-processing involves taking the output of a process-based model and feeding it into a data-driven model. In this paper, we applied a LSTM-based post-processor for NWM basin-scale streamflow predictions. This is a straightforward theory-guided ML approach. We tested a LSTM-based post-processor that uses the dynamic NWM model outputs (shown in 2.1 and described below in the methods section) and compared the results against the NWM itself. We also tested a post-processor that included both the NWM outputs and atmospheric forcings as inputs and compared against an LSTM model trained only with atmospheric forcings (no NWM outputs).

Table 2.1: National Water Model (NWM) output data.

Feature name	Feature	NWM model component	Resolution
ACCET	Accumulated evapotranspiration	LDAS	1 km
FIRA	Total net long-wave (LW) radiation to atmosphere	LDAS	1 km
FSA	Total absorbed short-wave (SW) radiation	LDAS	1 km
FSNO	Snow cover fraction on the ground	LDAS	1 km
HFX	Total sensible heat to the atmosphere	LDAS	1 km
LH	Latent heat to the atmosphere	LDAS	1 km
SNEQV	Snow water equivalent	LDAS	1 km
SNOWH	Snow depth	LDAS	1 km
SOIL M (4 layers)	Volumetric soil moisture	LDAS	1 km
SOIL W (4 layers)	Liquid volumetric soil moisture	LDAS	1 km
TRAD	Surface radiative temperature	LDAS	1 km
UGDRNOFF	Accumulated underground runoff	LDAS	1 km
streamflow	River Flow	CHRT	Point
q_lateral	Runoff into channel reach	CHRT	Point
velocity	River Velocity	CHRT	Point
qSfcLatRunoff	Runoff from terrain routing	CHRT	Point
qBucket	Flux from groundwater bucket	CHRT	Point
qBtmVertRunoff	Runoff from bottom of soil to groundwater bucket	CHRT	Point
Sfcheadsubrt (mean and max)	Ponded water depth	RTOUT	250 km
Zwattablrt (mean and max)	Water table depth	RTOUT	250 km

We applied the LSTM post-processors to 531 basins across the CONUS. The basins chosen for this large-scale analysis are mostly headwater catchments without engineered control structures, such as dams, canals, and levees. This was a deliberate choice made for the purpose of simulating a close-to-natural rainfall–runoff response. Our goal was to use the post-processor to learn systematic corrections to simulated basin-scale rainfall–runoff processes that can improve forecasts of streamflow, rather than the hydraulic engineering implications resulting from simulated controlled flow, for example a reservoir release. Kim et al. (2020) showed the limitation of the NWM to predict streamflow in a highly engineered watershed and the need for representing controlled releases. Thus, we are using some of the simplest, and top performing, applications of the NWM for these experiments.

2.3 Methods

2.3.1 Data and models

2.3.1.1 CAMELS Catchments

This study used the Catchment Attributes and Meteorological dataset for Large Sample Studies (CAMELS) (Newman et al., 2015; Addor et al., 2017). The U.S. National Center for Atmospheric Research curated these data (NCAR; https://ral.ucar.edu/solutions/products/camels, accessed March 2020), and we used the 531 (out of 671) basins that Newman et al. (2015) chose for model benchmarking. Newman et al. (2015) excluded basins with large discrepancies in different methods for measuring basin area and also basins larger than $2,000 km^2$. CAMELS data include corresponding daily streamflow records from USGS gauges, and meteorological forcing data (precipitation, max/min temperature, vapor pressure, and total solar radiation) come from North American Land Data Assimilation System (NLDAS) (Xia et al., 2012).

2.3.1.2 National Water Model

We used the NWM version 2.0 reanalysis, which contains output from a 25-year (January 1993 through December 2019) retrospective simulation (https://docs.opendata.aws/nwm-archive/readme.html, accessed June 2020). The NWM retrospective ingests rainfall and other meteorological forcings from atmospheric reanalyses (https://water.noaa.gov/about/nwm, accessed June 2020). NWM reanalysis output includes channel outputs (point fluxes: CHRT) and land surface (gridded states and fluxes: LDAS and RT) outputs. The specific features that we used from the NWM reanalysis are shown in Table 2.1. To be compatible with the LSTM model, which uses a one-day timestep and was trained using all basins simultaneously, we took the mean values of these model outputs across UTC calendar days (midnight–2300) to produce daily records from the hourly NWM when used as input to the LSTM, but for NWM streamflow diagnostics we used the local calendar day (based on U.S. time zone) to be compatible with the USGS gauge records. We collected channel routing point data (CHRT) at each individual, NWM stream reach that corresponds to the stream gauge associated with each CAMELS catchment. We collected the gridded land surface data (LDAS) from each $1km^2$ Noah-MP cell (Niu et al., 2011) contained within the boundaries of each CAMELS catchment, and then calculated the averaged to produce a single representative (lumped) value for each catchment. We collected Gridded routing data (RT) from each $250m^2$ cell, and we included the mean and maximum value within the catchment boundary. We did not include lake input and output fluxes because these would be inconsistent across basins (some basins have zero and some basins have multiple lakes). Note that the units of the NWM outputs are not required for the LSTM post-processor.

2.3.1.3 LSTM network

The LSTM is a recurrent neural network that is able to maintain a memory of the system state and dynamics through a period of time (in this case 365 days). This recurrent state space is the main advantage for hydrologic applications over other types of neural networks. We developed our LSTM network from Kratzert et al. (2018, 2019b,a) using a codebase that is now referred to as NeuralHydrology (https://neuralhydrology.github.io/ accessed March 2021). NeuralHydrology was written in the Python programming language and is based primarily on the Pytorch ML library.

The LSTM in previous studies used two types of inputs: daily meteorological forcings and static catchment attributes. Again, note that the units of the forcing data are irrelevant when used as inputs for the LSTM, which does not include a mass or energy balance. We normalized all inputs to the LSTM, including static and dynamic inputs by subtracting the mean and dividing by the standard deviation of the training data. We used 18 catchment attributes from the CAMELS dataset related to climate, vegetation, topography, geology, and soils. These are described in more detail by Addor et al. (2017) and listed here in Table 2.2. Catchment attributes are static for each basin (do not change in time). LSTMs are trained to make predictions that are appropriate for individual basins according to their static attributes (Kratzert, Klotz, Shalev, et al. 2019), allowing us to train a single model that can be applied on any basin (we tested them on 531 CAMELS basins). The static attributes position a particular basin within an input space that is suitable for a particular hydrologic response (Nearing et al. 2021). For instance, the geologic permeability may influence the mass difference between total rainfall and runoff in a particular basin, as it would as a parameter in a process-based model. For the post-processing runs, we added the NWM model output predictions from version 2.0 of the NWM shown in Table 2.1.

We trained the LSTM models to make predictions at all 531 CAMELS catchments used in the analysis. We split the data temporally into a training period and testing period, and we present no results from the training period as these results are unrepresentative of the out-of-sample predictions. We trained the LSTMs on water years 2004 through 2014 and tested on water years 1994 through 2002. We included no spatial splits in the training procedure. The LSTMs used a 365-day LSTM look-back period, so a full year gap was left between training and testing to prevent bleedover (i.e., information exchange) between the two periods. We trained separate LSTMs with 10 unique random seeds for initializing weights and biases, and calculated benchmarking statistics using the ensemble mean hydrograph. The LSTMs make predictions representing runoff in units [mm], reflecting an area normalized volume of water that moves through a stream at each model time step. USGS gauge records (and the NWM predictions) are in streamflow units [L3/T]. We used the geospatial fabric estimate of the catchment area provided in the

Table 2.2: North American Land Data Assimilation System forcings and static catchment attributes.

Meteorological forcing data (used only in models denoted with an "A")	
Maximum air temp (TMax)	2-m daily maximum air temperature
Minimum air temp (TMin)	2-mr daily minimum air temperature
Precipitation (PRCP)	Average daily precipitation
Radiation (SRAD)	Surface-incident solar radiation
Vapor pressure (Vp)	Near-surface daily average
Static catchment attributes (used in each of the LSTM models)	
Precipitation mean	Mean daily precipitation
PET mean	Mean daily potential evapotranspiration
Aridity index	Ratio of mean PET to mean precipitation
Precipitation seasonality	Estimated by representing annual precipitation and temperature as sin waves positive (negative) values indicate precipitation peaks during the summer (winter). Values of approx. 0 indicate uniform precipitation throughout the year
Snow fraction	Fraction of precipitation falling on days with temp [C]
High precipitation frequency	Frequency of days with $\leq$ $5x$ mean daily precipitation.
Low precipitation frequency	Frequency of dry days ($< 1mm/day$)
Low precipitation duration	Average duration of dry periods (number of consecutive days with precipitation ($< 1mm/day$)
Elevation	Catchment mean elevation
Slope	Catchment mean slope
Area	Catchment area
Forest fraction	Fraction of catchment covered by forest
LAI max	Maximum monthly mean of leaf area index
LAI difference	Difference between the max. and min. mean of the leaf area index
GVF max	Maximum monthly mean of green vegetation fraction
GVF difference	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 fraction	Fraction of sand in the soil
Silt fraction	Fraction of silt in the soil
Clay fraction	Fraction of clay in the soil
Carbonate rocks fraction	Fraction of the catchment area characterized as "carbonate sedimentary rocks"
Geological permeability	Surface permeability (log10)

CAMELS dataset to convert all streamflow to units [L] for our diagnostic comparison. We trained the LSTMs with the protocol and features described in Appendix B of Kratzert et al. (2019c): this includes 30 epochs, a hyperbolic tangent activation function, a hidden layer size of 256 cell states, a look-back of 365 days, variable learning rates set at epoch 0 to 0.001, epoch 11 to 0.005 and epoch 21 to 0.0001, dropout rate of 0.4 and an input sequence length: 270.

Overfitting of deep learning models can lead to poor performance when the models make predictions on data that is not part of the training set. The methods described above to ensure that information in the testing set (water years 1994 through 2002) is not part of the training set helps build confidence in our modeling results. In addition, the dropout rate is an important hyper-parameter for preventing overfitting. The dropout probabilistically removed some connections from the LSTM network during training, in our case with a probability of 0.4. This avoids the network relying too heavily on specific connections. Model runs during testing did not include dropout.

2.3.1.4 Experimental Design

We tested the results from LSTM post-processing against the NWM and also against a LSTM trained with atmospheric forcings as dynamic inputs to the model, with no inputs from the NWM model outputs (referred to as LSTM_A, in which the A stands for atmospheric forcing). Table 2.3 will guide the reader through the setup of each model.

Table 2.3: List of models for post processing analysis.

Model label	Number of dynamic LSTM inputs	Model description
NWM	N/A	NWM mean daily streamflow predictions
LSTM_PP	28	LSTM trained with NWM output for post-processing
LSTM_PPA	33	LSTM trained with NWM output and atmospheric forcings for post-processing
LSTM_A	5	LSTM trained with atmospheric forcing conditions

Simple schematics of the LSTMs used in this study are shown in Figure 2.1. The LSTM post-processors (LSTM_PP and LSTM_PPA) used NWM outputs as LSTM inputs, and the processbased NWM predictions influenced the LSTMbased streamflow predictions.

This is a straightforward method of theoryguided (or physics-informed) ML, but is commonly referred to as post-processing (Han 2021).

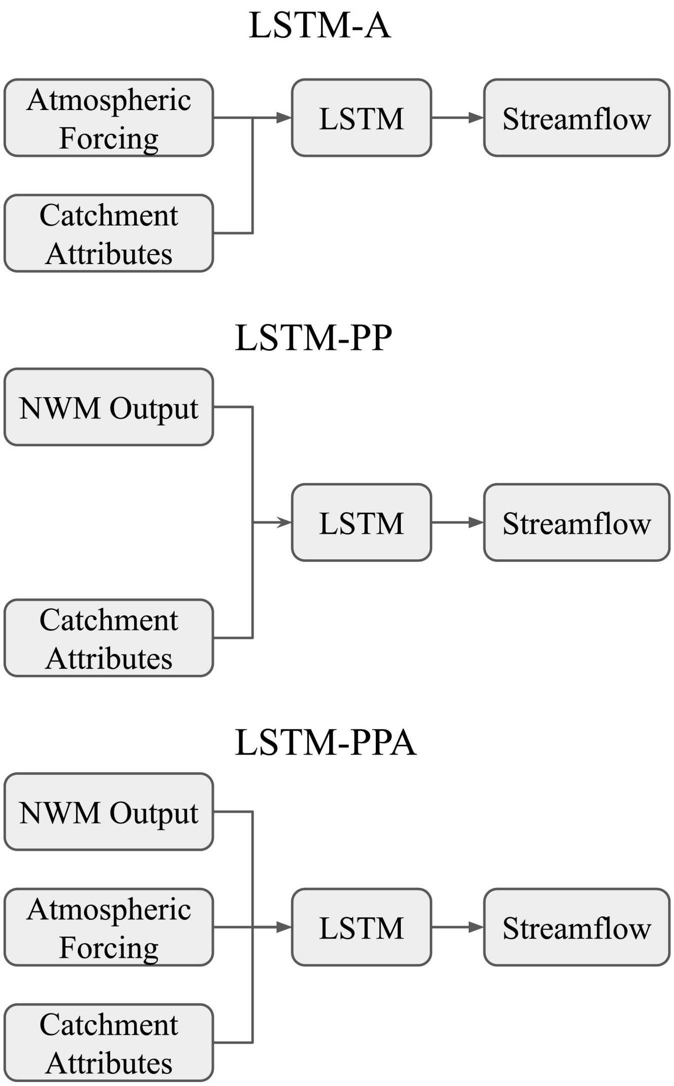

Figure 2.1: Flow chart showing the LSTM_A and the LSTM post-processors with NWM data as inputs (LSTM_PP and LSTM_PPA). LSTM_PP is the post-processor which used only NWM outputs as input to an LSTM, and LSTM_PPA used both the NWM outputs and atmospheric forcings.

As a quality check, we compared the results from each LSTM ensemble member, and found a relative standard error of the mean streamflow about 1%, and relative standard error of the Nash–Sutcliffe efficiency (NSE) value of about 0.01%. This means that all LSTM solutions are similar between random initialization seeds. Gauch et al.

(2021) attributed a 0.01 discrepancy in NSE values of the LSTM predictions to nondeterminism of the loss function minimization. In our experiments discrepancies in the loss function occur between different random seed initializations, but running the training procedure twice with the same random seed gives an identical solution, satisfying the definition of determinism.

Model comparisons. We tested/evaluated all models (NWM and all LSTMs) on the same daily data and the same time period (years 1994–2002). We trained the LSTMs on data from years 2004–2014 and evaluated them on years 1994–2002. The NWM was calibrated by NOAA on the time period 2007–2013 (https://ral.ucar.edu/sites/default/files/public/9_RafieeiNasab_CalibOverview_CUAHSI_Fall019_0.pdf, accessed August 2021), though no journal publications thoroughly describe the details of this calibration. For this study, we tested the performance of the NWM reanalysis only on the time period 1994–2002 (the same time period as the LSTM).

2.3.1.5 Performance metrics

We calculated several metrics to evaluate predictive performance, including the NSE and Kling–Gupta efficiency (KGE) values (Gupta et al., 2009). We calculated the variance, bias, and Pearson correlation metrics separately as components of the NSE (Gupta et al., 2009); these tell us about relative variability, mass conservation, and linear correlation between the modeled/observed streamflow values, respectively. Observed streamflow values are from the USGS streamflow gauges associated with each of the CAMELS basins. We calculated the metrics in two ways: (1) at each basin and then averaged together, and (2) using all of the flows from all basins combined.

Our graphical results focus on three performance metrics: (1) NSE measures the overall predictive performance as a correlation coefficient for the $1:1$ linear fit between simulations and observations, (2) Peak timing error measures the absolute value of differences (in units days) between simulated and observed peak flows for a given event, and (3) total (absolute) bias measures the overall bias of the simulated hydrograph relative to observations and represents how well the model matches the total volume of partitioned rainfall that passes through the stream gauge at each basin.

We also calculated performance metrics on different flow regimes. Rising limbs and falling limbs were characterized by a one-day derivative, where positive derivatives were categorized as rising limb, and negative derivatives as falling limb. High flows were characterized as all flow above the $80th$ percentile in a given basin, and low flows as below the $20th$ percentile in a given basin.

We tested the performance of the LSTM post-processors in different regions. We split the basins by USGS designated "water resource regions" (https://water.usgs.gov/GIS/regions.html, accessed July 2020). To analyze the regions individually we averaged the NSE, bias, and timing error of the CAMELS basins within each region.

We set an alpha value for statistical significance to $\alpha = 0.05$. To control for multiple comparisons we adjusted the alpha values using family-wise error rate equal to $1 - (1 - \alpha)m$, with m being the number of significance tests (86 in total), which brought our effective alpha value down to 0.049. We tested for statistical significance with a Wilcoxon signed-rank test against the null hypothesis that our test models (LSTM post-processors) performance across basins came from the same distribution as our base models (NWM and LSTM_A).

2.3.1.6 Simulated hydrograph representation of hydrologic signatures

Hydrologic signatures help us understand how well a model represents important aspects of real-world streamflow, and where improvement should be made to the model's conceptualization (Gupta et al., 2008). We analyzed the hydrologic signatures described by Addor et al. (2018), and these are listed below in Table 2.4. We calculated the true signatures with USGS streamflow observations, and calculated model representations with predicted values of daily streamflow. We compared true values and predicted values with a correlation coefficient (r2) across basins (one value of the observed and predicted hydrologic signatures were calculated per basin), higher values indicate a better representation of hydrologic signature across basins by the model. We used the Steiger method to test for statistically significant changes between the LSTM_A, NWM, and the LSTM post-processor (Steiger and Browne, 1984).

Table 2.4: Hydrologic signatures adapted from Addor et al. (2018).

Signature description	Signature name
Average duration of low-flow events	low_q_dur
Frequency of days with zero flow	zero_q_freq
Average duration of high-flow events	high_q_dur
Streamflow precipitation elasticity	stream_elas
Frequency of high-flow days	high_q_freq
Slope of the flow duration curve	slope_fdc
Frequency of low-flow days	low_q_freq
Baseflow index	baseflow_index
Runoff ratio	runoff_ratio
Mean half-flow date	hfd_mean
5% flow quantile	$q5$
95% flow quantile	$q95$
Mean daily discharge	q_mean

2.3.1.7 Identifying basins best suited for post-processing with multi-linear regression

The LSTM post-processors did not improve performance at every basin. It therefore would be valuable to know if a LSTM post-processor will work in any particular basin before implementation. We trained a multi-linear regression, using the Scikit-learn library in Python, to predict the performance changes between the NWM and the LSTM post-processors (LSTM_PP and LSTM_PPA) at each individual basin. The multi-linear regression analysis included performance scores of the NWM streamflow predictions, hydrologic signatures, and catchment characteristics as inputs. These regressors are useful to help interpret what basins might benefit most from an LSTM post-processor. We trained and tested multi-linear regression models using k-fold cross-validation with 20 splits ($k = 20$) over the 531 basins. We report the correlation (r^2) of out-of-sample regression predictions of post-processing changes vs. actual post-processing changes.

2.3.1.8 Interpretation of LSTM with integrated gradients

We aim to explain the relationship between a model's predictions in terms of its features. This will help us understand feature importance, identifying data issues, and inform NWM process diagnostics from the post-processors. We calculated integrated gradients (Sundararajan et al., 2017) to attribute the LSTM inputs (both atmospheric forcings and NWM outputs) to the total prediction of streamflow. Integrate gradients are a type of sensitivity analysis that are relatively insensitive to low gradients (e.g., at the extremes of neural network activation functions). We calculated integrated gradients separately for each input, at each timestep, for each lookback timestep, in each basin. This means that for nine years of test data with a 365-day lookback there were about 1.2 million integrated gradients per input, per basin. The unit of the integrated gradient is technically normalized streamflow, but we were mostly interested in the relative values of integrated gradients of each individual LSTM input.

2.3.1.9 Interpretation of LSTM with correlations between performance and NWM inputs

Of the NWM calibrated basins, 480 overlap with the 531 CAMELS catchments used in this study. In a separate set of experiments, we trained the LSTM_A and the LSTM post-processors LSMT_PP and LSTM_PPA) on only the 480 calibrated basins. We then used the full set of 531 catchments to test the performance out-of-sample. We analyzed the 480 in-sample basins and 51 out-of-sample basins separately using the NSE, bias, and timing error metrics. This allowed us to determine if the LSTM is a suitable post-processing method to use in uncalibrated basins. If the post-processors trained only on calibrated basins can improve streamflow predictions at uncalibrated basins, then they would be considered suitable, particularly if there is no statistical difference between the post-processor's performance improvement over the NWM and/or LSTM_A.

2.3.1.10 Sensitivity analysis and NWM process diagnostics

We trained a set of LSTM post-processors using different combinations of NWM outputs as input to the LSTM, as described in Table 2.5. To test the sensitivity to the NWM streamflow prediction itself, we trained an LSTM with only streamflow (LSTM_Q_only), and excluded it from another (LSTM_PP_noQ). We tested the sensitivity to the channel routing (LSTM_chrt) and land surface (LSTM_ldas) components of the NWM by training LSTMs with only these dynamic inputs. We trained these models with the same specifications as theLSTM_A, LSTM_PPA, and LSTM_PP.

Table 2.5: Additional models for sensitivity analysis and NWM diagnostics.

Model label	Number of dynamic LSTM parameters	Model description
LSTM_PP_noQ	26	LSTM post-processor (LSTM_PP) but without streamflow or velocity
LSTM_Q_only	1	LSTM trained with NWM streamflow only
LSTM_chrt	6	LSTM trained with NWM channel routing outputs only
LSTM_ldas	18	LSTM trained with NWM land surface outputs only

Each of these models (Table 2.5), in addition to the main post-processing models presented in Table 2.3, have a distinct flow of information that we can use to diagnose NWM model processes. Figure 2.2 shows the information flow of each of the model subcomponents. We used the performance results of the different post-processing models to assess how much information passes between the model components. Nearing et al. (2018) described the method to quantify the information exchange down a modeling chain (i.e., integrating over the expected effect of the conditional probability), but since we used limited outputs from the NWM reanalysis, rather than the full state space, we examined the NWM only qualitatively for information loss between the major NWM subcomponents (land surface runoff, overland router, and channel router). The LSTM extracts information from its input to make predictions about its target, in our case streamflow, and we assumed higher streamflow prediction accuracy indicated more information is available in the NWM components used as input. If a post-processor made less accurate streamflow predictions than the LSTM_A, then this indicates that the NWM modeling chain lost information from the atmospheric forcings.

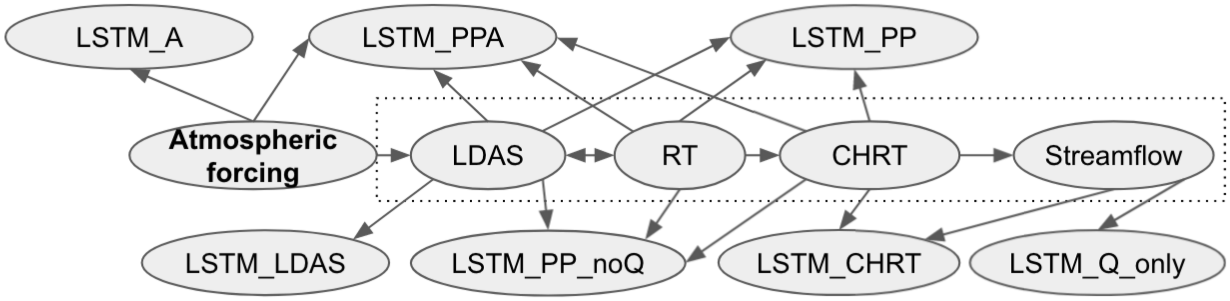

Figure 2.2: Process network diagram showing the information flow of each of these models. Arrows indicate the information flow from one component of the model to another. The NWM components are outlined with the dashed box. This is also a good guide for understanding the inputs to the different post-processing models.

2.4 Results

2.4.1 Overall model performance

Post-processing the NWM with LSTMs significantly improved predictive performance, both with or without including the atmospheric forcings as inputs into the model. The LSTM_A, however, is the overall better performing model. Figure 2.3 shows the cumulative distributions of three performance metrics (NSE, peak timing error, and total bias).

The LSTM_PP improved the NSE score of the NWM mean daily streamflow at a total of 465 (88%) and reduced accuracy in 66 basins (12%) of the total 531 CAMELS basins, improved the total bias of the NWM mean daily streamflow at a total of 325 (61%) of basins and improved the peak timing error at a total of 488 (92%) of basins. The LSTM_PPA post-processor improved the NSE score of the NWM mean daily streamflow at a total of 488 (92%) and reduced accuracy in 43 basins (8%) of the total 531 CAMELS basins. The LSTM_PPA post-processor improved the total bias of the NWM mean daily

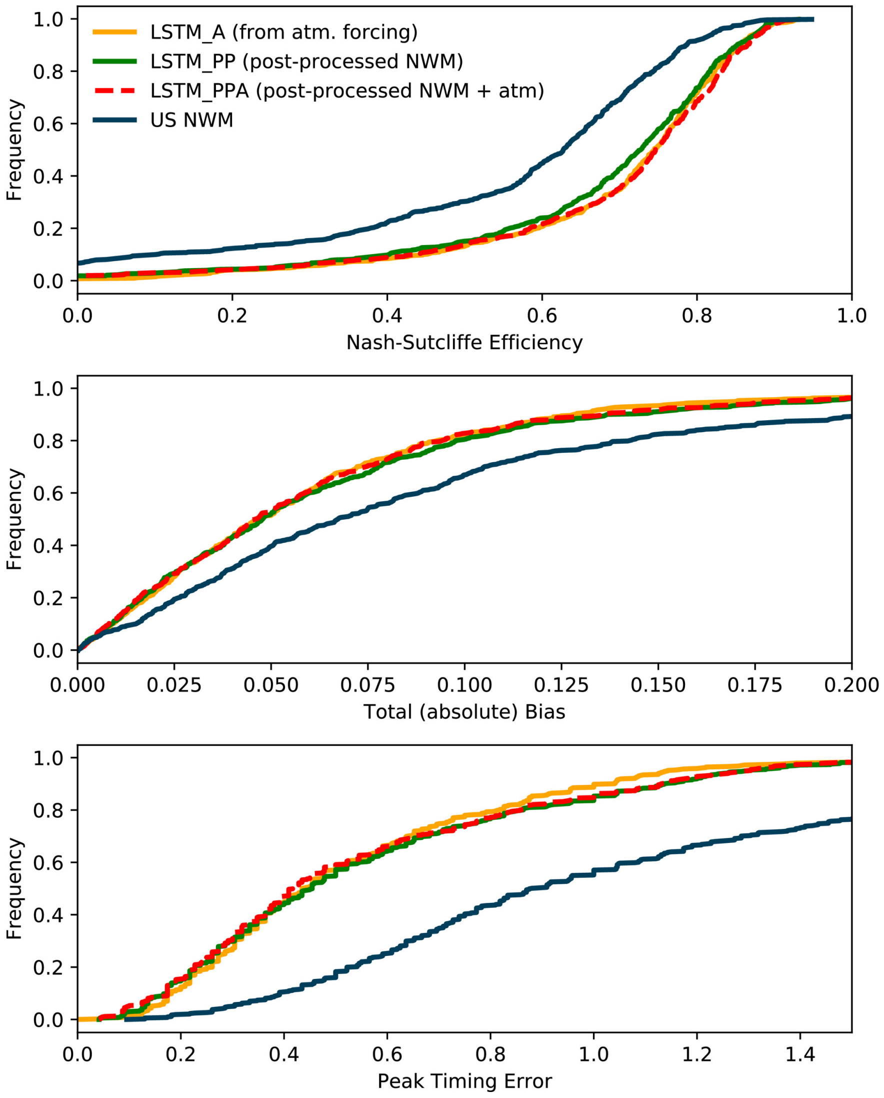

Figure 2.3: Results showing the cumulative distributions of model performance calculated as Nash-Sutcliffe Efficiency (NSE), total bias, and peak timing error over a 10-year test period in 531 CAMELS catchments. The National Water Model (NWM) reanalysis streamflow was averaged daily, long short-term memory (LSTM) networks shown used (i) the original atmospheric inputs (LSTM_A), (ii) NWM states and fluxes only (LSTM_PP), and (iii) both atmospheric forcings and NWM states and fluxes (LSTM_PPA). These figures omit the distribution tails for clarity.

streamflow at a total of 331 (62%) of basins and improved the peak timing error at a total of 494 (93%) of basins. The LSTM_A (without NWM model output) outperformed the NWM at a total of 473 (89%) and reduced accuracy in 58 basins (11%), improved the total bias of the NWM mean daily streamflow at a total of 339 basins (64%) and improved the peak timing error at a total of 484 basins (91%). The LSTM_PPA improved the greatest number of basins in terms of NSE and peak timing error and the LSTM_A was the best performing model in terms of total bias. Figure 2.4 shows scatter plots of the post-processor performance at individual basins against the performance of the NWM and LSTM_A.

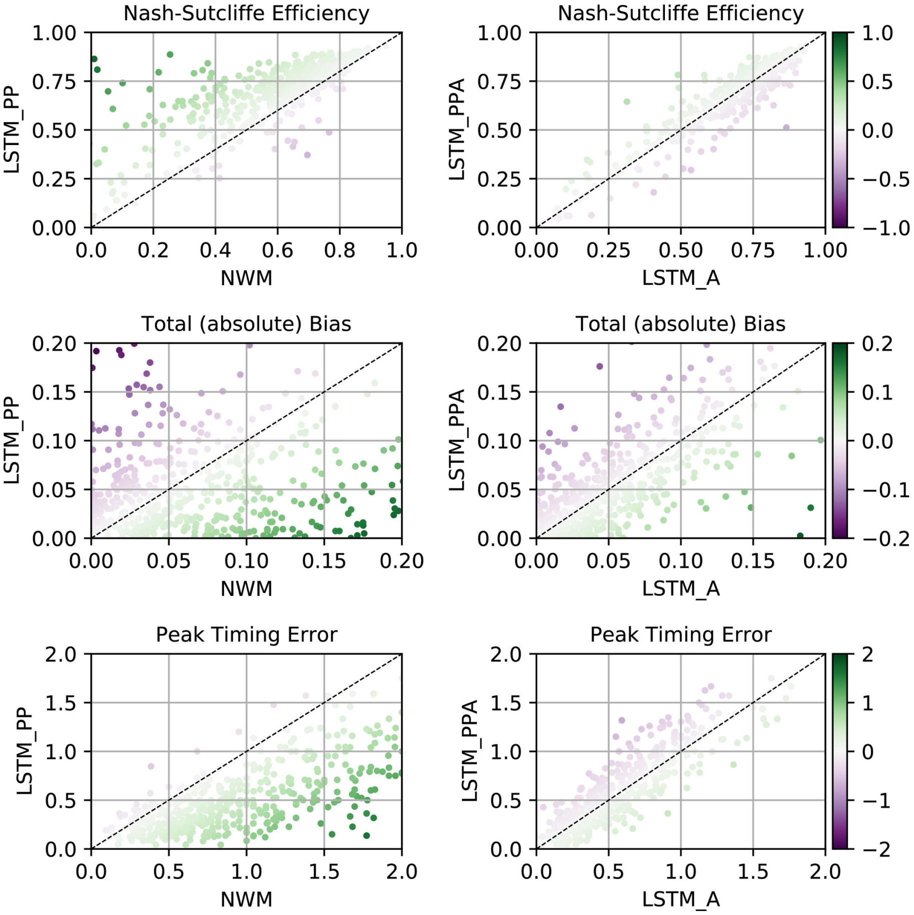

Figure 2.4: Performance differences of the post-processors against the NWM and LSTM_A) in 531 CAMELS basins across CONUS. Green indicates basins where post-processing improved performance over the NWM and LSTM_A (darker indicates larger relative improvement), and purple indicates basins where there was a decrease in performance (darker indicating worse relative detriment). The first column shows the performance difference between the LSTM_PP and the NWM. The second column shows the performance difference between the LSTM_PPA and the LSTM_A.

The post-processing models (LSTM_PP and LSTM_PPA) improved relative to the NWM in similar basins. The improvements of the two post-processing methods are correlated across all basins ($r2 = 0.995$). Performance comparisons between the LSTM models and the NWM for each basin are plotted spatially in Figure 2.5. Notice that some of the highest NSE improvements between the LSTM_PP and the NWM are the worst NSE detriments between the LSTM_PPA and the LSTM_A, particularly in the northern plains. This indicates that although the post-processor greatly improves the NWM, the information from the NWM at bad basins hinders the performance of the LSTM, or in other words, the NWM passes bad information to the LSTM.

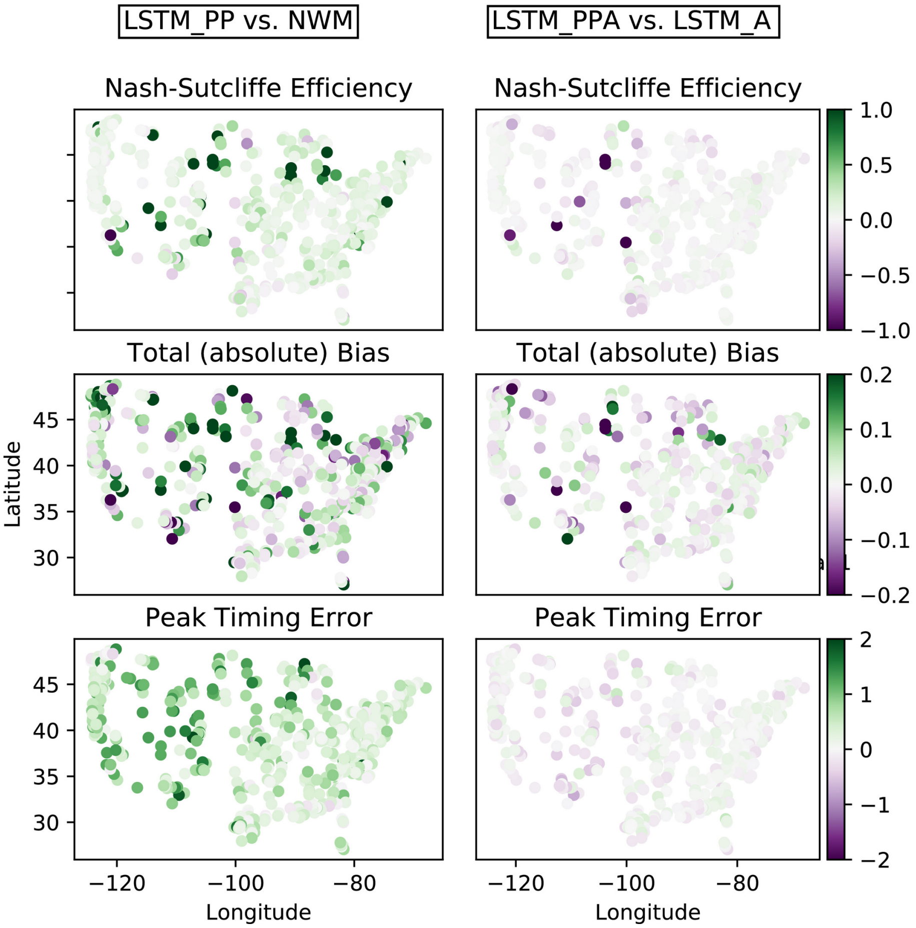

Figure 2.5: Per-basin performance change between the post-processors and NWM and LSTM_A) in 531 CAMELS basins across CONUS. Green indicates basins where post-processing improved performance over the NWM and LSTM_A (darker indicates larger relative improvement), and purple indicates basins where there was a decrease in performance (darker indicating worse relative detriment). The first column (a-c) shows the performance change between the LSTM_PP and the NWM. The second column (d-f) shows the performance change between the LSTM_PPA and the LSTM_A.

2.4.2 Performance by flow regime

The LSTM post-processors improved the predictive performance of the NWM according to the NSE and KGE metrics, as well as their components (variance and correlation). A full set of performance metrics broken down by flow regime are shown in Table 2.4.2. The left side of the table shows the average of metrics calculated individually at each basin, and the right side of the table shows the metrics as calculated by combining the flows from all basins. The NSE includes both mean and median averages, but the rest of the metrics are only averaged by the median.

Table 2.6: Predictive performance for NWM, LSTM_A, and the LSTM post-processors during various flow regimes. The NSE and Kling–Gupta efficiency (KGE) are overall performance metrics of prediction quality. Variance, bias, and correlation (R) are the components of the NSE. We calculated these in two ways: (1) at each basin and averaged across all basins, and (2) once using the observed and predicted streamflow values from all basins combined. Note that calculations done once across all basins do not include a test of significance.

Flow categories	Calculated per-basin						All basins			
	NSE (mean)	NSE (median)	KGE	Variance	Bias	R	NSE	Variance	Bias	R
				All flows						
NWM	0.46	0.62	0.64	0.82	-0.01	0.82	0.75	0.85	0.02	0.87
LSTM_PP	0.65**	0.73**	0.74**	0.86	0.02	0.87**	0.81	0.92	0.02	0.9
LSTM_A	0.69	0.74	0.74	0.83	0.02	0.88	0.82	0.89	0.01	0.9
LSTM_PPA	0.67	0.75	0.76	0.87	0.02	0.88	0.82	0.93	0.02	0.91
				Rising limbs						
NWM	0.47	0.60	0.60	0.77	-0.07	0.81	0.73	0.82	-0.05	0.85
LSTM_PP	0.64**	0.70**	0.72**	0.83**	0.00**	0.86**	0.78	0.88	0	0.88
LSTM_A	0.66	0.71	0.72	0.80	-0.01	0.86	0.78	0.85	-0.01	0.88
LSTM_PPA	0.65	0.72	0.74	0.85	0.00	0.87	0.79	0.89	0.00	0.89
				Falling limbs						
NWM	0.29	0.62	0.64	0.94	0.03	0.83	0.78	0.90	0.00	0.88
LSTM_PP	0.62**	0.75**	0.76**	0.95**	0.07	0.90**	0.87	0.99	0.04	0.93
LSTM_A	0.69	0.78	0.77	0.92	0.05	0.90	0.87	0.96	0.03	0.93
LSTM_PPA	0.65	0.77	0.77	0.94	0.05	0.90	0.87	0.98	0.03	0.93
				Above 80th percentile						
NWM	0.17	0.41	0.54	0.80	-0.13	0.73	0.69	0.83	-0.10	0.84
LSTM_PP	0.47**	0.57**	0.64**	0.82	-0.08**	0.80**	0.76	0.89	-0.04	0.90
LSTM_A	0.53	0.58	0.67	0.81	-0.08	0.81	0.78	0.86	-0.06	0.88
LSTM_PPA	0.50	0.60	0.69	0.84	-0.07	0.81	0.79	0.90	-0.04	0.89
				Below 20th percentile						
NWM	-18,384.37	-17.47	-1.96	3.79	1.89*	0.36	0.37	1.31	0.22	0.81
LSTM_PP	-6,941.62**	-15.66**	-1.28**	2.84**	3.21	0.432	0.53	1.3	0.33	0.90
LSTM_A	-4,749.68	-16.35	-1.31	2.85	3.27	0.43	0.56	1.26	0.33	0.89
LSTM_PPA	-5,147.62	-14.66	-1.24	2.85	2.87	0.43	0.58	1.28	0.30	0.90

Note: * Post-processing significantly hurts the NWM. ** Post-processing significantly helps the NWM.

In general Table 2.4.2 shows that the LSTM post-processors improved over the NWM in nearly all flow regimes according to most metrics. The LSTM_PPA also improved upon the LSTM_A in more than half the basins, and by most metrics, though not significantly. The prediction of rising limb and high flow regimes was improved upon by the LSTM post-processors according to every metric.

Bias was the only metric that was reduced due to post-processing, and the difference was highest in low flow regimes. All models poorly predicted flows below the 20th percentile. This is likely due to the fact that all models tend to have difficulty predicting zero streamflow, and the 101 basins with periods of zero streamflow affected the average performance metrics. This will be discussed further in terms of hydrologic signatures.

The right side of the table has better performance values than the average of metrics calculated individually at each basin. This is a result of some of the better performing basins compensating for poorer performing basins, or from a different perspective, some basins have a relatively poor performance which weighs down the average.

2.4.3 Performance by region

Results from a regional analysis of performance are shown below in Figure 2.6. The LSTM post-processors significantly improved the NSE over the NWM in 15 of the 18 regions, the peak timing error in 16 regions (all regions with enough basins for a statistical evaluation) and significantly improved bias in only one region. Note that region 9 was represented by only two CAMELS basins, which is not sufficient for statistical evaluation. The bias was better represented by the NWM than the post-processor in five of the 18 regions, including the entire East Coast (regions 01, 02, and 03), the Pacific Northwest (17), and the Lower-Colorado River (15).

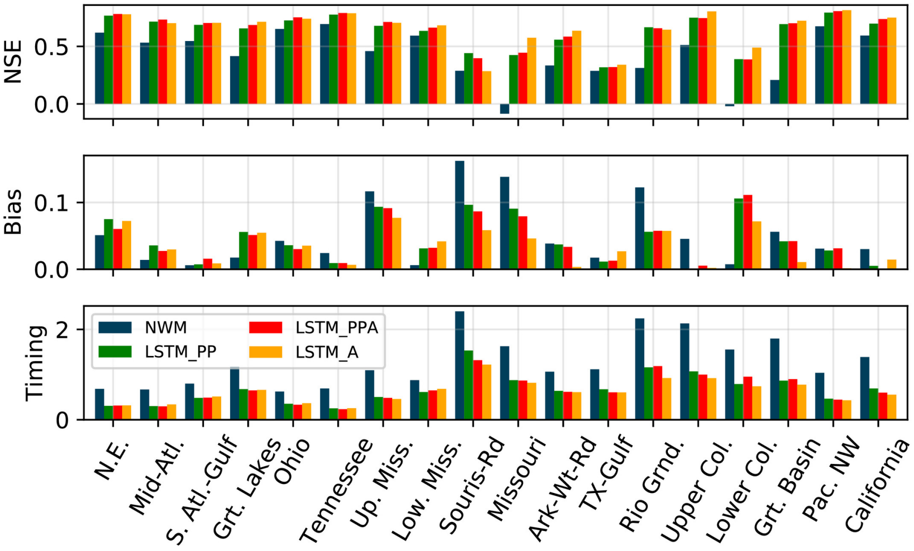

Figure 2.6: Regionally averaged performance metrics for NWM, LSTM_A, and the LSTM post-processors (LSTM_PP and LSTM_PPA) in different USGS water resources regions.

The regional performance of the LSTM post-processors and the regional performance of the LSTM_A were correlated with the regional performance of the NWM in terms of NSE ($r^2 = 0.78$ for post-processors and 0.63 for LSTM_A) and peak timing error ($r^2 = 0.96$ for post-processors and 0.92 for LSTM_A), but not in terms of bias ($r^2 = 0.24$, calculated on bias although absolute bias is plotted for clarity). The post-processors and the LSTM_A are correlated in terms of their bias ($r^2 = 0.91$). A better model has a higher NSE, bias closer to zero, and a lower timing error.

2.4.4 Regression to predict post-processing performance improvement

The performance of the LSTM_A was more predictable than the post-processors. We performed a multi-linear regression on the target of performance improvement over the NWM, with inputs being the catchment attributes and hydrologic signatures, as well as the NWM performance itself. Figure 2.7 shows the results predicting the LSTM improvement over the NWM at each basin with an r^2 value of 0.97, 0.88, and 0.89 for the LSTM_A, LSTM_PPA, and LSTM_PP, respectively. The high r^2 value is due in part to the outlier basins with abnormally large performance improvements from the LSTM models (LSTM_A, LSTM_PPA, and LSTM_PP). This means that the magnitude of the LSTM_A and post-processors improvement is directly related to the performance of the NWM.

The aim of these results is to understand whether it is possible to predict where post-processing might be beneficial (remember that post-processing helped in most basins). Although we found relatively high predictability in the improvement expected from post-processing, a problem is that this requires knowing ahead of time the NWM performance. This prevents us from predicting post-processing improvement in ungauged basins, since calculating the NWM performance requires streamflow observations. The correlation analysis below may help inform future efforts to learn general patterns of post-processor improvement over both the NWM and the LSTM_A.

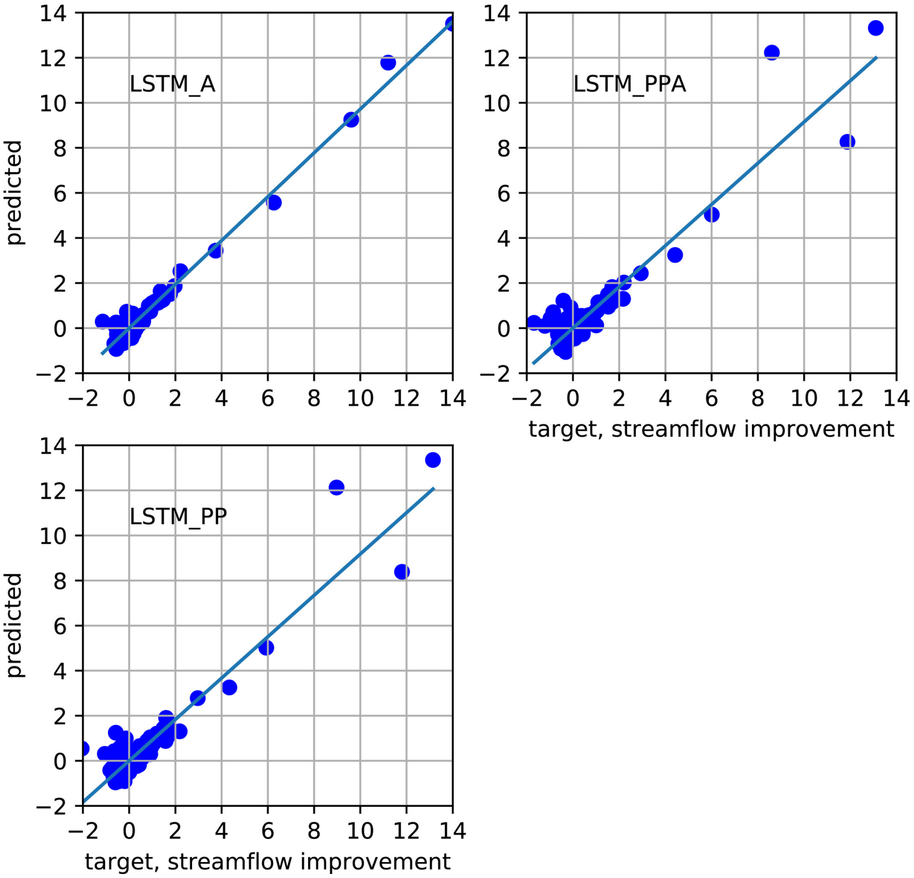

Figure 2.7: Predicting LSTM_A, LSTM_PP and LSTM_PPA performance over the NWM at each basin using a linear regression with NWM performance and hydrologic signatures as inputs. Scatter plots with all of the 531 basins.

2.4.5 Correlations between NWM inputs and improvements

Figure 2.8 shows correlations (over 531 basins) between the time-averaged NWM inputs and changes in performance metric scores of the post-processor relative to the NWM and LSTM_A. The LSTM_PP was compared against the NWM and the LSTM_PPA was compared against the LSTM_A, although qualitatively both post-processor models were similar. The rows of this figure show that correlation was weaker for differences in NSE score than total bias and peak timing error. Performance differences between the NWM and the post-processor were most strongly (anti)correlated with stream velocity from the channel router and accumulated underground runoff from the land surface model component: basins with lower stream velocity (velocity) and less underground runoff (UGDRNOFF) saw greater performance improvement from (daily) post-processing. This means that in basins with high underground runoff and/or high stream velocity the post-processor improvements were smaller. In contrast, basins with higher total radiation (TRAD) and higher latent heat flux (LH) saw greater improvement due to post-processing. This means that in basins with more radiation and heat flux the post-processor improvements were larger. A direct interpretation of this could be that a flat meandering stream in the Southwest will benefit from post-processing, which is consistent with the findings of Salas et al. (2018) that WRF-Hydro's performance is generally poor in the Southwest. Performance differences between the LSTM_A and the post-processor were most strongly correlated with snow water equivalent and snow depth. This is consistent with the findings of Hansen et al. (2019) that the NWM represents snowpack hydrology well.

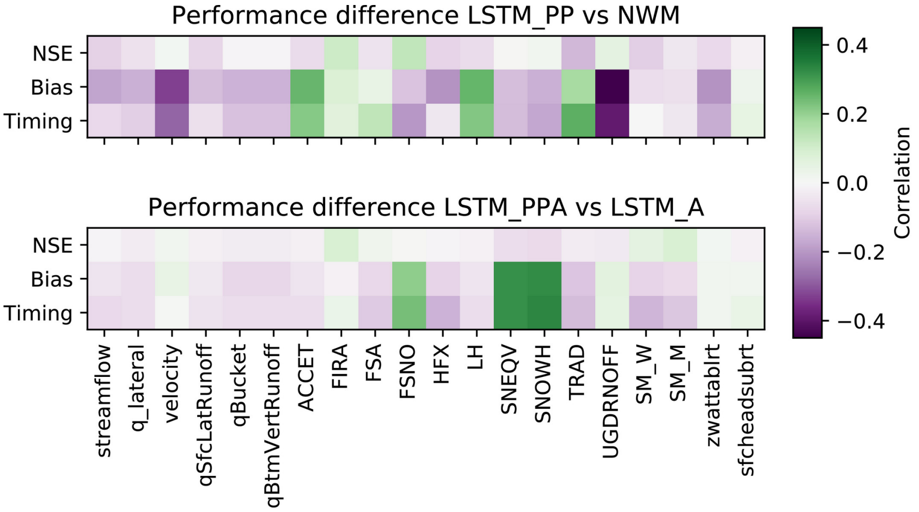

Figure 2.8: Correlations between the time-averaged NWM related inputs vs. performance metric differences between the LSTM post-processors (LSTM_PP and LSTM_PPA) and NWM and LSTM_A.

2.4.6 Integrated gradients

Figure 2.9 shows the relative strength of the total attribution of the dynamic inputs to the LSTM_PPA averaged across the entire validation period and across basins. The ordered magnitudes of the integrated gradients can be interpreted as corresponding to the order of importance of inputs. The most important dynamic features for the LSTM_PPA were: (1) precipitation from NLDAS, and (2) routed streamflow from the NWM point data. Precipitation inputs were weighted higher than the NWM streamflow output itself, which means that even when NWM streamflow data were available, the LSTM_PPA generally learned to get information directly from forcings rather than from the NWM streamflow output. This indicates that the LSTM_PPA generated a new rainfall–runoff relationship rather than relying on the NWM, which is consistent with the overall results (Figure 2.2) that showed similar performance between the LSTM_A and LSTM_PPA.

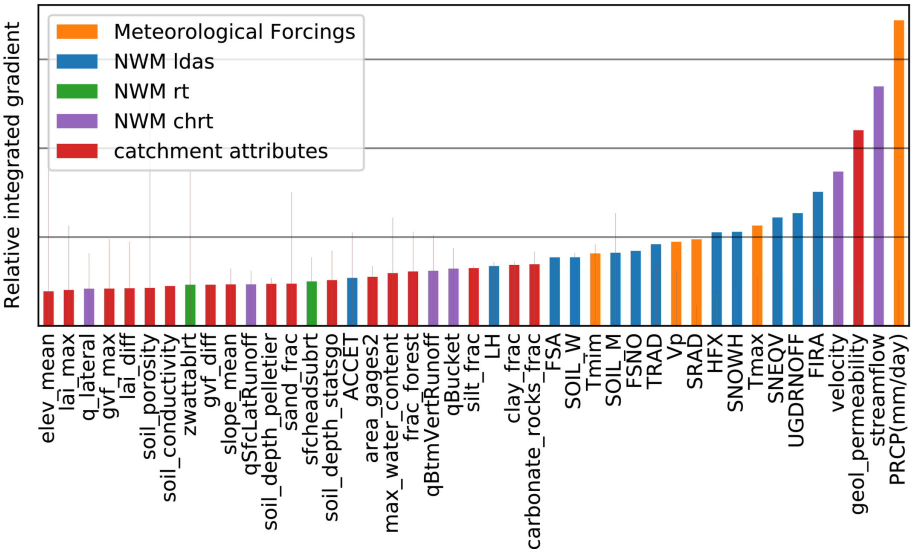

Figure 2.9: Attributions to the LSTM_PPA predictions. The vertical axis shows the relative magnitude of attribution (importance) for each input, with precipitation (PRCP) as the top contributor and NWM-predicted runoff into channel reach (q_lateral) contributing the least.

Figure 2.10 shows the relative strength of the total attribution of the dynamic inputs to the LSTM_PP. Without the atmospheric forcings included in the post-processor inputs, the NWM streamflow output was by far the highest contributing dynamic input feature to the LSTM_PP. The static permeability of the catchment was the next highest.

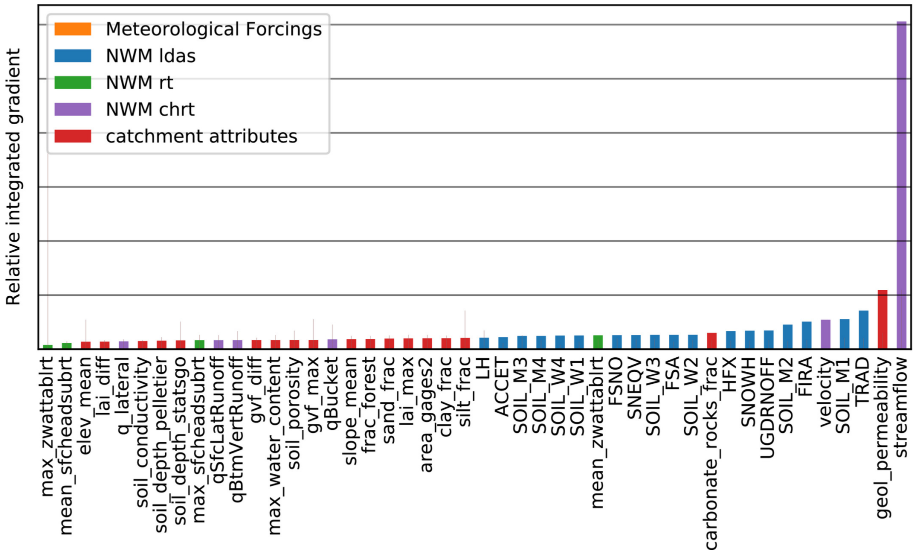

Figure 2.10: Attributions for the LSTM_PP model. Color coded by LSTM input source. The streamflow is overwhelmingly the highest contributor to the post-processed streamflow prediction.

2.4.7 Representations of hydrologic signatures

Results of the analysis of hydrologic signature representation are shown in Figure 2.11, which also shows that the hydrologic signatures best represented by the NWM were similarly those best represented by the LSTM_PPA. The same was true for the most poorly represented hydrologic signatures in both models.

The LSTM post-processors hurt the representation of the frequency of days with zero flow. There were 101 basins with any periods of zero flow. None of these models do well simulating zero flow, but the NWM is better at handling this situation, predicting zero flow periods in 56 of the 101 basins. The LSTM_A, LSTM_PPA, and LSTM_PP only predicted periods of zero flows at 35, 29, and 25 basins, respectively. This is an important characteristic in basins in the Southwest, where the NWM could use the benefit of a LSTM post-processor, so this would be a good place to focus future research of theory-guided ML for hydrology.

The LSTM post-processor made a significant improvement over the NWM for several signatures. The improvement to runoff ratio, which is the fraction of precipitation

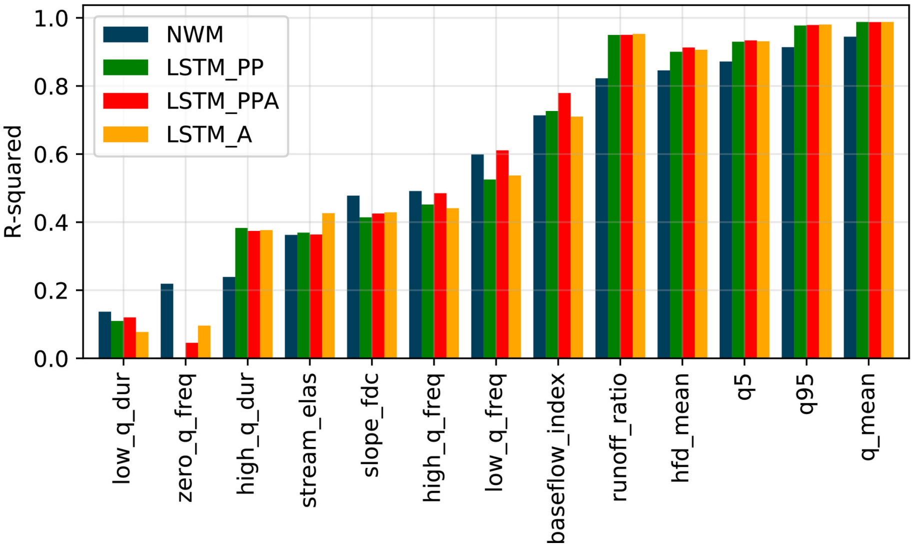

Figure 2.11: Correlation between simulated and observed per-basin hydrologic signatures from the NWM (blue), LSTM_A (orange), LSTM_PPA (green), and LSTM_PP (red). Larger values indicates better performance.

that makes it through the stream gauge at the surface, could be a compensation for the uncalibrated soil parameters in the NWM mentioned by Salas et al. (2018). The LSTM post-processor improved both high and low flow predictions (5% and 95% flow quantiles), which are important for natural resources management. The mean daily discharge was the best represented hydrologic signature by all models.

The LSTM_PPA post-processor made significant improvements over the LSTM for baseflow index. This is the only sign that an LSTM post-processor improved over both the NWM and the LSTM_A. This signature estimates the contribution of baseflow to the total discharge, which is computed by hydrograph separation. Klemeš (1986) (summarizing Lindsly's Applied Hydrology) cautioned strongly against using hydrograph separation, because there is no real basis for distinguishing the source of flow in a stream.

2.4.8 Results comparing gauged basins vs. ungauged basins

Results in Table 2.7 summarize an analysis designed to replicate prediction in ungauged basins. The table has metrics from predictions by the NWM, LSTM_A and the LSTM post-processors (LSTM_PP and LSTM_PPA) calculated only at basins that were either calibrated or uncalibrated, but not both. There was no statistical difference between the calibrated and uncalibrated samples. This indicates that the LSTM post-processor works in uncalibrated basins. When post-processors were trained only in calibrated basins (denoted with a “C” in Table 2.7), however, the performance in uncalibrated basins significantly deteriorated. But this is true for the LSTM_A as well, so it is not a result of the calibration (as calibration would not influence the LSTM_A), but a result of prediction at ungauged type basins. However, the median performance of the post-processor predictions at ungauged type basins when trained at only calibrated basins was still significantly better than the NWM in the uncalibrated basins.

Table 2.7: Performance of the LSTM and the LSTM post-processor split between basins where the NWM was calibrated vs. uncalibrated. The “C” in the model name denotes that the model training set only included calibrated basins.

	Calibrated basins				Uncalibrated basins			
Mean	Median	Max	Min	Mean	Median	Max	Min	
NSE								
NWM	0.49	0.64	0.95	-10.81	0.18	0.48	0.79	-7.10
LSTM_PP	0.65	0.73	0.93	-3.32	0.69	0.71	0.89	0.38
LSTM_A	0.68	0.74	0.93	-0.64	0.73	0.75	0.89	0.43
LSTM_PPA	0.66	0.75	0.93	-3.61	0.71	0.73	0.89	0.42
LSTM_PP(C)	0.65	0.73	0.93	-1.86	0.21	0.57	0.75	-8.12
LSTM_A(C)	0.67	0.74	0.93	-1.13	0.51	0.67	0.84	-2.54
LSTM_PPA(C)	0.67	0.75	0.94	-2.71	0.13	0.58	0.84	-14.07
Total bias								
NWM	0.01	-0.01	2.57	-0.63	0	-0.06	1.84	-0.58
LSTM_PP	0.04	0.02	1.05	-0.24	0.02	0.01	0.27	-0.12
LSTM_A	0.02	0.02	0.56	-0.22	0.02	0.01	0.2	-0.11
LSTM_PPA	0.03	0.02	0.98	-0.21	0.01	0	0.22	-0.11
LSTM_PP(C)	0.01	-0.01	0.92	-0.25	0.06	-0.04	2.15	-0.51
LSTM_A(C)	0.02	0.02	0.62	-0.21	0.09	0.04	0.99	-0.20
LSTM_PPA(C)	0.01	0	0.95	-0.22	0.06	-0.05	2.89	-0.41
Peak timing error								
NWM	1.06	0.91	3	0.1	1.04	0.77	2.7	0.25
LSTM_PP	0.55	0.45	1.95	0.04	0.52	0.35	1.59	0.04
LSTM_A	0.53	0.43	1.76	0	0.51	0.41	1.5	0.04
LSTM_PPA	0.54	0.42	1.75	0.04	0.51	0.36	1.45	0.05
LSTM_PP(C)	0.55	0.45	2.1	0	0.59	0.41	1.76	0.09
LSTM_A(C)	0.52	0.43	1.77	0	0.57	0.5	1.5	0.08
LSTM_PPA(C)	0.54	0.41	1.83	0.04	0.57	0.41	1.65	0.13

The NWM, LSTM_A, and the LSTM_PPA had higher NSE scores in calibrated basins than the uncalibrated basins. Note that these results are from the LSTMs (with and without NWM model outputs) trained on only basins where the NWM was calibrated. In the case of the LSTM post-processors the mean NSE scores in uncalibrated basins were very low for NSE. This is a result of two outlier basins (1466500, MCDONALDS BRANCH, Lat 39.9, Lon −74.5, Area 5.7 km; and 01484100 BEAVERDAM BRANCH, Lat 38.9, Lon −75.5, Area 7.8 km). Both of those outlier basins are much smaller, and have lower flows, than the average of the training set. Without these basins the mean NSE scores were 0.32, 0.51, 0.56 and 0.56 for the NWM, LSTM_PP, LSTM_A, and LSTM_PPA, respectively. Table 2.7 also shows that the median value of the LSTM_PPA was higher than the NWM, as was the maximum NSE value, but the minimum value was exceptionally low.

The total bias in calibrated basins was generally better (lower) than the uncalibrated basins. The timing error of the NWM was actually better in the uncalibrated basins, but the LSTM_A and LSTM post-processors had better performance in the calibrated basins. The NSE values for the NWM, LSTM_A, and the LSTM post-processors (LSTM_PP and LSTM_PPA) were significantly different in the calibrated basins vs. the uncalibrated basins, as were the differences between the LSTM_A and LSTM post-processors (LSTM_PP and LSTM_PPA) compared to the NWM. The bias values were significantly different between the two samples (calibrated vs. uncalibrated), but the differences between LSTM_A and LSTM post-processors vs. the NWM were not statistically different. This means that the LSTM models were successful at predicting streamflow at basins outside of the calibration set.

2.4.9 LSTM post-processor sensitivity to inputs and application for process representation diagnostics

Figure 2.12 shows results from the LSTM models with inputs from different parts of the NWM (land surface model only, channel router only, predicted streamflow only, and all states and fluxes). The best performing LSTM models (LSTM_A and LSTM_PPA) were the ones trained with inputs that included the five atmospheric forcing variables with (LSTM_PPA) and without (LSTM_A) the NWM output (these are the same models discussed in previous sections above). This implies that LSTM in general was able to extract more information from the atmospheric forcings than the NWM. Each of the LSTM post-processors made better average daily streamflow predictions than the NWM itself, indicating that information from the atmospheric forcings is lost in the NWM model structure before the streamflow prediction is made. For example, the LSTM that took as inputs only the LDAS model output from the NWM made better predictions than the NWM itself, indicating that there is more information in the LDAS states and fluxes than the NWM is able to translate into streamflow predictions. The same was true for the states and fluxes of the CHRT component of the NWM, meaning that information is also lost in the CHRT component of the NWM model structure.

2.5 Discussion

2.5.1 Comparison between the LSTM_A and the post-processors (LSTM_PP and LSTM_PPA)

The LSTM_A, trained only on atmospheric forcings as dynamic inputs, was better at extrapolating hydrologic conditions outside the training set than the LSTM post-processors (LSTM_PP and LSTM_PPA), and thus LSTM_A is the better performing model. This is shown in the analysis of prediction in ungauged basins, specifically Table 2.7. The post-processors both failed to make reasonable predictions at two basins that were much smaller than any basins included in the training set. The LSTM_A was able to make good predictions in these basins. Including the NWM output as dynamic inputs to the LSTM constrained the model and prevented it from learning general hydrologic relationships that can be extracted to basins with characteristics that might be unrecognizable.

2.5.2 Potential for improving the performance of both the NWM and ML

Results presented here show that the LSTM post-processors are unreliable for improving predictions of the NWM. The LSTM post-processors did provide significant benefit to the NWM streamflow predictions at almost all (88% and 92% for LSTM_PP and

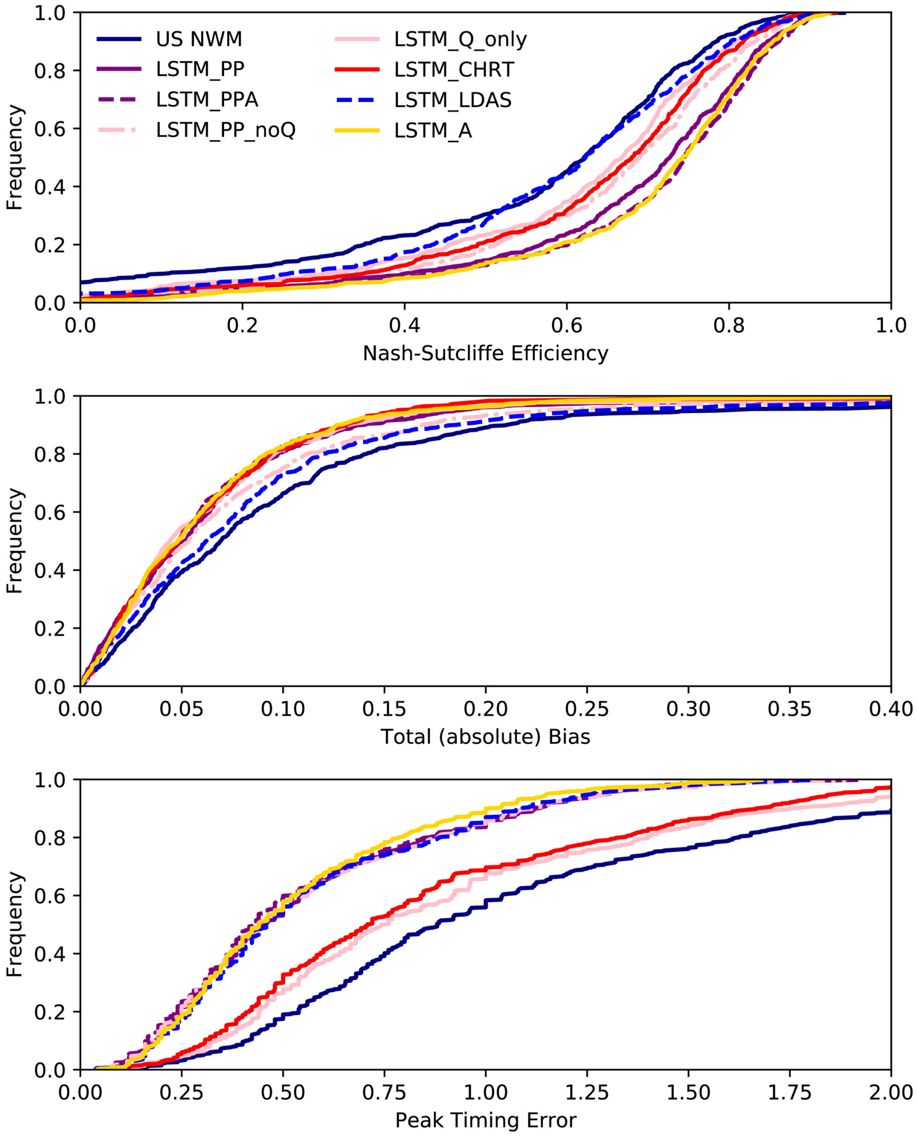

Figure 2.12: Performance of the LSTM post-processor trained with different sets of NWM output. Each of these post-processors outperform the NWM. LSTM_A is the LSTM trained with atmospheric forcings as dynamic inputs. LSTM_PP is the NWM post-processor trained with the outputs of the NWM as dynamic inputs. LSTM_PPA used both the NWM outputs and atmospheric forcings as inputs. LSTM_PP_noQ used all the NWM outputs except for streamflow and velocity from the channel router. LSTM_Qonly used only streamflow from the NWM output. LSTM_chrt used only the NWM channel router outputs. LSTM_ldas used only the land surface fluxes as inputs.

LSTM_PPA, respectively) of the 531 basins analyzed here, but was severely detrimental to two basins in our tests of ungauged basins. In the basins where this was not the case, it may be possible to use fine tuning a version of the post-processor that is specific to each gauge location (as would be done in traditional model calibration); however, the LSTM_A did not have this problem and is more reliable. We trained the LSTMs on headwater basins, so further work would be needed to include reservoirs, urban areas, and other management practices. It is worth noting that these LSTM models can be trained on a laptop computer in a few hours, a relatively minor computational cost, and the computational cost of forward prediction is negligible. By comparison the computational cost of calibrating the NWM is much higher — typically requiring HPC or cloud systems.

The NWM performance and the performance improvement from the LSTM post-processors (LSTM_PP and LSTM_PPA) were negatively correlated: basins with a low performance by the NWM have the highest performance change from the LSTM post-processors. This means that post-processing can be expected to correct situations where the NWM gives bad predictions. Conversely, the performance of the NWM and the LSTM_A (the LSTM trained without NWM model outputs) were minimally correlated ($r2 = 0.42$, 0.30, and 0.67 for NSE, bias, and timing, respectively). Considering also that the overall performance of the LSTM_A changed only minimally from the addition of the NWM inputs (as shown in Figures 2.3 - 2.5; Table 2.6) and that the LSTM_PPA still preferred to extract more information from precipitation forcings (shown in Figure 2.9), we might conclude that the LSTM post-processors learned new patterns of the rainfall–runoff response, which are not fully represented by the NWM. But this relationship is also learned by LSTM_A, without the influence of the NWM. The overall improvement in the representation of hydrologic signatures indicates the post-processor may be a better representation of physical flow patterns than either the NWM or the LSTM_A, though not significantly. The interpretation of the integrated gradient (Figures 2.9 and 2.10) and the correlations between improvement and NWM features (Figure 2.8) indicate that this improvement of flow patterns comes from information in the NWM representation of streamflow and snow states.

2.5.3 Application to real-time forecasting

The NWM is not simply a rainfall–runoff simulator; it simulates flow through 2.7 million river reaches around CONUS, dam operations, land surface processes, hydraulics, and other complications of large domain hydrology. The nature of the CAMELS catchments selected in these experiments is such that they have few engineered control structures and are under $20,000 km^2$. The results presented in this paper show that the LSTMs improved streamflow predictions in the catchments studied here, which all had limited human disturbance (e.g., dams, reservoirs, etc.). Kratzert et al. (2019a) showed that LSTM_A predictions extend into ungauged basins, and this is consistent with our results. Our results (section "Results comparing calibrated basins vs. uncalibrated basins") show that the LSTM_A is a much better choice than the post-processors in ungauged basins, which is the majority of the NWM domain. The immediate potential for improving real-time forecasting could be deploying an LSTM_A for streamflow prediction in undisturbed catchments, and undisturbed subcatchments upstream of unnatural hydrologic conditions such as dams, agriculture lands, and urban centers. This would allow for retaining conceptual representations of lakes and reservoirs that already exist in the NWM.

2.5.4 Diagnosing process-based models, physical processes, and data concerns

The sensitivity analysis reported in Figure 2.12 showed that some components of the NWM caused poor predictions. Specifically, information was lost in channel router (CHRT) component of the model. This diagnostic method could be used to compare different schemes for future versions of the NWM. For instance, changing the routing function might conserve timing information from the land surface fluxes, or modifying the evapotranspiration options in Noah-MP may conserve mass bias information from the NWM forcing engine. Such improvements could be quantified with this post-processing method.

Each of the post-processing models tested for sensitivity (Figure 2.12) fall, roughly and inclusively, between the NWM and the LSTM_A. Based on the relative positions between those bounding curves, we can identify sources of information loss through the NWM modeling chain:

- The channel routing outputs contain more information of simulation bias than timing, meaning the channel router moves with poor timing, but conserves mass well.
- The land surface outputs contain more information of simulation timing than bias, meaning the land surface component does not conserve mass well, but delivers water to the channel at appropriate times.
- Information is lost during channel routing after the mass is delivered, indicating the channel router is not functioning properly.

There is potential to expand this analysis, breaking down the NWM components even further. Quantification can be done with the full state space from the NWM. Retrospective runs using new versions of the NWM should output the full state space for these types of analysis. This diagnostics analysis using ML post-processing is possible with any physics-based, conceptual or process-based dynamics model.

2.5.5 Moving forward with theory-guided ML

The post-processing procedure presented here is one of the cruder techniques currently available for combining process-based and data-driven models. Several other methods of combining the benefits of ML (predictability) with the benefits of physically realistic hydrologic theory (robustness) are in development. For example, (Pelissier et al., 2019) integrated a trained Gaussian Processes into the state-space dynamics of a process-based land surface model for predicting soil moisture time series. Another example is using physical principles to constrain the loss function of an ML model during training — for example, Hoedt et al. (2021) integrated mass balance constraints into an LSTM and applied this model to the same 531 basins used in this study. Implementing post-processing is relatively straightforward compared to other techniques such as adding physics into ML code or using ML to dynamically update the state variables, but is unreliable when the process-based models used as input is uncalibrated.

Using ML for post-processing has the potential for advancing the explainability of data-driven models. We showed that the LSTM model representation of hydrologic signatures (with and without NWM model outputs) is highly correlated with the NWM. This indicates that the "learned" functions mapping inputs to streamflow are actually quite similar. We might have trouble expressing the "learned" LSTM with compact formulas (e.g., PDEs), given the high number of trained model weights, but we can use them with confidence knowing their structural similarities with process-based models like the NWM.

2.6 Conclusion

The LSTM post-processors (LSTM_PPA and LSTM_PP) significantly outperformed the NWM, but did not consistently, nor significantly, outperformed the LSTM_A (the LSTM model trained without the NWM model outputs as LSTM inputs). LSTMs, in general, are capable of learning the dynamics of rainfall–runoff processes, gaining little additional information from the conceptualizations coded within the NWM. The "pure" post-processing model (LSTM_PP) outperformed the NWM in terms of bias, and significantly outperformed the NWM in terms of NSE and timing. A decision to use the LSTM as a post-processor for the NWM should be made with professional judgment, considering the comparison of the NWM, LSTM, and LSTM post-processor's performance. In locations where the NWM is not calibrated, or the hydrologic conditions are not well understood, it would be best to use the LSTM without the influence from the NWM.

The results indicate that there is more information in the atmospheric forcings about streamflow observations than in the NWM outputs, including the NWM streamflow prediction. The NWM loses information between the atmospheric forcing inputs and the outputs. The NWM land surface component (LDAS) loses information about mass conservation (shown from the bias error), and the channel router (CHRT) loses information about streamflow timing. The NWM routing scheme should be considered as a priority for improving the NWM.

2.7 Code and data availability

All data and code used in this paper are publicly available in the following locations: U.S. National Water Model: https://docs.opendata.aws/nwm-archive/readme.html. CAMELS data: https://ral.ucar.edu/solutions/products/camels. Data processing code:https://github.com/jmframe/nwm-reanalysis-model-data-processing; https://doi.org/10.5281/zenodo.4642605. LSTM code: https://github.com/kratzert/ealstm_regional_modeling. Post-processing and analysis code: https://github.com/jmframe/nwm-post-processing-with-lstm; https://doi.org/10.5281/zenodo.4642603.

REFERENCES

Addor, N., Nearing, G., Prieto, C., Newman, A. J., Le Vine, N., and Clark, M. P. (2018). A Ranking of Hydrological Signatures Based on Their Predictability in Space. *Water Resources Research*, 54(11):8792–8812.

Addor, N., Newman, A. J., Mizukami, N., and Clark, M. P. (2017). The CAMELS data set: catchment attributes and meteorology for large-sample studies. *Earth Syst. Sci*, 21:5293–5313.

Chadalawada, J., Herath, H. M., and Babovic, V. (2020). Hydrologically Informed Machine Learning for Rainfall-Runoff Modeling: A Genetic Programming-Based Toolkit for Automatic Model Induction. *Water Resources Research*, 56(4):1–23.

Cosgrove, B., Gochis, D., Clark, E. P., Cui, Z., Dugger, A. L., Fall, G. M., Feng, X., Fresch, M. A., Gourly, J. J., Khan, S., Kitzmiller, D., Lee, H. S., Liu, Y., McCreight, J. L., Newman, A. J., Oubeidillah, A., Pan, L., Pham, C., Salas, F., and Sampson, K. M. (2015). Hydrologic modeling at the national water center: Operational implementation of the wrf-hydro model to support national weather service hydrology. In *AGU Fall Meeting Abstracts*.

Daw, A., Thomas, R. Q., Carey, C. C., Read, J. S., Appling, A. P., and Karpatne, A. (2020). Physics-Guided Architecture (PGA) of Neural Networks for Quantifying Uncertainty in Lake Temperature Modeling. *Proceedings of the 2020 SIAM International Conference on Data Mining*, pages 532–540.

Elmer, N. J. (2019). *Using Satellite Observations of River Height and Vegetation To Improve National Water Model Initialization and Streamflow Prediction.* PhD thesis, The University of Alabama in Huntsville.

Gauch, M., Kratzert, F., Klotz, D., Nearing, G., Lin, J., and 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.

Gauch, M., Mai, J., and Lin, J. (2019). The Proper Care and Feeding of CAMELS: How Limited Training Data Affects Streamflow Prediction. 2342:0–2.

Gupta, H. V., Kling, H., Yilmaz, K. K., and 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., Wagener, T., and Liu, Y. (2008). Reconciling theory with observations: elements of a diagnostic approach to model evaluation. *Hydrological Processes*, 2274(November 2008):2267–2274.

Hansen, C., Shafiei Shiva, J., McDonald, S., and Nabors, A. (2019). Assessing Retrospective National Water Model Streamflow with Respect to Droughts and Low Flows in the Colorado River Basin. *Journal of the American Water Resources Association*, 55(4):964–975.

Hochreiter, S. (1991). Untersuchungen zu dynamischen neuronalen Netzen. *Unpublished doctoral dissertation, Institut für Informatik, Technische Universität, Munchen*, pages 1–71.

Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Hoedt, P.-J., Kratzert, F., Klotz, D., Halmich, C., Holzleitner, M., Nearing, G., Hochreiter, S., and Klambauer, G. (2021). MC-LSTM: Mass-Conserving LSTM. pages 1–32.

Karpatne, A., Watkins, W., Read, J., and Kumar, V. (2017). Physics-guided Neural Networks (PGNN): An Application in Lake Temperature Modeling.

Kim, J., Read, L., Johnson, L. E., Gochis, D., Cifelli, R., and Han, H. (2020). An experiment on reservoir representation schemes to improve hydrologic prediction: coupling the National Water Model with the HEC-ResSim. *Hydrological Sciences Journal*, 0(0):1.

Klemeš, V. (1986). Dilettantism in hydrology: Transition or destiny? *Water Resources Research*, 22(9 S):177S–188S.

Kratzert, F., Klotz, D., Brenner, C., Schulz, K., and 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., and Nearing, G. S. (2019a). Towards Improved Predictions in Ungauged Basins: Exploiting the Power of Machine Learning. *Water Resources Research*, page 2019WR026065.

Kratzert, F., Klotz, D., Shalev, G., Klambauer, G., Hochreiter, S., and Nearing, G. (2019b). 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., Klotz, D., Shalev, G., Klambauer, G., Hochreiter, S., and Nearing, G. (2019c). 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.

Nearing, G. S., Kratzert, F., Sampson, A. K., Pelissier, C. S., Klotz, D., Frame, J. M., Prieto, C., and Gupta, H. V. (2020). What Role Does Hydrological Science Play in the Age of Machine Learning? *Water Resources Research*.

Nearing, G. S., Ruddell, B. L., Clark, M. P., Nijssen, B., and Peters-Lidard, C. (2018). Benchmarking and process diagnostics of land models. *Journal of Hydrometeorology*, pages JHM–D–17–0209.1.

Newman, A. J., Clark, M. P., Sampson, K., Wood, A., Hay, L. E., Bock, A., Viger, R. J., Blodgett, D., Brekke, L., Arnold, J. R., Hopson, T., and Duan, Q. (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.

Niu, G.-Y., Yang, Z.-L., Mitchell, K. E., Chen, F., Ek, M. B., Barlage, M., Kumar, A., Manning, K., Niyogi, D., Rosero, E., et al. (2011). The community noah land surface model with multiparameterization options (noah-mp): 1. model description and evaluation with local-scale measurements. *Journal of Geophysical Research: Atmospheres*, 116(D12).

NOAA (2019). Stakeholder engagement to inform national weather service hydrologic products and services to meet user needs noaa national weather service water resources services branch and office of water prediction.

Pelissier, C., Frame, J., and Nearing, G. (2019). Combining Parametric Land Surface Models with Machine Learning.

Read, J. S., Jia, X., Willard, J., Appling, A. P., Zwart, J. A., Oliver, S. K., Karpatne, A., Hansen, G. J., Hanson, P. C., Watkins, W., Steinbach, M., and Kumar, V. (2019). Process-Guided Deep Learning Predictions of Lake Water Temperature. *Water Resources Research*, 55(11):9173–9190.

Salas, F. R., Somos-Valenzuela, M. A., Dugger, A., Maidment, D. R., Gochis, D. J., David, C. H., Yu, W., Ding, D., Clark, E. P., and Noman, N. (2018). Towards Real-Time Continental Scale Streamflow Simulation in Continuous and Discrete Space. *Journal of the American Water Resources Association*, 54(1):7–27.

Steiger, J. and Browne, M. (1984). The comparison of interdependent correlations between optimal linear composites. *Psychometrika*, 49(1):11–24.

Sundararajan, M., Taly, A., and Yan, Q. (2017). Axiomatic attribution for deep networks. *34th International Conference on Machine Learning, ICML 2017*, 7:5109–5118.

Tartakovsky, A. M., Marrero, C. O., Perdikaris, P., Tartakovsky, G. D., and Barajas-Solano, D. (2020). Physics-Informed Deep Neural Networks for Learning Parameters and Constitutive Relationships in Subsurface Flow Problems. *Water Resources Research*, 56(5):1–16.

Xia, Y., Mitchell, K., Ek, M., Sheffield, J., Cosgrove, B., Wood, E., Luo, L., Alonge, C., Wei, H., Meng, J., Livneh, B., Lettenmaier, D., Koren, V., Duan, Q., Mo, K., Fan, Y., and Mocko, D. (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(3).

Ye, A., Duan, Q., Yuan, X., Wood, E. F., and Schaake, J. (2014). Hydrologic post-processing of MOPEX streamflow simulations. *JOURNAL OF HYDROLOGY*, 508:147–156.

CHAPTER 3

DEEP LEARNING RAINFALL-RUNOFF PREDICTIONS OF EXTREME EVENTS

Jonathan M. Frame, Frederik Kratzert, Daniel Klotz, Martin Gauch, Guy Shalev, Oren Gilon, Logan M. Qualls, Hoshin V. Gupta and Grey S.Nearing

Published in Hydrology and Earth Systems Science

https://doi.org/10.5194/hess-26-3377-2022

3.1 Abstract

The most accurate rainfall-runoff predictions are currently based on deep learning. There is a concern among hydrologists that the predictive accuracy of data-driven models based on deep learning may not be reliable in extrapolation or for predicting extreme events. This study tests that hypothesis using Long Short-Term Memory networks (LSTMs) and an LSTM variant that is architecturally constrained to conserve mass. The LSTM (and the mass-conserving LSTM variant) remained relatively accurate in predicting extreme (high return-period) events compared to both a conceptual model (the Sacramento Model) and a process-based model (US National Water Model), even when extreme events were not included in the training period. Adding mass balance constraints to the data-driven model (LSTM) reduced model skill during extreme events.

3.2 Introduction

Deep learning (DL) provides the most accurate rainfall-runoff simulations available from the hydrological sciences community (Kratzert et al., 2019b,a). This type of finding is not new – Todini (2007) noted more than a decade ago, in his review of the history of hydrological modeling, that *"physical process-oriented modellers have no* *confidence in the capabilities of data-driven models' outputs with their heavy dependence on training sets, while the more system engineering-oriented modellers claim that data-driven models produce better forecasts than complex physically-based models."* Echoing this sentiment about the perceived predictive reliability of data-driven models, Sellars (2018) reported in their summary of a workshop on 'Big Data and the Earth Sciences' that *"[m]any participants who have worked in modeling physical-based systems continue to raise caution about the lack of physical understanding of ML methods that rely on data-driven approaches."*

The idea that the predictive accuracy of hydrological models based on physical understanding might be more reliable than machine learning (ML) based models in out-of-sample conditions was drawn from early experiments on shallow neural networks (e.g., Cameron et al., 2002; Gaume and Gosset, 2003). However, although this idea is still frequently cited (e.g., quotations above; Herath et al., 2020; Reichstein et al., 2019; Rasp et al., 2018), it has not been tested in the context of modern DL models, which are able to generalize complex hydrological relationships across space and time (Nearing et al., 2020b). Further, there is some evidence that this hypothesis might not be true. For example, Kratzert et al. (2019a) showed that DL can generalize to *ungauged* basins with better overall skill than calibrated conceptual models in *gauged* basins. Kratzert et al. (2019b) used a slightly modified version of a Long Short-Term Memory Network (LSTM) to show how the model learns to transfer information between basins. Similarly, Nearing et al. (2019) showed how an LSTM-based model learns *dynamic* basin similarity under changing climate, so that when the climate in a particular basin shifts (e.g., becomes wetter or drier), the model learns to adapt hydrological behavior based on different climatological neighbors. Further, because DL is currently the state-of-the-art for rainfall-runoff prediction, it is important to understand its potential limits.

The primary objective of this study is to test the hypothesis that data-driven models lose predictive accuracy in extreme events more than models based on process-understanding. We focus specifically on high return period (low probability) streamflow events, and compare four models: a standard deep learning model, a physics-informed deep learning model, a conceptual rainfall-runoff model, and a process-based hydrological model.

3.3 Methods

3.3.1 Data

The hydrological sciences community lacks community-wide standardized procedures for model benchmarking, which severly limits the effectiveness of new model development and deployment efforts (Nearing et al., 2020b). In previous studies, we used open community data sets and consistent training/test procedures that allow for results to be directly comparable between studies – we continue that practice here to the extent possible.

Specifically, we used the Catchment Attributes and Meteorological Large Sample (CAMELS) data set curated by the US National Center for Atmospheric Research (NCAR) (Newman et al., 2015; Addor et al., 2017). The CAMELS data set consists of daily meteorological and discharge data from 671 catchments in CONUS ranging in size from 4 km^2 to 25,000 km^2 that have largely natural flows and long streamflow gauge records (1980-2008). We used 498 of 671 CAMELS catchments – these were included in the basins that were used for model benchmarking by Newman et al. (2017), who removed basins with (i) large discrepancies between different methods of calculating catchment area, and (ii) areas larger than 2,000 km^2.

CAMELS includes daily discharge data from the USGS Water Information System, which are used as training and evaluation target data. CAMELS also includes several daily meteorological forcing data sets (Daymet, NLDAS, Maurer). We used NLDAS for this project because we benchmarked against the National Water Model retrospective (will be introduced in detail in 3.3.3.2), which also uses NLDAS. CAMELS also includes several static catchment attributes related to soils, climate, vegetation, topography, and geology (Addor et al., 2017) that are used as input features. We used the same input features (meteorological forcings and static catchment attributes) that were listed in Table 1 by Kratzert et al. (2019b).

3.3.2 Return period calculations

The return periods of peak annual flows provide a basis for categorizing target data in a hydrologically meaningful way. This results in a metric that is consistent while maintaining diversity across basins – e.g., a similar flow volume may be 'extreme' in one basin but not in another. Splitting model training and test periods by different return periods allows us to assess model performance on both rare and effectively unobserved events.

For return period calculations we followed guidelines in the U.S. Interagency Committee on Water Data Bulletin 17b (IACWD, 1982). The procedure is to fit all available annual peak flows (log transformed) for each basin to a Pearson Type III distribution using the method of moments:

$$f(y;\tau,\alpha,\beta) = \frac{(\frac{y-\tau}{\beta})^{\alpha-1}exp(-\frac{y-\tau}{\beta})}{|\beta|\Gamma(\alpha)}, \tag{3.1}$$

with $\frac{y-\tau}{\beta} > 0$ and distribution parameters τ, α, and β, where τ is the location parameter, α is the shape parameter, β is the scale parameter, and $\Gamma(\alpha)$ is the gamma function.

To calculate the return periods, we used annual peak flow observations taken directly from the USGS National Water Information System (WIS), instead of from the CAMELS data, because the Bulletin 17b guidelines require annual peak flows whereas CAMELS provides only daily averaged flows. The Bulletin 17b (IACWD, 1982) guidelines require using all available data, which for peak flows ranges from 26 to 116 years. After fitting the return period distributions for each basin, we classified each water year of the

CAMELS data from each basin (each basin-year of data) according to the return period of its observed peak annual discharge.

This return-period analysis does not account for nonstationarity – i.e., the return period of a given magnitude of event in a given basin could change due to changing climate or changing land use. There is currently no agreed upon method to account for nonstationarity when determining flood flow frequencies, so it would be difficult to incorporate this in our return period calculations. However, for the purpose of this paper (testing whether the LSTM is reliable in extreme events) this is not an issue because stationary return period calculations directly test predictability on large events that are out-of-sample *relative to the training period*, which for practical purposes can represent potential nonstationarity.

3.3.3 Models

3.3.3.1 ML models & training

We test two ML models: a pure LSTM and a physics-informed LSTM that is architecturally constrained to conserve mass – we call this a Mass-Conserving LSTM (MC-LSTM; Hoedt et al., 2021). These models are described in detail in Appendices 3.6 and 3.7.

Daily meteorological forcing data and static catchment attributes data were used as inputs features for the LSTM and MC-LSTM, and daily streamflow records were used as training targets with a normalized squared-error loss function that does not depend on basin-specific mean discharge (i.e., large and/or wet basins are not over-weighted in the loss function):

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

where B is the number of basins, N is the number of samples (days) per basin B, $\widehat{y}_n$ is the prediction for sample n ($1 \leq n \leq N$), y_n is the corresponding observation, and $s(b)$ is the standard deviation of the discharge in basin b ($1 \leq b \leq B$), calculated from the training period (see, Kratzert et al., 2019b).

We trained both the standard LSTM and the MC-LSTM using the same training and test procedures outlined by Kratzert et al. (2019b). Both models were trained for 30 epochs using sequence-to-one prediction to allow for randomized, small minibatches. We used a minibatch size of 256 and, due to sequence-to-one training, each minibatch contained (randomly selected) samples from multiple basins. The standard LSTM had 128 cell states and a 365-day sequence length. Input and target features for the standard LSTM were pre-normalized by removing bias and scaling by variance. For the MC-LSTM the inputs were split between auxiliary, which were pre-normalized, and the mass input (in our case precipitation), which was not pre-normalized. Gradients were clipped to a global norm (per minibatch) of 1. Heteroscedastic noise was added to training targets (resampled at each minibatch) with standard deviation of 0.005 times the value of each target datum. We used an Adam optimizer with a fixed learning rate schedule; the initial learning rate of 1e-3 was decreased to 5e-4 after 10 epochs and 1e-4 after 25 epochs. Biases of the LSTM forget gate were initialized to 3 so that gradient signals persisted through the sequence from early epochs.

The MC-LSTM used the same hyperparameters as the LSTM except that it used only 64 cell states, which was found to perform better for this model (see, Hoedt et al., 2021). Note that the memory states in an MC-LSTM are fundamentally different than those of the LSTM due to the fact that they are physical states with physical units instead of purely information states.

All ML models were trained on data from the CAMELS catchments simultaneously. We used three different train and test periods:

1. The first train/test period split was the same split used in previous studies (Kratzert et al., 2019b, 2021; Hoedt et al., 2021). In this case, the training period included nine water years from October 1, 1999 through September 30, 2008, and the test

period included ten water years 1990-1999 (i.e., from October 1, 1989 through September 30, 1999). This train/test split was used *only* to ensure that the models trained here achieved similar performance compared with previous studies.

2. The second train/test period split used a test period that aligns with the availability of benchmark data from the US National Water Model (see Section 3.3.3.2). The train period included water years 1981-1995, and the test period included water years 1996-2014 (i.e., from October 1, 1995 through September 30, 2014). This was the same training period used by Newman et al. (2017) and Kratzert et al. (2019a), but with an extended test period. This train/test split was used because the NWM-Rv2 data record is not long enough to accommodate the train/test split used by previous studies (item above in this list).

3. The third train/test period split used all water years in the CAMELS data set with five-year or lower return period peak flow for training, while the test period included water years with greater than five-year return period peak flow in the period 1996-2014 (to be comparable with the test period in the item above). This is to test whether the data-driven models can extrapolate to extreme events that are not included in the training data. Return period calculations are described in Section 3.3.2. To account for the 365-day sequence length for sequence-to-one prediction, we separated all train and test years in each basin by at least one year (i.e., we removed years with high return periods, and their preceding years, from the training set). A file containing the train/test year splits for each CAMELS basin based on return periods is available in the GitHub repository linked in the Code and Data Accessibility statement.

3.3.3.2 Benchmark models & calibration

The conceptual model that we used as a benchmark was the Sacramento Soil Moisture Accounting model (SAC-SMA) with SNOW-17 and a unit hydrograph routing function. This same model was used by (Newman et al., 2017) to provide standardized model benchmarking data as part of the CAMELS data set, however we re-calibrated SAC-SMA to be consistent with our training/test splits that are based on return periods. We used the Python-based SAC-SMA code and calibration package developed by (Nearing et al., 2020a), which uses the SpotPy calibration library (Houska et al., 2019). SAC-SMA was calibrated separately at each of the 531 CAMELS basins using the three train/test splits outlined in Section 3.3.3.1.

The process-based model that we used as a benchmark was the NOAA National Water Model (NWM) retrospective run version 2 (NWM-Rv2). The NWM is based on WRF-Hydro (Salas et al., 2018), which is a process-based model that includes Noah-MP (Niu et al., 2011) as a land surface component, kinematic wave overland flow, and Muskingum-Cunge channel routing. NWM-Rv2 was previously used as a benchmark for LSTM simulations in CAMELS by Kratzert et al. (2019a), Gauch et al. (2021) and Frame et al. (2020). Public data from NWM-Rv2 is hourly and CONUS-wide – we pulled hourly flow estimates from the USGS gauges in the CAMELS data set and averaged these hourly data to daily over the time period October 1, 1980 through September 30, 2014. As a point of comparison, Gauch et al. (2021) compared hourly and daily LSTM predictions against the NWM-Rv2 and found that the NWM-Rv2 was significantly more accurate at the daily timescale than at the hourly timescale, whereas the LSTM did not lose accuracy at the hourly timescale vs. the daily timescale. All experiments in the present study were done at the daily timescale.

The NWM-Rv2 was calibrated by NOAA personnel on about 1400 basins with NLDAS forcing data on water years 2009-2013. Part of our experiment and analysis includes data-driven models trained on irregular years, specifically with water years that include peak flow annual return period less than 5 years, and the calibration of the conceptual model (SAC-SMA) was also done on these years. Without the ability to re-calibrate the NWM-Rv2 on the same time period as the LSTM, MC-LSTM and

SAC-SMA, we cannot directly compare the performance of the NWM-Rv2 with the other models. This model still provides a useful benchmark for the data-driven models, even if it does have a slight advantage over the other models due to the calibration procedure.

3.3.3.3 Performance metrics and assessment

We used the same set of performance metrics that were used in previous CAMELS studies (Kratzert et al., 2019b,a, 2021; Gauch et al., 2021; Klotz et al., 2021). A full list of these metrics is given in Table 3.1. Each of the metrics was calculated for each basin separately on the whole test period for each of the training/test splits described in Section 3.3.3.1 except for the return-period based training/test split. In the former case (contiguous training/test periods) our objective is to maintain continuity with previous studies that report statistics calculated over entire test periods. In the latter case (return-period based training/test splits) our objective is to report statistics separately for different return periods, and it is therefore necessary to calculate separate metrics for each water year and each basin in the test period. The last metric outlined in Table 3.1, the absolute percent bias of peak flow only for the largest streamflow event in each water year, lets us assess the ability to extrapolate to high-flow events. The metric was calculated separately for each annual peak flow event in all three training/test splits.

Table 3.1: Overview of evaluation metrics.

Metric	Description	Reference/Equation	Range of values and best fit
NSE	Nash-Sutcliff efficiency	Eq. 3 in Nash and Sutcliffe (1970)	$(-\infty, 1]$, best: 1.
KGE	Kling-Gupta efficiency	Eq. 9 in Gupta et al. (2009)	$(-\infty, 1]$, best: 1.
Pearson-r	Pearson correlation between observed and simulated flow		$(-\infty, 1]$, best: 1.
α-NSE	Ratio of standard deviations of observed and simulated flow	From Eq. 4 in Gupta et al. (2009)	$(0, \infty)$, best: 1.
β-NSE	Ratio of the means of observed and simulated flow	From Eq. 10 in Gupta et al. (2009)	$(-\infty, \infty)$, best: 0.
FHV	Top 2% peak flow bias	Eq. A3 in Yilmaz et al. (2008)	$(-\infty, \infty)$, best: 0.
FLV	Bottom 30% low flow bias	Eq. A4 in Yilmaz et al. (2008)	$(-\infty, \infty)$, best: 0.
FMS	Bias of the slope of the flow duration curve between the 20% and 80% percentile	Eq. A2 Yilmaz et al. (2008)	$(-\infty, \infty)$, best: 0.
Peak-Timing	Mean peak time lag (in days) between observed and simulated peaks	Appendix B in Kratzert et al. (2021)	$(-\infty, \infty)$, best: 0.
Abs. error peak Q	Absolute percent error of peak flow	$(\frac{\lvert Q_{obs}-Q_{sim}\rvert}{Q_{obs}})$.	$(0, \infty)$, best: 0.

3.4 Results

3.4.1 Benchmarking whole hydrographs

Table 3.2 provides performance metrics for all models (Section 3.3.3.2) on the three test periods (Section 3.3.3.1). Appendix 3.8 provides a breakdown of the metrics in Table 3.2 by annual return period.

The first test period (1989-1999) is the same period used by previous studies, which allows us to confirm that the DL-based models (LSTM and MC-LSTM) trained for this project perform as expected relative to prior work. The performance of these models (according to the metrics) are broadly equivalent to those reported for single models (not ensembles) by Kratzert et al. (2019b) (LSTM) and Hoedt et al. (2021) (MC-LSTM).

The second test period (1995-2014) allows us to benchmark against the NWM-Rv2, which does not provide data prior to 1995. Most of these scores are broadly equivalent to the metrics for the same models reported for the test period 1989-1999, with the exception of the FHV (high flow bias), FLV (low flow bias), add FMS (flow duration curve bias). These metrics depend heavily on the observed flow characteristics during a particular test period and, because they are less stable, are somewhat less useful in terms of drawing general conclusions. We report them here primarily for continuity with previous studies (Kratzert et al., 2019b,a, 2021; Frame et al., 2020; Nearing et al., 2020a; Klotz et al., 2021; Gauch et al., 2021), and because one of the objectives of this paper (Section 3.3.2) is to expand on the high flow (FHV) analysis by benchmarking on annual peak flows.

The third test period (based on return periods) allows us to benchmark only on water years that contain streamflow events that are larger (per basin) than anything seen in the training data ($\leq$ 5-year return periods in training and $>$5-year return periods in testing). Model performances generally improve overall in this period according to the three correlation-based metrics (NSE, KGE, Pearson-r), but degrade according to the variance-based metric (alpha-NSE). This is expected due to the nature of the metrics themselves – hydrology models generally exhibit higher correlation with observations under wet conditions, simply due to higher variability. However, the data-driven models remained better than both benchmark models against all four of these metrics, and while the bias metric (beta-NSE) was less consistent across test periods, the data-driven models had less overall bias than both benchmark models in the return-period test period.

The results in Table 3.2 indicate broadly similar performance between the LSTM and MC-LSTM across most metrics in the two nominal (i.e., unbiased) test periods. However, there were small differences. The MC-LSTM generally performed slightly worse according to most metrics and test periods. The cross-comparison was mixed according to the timing-based metric (Peak-Timing). Notably, differences between the two ML-based models were small compared to the differences between these models and the conceptual (SAC-SMA) and process-based (NWM-Rv2) models, which both performed substantively worse across all metrics except FLV and FMS. The results also indicate that the MC-LSTM performs much worse according to the FLV metric, but we caution that the FLV metric is fragile, particularly when flows approach zero (due to dry or frozen conditions). The large discrepancy comes from several outlier basins that are regionally clustered, mostly, around the south-west. The FLV equation includes a log value of the simulation and observed flows. This causes a very large instability in the calculation. Flow duration curves (and flow duration curve of the minimum 30% of flows) of the LSTM and the MC-LSTM are qualitatively similar, but they diverge on the low flow in terms of log values.

There were clear differences between the physics-constrained (MC-LSTM) and unconstrained (LSTM) data-driven models in the high-return period metrics. While both data-driven models performed better than both benchmark models in these out-of-sample events, adding mass balance constraints resulted in *reduced* performance in the out-of-sample years.

The MC-LSTM includes a flux term that accounts for unobserved sources and sinks (e.g., evapotranspiration, sublimation, percolation). However, it is important to note that most or all hydrology models that are based on closure equations include a residual term in some form. Like all mass balance models, the MC-LSTM explicitly accounts for all water in and across the boundaries of the system. In the case of the MC-LSTM, this residual term is a single, aggregated flux that is parameterized with weights that are *shared* across all 498 basins. Even with this strong constraint, the MC-LSTM performs significantly better than the physically-based benchmark models. This result indicates that classical hydrology model structures (conceptual flux equations) actually cause larger prediction errors than can be explained as being due to errors in the forcing and observation data.

3.4.2 Benchmarking peak-flows

Figure 3.1 shows the average absolute percent bias of annual peak flows for water years with different return periods. The training/calibration period for these results is the contiguous test period (water years 1996-2014). All models had increasingly large average errors with increasingly large extreme events. LSTM average error was lowest in all the return period bins. SAC-SMA was the worst performing model in terms of average error. SAC-SMA was trained (calibrated) on the same data as the LSTM and MC-LSTM, and its performance decreased substantively with increasing return period while that of the LSTM did not.

Figure 3.2 shows the average absolute percent bias of annual peak flows for water years with different return periods, from models with train/test split based on return periods, with all test data coming from water years 1996-2014. This means that Figures 3.1 and 3.2 are only partially comparable – all statistics for each return period bin were calculated on the same observation data. All of the data shown in Figure 3.1 come from the test period. However since all water years with return periods of less than 5 years were used for training in the return-period based train/test split, the 1-5 year return period

Table 3.2: Median performance metrics across 498 basins on two separate time split test periods and test period split by return period (or probability) of the annual peak flow event (i.e., testing across years with an a peak annual event above 5 year return period, or a 20 percent probability of annual exceedance).

	Test period: 1989 - 1999			Test period: 1996 - 2014				Test period: low probability years			
Metric	LSTM	MC-	SAC-	LSTM	MC-	SAC-	NWM-	LSTM	MC-	SAC-	NWM-
		LSTM	SMA		LSTM	SMA	Rv2		LSTM	SMA	Rv2
NSE	0.72	0.71	0.64	0.71	0.72	0.63	0.63	0.81	0.77	0.66	0.67
KGE	0.73	0.73	0.67	0.77	0.74	0.68	0.67	0.77	0.71	0.62	0.64
Pearson-r	0.86	0.86	0.82	0.86	0.86	0.81	0.82	0.91	0.9	0.84	0.85
Alpha-NSE	0.82	0.82	0.79	0.94	0.87	0.83	0.85	0.82	0.77	0.7	0.79
Beta-NSE	-0.04	-0.02	-0.01	0.01	-0.01	-0.01	-0.01	-0.03	-0.04	-0.03	-0.04
FHV	-17.95	-16.76	-19.74	-7.17	-13.1	-15.55	-13.02	-17.37	-24.08	-31.08	-20.42
FLV	-8.37	-33.74	31.18	-9.49	-27.23	28.56	2.85	-2.49	-39.39	27.1	10.81
FMS	-7.28	-8.79	-14.27	-9.67	-8.65	-8.38	-5.23	-6.37	-4.87	-11.29	-4.31
Peak-Timing	0.33	0.33	0.43	0.38	0.4	0.53	0.54	0.36	0.42	0.72	0.62

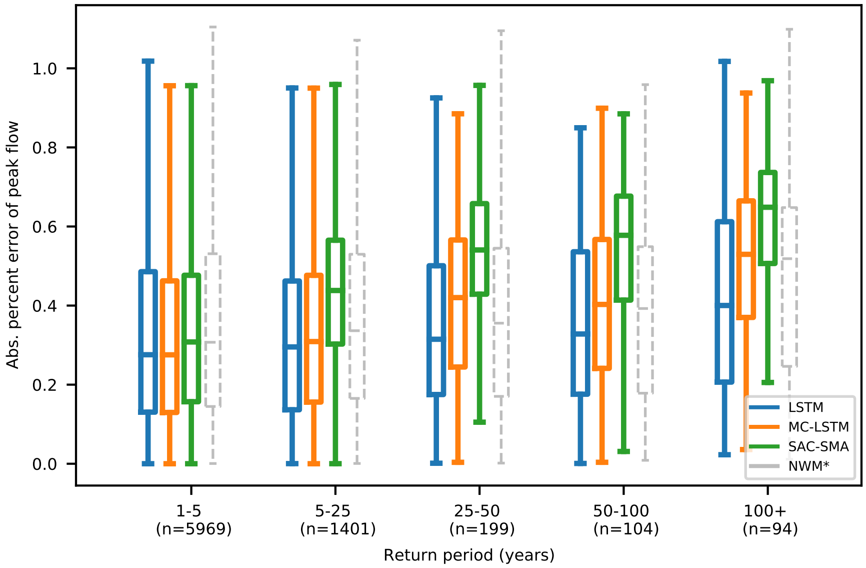

Figure 3.1: Average absolute percent bias of daily peak flow estimates from four models binned down by return period, showing results from models trained on a contiguous time period that contains a mix of different peak annual return periods. All statistics shown are calculated on test period data. The LSTM, MC-LSTM, and SAC-SMA models were all trained (calibrated) on the same data and time period. The NWM was calibrated on with the same forcing data, but on a different time period.

category on Figure 3.2 shows metrics calculated on training data. What is comparable from these two figures are relative trends between models.

For the return-period test (Figure 3.2) the LSTM, MC-LSTM, and SAC-SMA were trained on data from all water years in 1980-2014 with return periods smaller or equal to 5 years, and all of the models showed substantively better average performance in the low return period (high probability) events than in the high return period (low probability) events. SAC-SMA performance deteriorated faster than LSTM and MC-LSTM performance with increasingly extreme events. The unconstrained data-driven model (LSTM) performed better on average than all physics-informed and physically-based models in predicting extreme events in all out-of-sample training cases except for the 25-50 and 50-100, where the NWM-Rv2 performed slightly better on average. However, remember that the NWM-Rv2 calibration data was not segregated by return period.

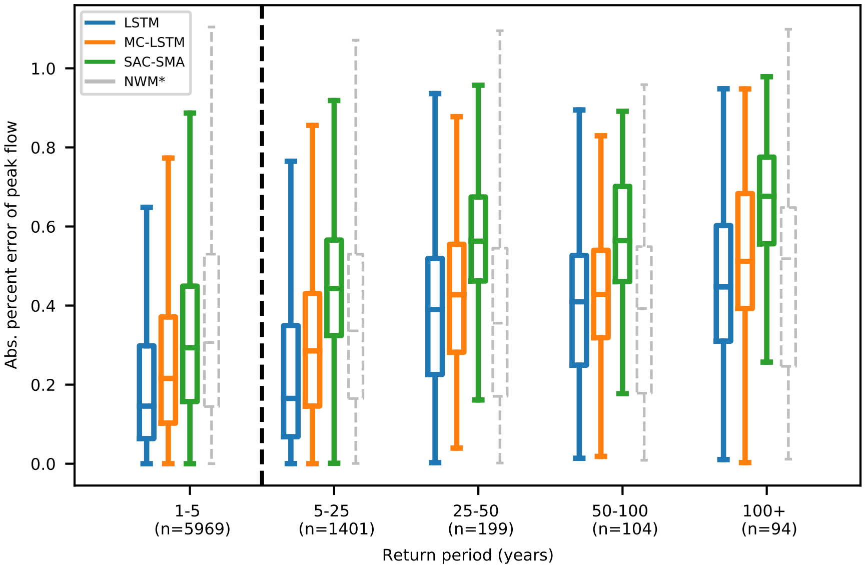

Figure 3.2: Average absolute percent bias of daily peak flow estimates from four models binned down by return period, showing results from models trained only on water years with return periods less than 5 years. The 1-5 year return period bin (left of the black dashed line) show statistics calculated on training data, while bins with return period years 5+ (to the right of the black dashed line) show statistics calculated on testing data. The LSTM, MC-LSTM, and SAC-SMA models were all trained (calibrated) on the same data and time period. The NWM was calibrated on with the same forcing data, but on a contiguous time period that does not exclude extreme events, as described in section 3.3.3.2

3.5 Conclusions & discussion

The hypothesis tested in this work was that predictions made by data-driven streamflow models are likely to become unreliable in extreme or out-of-sample events. This is an important hypothesis to test because it is a common concern among physical scientists and among users of model-based information products (e.g., Todini, 2007), however prior work (e.g., Kratzert et al., 2019b; Gauch et al., 2021) demonstrated that predictions made by data-based rainfall-runoff models were more reliable than other types of physically-based models, even in extrapolation to ungauged basins (Kratzert et al., 2019a). Our results indicate that this hypothesis is incorrect – the data-driven models (both the pure ML model and the physics-informed ML model) were better than benchmark models at predicting peak flows in almost all conditions, including extreme events and including when extreme events were not included in the training data set.

It was somewhat surprising to us that the physics-constrained LSTM did not perform as well as the pure LSTM at simulating peak flows and out-of-sample events. This surprised us for two reasons. First, we expected that adding closure would help in situations where the model sees rainfall events that are larger than anything it had seen during training. In this case, the LSTM could simply ‘forget’ water while the MC-LSTM would have to do something with the excess water – either store it in cell states or release it through one of the output fluxes. Second, Hoedt et al. (2021) reported that the MC-LSTM had lower bias than the LSTM on 98th percentile streamflow events (this is our FHV metric). Our comparison between different training/test periods showed that FHV is a volatile metric, which might account for this discrepancy. The analysis by Hoedt et al. (2021) also did not consider whether a peak flow event was similar or dissimilar to training data, and we saw the greatest differences between the LSTM and MC-LSTM when predicting out-of-sample return period events.

This finding (differences between pure ML and physics-informed ML) is worth discussing. The project of adding physical constraints to ML is an active area of research across most fields of science and engineering (Karniadakis et al., 2021), including hydrology (e.g., Zhao et al., 2019; Jiang et al., 2020; Frame et al., 2020). It is important to understand that there is only one type of situation in which adding any type of constraint (physically-based or otherwise) to a data-driven model can add value: if constraints help optimization. Helping optimization is meant here in a very general sense, which might include processes such as smoothing the loss surface, casting the optimization into a convex problem, restricting the search space, etc. Neural networks (and recurrent neural networks) can emulate large classes of functions (Hornik et al., 1989; Schäfer and Zimmermann, 2007), and by adding constraints to this type of model we can only *restrict* (not expand) the space of possible functions that the network can emulate. This form of regularization is valuable *only* if it helps locate a better (in some general sense) local minimum on the optimization response surface (Mitchell, 1980). And it is *only* in this sense that constraints imposed by physical theory can add information relative to what is available purely from data.

3.6 Appendix: LSTM

Long Short Term Memory networks (Hochreiter and Schmidhuber, 1997) represent time-evolving systems using a recurrent network structure with an explicit state space. Although LSTMs are not based on physical principles, Kratzert et al. (2018) argued that they are useful for rainfall-runoff modeling because they represent dynamic systems in a way that corresponds with physical intuition – specifically, LSTMs are Markovian in the (weak) sense that the future depends on the past only conditionally through the present state and future inputs. This type of temporal dynamics is implemented in an LSTM using an explicit input-state-output relationship that is conceptually similar to most hydrology models.

The LSTM architecture (Figure 3.3) takes a sequence of input features $\boldsymbol{x} = [\boldsymbol{x}[1], ..., \boldsymbol{x}[T]]$ of data over T time steps, where each element $\boldsymbol{x}[t]$ is a vector containing features at time step t. A vector of recurrent *cell states* $\boldsymbol{c}$ is updated based on the input features and current cell state values at time t. The cell states also determine LSTM outputs or hidden states, $\boldsymbol{h}[t]$, which are passed through a *head layer* that combines the LSTM outputs (that are not associated with any physical units) into predictions $\hat{\boldsymbol{y}}[t]$ that attempt to match the target data (which may or may not be associated with physical units).

The LSTM structure (without the head layer) is as follows:

$$\boldsymbol{i}[t] = \sigma(\boldsymbol{W_i x}[t] + \boldsymbol{U_i h}[t-1] + \boldsymbol{b_i}) \tag{3.3}$$

$$\boldsymbol{f}[t] = \sigma(\boldsymbol{W_f x}[t] + \boldsymbol{U_f h}[t-1] + \boldsymbol{b_f}) \tag{3.4}$$

$$\boldsymbol{g}[t] = \tanh(\boldsymbol{W_g x}[t] + \boldsymbol{U_g h}[t-1] + \boldsymbol{b_g}) \tag{3.5}$$

$$\boldsymbol{o}[t] = \sigma(\boldsymbol{W_o x}[t] + \boldsymbol{U_o h}[t-1] + \boldsymbol{b_o}) \tag{3.6}$$

$$\boldsymbol{c}[t] = \boldsymbol{f}[t] \odot \boldsymbol{c}[t-1] + \boldsymbol{i}[t] \odot \boldsymbol{g}[t] \tag{3.7}$$

$$\boldsymbol{h}[t] = \boldsymbol{o}[t] \odot \tanh(\boldsymbol{c}[t]), \tag{3.8}$$

The symbols $\boldsymbol{i}[t]$, $\boldsymbol{f}[t]$ and $\boldsymbol{o}[t]$ refer to the *input gate*, *forget gate*, and *output gate* of the LSTM respectively, $\boldsymbol{g}[t]$ is the *cell input* and $\boldsymbol{x}[t]$ is the *network input* at time step t, $\boldsymbol{h}[t-1]$ is the LSTM output, which is also called the *recurrent input* because it is used as inputs to all gates in the next timestep, and $\boldsymbol{c}[t-1]$ is the cell state from the previous time step.

Cell states represent the memory of the system through time, and are initialized as a vector of zeros. $\sigma(\cdot)$ are sigmoid activation functions, which return values in $[0, 1]$. These sigmoid activation functions in the forget gate, input gate, and output gate are used in a way that is conceptually similar to on/off switches – multiplying anything by values in $[0, 1]$ is a form of attenuation. The forget gate controls the memory timescales of each of the cell states, and the input and output gates control flows of information from the input features to the cell states and from the cell states to the outputs (recurrent inputs), respectively. $\boldsymbol{W}$, $\boldsymbol{U}$ and $\boldsymbol{b}$ are calibrated parameters, where subscripts indicate which gate the particular parameter matrix/vector is associated with. $\tanh(\cdot)$ is the hyperbolic tangent activation function, which serves to add nonlinearity to the model in the cell input and recurrent input, and $\odot$ indicates element-wise multiplication. For a hydrological interpretation of the LSTM, see Kratzert et al. (2018).

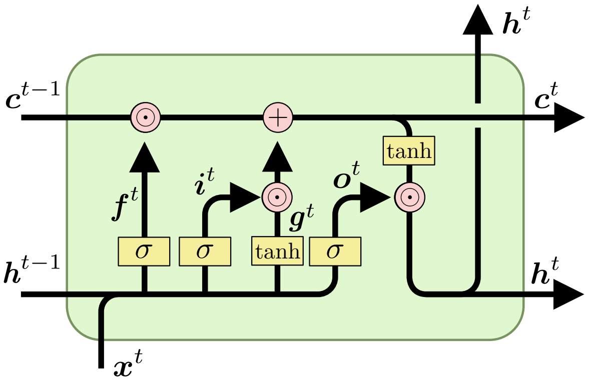

Figure 3.3: A single timestep of a standard LSTM with timesteps marked as superscripts for clarity. $\boldsymbol{x}^t$, $\boldsymbol{c}^t$, and $\boldsymbol{h}^t$ are the input features, cell states, and recurrent inputs at time t, respectively. $\boldsymbol{f}^t$, $\boldsymbol{i}^t$, and $\boldsymbol{o}^t$ are the forget-, input- and output-gate and $\boldsymbol{g}^t$ denotes the cell input. Boxes labeled σ and tanh represent single sigmoid and hyperbolic tangent activation layers with the same number of nodes as cell states. The addition sign represent element-wise addition and $\odot$ represents element-wise multiplication.

3.7 Appendix: Mass conserving LSTM

The LSTM has an explicit input-state-output structure that is recurrent in time and is conceptually similar to how physical scientists often model dynamical systems. However the LSTM does not obey physical principles, and the internal cell states have no physical units. We can leverage this input-state-output structure to enforce mass conservation, in a manner that is similar to discrete-time explicit integration of a dynamical systems model, as follows:

$$\textit{New States} = \textit{Old States} + \textit{Inputs} - \textit{Outputs}. \tag{3.9}$$

Using the notation from Appendix 3.6, this is:

$$\boldsymbol{c}^*[t] = \boldsymbol{c}^*[t-1] + \boldsymbol{x}^*[t] - \boldsymbol{h}^*[t], \tag{3.10}$$

where $\boldsymbol{c}^*[t]$, $\boldsymbol{x}^*[t]$ and $\boldsymbol{h}^*[t]$ are components of the cell states, input features, and model outputs (recurrent inputs) that contribute to a particular conservation law.

As presented by Hoedt et al. (2021), we can enforce conservation in the LSTM by doing two things. First, we use special activation functions in some of the gates to guarantee that mass is conserved from the inputs and previous cell states. Second, we subtract the outgoing mass from the cell states. The important property of the special activation functions is that the sum of all elements sum to one. This allows the outputs of each activation node to be scaled by a quantity that we want to conserve, so that each scaled activation value represents a fraction of that conserved quantity. In practice, we can use any standard activation function (e.g., sigmoid, ReLU), as long as we normalize the activation. With positive activation functions we can, for example, normalize by the L1 norm (see Eq. 3.11 and 3.12). Another option would be to use the softmax activation function, which sums to one by definition.

$$\widehat{\sigma}(s_k) = \frac{\sigma(s_k)}{\sum_k \sigma(s_k)} \tag{3.11}$$

$$\widehat{\mathrm{ReLU}}(s_k) = \frac{\max(s_k, 0)}{\sum_k \max(s_k, 0)} \tag{3.12}$$

The constrained model architecture is illustrated in Fig. 3.4. An important difference with the standard architecture is that the inputs are separated into *mass inputs* $\boldsymbol{x}$ and *auxiliary inputs* a. In our case, the mass input is precipitation and the auxiliary inputs are everything else (e.g. temperature, radiation, catchment attributes). The input gate (sigmoids) and cell input (hyperbolic tangents) in the standard LSTM are (collectively) replaced by one of these normalization layers, while the output gate is a standard sigmoid gate, similar to the standard LSTM. The forget gate is also replaced by a normalization layer, with the important difference that the output of this layer is a square matrix with dimension equal to the size of the cell state. This matrix is used to "reshuffle" the mass between the cell states at each timestep. This *reshuffling matrix* is column-wise normalized so that the dot product with the cell state vector at time t results in a new cell state vector having the same absolute norm (so that no mass is lost or gained).

We call this general architecture a *Mass-Conserving LSTM* (MC-LSTM), even though it works for any type of conservation law (mass, energy, momentum, counts, etc.). The architecture is illustrated in Figure 3.4 and is described formally as follows:

$$\hat{\boldsymbol{c}}[t-1] = \frac{\boldsymbol{c}[t-1]}{||\boldsymbol{c}[t-1]||_1} \tag{3.13}$$

$$\boldsymbol{i}[t] = \widehat{\sigma}(\boldsymbol{W}_i\boldsymbol{x}[t] + \boldsymbol{U}_i\hat{\boldsymbol{c}}[t-1] + \boldsymbol{V}_i\boldsymbol{a}[t] + \boldsymbol{b}_i) \tag{3.14}$$

$$\boldsymbol{o}[t] = \sigma(\boldsymbol{W}_o\boldsymbol{x}[t] + \boldsymbol{U}_o\hat{\boldsymbol{c}}[t-1] + \boldsymbol{V}_o\boldsymbol{a}[t] + \boldsymbol{b_o}) \tag{3.15}$$

$$\boldsymbol{R}[t] = \widehat{\mathrm{ReLU}}(\mathbf{W}_R\boldsymbol{x}[t] + \mathbf{U}_R\hat{\boldsymbol{c}}[t-1] + \mathbf{V}_R\boldsymbol{a}[t] + \boldsymbol{b}_R) \tag{3.16}$$

$$\boldsymbol{m}[t] = \boldsymbol{R}[t]\boldsymbol{c}[t-1] + \boldsymbol{i}[t]\boldsymbol{x}[t] \tag{3.17}$$

$$\boldsymbol{c}[t] = (1 - \boldsymbol{o}[t]) \odot \boldsymbol{m}[t] \tag{3.18}$$

$$\boldsymbol{h}[t] = \boldsymbol{o}[t] \odot \boldsymbol{m}[t] \tag{3.19}$$

Learned parameters are $\boldsymbol{W}$, $\boldsymbol{U}$, $\boldsymbol{V}$, and $\boldsymbol{b}$ for all of the gates. The normalized activation functions are, in this case, $\widehat{\sigma}$ (see Eq. 3.11) for the input gate and $\widehat{\mathrm{ReLU}}$ (see Eq. 3.12) for the redistribution matrix $\boldsymbol{R}$, as in the hydrology example of Hoedt et al. (2021). The product of $\boldsymbol{i}[t]\boldsymbol{x}[t]$ and $\boldsymbol{o}[t] \odot \boldsymbol{m}[t]$ are input and output fluxes, respectively.

Because this model structure is fundamentally conservative, all cell states and information transfers within the model are associated with physical units. Our objective in this study was to maintain the overall water balance in a catchment – our conserved input feature, $\boldsymbol{x}$, is precipitation in units $[mm/day]$ and our training targets are catchment discharge also in units of $[mm/day]$. Thus, all input fluxes, output fluxes, and cell states in the MC-LSTM have units of $[mm/day]$.

In reality, precipitation and streamflow are not the only fluxes of water into or out of a catchment. Because we did not provide the model with (for example) observations of evapotranspiration, aquifer recharge, or baseflow, we accounted for unobserved sinks in the modeled systems by allowing the model to use one cell state as a *trash cell.* The output of this cell is ignored when we derive the final model prediction as the sum of the outgoing mass $\sum \boldsymbol{h}$.

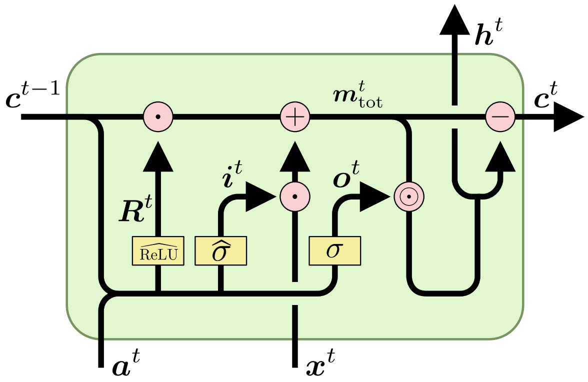

Figure 3.4: A single timestep of a Mass-Conserving LSTM with timesteps marked as superscripts for clarity. As in Figure 3.3, $\boldsymbol{c}^t$, $\boldsymbol{a}^t$, $\boldsymbol{x}^t$, $\boldsymbol{i}^t$, $\boldsymbol{o}^t$, and $\boldsymbol{R}^t$ are the cell states, conserved inputs, input features, input fluxes, output fluxes, and reshuffling matrix at time t, respectively. σ represents a standard sigmoid activation layer, $\widehat{\sigma}$ and $\widehat{\text{ReLU}}$ represent normalized sigmoid activation layers and normalized ReLU activation layer respectively. Addition and subtraction signs represent element-wise addition and subtraction, $\odot$ represents element-wise multiplication and the $\cdot$ sign represents the dot-product.

3.8 Appendix: Benchmarking annual return period metrics

Figure 3.5 shows nine performance metrics calculated on model test results split into bins according to the return period of the peak annual flow event. The LSTM, MC-LSTM and SAC-SMA were calibrated/trained on water years 1981-1995. The results shown in this figure are for water years 1996-2014. The LSTM and MC-LSTM performs better than the benchmark models according to most metrics, and during most return period bins. There are a few instances where the NWM performs better than the LSTM and/or the MC-LSTM. The NWM calibration does not correspond to the training/calibration period of SAC-SMA, LSTM or the MC-LSTM.

Figure 3.6 shows the nine performance metrics calculated on model test results split into bins according to the return period of the peak annual flow event. The LSTM,

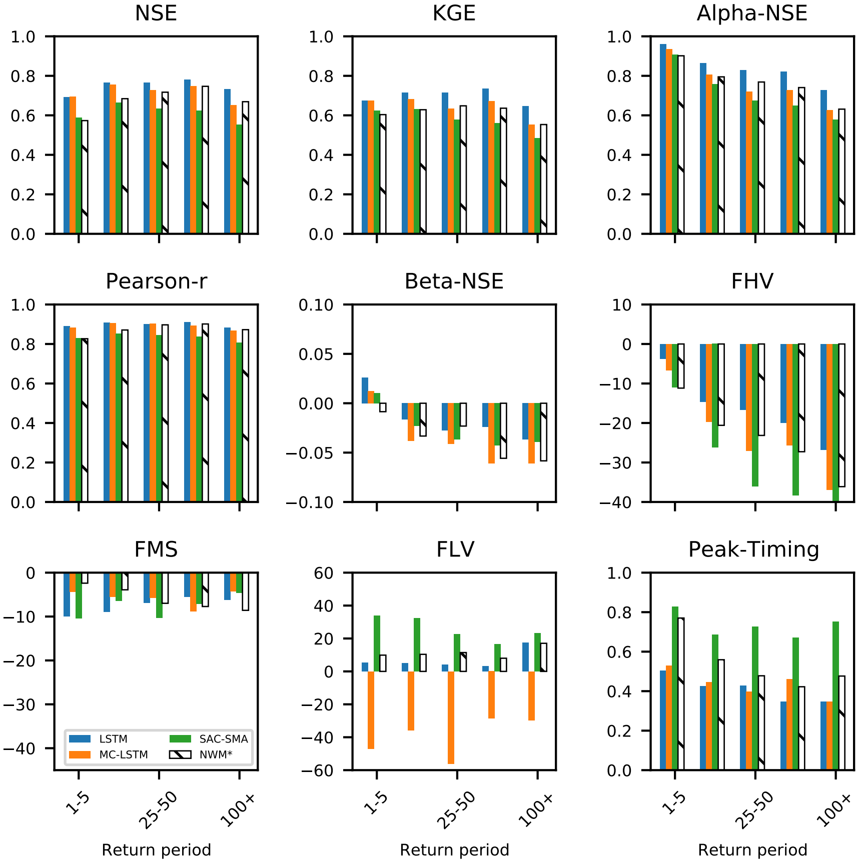

Figure 3.5: Metrics for training only on a standard time split; train period was water years 1981-1995 and test period (shown here) was water years 1996-2014. The total number of samples in each bin are as follows: n=5969 for 1-5, n=1260 for 5025, n=185 for 25-50, n=91 for 50-100 and n=84 for 100+.

MC-LSTM and SAC-SMA were calibrated/trained on water years with a peak annual flow event that had a return period of less that five years (i.e., bin 1-5 indicated by the dashed line). The results shown in this figure are for water years 1996-2014. The LSTM and MC-LSTM performs better than the SAC-SMA model according every metric, and during all bins. There are a few instances where the NWM performs better than the LSTM and/or the MC-LSTM. The NWM calibration does not correspond to the training/calibration period of SAC-SMA, LSTM or the MC-LSTM.

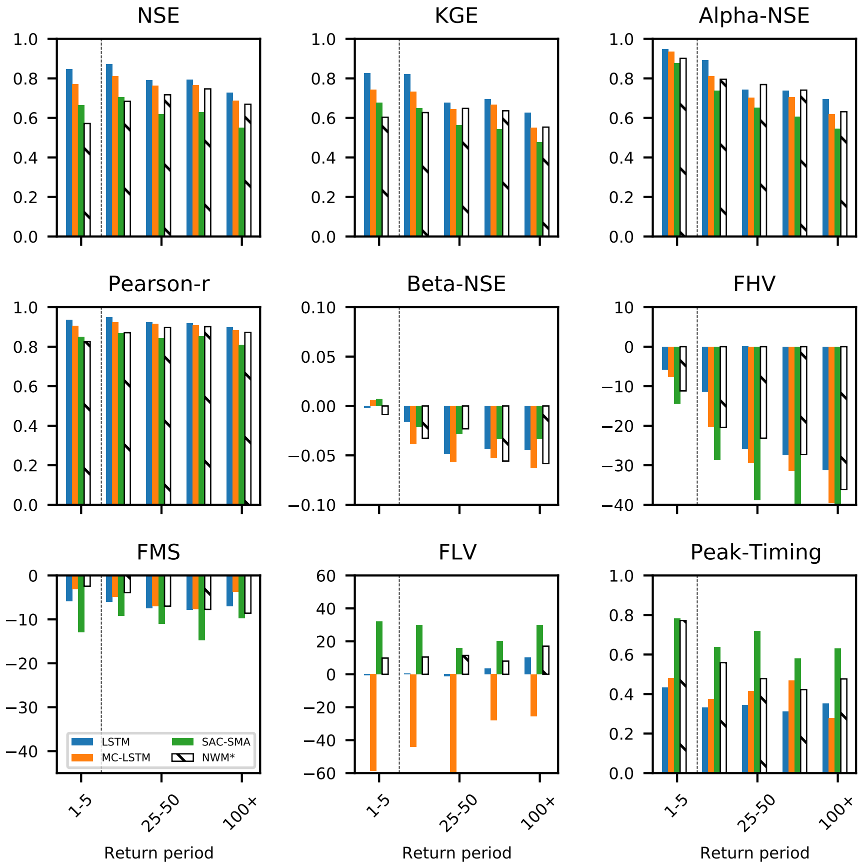

Figure 3.6: Metrics for the models trained only on high-probability years. The bins of return periods greater than 5 are out-of-sample for the LSTM, MC-LSTM and SAC-SMA. The total number of samples in each bin are as follows: n=5969 for 1-5, n=1260 for 5025, n=185 for 25-50, n=91 for 50-100 and n=84 for 100+.

3.9 Code and data availability

All LSTMs and MC-LSTMs were trained using the NeuralHydrology Python library available at https://github.com/neuralhydrology/neuralhydrology. A snapshot of the exact version that we used is available at https://github.com/jmframe/mclstm_2021_extrapolate/neuralhydrology and under DOI number 10.5281/zenodo.5051961. Code for calibrating SAC-SMA is from https://github.com/Upstream-Tech/SACSMA-SNOW17, which includes the SpotPy calibration library https://pypi.org/project/spotpy/. Input data for all model runs except the NWM-Rv2 came from the public NCAR CAMLES repository https://ral.ucar.edu/solutions/products/camels and were used according to instructions outlined in the NeuralHydrology readme. NWM-Rv2 data are available publicly from https://registry.opendata.aws/nwm-archive/. Code for the return period calculations is publicly available from https://www.mathworks.com/matlabcentral/fileexchange/22628-log-pearson-flood-flow-frequency-using-usgs-17b (Burkey, 2009), and daily USGS peak flow data extracted from the USGS Water Information System for the CAMELS return period analysis were collected and archived on the CUAHSI HydroShare platform under DOI number 10.4211/hs.c7739f47e2ca4a92989ec34b7a2e78dd. All model output data generated by this project will be available on the CUAHSI HydroShare platform under a DOI number https://doi.org/10.4211/hs.d750278db868447dbd252a8c5431affd. Interactive Python scripts for all post-hoc analysis reported in this paper, including calculating metrics and generating tables and figures, are available at https://github.com/jmframe/mclstm_2021_extrapolate and under DOI number 10.5281/zenodo.5165216.

REFERENCES

Addor, N., Newman, A. J., Mizukami, N., and Clark, M. P. (2017). The camels data set: catchment attributes and meteorology for large-sample studies. *Hydrology and Earth System Sciences (HESS)*, 21(10):5293–5313.

Burkey, J. (2009). Log-pearson flood flow frequency using usgs 17b.

Cameron, D., Kneale, P., and See, L. (2002). An evaluation of a traditional and a neural net modelling approach to flood forecasting for an upland catchment. *Hydrological Processes*, 16(5):1033–1046.

Frame, J., Nearing, G., Kratzert, F., and Rahman, M. (2020). Post processing the us national water model with a long short-term memory network.

Gauch, M., Kratzert, F., Klotz, D., Nearing, G., Lin, J., and 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.

Gaume, E. and Gosset, R. (2003). Over-parameterisation, a major obstacle to the use of artificial neural networks in hydrology? *Hydrology and Earth System Sciences*, 7(5):693–706.

Gupta, H. V., Kling, H., Yilmaz, K. K., and 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.

Herath, H. M. V. V., Chadalawada, J., and Babovic, V. (2020). Hydrologically Informed Machine Learning for Rainfall-Runoff Modelling: Towards Distributed Modelling. *Hydrology and Earth System Sciences Discussions*, (October):1–42.

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

Hoedt, P.-J., Kratzert, F., Klotz, D., Halmich, C., Holzleitner, M., Nearing, G. S., Hochreiter, S., and Klambauer, G. (2021). Mc-lstm: Mass-conserving lstm. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 4275–4286. PMLR.

Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366.

Houska, T., Kraft, P., Chamorro-Chavez, A., and Breuer, L. (2019). Spotpy: A python library for the calibration, sensitivity-and uncertainty analysis of earth system models. In *Geophysical Research Abstracts*, volume 21.

IACWD (1982). Guidelines for determining flood flow frequency: Bulletin 17b. Technical report, Washington, D.C.

Jiang, S., Zheng, Y., and Solomatine, D. (2020). Improving ai system awareness of geoscience knowledge: Symbiotic integration of physical approaches and deep learning. *Geophysical Research Letters*, 47(13):e2020GL088229.

Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., and Yang, L. (2021). Physics-informed machine learning. *Nature Reviews Physics*, (May).

Klotz, D., Kratzert, F., Gauch, M., Keefe Sampson, A., Brandstetter, J., Klambauer, G., Hochreiter, S., and Nearing, G. (2021). Uncertainty estimation with deep learning for rainfall–runoff modelling. *Hydrology and Earth System Sciences Discussions*, pages 1–32.

Kratzert, F., Klotz, D., Brenner, C., Schulz, K., and 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., and Nearing, G. S. (2019a). Toward improved predictions in ungauged basins: Exploiting the power of machine learning. *Water Resources Research*, 55(12):11344–11354.

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

Kratzert, F., Klotz, D., Shalev, G., Klambauer, G., Hochreiter, S., and Nearing, G. (2019b). 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.

Mitchell, T. M. (1980). *The need for biases in learning generalizations.* Department of Computer Science, Laboratory for Computer Science Research ....

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

Nearing, G., Pelissier, C., Kratzert, F., Klotz, D., Gupta, H., Frame, j., and Sampson, A. (2019). Physically informed machine learning for hydrological modeling under climate nonstationarity. *44th NOAA Annual Climate Diagnostics and Prediction Workshop.*

Nearing, G., Sampson, A. K., Kratzert, F., and Frame, J. (2020a). Post-processing a conceptual rainfall-runoff model with an lstm.

Nearing, G. S., Kratzert, F., Sampson, A. K., Pelissier, C. S., Klotz, D., Frame, J. M., Prieto, C., and Gupta, H. V. (2020b). What role does hydrological science play in the age of machine learning? *Water Resources Research*, page e2020WR028091.

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.

Newman, A. J., Mizukami, N., Clark, M. P., Wood, A. W., Nijssen, B., and Nearing, G. (2017). Benchmarking of a physically based hydrologic model. *Journal of Hydrometeorology*, 18(8):2215–2225.

Niu, G.-Y., Yang, Z.-L., Mitchell, K. E., Chen, F., Ek, M. B., Barlage, M., Kumar, A., Manning, K., Niyogi, D., Rosero, E., et al. (2011). The community noah land surface model with multiparameterization options (noah-mp): 1. model description and evaluation with local-scale measurements. *Journal of Geophysical Research: Atmospheres*, 116(D12).

Rasp, S., Pritchard, M. S., and Gentine, P. (2018). Deep learning to represent subgrid processes in climate models. *Proceedings of the National Academy of Sciences of the United States of America*, 115(39):9684–9689.

Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., et al. (2019). Deep learning and process understanding for data-driven earth system science. *Nature*, 566(7743):195–204.

Salas, F. R., Somos-Valenzuela, M. A., Dugger, A., Maidment, D. R., Gochis, D. J., David, C. H., Yu, W., Ding, D., Clark, E. P., and Noman, N. (2018). Towards real-time continental scale streamflow simulation in continuous and discrete space. *JAWRA Journal of the American Water Resources Association*, 54(1):7–27.

Schäfer, A. M. and Zimmermann, H.-G. (2007). Recurrent neural networks are universal approximators. *International journal of neural systems*, 17(04):253–263.

Sellars, S. (2018). “grand challenges” in big data and the earth sciences. *Bulletin of the American Meteorological Society*, 99(6):ES95–ES98.

Todini, E. (2007). Hydrological catchment modelling: past, present and future. *Hydrology and Earth System Sciences*, 11(1):468–482.

Yilmaz, K. K., Gupta, H. V., and Wagener, T. (2008). A process-based diagnostic approach to model evaluation: Application to the nws distributed hydrologic model. *Water Resources Research*, 44(9).

Zhao, W. L., Gentine, P., Reichstein, M., Zhang, Y., Zhou, S., Wen, Y., Lin, C., Li, X., and Qiu, G. Y. (2019). Physics-constrained machine learning of evapotranspiration. *Geophysical Research Letters*, 46(24):14496–14507.

CHAPTER 4

ON STRICTLY ENFORCED MASS CONSERVATION CONSTRAINTS FOR MODELING THE RAINFALL RUNOFF PROCESS

Jonathan M. Frame, Frederik Kratzert, Hoshin V. Gupta, Paul Ullrich and Grey S.Nearing

In review for publication in Hydrologic Processes

https://doi.org/10.31223/X5BH0P

4.1 Abstract

It has been proposed that conservation laws might not be beneficial for accurate hydrological modeling due to errors in input (precipitation) and target (streamflow) data (particularly at the event time scale), and this might explain why deep learning models (which are not based on enforcing closure) can out-perform catchment-scale conceptual models at predicting streamflow. We test this hypothesis at the event and multi-year time scale using physics-informed (mass conserving) machine learning and find that: (1) enforcing closure in the rainfall-runoff mass balance does appear to harm the overall skill of hydrological models, (2) deep learning models learn to account for spatiotemporally variable biases in data, however (3) this "closure" effect accounts for only a small fraction of the difference in predictive skill between deep learning and conceptual models.

4.2 Introduction

Deep learning (DL) models are becoming the standard benchmark for predictive hydrologic modeling in the current literature because of their high accuracy relative to conceptual models (Nearing et al., 2020c), as well as their ability to extrapolate to new locations (Kratzert et al., 2019a) and extreme events (Frame et al., 2022). There has been a recent push to combine deep learning with physical theory to (i) gain better process understanding, and (ii) improve predictive accuracy, especially under out-of-sample conditions (Jia et al., 2020; Reichstein et al., 2019; Shen et al., 2021; Willard et al., 2021). There have been several recent attempts to build hybrid PB-DL models (sometimes referred to "physics-informed" or "theory-guided", e.g., Bennett and Nijssen, 2021; Karniadakis et al., 2021; Pelissier et al., 2019; Daw et al., 2020; Tsai et al., 2020; Zhao et al., 2019; Jiang et al., 2020; Xie et al., 2021; Hoedt et al., 2021; Nearing et al., 2020a). We therefore think it is important to take a step back and explore if (and what) basic components of physical theory might actually be beneficial for hydrologic prediction.

In this paper we test one hypothesis in particular: we use physics-informed machine learning to explore the longstanding assumption that mass conservation should be the foundation of hydrological models. The first physical law introduced formally by Chow et al. (1988, equation 1.3.5) (a standard introductory hydrology textbook) is:

$$dS/dt = I(t) - Q(t), \tag{4.1}$$

where the change of a system's mass storage (S) with respect to time (t) is equal to total mass input (I) minus total mass output (Q). This is the first physical constraint placed on the transfer function between inputs and output of a hydrological system (i.e., Chow et al., 1988, equation 1.3.1).

While conservation laws are considered to be a fundamental truth about (classical scale) systems in our physical world, it is not necessarily the case that this makes them a proper or useful foundation for either understanding or modeling watershed systems. This distinction is motivated by Beven (2020), who proposed that the closure problem might explain the *poor* performance of conceptual and physically-based (PB) hydrology models relative to DL: *"given the epistemic uncertainties in water and energy balances, then this* [*conservation constraints*] *might not necessarily be advantageous in obtaining better DL* *predictions if, for example, the observational data do not themselves provide consistent mass and energy balance closure".* In other words, conceptual and PB models typically demand a degree of closure that may not necessarily be achievable given sparse and error-prone observation data, and (Beven hypothesized that) the superior performance of DL might be due to its ability to learn and account for consistent error structures present in the input–output data. In practice, PB models sometimes account for error prone data with pre- and post-processing, as well as data assimilation, however pre- and post-processing is not necessary when using DL, as these steps can be learned directly from training data (e.g., Frame et al., 2021).

The proposal explains poor rainfall-runoff model calibration and performance as being a consequence of so-called "disinformation" in data (e.g., Beven et al., 2008; Beven and Westerberg, 2011; Sivapalan et al., 2003). In addition to the observational uncertainty present in data used for driving and evaluating models, there is also uncertainty regarding what actually constitute the true physical inputs and losses from a hydrologic system – for example, mass contributions to the system through natural springs and anthropogenic water resources can come from outside of the watershed "boundary" and are not often directly observable or represented in the available data set. Beven's hypothesis is that these types of effects might explain the relative accuracy of DL streamflow models, due to their not being constrained to conserve mass.

In this paper, we place a bound on this "closure" effect (i.e., on the information loss due to enforcing closure over error-prone data), and show two things:

1. DL is able to learn and account for systematic (but spatiotemporally dynamic) errors in data, and
2. the closure effect does not explain the majority of the performance gap between PB models and DL models of streamflow.

The Long Short-Term Memory (LSTM) was chosen as the deep learning architecture for this study because 1) it is the best performing deep learning model for the rainfall-runoff process, and 2) because we have a directly comparable mass conserving, and non-mass conserving version available (MC-LSTM). We test the ability and performance of the mass conserving MC-LSTM, a DL model with an architecture designed to strictly enforce mass conservation at every timestep, to performing its namesake task (mass conservation), by assessing the long-term bias of predicted runoff in a large-sample dataset (Gupta and Nearing, 2014) and event-based runoff coefficients (Beven, 2019). To be clear, we are not questioning whether hydrologic processes in the real world are governed by the physical concept of mass conservation. What we are questioning is whether a testable, scale-relevant theory of watersheds should be based on this principle. Alternatively, it is possible that no successful scale-relevant theory of watersheds has been developed to-date because the fundamental conceptual basis for a "watershed" is itself incorrect; the typical "fixed catchment control-volume" represented by our watershed delineations with closed internal states cannot represent mesoscale storm-system scales and groundwater aquifer scales.

4.3 Methods

We designed an experiment to test the hypothesis proposed by Beven (2020) that the lack of mass conservation in DL rainfall-runoff models explains the difference in skill relative to PB models that are constrained by closure. The basic experiment is as follows. We use two meteorological data sets, one with a large, nonlinear, and location-specific bias and one without such a bias, to benchmark three models: (i) a standard PB model calibrated per-basin, (ii) a standard DL model trained regionally (over all basins, not per-basin), and (iii) a physics-informed DL model that is constrained to enforce mass conservation trained regionally (in the same way as the standard DL model). Our goal is to understand how much of the difference in skill between the PB and DL models can be accounted for by forcing closure on biased data.

4.3.1 Data

Several recent modeling studies used open community data sets and consistent training/test procedures that allow for results to be directly comparable (Kratzert et al., 2019b, 2021; Klotz et al., 2021; Gauch et al., 2021b,a; Newman et al., 2017; Frame et al., 2021, 2022). We continue that practice here. Specifically, we used the Catchment Attributes and Meteorological Large Sample (CAMELS) data set curated by the US National Center for Atmospheric Research (NCAR) (Newman et al., 2015; Addor et al., 2017). The CAMELS data set consists of daily meteorological and discharge data from 671 catchments in CONUS ranging in size from 4 km^2 to 25,000 km^2 that have largely natural flows and long streamflow gauge records (1980-2008). Newman et al. (2017) developed CAMELS as a data set for community model benchmarking and by excluding basins with (i) large discrepancies between different methods of calculating catchment area, and (ii) areas larger than 2,000 km^2. This results in the large-sample (Gupta and Nearing, 2014) data set with 531 basins that has been used by all of the benchmarking studies cited above. In the current study, we had to omit one of these 531 basins due to a data constraint that will be explained below in Section 4.3.3.

CAMELS includes daily discharge data from the USGS Water Information System, which are used as training and evaluation targets. CAMELS also includes multiple daily meteorological forcing data products that are used as model inputs, shown in Table 4.1. CAMELS also includes several static catchment attributes related to soils, climate, vegetation, topography, and geology (Addor et al., 2017) that are used as input features to the DL models. We used the same input features (meteorological forcings and static catchment attributes) that are listed in Table 1 by Kratzert et al. (2019b).

We used Daymet and NLDAS for this project. The reason that we used these two meteorological forcing data sets is because Daymet exhibits a large positive mass bias in the Eastern US relative to USGS streamflow data, while NLDAS does not. This bias can be seen clearly in some of our results presented in Section 4.4.2. In that section we describe a regional analysis of the total cumulative streamflow bias grouped for different regions of CONUS. The regions were delineated according the United States Geological Survey (USGS) Water Resources Regions outlined in Water-Supply Paper 2294 (USGS, 1987). This includes 18 distinct regions, but only 17 of which have enough CAMELS basins for meaningful statistics (leaving out Souris-Red-Rainy, hydrologic unit code 09).

Table 4.1: Forcing products from the CAMELS dataset

Forcing product	Description	Citation
NLDAS	North American Land Data Assimilation System. Spatial resolution is 1/8th-degree, and the temporal resolution is hourly. The data span 1979 to present. Data can be downloaded in their native GRIB format, but CAMELS provides basin averages. This product is oriented toward land/hydrology modeling. The non-precipitation land-surface forcing fields are derived from the analysis fields of a North American Regional Reanalysis (NARR). Surface pressure, longwave radiation, air temperature and specific humidity are adjusted vertically to account for terrain height.	Xia et al. (2012)
Daymet	Daily Surface Weather Data for North America. Spatial resolution is 1-km x 1-km in Lambert Conformal Conic projection. The data span 1980 through 2015. Data can be downloaded in their native netCDF file formats, but CAMELS provides basin averages. Several of the variables are derived from selected meteorological station data by interpolation and extrapolation algorithms. Data are assembled by parameter and year with each yearly file containing a time dimension of 365 days.	Thornton et al. (2014)

4.3.2 Models

4.3.2.1 Models inspired by physical concepts

The conceptual model that we used as a benchmark was the Sacramento Soil Moisture Accounting model (SAC-SMA) with SNOW-17 and a unit hydrograph routing function. This is the model used by (Newman et al., 2017) as a basis for standardized benchmarking with the CAMELS data set, however we re-calibrated SAC-SMA to be consistent with our training/test splits. We used the Python-based SAC-SMA code and calibration package developed by (Nearing et al., 2020b), which uses the SpotPy calibration library (Houska et al., 2019). We use the Dynamically Dimensioned Search algorithm with ten thousand model runs. SAC-SMA was calibrated separately at each of the 531 CAMELS basins using the three train/test splits outlined in Section 4.3.2.3, and get results comparable to (Newman et al., 2017).

We also benchmarked the U.S. National Water Model (NWM) using the NOAA National Water Model CONUS Retrospective Dataset (https://registry.opendata.aws/nwm-archive/ accessed December 2021). The NWM is based on physics-inspired equations, but it has been argued that these types of models are still conceptual in nature, but applied to the grid scale (Beven, 1989), so we will refer to all non-DL models as conceptual. We present the results of the NWM benchmarks in Appendix 4.9, rather than the main body of this paper because (i) the NWM is only available for NLDAS forcing, and (ii) we are not able to calibrate the NWM to match our other models, so the NWM results aren't directly comparable. A complete description of the NWM is provided in Appendix 4.9 along with a complete set of figures.

4.3.2.2 Deep learning models

The Long Short Term Memory (LSTM) network is the current state-of-the-art model for predicting streamflow at the watershed scale. The LSTM is a recurrent neural network with an explicit state space, and explicit controls on input-state and state-output relationships, as well as explicit controls on memory timescales, which makes it suitable for at least many dynamical systems applications. The LSTM does not enforce conservation laws, which means that there is potential for predicted runoff to violate Equation 4.1.

The Mass-Conserving LSTM (MC-LSTM) is also a recurrent neural network with an explicit state space and explicit input-state and state-output relationships. The internal calculations of the MC-LSTM ensure mass-conservation between any number of inputs (here precipitation) and outputs (here streamflow). In reality, precipitation and streamflow are not the only fluxes of water into or out of a catchment. The MC-LSTM accounts for unobserved sinks (e.g., evapotranspiration, aquifer recharge and anthropogenic water resources) using a subset of cell states to accumulate mass that does not translate to streamflow.

Both the LSTM and MC-LSTM use the same forcing variables, but the MC-LSTM distinguishes between mass inputs (with a specific unit of mass to conserve through the modele) and auxiliary (no enforced conservation enforced) forcing inputs, shown in Table 4.2. Appendixes 4.7 and 4.8 provide additional details of the LSTM and MC-LSTM, respectively.

Table 4.2: Forcing variables for LSTM and MC-LSTM

Forcing variable	Role in MC-LSTM (unit)
Average daily precipitation	Mass conserving (mm)
Daily maximum air temperature	Auxiliary
Daily minimum air temperature	Auxiliary
Solar radiation	Auxiliary
Vapor pressure	Auxiliary

4.3.2.3 Training

We used daily meteorological forcing data and static catchment attributes data as inputs features for the LSTM and MC-LSTM, and we used daily streamflow records as training targets with a normalized squared-error (NSE*) loss function that does not depend on basin-specific mean discharge (i.e., large and/or wet basins are not over-weighted in the loss function):

$$\text{NSE*} = \frac{1}{B}\sum_{b=1}^{B}\sum_{n=1}^{N}\frac{(\widehat{y}_n - y_n)^2}{(s(b)+\epsilon)^2}, \tag{4.2}$$

where B is the number of basins, N is the number of samples (days) per basin B, $\widehat{y}_n$ is the prediction for sample n ($1 \leq n \leq N$), y_n is the corresponding observation, $s(b)$ is the standard deviation of the discharge in basin b ($1 \leq b \leq B$), and ϵ is a small constant for numerical stability (we used 0.1), calculated from the training period (see Kratzert et al., 2019b).

We trained both the standard LSTM and the MC-LSTM using the same training and test procedures outlined by Kratzert et al. (2019b). Both models were trained for 30 epochs using sequence-to-one prediction to allow for randomized, small minibatches. We used a minibatch size of 256 and, due to sequence-to-one training, each minibatch contained (randomly selected) samples from multiple basins. The standard LSTM had 128 cell states and a 365-day sequence length. Input and target features for the standard LSTM were pre-normalized by removing bias and scaling by variance. For the MC-LSTM the inputs were split between auxiliary, which were pre-normalized, and the mass input (in our case precipitation), which was not pre-normalized. Gradients were clipped to a global norm (per minibatch) of 1. Heteroscedastic noise was added to training targets (resampled at each minibatch) with standard deviation of 0.005 times the value of each target datum. We used an Adam optimizer with a fixed learning rate schedule; the initial learning rate of 1e-3 was decreased to 5e-4 after 10 epochs and 1e-4 after 25 epochs. Biases of the LSTM forget gate were initialized to 3 so that gradient signals persisted through the sequence from early epochs. The MC-LSTM used the same hyperparameters as the LSTM except that it used only 64 cell states, which was found to perform better for this model (see, Hoedt et al., 2021). Note that the memory states in an MC-LSTM are fundamentally different than those of the LSTM due to the fact that they are physical states with physical units instead of purely information states.

Both the LSTM and MC-LSTM were trained on data from 531 CAMELS catchments simultaneously. The train/test period split was the same split used in previous studies (Kratzert et al., 2019b, 2021; Hoedt et al., 2021). In this case, the training period included nine water years from October 1, 1999 through September 30, 2008, and the test period included ten water years 1990-1999 (i.e., from October 1, 1989 through September 30, 1999). This train/test split was used *only* to ensure that the models trained here achieved similar performance compared with previous studies. Appendix 4.9 includes an analysis of a different time period (the train period included water years 1981-1995, and the test period included water years1996-2014), which was chosen to overlap with the NWM-Rv2 retrospective run.

4.3.3 Performance metrics

We report two sets of performance metrics. The first set are standard benchmarking metrics that we report for two reasons: (i) to show that the models perform similarly with previous benchmarking studies, and (ii) to allow us to demonstrate a distinction between model performance and consistency of long-term mass balance. The second set of metrics are related to long-term streamflow biases, and allow us to test our primary hypothesis. These metrics are described in the following two subsections.

4.3.3.1 Standard performance metrics

We benchmarked all models using the same set of performance metrics that were used in previous CAMELS studies (Kratzert et al., 2019b,a, 2021; Gauch et al., 2021a; Klotz et al., 2021). A full list of these metrics is given in Table 4.3. Each of the metrics was calculated for each basin separately on the whole test period for the training/test splits described in Section 4.3.2.3 (the test period consists of water years 1990-1999).

Table 4.3: Overview of performance benchmarking evaluation metrics for hydrological models. The notation of the original publications is kept.

Metric	Description	Reference/Equation
NSE	Nash-Sutcliff efficiency	Eq. 3 in Nash and Sutcliffe (1970)
KGE	Kling-Gupta efficiency Skill Score	Eq. 9 in Gupta et al. (2009)
Pearson-r	Pearson correlation between observed and simulated flow	
α-NSE	Ratio of standard deviations of observed and simulated flow	From Eq. 4 in Gupta et al. (2009)
β-NSE	Ratio of the means of observed and simulated flow	From Eq. 10 in Gupta et al. (2009)
Peak-Timing	Mean peak time lag (in days) between observed and simulated peaks	Appendix B in Kratzert et al. (2021)

4.3.3.2 Long term mass balance

We conducted a long-term mass balance analysis using the absolute mass bias error for each basin:

$$total\ absolute\ mass\ bias = \frac{|\sum obs.\ Q - \sum sim.\ Q|}{\sum obs.\ Q} \tag{4.3}$$

where Q is the mass flux of streamflow. The positive and negative mass bias error for each basin is calculated as

$$positive\ mass\ bias = \begin{cases} x = \frac{\sum obs.\ Q - \sum sim.\ Q}{\sum obs.\ Q}, & \text{if } x > 0 \\ 0, & \text{otherwise} \end{cases} \tag{4.4}$$

and

$$negative\ mass\ bias = \begin{cases} x = -\frac{\sum obs.\ Q - \sum sim.\ Q}{\sum obs.\ Q}, & \text{if } x < 0 \\ 0, & \text{otherwise} \end{cases} . \tag{4.5}$$

We used these metrics to provide a general measure the ability of each model to close the mass balance between precipitation and streamflow. These metrics require a continuous observation record, however only 530 of the 531 CAMELS benchmarking basins satisfy this for the test period of water years 1990-1999. It is worth noting that these are technically volume calculations, and that we assume a constant liquid density for mass balance.

4.3.3.3 Measuring information loss from modeling constraints

Following the discussion by (Nearing and Gupta, 2015), we anticipate an ordering of information content like:

$$H_{streamflow} \geq I_{inputdata} \geq I_{LSTM} \geq I_{MC-LSTM} \geq I_{SAC-SMA} \tag{4.6}$$

H indicates the total entropy of whatever target data we are trying to predict (here a hydrograph in an individual basin). There is some amount of information in the input data (meteorological forcings and basin attributes), however the data processing inequality Cover and Thomas (2005, equation 2.122, page 35) indicates that information is lost by any model which means that any model prediction contains less than, or equal to, information about the target data than is contained in the raw inputs (see Nearing and

Gupta, 2015, for further discussion). Finally, we hypothesize that the constraints in the MC-LSTM (mass conservation) and the conceptual SAC-SMA model will mean that these two models provide less information than the LSTM. It is important to point out that the latter two terms of Equation/Inequality 4.6 are only hypotheses – it is possible that adding constraints to a trained model (either a neural network or a calibrated conceptual model) will improve performance. We consider this unlikely, since adding constraints to a DL model serves only to restrict the space of functions that the model can emulate, however it is always possible that regularization will help avoid local minima during training, or otherwise compensate for limited information content of training data.

We quantified this (hypothesized) chain of inequalities using two difference metrics. The first metric is the standard mutual information (MI) metric calculated by histograms with 100 bins:

$$MI(U,V) = \sum_{i=1}^{|U|}\sum_{j=1}^{|V|} \frac{|U_i \bigcap V_j|}{N} log \frac{N|U_i \bigcap V_j|}{|U_i||V_j|} \tag{4.7}$$

where U is the observed streamflow, V is the simulated streamflow and N is the number of records. Mutual information obeys the data processing inequality, so that the first and second terms of Equation/Inequality 4.6 apply strictly. We calculated the MI in two ways: (1) at each basin individually for a distribution of values, and (2) using all of the flows from all basins combined for an overall MI score that does not account for distinctions between basins.

We also report the skill score outlined by Knoben et al. (2019) based on KGE metrics:

$$KGE_{skillscore} = \frac{KGE_{model} - KGE_{baseline}}{1 - KGE_{baseline}} \tag{4.8}$$

where the skill score compares the performance of a candidate model with a baseline. This lets us draw an intuitive connection between the benchmarking metrics in Section 4.3.3.1 and Equation 4.6, however we do omit proof that the KGE obeys the data processing inequality, and therefore the relationship is only intuitive. The first level of constraint that we test is the strictly enforced mass conservation in the MC-LSTM and SAC-SMA. This is analogous to the third inequality in Equation 4.6, and in this case KGE_{model} and $KGE_{baseline}$ in Equation 4.8 are the MC-LSTM and the LSTM, respectively. The second level of constraint is the conceptualization of the watershed as implemented by SAC-SMA model architecture. This is analogous to the third term in Equation/Inequality 4.6, and in this case KGE_{model} and $KGE_{baseline}$ in Equation 4.8 are SAC-SMA and the LSTM, respectively (we use the LSTM instead of the MC-LSTM as the baseline in order to plot a direct comparison between information lost by the MC-LSTM and SAC-SMA).

4.3.4 Events based analysis of mass balance

We analyze the performance for individual runoff events. This type of analysis requires additional assumptions about the nature of the rainfall-runoff relationship. The first assumption is that we can resolve a mass balance of individual runoff events. This is not a general assumption about individual watersheds, as water mass can accumulate in the form of snowpack, surface water ponding, groundwater, etc. In other words we lack the ability to directly analyze a model's accuracy in representing mass conservation of the rainfall-runoff process of an individual event because there is uncertainty about the complete water mass state within a watershed, and potentially extra-watershed mass sources and sinks. Beven (2019) describes a methodology that attempts to address the assumptions described above, and the fact that the hydrologic data includes significant epistemic uncertainties. Beven (2019) produce sample plots showing histograms of runoff coefficients (runoff ratios) from 100 nearest-neighbour events; empirical membership values of potential runoff coefficients from nearest-neighbour storms based on Mahalabonis distances; and event hydrographs. We produce similar plots for every event in the record where the event criteria is met by both NLDAS and Daymet forcings. But for the sake of not cherry picking any particular event as an example, we offer them as a supplement available on Hydroshare (Frame, 2022). We use Beven's 2019 methodology for analyzing the mass balance between model and observation based on events, as opposed to the intended purpose of eliminating events from the training period. Another assumption is that the liquid density of the input and output is identical for all events, which means any phase change and/or volume change is done internally by each model. This assumption is based on the units of the input and output data provided by CAMELS.

4.3.4.1 Defining a runoff event

Since we included a diverse range of basins in our analysis, we needed to define a runoff event that could be representative across many different types of hydrologic regimes. We defined an event as consecutive days with precipitation, so long as one of those days has a total daily precipitation greater than the 25th percentile of the (non-zero) precipitation within the record. Distinct precipitation events were defined as separated by at least one day where the precipitation is less than 5th percentile of the (non-zero) precipitation within the record. The 25th percentile as the ”event” threshold and the 5th percentile as the ”event separation” threshold was tuned in order to get a sufficient number of rainfall events (100) in every basin. In other words an event starts when precipitation is greater than the 25th percentile and stops when less than the 5th percentile.

4.3.4.2 Mahalanobis distance

For every streamflow event we calculated the 100 most similar events using the Mahalanobis distance in the space of antecedent runoff conditions and total precipiutation of the runoff event. The Mahalanobis distance between each event individually with the other runoff events is calculated as:

$$Mahalanobis\ distance = \sqrt{(u-v)V^{-1}(u-v)^T} \tag{4.9}$$

where u is the antecedent streamflow, v is the runoff ration of the particular event, and V is the covariance matrix.

4.3.4.3 Comparing observed and predicted runoff ratios for individual events

For each runoff event, at each individual basins, we compared only with the closest 100 nearest neighbors according to the Mahalanobis distance (4.9) described above. For every precipitation event we compare the distribution of nearest neighbor modeled streamflow predictions with the distribution of nearest neighbor observed streamflow predictions using the mutual information calculation described in 4.3.3.3. The distributions are compared with the Mutual Information (Equation 4.7) and the coefficient of determination.

4.3.4.4 Analyzing the distributions of predicted runoff ratios

We calculated the location of the predicted runoff ratio for each runoff event within the distribution of the 100 nearest neighbor events. We then consider the percentile bins of the event distributions with an idealized set of bins, where each tenth percentile of the events fall exactly in that tenth percentile. We then plot the count of the events against the corresponding idealized percentile count. This is commonly referred to as a quantile-quantile plot. We then also plot the distance (and cumulative distance) of these quantiles from the one-to-one line. We then summarize the models on a regional basis by plotting the absolute, positive and negative divergence from the one-to-one line of the quantile-quantile plot.

4.3.5 Conditionality of the modelling analysis

Uncertainty in this experiment comes from three primary sources: data, models, and training. These sources are analogous to standard sources of uncertainty in most hydrology modeling studies: data, model structure, and model parameters (training is analogous to calibration).

The hypothesis that we are testing is related to understanding relationships between data and model uncertainty. Our objective is to understand how different models deal with uncertainty in data. We do not explicitly represent uncertainty in data (e.g., probabilistically), because our experiment does not require this for testing the hypothesis. We treat training/calibration uncertainty by using an ensemble of eight models, where we take the ensemble mean of streamflow as our final model estimate. This is approach is discussed explicitly for DL models by (Kratzert et al., 2019a), and for the SAC-SMA model by (Newman et al., 2015). Our analysis in Section 4.4.2 includes a box-and-whisker plot showing the mean, standard deviation and outliers of each performance metric. We also include a complete (second) set of results of the same analysis on a different time period in Appendix 4.9, with results that are nearly identical, indicating that our results are not the result of an anomalous time period.

Our models and training are consistent with previous studies. Benchmarking results like what are reported in Section 4.4.1 have been repeated by several research groups using different basins and different data products. Results presented here are consistent with previous large-sample studies for all models, which provides a degree of confidence about the modeling results in general. We included 5/95% confidence intervals of the summary statistics, and these are relatively low given our large sample size.

4.4 Results

4.4.1 Model performance

Table 4.4 provides performance metrics (Section 4.3.3.1 for the LSTM, MC-LSTM and SAC-SMA model simulations over the test period (water years 1990-1999). Most of these scores are broadly equivalent to the metrics for the same models reported by other studies Kratzert et al. (2019b, e.g.,). More importantly, these metrics allow us to test the hypothesis that explicit mass conservation degrades performance (as a reminder, this hypothesis was proposed by (Beven, 2020)). What we are looking for in these metrics is that either all the mass conserving models perform worse than the non-constrained LSTM, which would support the hypothesis that mass conservation is detrimental to models, or that the MC-LSTM with an explicit mass conserving constraint does as well or better than the LSTM, which would indicate that the problem with the conceptual model is *not* a matter of enforcing closure over erroneous data.

Results show similar average performance between the LSTM and MC-LSTM, however there were largely small differences. The LSTM had a higher KGE score in 323 basins, with an average difference of 0.06, and the MC-LSTM had a higher KGE score in 208 basins, also with an average difference of 0.06. In general, the median performance metrics of the two models were broadly comparable. Both models were, on average, better across all metrics than SAC-SMA. This suggests that enforcing closure does *not* explain the differences between data-based and conceptual models.

4.4.2 Long-term cumulative discharge

Figure 4.1 shows the cumulative density functions (CDFs) of long-term cumulative discharge from the 530 CAMELS basins from the models during the 1989-1999 test period. The LSTM, MC-LSTM and SAC-SMA all have a similar total mass bias with the NLDAS forcing. SAC-SMA has the lowest negative mass error, but the highest positive mass error. The LSTM has the highest negative mass error, but the lowest positive mass error. The MC-LSTM is generally in between the LSTM and SAC-SMA. Overall, the LSTM and MC-LSTM predicted streamflows that result in more accurate long-term cumulative discharge than the calibrated SAC-SMA model. The LSTM and the MC-LSTM performed roughly similarly on NLDAS, and MC-LSTM slightly outperformed the LSTM on Daymet. With Daymet forcing SAC-SMA's streamflow predictions are biased towards a very high positive mass error.

Figure 4.2 shows the long term positive or negative mass biases distributed across the Contiguous United States (CONUS) from for the three models with both Daymet and NLDAS forcings. The result of the SAC-SMA simulation with Daymet forcings shows a clear positive mass bias error in the eastern half of CONUS. The result of the SAC-SMA simulation with NLDAS forcings shows a mix of positive and negative mass bias throughout CONUS. The LSTM and the MC-LSTM look relatively similar, to each other

Table 4.4: Median performance metrics (plus or minus the 95% confidence interval) across 530 basins calculated on the test period 1990-1999 with two separate forcing products.

	Daymet forcing			NLDAS forcing		
Metric	LSTM	MC-LSTM	SAC-SMA	LSTM	MC-LSTM	SAC-SMA
NSE	0.77 ± -0.02	0.76 ± -0.01	0.65 ± -0.03	0.74 ± -0.01	0.74 ± -0.01	0.67 ± − 0.02
KGE	0.76 ± -0.02	0.76 ± -0.02	0.59 ± n/a	0.74 ± -0.02	0.74 ± -0.02	0.68 ± − 0.02
Pearson-r	0.89 ± -0.01	0.88 ± -0.01	0.83 ± n/a	0.88 ± -0.01	0.87 ± -0.01	0.83 ± − 0.01
Alpha-NSE	0.85 ± -0.01	0.84 ± -0.01	0.76 ± -0.02	0.81 ± -0.02	0.81 ± -0.02	0.78 ± − 0.02
Beta-NSE	-0.04 ± -0.01	-0.03 ± -0.01	0.06 ± -0.01	-0.03 ± -0.01	-0.02 ± -0.01	-0.01 ± − 0.01
Peak-Timing	0.3 ± − 0.03	0.3 ± − 0.03	0.38 ± − 0.06	0.32 ± − 0.03	0.31 ±-0.03	0.41 ± -0.06

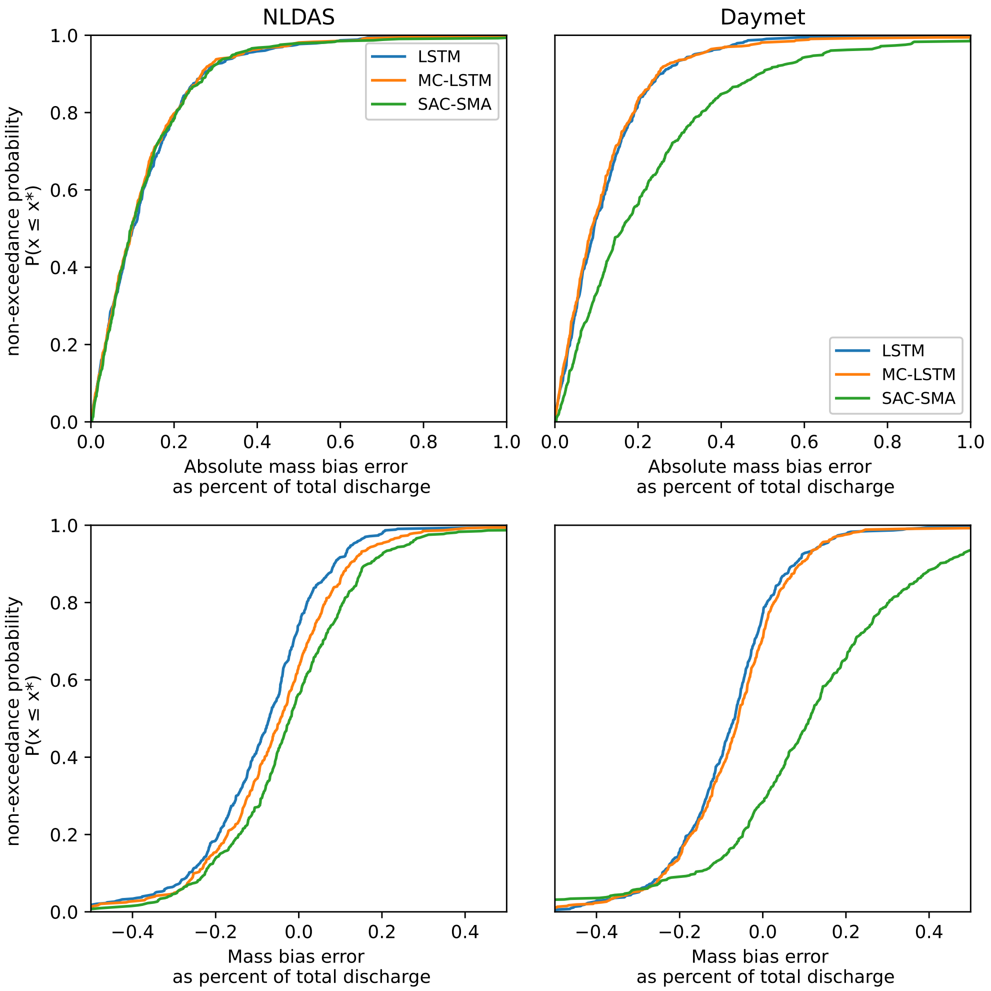

Figure 4.1: Distribution of mass balance error across the 530 basins. Top: Cumulative distribution curves of the absolute mass error from models forced with NLDAS (left) and Daymet (right). Bottom: Cumulative distributions of mass error from models forced with NLDAS (left) and Daymet (right).

and for both NLDAS and Daymet forcing. This Central CONUS (CenCon) region (i.e., Missouri, Arkansas-White-Red and Texas-Gulf) is generally tough to predict, with conceptual, physical and deep learning models. SAC-SMA also shows a negative mass bias pattern in the same central CONUS region, though to a lesser spatial extent and higher magnitude, with Daymet forcings, but not so much with NLDAS forcings.

Figure 4.3 shows the mass bias errors for the model runs with Daymet forcings in box and whisker plots for the U.S. Water Resources Regions. SAC-SMA shows a very high

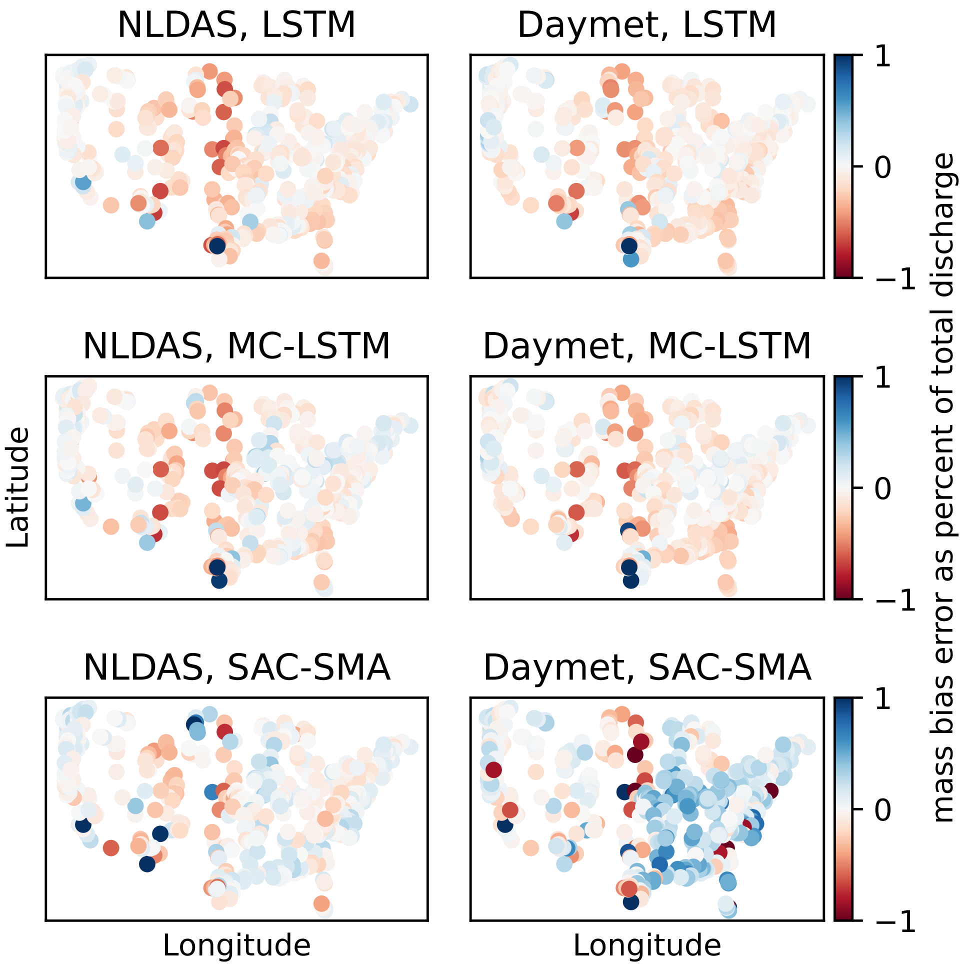

Figure 4.2: Geospatial distribution of long term positive or negative mass bias error. The left and right columns show the results with NLDAS and Daymet meteorological forcing data, respectively. The three rows are associated (from top to bottom) with LSTM, MC-LSTM and SAC-SMA.

mass balance error in the eleven eastern regions, but does much better in the western regions. The LSTM shows a high mass balance error in the Lower Colorado region, as compared to the MC-LSTM, and the MC-LSTM shows a higher mass balance error in the Rio Grande region, but the LSTM and MC-LSTM are relatively similar (more or less) in the other regions. All three models show relatively high mass bias errors in the CenCon region, which is the contribution from negative mass bias shown in Figure 4.2. SAC-SMA

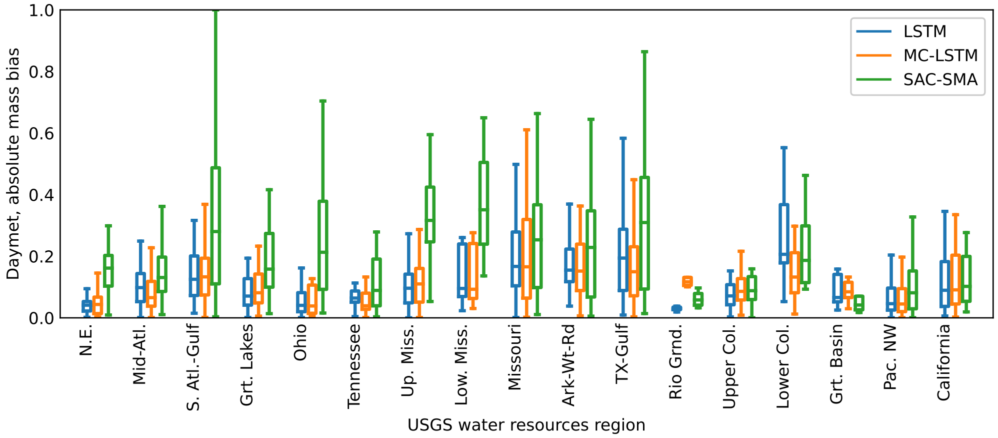

Figure 4.3: Regional mass balance errors from LSTM, MC-LSTM and SAC-SMA with Daymet forcings. The main body of each box shows the median and confidence intervals. The vertical lines extending to the most extreme, non-outlier data points. Souris-Red-Rainy region (Hydrologic Unit Code 09) is absent due to a lack of sufficient basins.

performs worse than the LSTM and the MC-LSTM in the CenCon region with Daymet forcings.

Figure 4.4 shows the mass bias errors for the model runs with NLDAS forcings in box and whisker plots for the U.S. Water Resources Regions. With NLDAS forcing, SAC-SMA does not have a consistent mass bias error, as with Daymet. The pattern of SAC-SMA mass bias error in the western U.S. is generally similar between Daymet (Figure 4.3) and NLDAS (Figure 4.4). The differences between the LSTM, MC-LSTM and SAC-SMA does not show any obvious patterns. SAC-SMA and LSTM shows a high mass bias error outlier in the Lower Colorado region, but MC-LSTM does not. All three models show relatively high mass bias errors in the CenCon region, although SAC-SMA has a lower mean mass bias error than the LSTM and the MC-LSTM, but has a higher outlier in Missouri. Kratzert et al. (2019a) shows in their Figure 4 that the LSTM scores better in terms of Nash-Sutcliffe Efficiency than SAC-SMA, which seems to indicate that mass bias error in the catchment data does not explain the difference in predictive skill between deep learning and conceptual models.

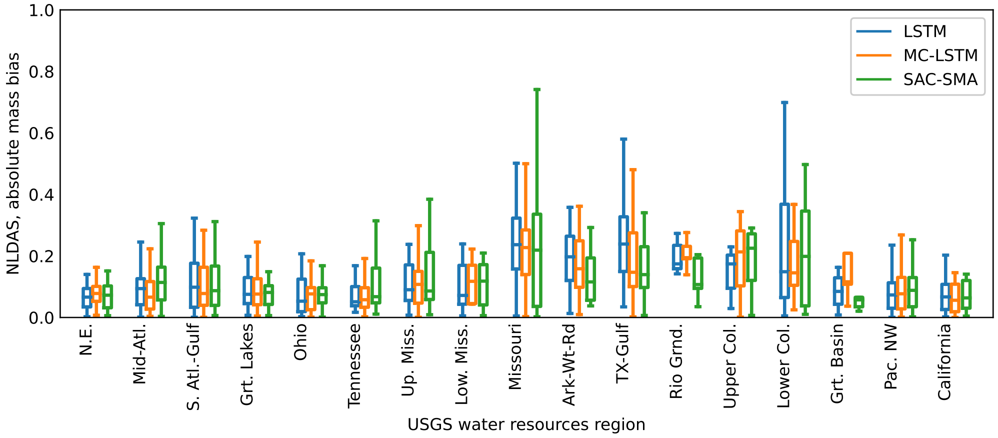

Figure 4.4: Regional mass balance errors from LSTM, MC-LSTM and SAC-SMA with NLDAS forcings. The main body of each box shows the median and confidence intervals. The vertical lines extending to the most extreme, non-outlier data points. Souris-Red-Rainy region (Hydrologic Unit Code 09) is absent due to a lack of sufficient basins.

Appendix 4.9 includes results from a separate time period, where the NWM can be compared (with caveats of the inconsistent calibration period) on the NLDAS forcing data. The overall, spatial and regional results are roughly similar for the LSTM, MC-LSTM and SAC-SMA.

4.4.3 Information loss due to modeling constraints

4.4.3.1 Mutual information

The mutual information scores of the combined 530 basins (concatenated and calculated once across all basins) with NLDAS forcings are: 0.39 (LSTM), 0.37 (MC-LSTM) and 0.34 (SAC-SMA), respectively. The mutual information scores of the combined 530 basins (concatenated and calculated once across all basins) with Daymet forcings for models LSTM, MC-LSTM and SAC-SMA are 0.40, 0.37 and 0.33, respectively. Figure 4.5 shows the CDF plots with mutual information scores *calculated individually for each of the 530 basin*. For both Daymet and NLDAS the CDF curves show that LSTM has the most mutual information with the observed runoff, followed by MC-LSTM and then by SAC-SMA.

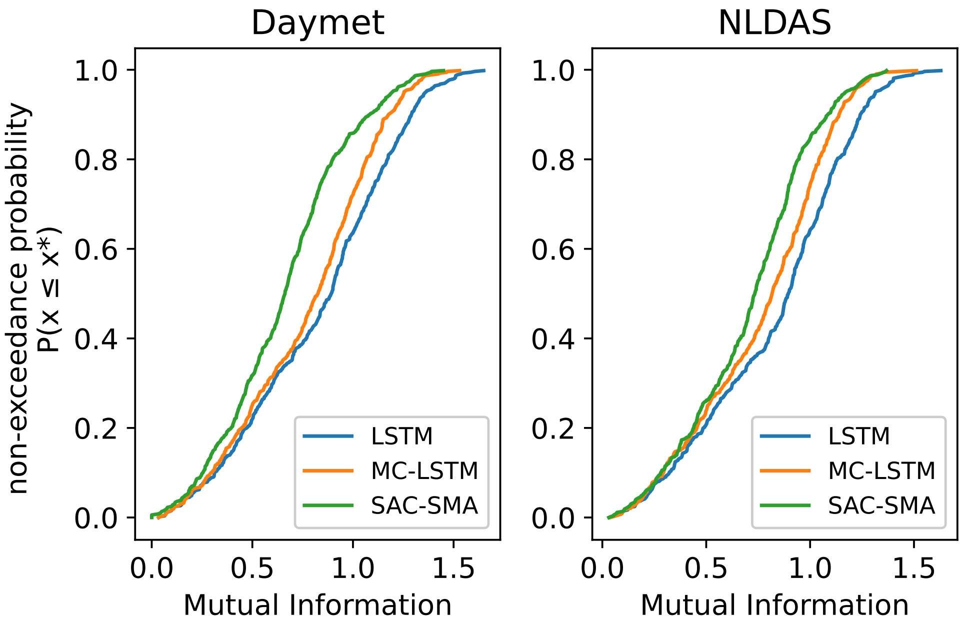

Figure 4.5: Left: Cumulative distribution of the mutual information for LSTM, MC-LSTM and SAC-SMA with Daymet forcing. Right: Cumulative distribution of the mutual information for LSTM, MC-LSTM and SAC-SMA with NLDAS forcing

4.4.3.2 KGE Skill Score

The unconstrained LSTM was used as a baseline model to measure information loss in the MC-LSTM and SAC-SMA. Results of the KGE skill score (KGE_{ss}) analysis are shown in Figure 4.6. The left subplot of this figure shows a clear ordering of model performance that agrees with what we hypothesized in Equation/Inequality 4.6 – generally, model performance degrades as more constraints are added. The left subplot also shows that DL models perform better when trained and forced with Daymet data than with NLDAS data. This is somewhat counter-intuitive given the large, nonstationary bias that we saw in the previous section (Figure 4.1). SAC-SMA, however, performed significantly worse with the biased data. While the DL models (even those constrained to conserve mass) were able to learn to accommodate the spatially heterogeneous biases in the input data, the PB model was not, even when trained on the biased data in each individual catchment. Daymet is the more informative precipitation product overall and a

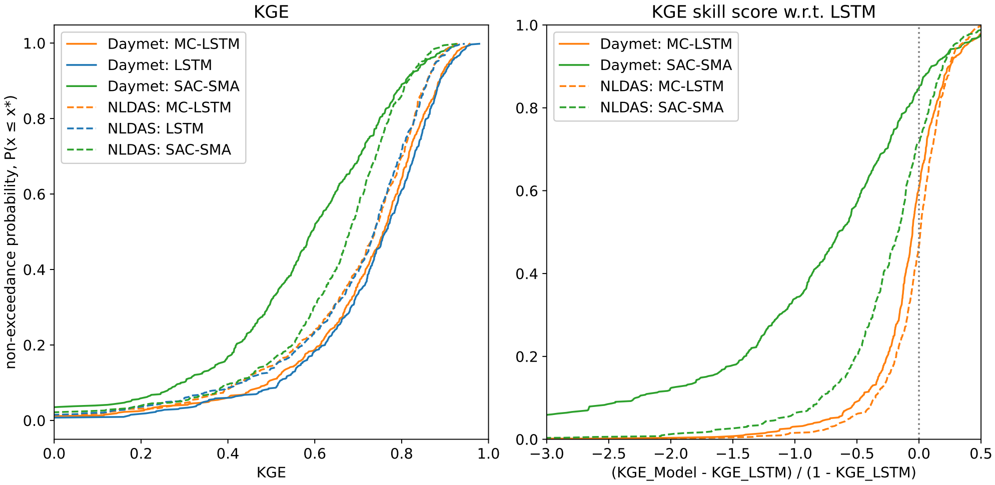

Figure 4.6: Left: Cumulative distribution of the Kling-Gupta Efficiency (KGE) for three models on two different forcing products. This subplot shows (i) that in general Daymet is more informative than NLDAS, and (ii) that the ordering of the inequality in Equation 4.6 is generally correct. Right: KGE skill scores (Equation 4.8) of SAC-SMA and MC-LSTM with respect to the unconstrained LSTM (positive values to the right of the dotted gray line mean that the MC-LSTM(SAC-SMA) performs better than the LSTM, and negative values to the left of the gray line mean that the MC-LSTM(SAC-SMA) performs worse than the LSTM). This subplot shows that adding mass balance constraints to the LSTM has more benefit when using NLDAS inputs than when using Daymet inputs.

flexible DL model is able to learn and extract this information while the PB model cannot (even though the PB model is locally calibrated), however it is *not* the mass balance constraints that cause the problem.

The right subplot of Figure 4.6 plots the CDF of the skill scores (Equation 4.8) of the MC-LSTM and SAC-SMA relative to the unconstrained LSTM. The gray dotted vertical line represents a skill score of zero, indicating that the test model (MC-LSTM, SAC-SMA) performs equally well as the baseline (LSTM). The main takeaway from this figure is that adding mass balance constraints (both in the MC-LSTM and in SAC-SMA) helps more when using the NLDAS data, even though it was the Daymet data that showed biases.

Figure 4.7 plots the CDF of the difference between the MC-LSTM and the LSTM. In each basin, this difference represents an *upper bound* on the error introduced by mass

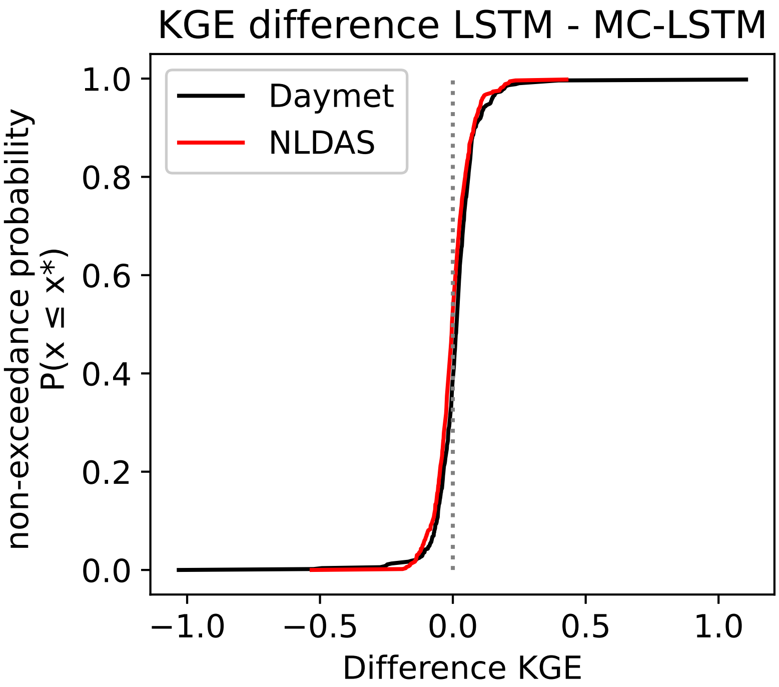

Figure 4.7: CDF of an estimated upper bound on the error introduced into DL streamflow predictions by adding a mass balance constraint.

balance constraints, relative to the LSTM. There are other possible reasons why the MC-LSTM might not perform as well as the LSTM (e.g., the way that it handles unobserved sources and sinks), however this difference (which is sometimes negative) is a conservative estimate of the error due to mass conservation in DL rainfall runoff models.

4.4.4 Comparing mass runoff coefficients of events as a proxy for short term mass balance

This section analyzes the mass balance of short term events (individual precipitation events over one to several days) based on the antecedent runoff conditions, and the runoff coefficient from the particular event.

4.4.4.1 Mutual information between runoff coefficient distributions

Table 4.5 shows the mutual information between runoff coefficient distributions predicted by the model simulations and the observed streamflow. Using both the mutual information between runoff ratio distributions and the r-squared of runoff ratio values, the LSTM matches the runoff ratio best across individual events, followed by the MC-LSTM and then SAC-SMA. This is consistent with the results of the long term mass balance analysis. The mutual information is much higher for Daymet forcing than for NLDAS. The coefficient of determination is higher for NLDAS for the LSTM and SAC-SMA, but is lower for MC-LSTM.

4.4.4.2 Quantile comparison of runoff coefficients within nearest neighbor distributions

Figure 4.8 shows the modeled vs observed quantiles of the runoff ratios. The NLDAS quantiles are very close to the 1 to 1 line for LSTM, MC-LSTM and SAC-SMA. The Daymet quantiles for all models bow out from the 1 to 1 line for the lower to middle quantiles, similarly for all three models, but the scale of that artifact is lowest for SAC-SMA and highest for MC-LSTM.

Figure 4.9 shows the q-q plot total absolute, positive and negative divergence from the 1 to 1 line on a per region basis. The NLDAS plots show little divergence and similar regional trends for each model. The Daymet plot show a lot of divergence. Specifically, Daymet forcings cause SAC-SMA to diverge positively in the eastern U.S. (HUCs 1-10), while the LSTM and MC-LSTM diverge negatively in the most eastern U.S. regions (excluding HUC regions 6 and 9).

4.5 Conclusions

The hypothesis tested in this paper is that errors in input/output (precipitation/streamflow) data cause apparent violations of closure that may largely explain the poor performance of conceptual models relative to deep learning. Given that the physical principle of mass balance over a control volume is one of the most fundamental components of hydrological theory, and is the first assumption we take for

Table 4.5: Comparing modeled and observed runoff ratios of individual events

	NLDAS forcing (n=52359)			Dayment forcing (n=42858)		
Metric	LSTM	MC-LSTM	SAC-SMA	LSTM	MC-LSTM	SAC-SMA
Mutual Information	0.165	0.150	0.120	0.602	0.523	0.439
Coefficient of determination (r^2)	0.903	0.849	0.831	0.897	0.866	0.752

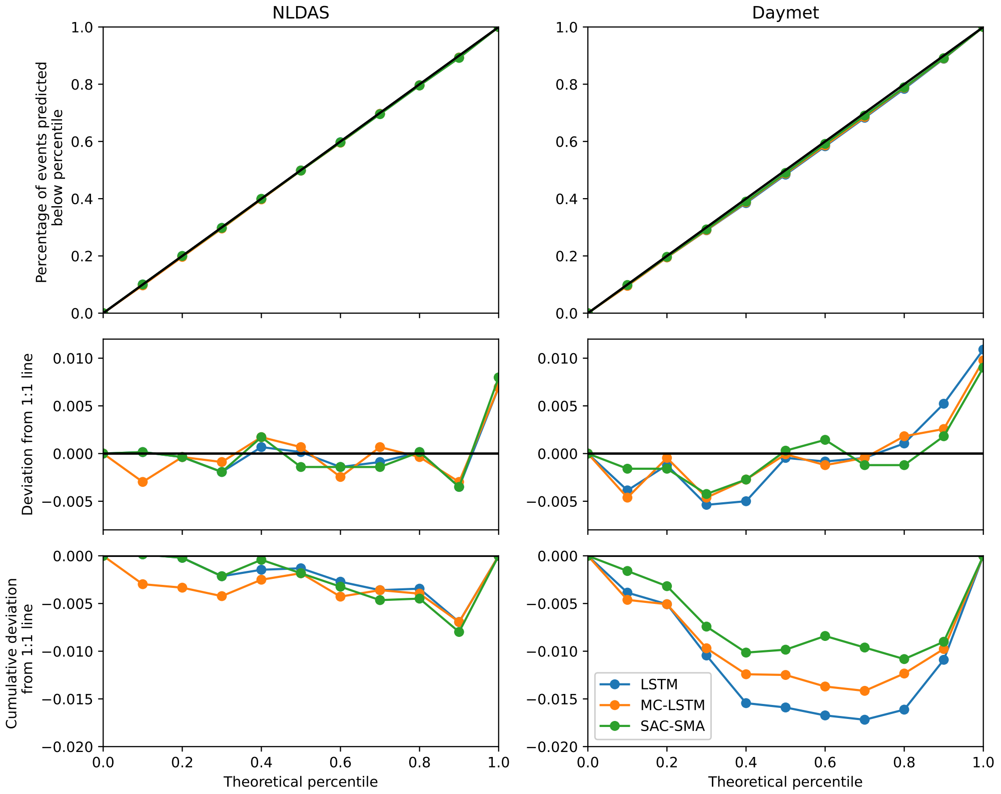

Figure 4.8: Top row: Quantile-quantile comparisons where the theoretical percentiles on the x-axis represent an ideal distribution and the predicted percentiles are on the y-axis. Middle row: Deviation from the 1 to 1 line of the quantile-quantile distributions shown on the top row. Bottom row: Cumulative deviation from the 1 to 1 line of the quantile-quantile distributions.

granted when developing theory-based hydrological models, it is arguably the first physical "law" that we might test when developing physics-informed ML strategies for hydrology.

For Beven's closure-violation hypothesis to be true as an explanation for the general failure of traditional hydrology models, the errors in rainfall or discharge data must necessarily be systematic in a way that can be learned by a neural network, but not by a calibrated conceptual model. Our results indicate that this is indeed the case: the DL-based LSTM network was able to learn the non-uniform patterns of biases in input–output data, and thereby extract useful information from different (imperfect) precipitation products in spatially and temporally heterogeneous ways. This was true even when using heavily biased rainfall products that contribute significant so-called

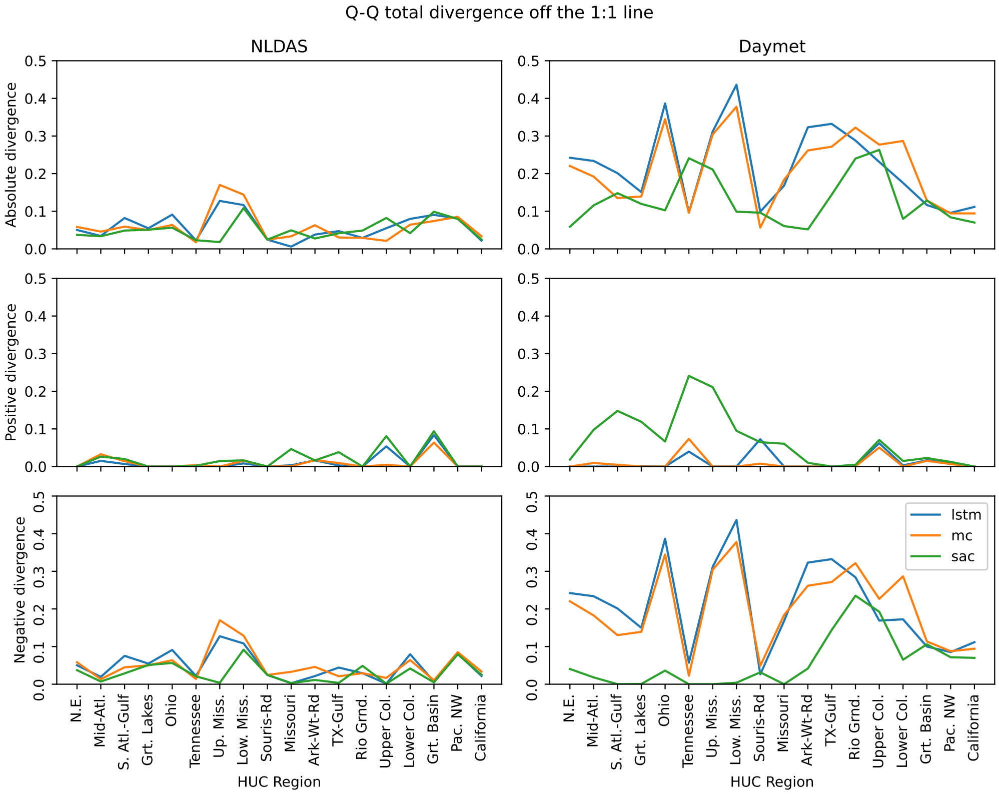

Figure 4.9: Top row: The absolute divergence. Middle row: The positive divergence, where there are more basins in the quantile bin. Bottom row: The negative divergence, where there are fewer basins in the quantile bin.

“disinformation” when calibrating a conceptual model; e.g., Daymet seems to actually contain more information about streamflow than the less biased NLDAS precipitation product. It is important to note that these data biases are not simple additive shifts in the mean – instead they are complex and heterogeneous throughout our large-sample dataset, and the DL models are able largely to learn this heterogeneity.

The long term mass balance analysis (Section 4.4.2) shows that SAC-SMA is strongly influenced by the strong positive mass bias in the Eastern U.S. from Daymet forcings. The event-based analysis (Section 4.4.4) shows that SAC-SMA responds to that mass bias by diverging positively from the simulated runoff coefficients within nearest neighbor distributions for those Eastern U.S. basins, whereas both the LSTM and the MC-LSTM respond with a negative divergence for those distributions. Given that both

SAC-SMA and the MC-LSTM both are constrained to conserve mass in a similar manner, it is likely not a mass conservation constraint that causes that positive vs. negative divergence.

While Beven (2020) was correct that the imposition of conservation laws is generally harmful for hydrologic prediction, this fact does not help to explain most of the significantly better skill provided by DL over traditional (theory-based) rainfall-runoff models. These findings demonstrate two things:

1. that conservation laws may not be a good foundation for scale-relevant hydrological theory,
2. that Beven and Westerberg's 2011 claim that disinformative data is a major source of modeling error is, in general, incorrect,
3. that DL models compensate for systematic biases in the input data on a per-event basis.

In fact, data that are "disinformative" when used in the context of calibrating poorly conceived models might actually contain significant amounts of useful information that is accessible when used in the context of better conceived models. In other words, for catchment-scale rainfall-runoff prediction it is arguably the current hydrological theory that is (more) disinformative, not the hydrological data. In summary, model performance degrades as constraints are added, which causes loss of information between the inputs (atmospheric forcings) and the target (streamflow), as shown in 4.3.3.3.

4.6 Discussion

There is some subtlety to this conclusion due to the fact that the MC-LSTM includes a flux term that accounts for unobserved sinks (e.g., evapotranspiration, sublimation, aquifer recharge). However, it is important to note that most or all hydrology models that are based on closure equations include a residual term in some form. Like all mass balance models, the MC-LSTM explicitly accounts for all water in and across the boundaries of the system. In the case of the MC-LSTM, this residual term is a single aggregated flux. Even with this strong constraint, the MC-LSTM performs significantly better than the mass-conserving benchmark conceptual model. This result indicates that classical hydrology model structures (conceptual architectures and flux equations) actually cause prediction errors that are larger than can be explained as being due to errors in the forcing and observation data.

Our ability to properly conduct a more rigorous and detailed analysis of long-term water balances is limited by the fact that accurate evapotranspiration and percolation data (etc.) are not readily available at watershed scales. Nonetheless, what our analysis based on examining cumulative discharge shows is that an LSTM architecture not constrained to conserve mass is able to extract information from the available data that enables it to learn "effective" water balances that are similar to those learned by a similar model architecture (MC-LSTM) that is explicitly constrained to enforce such closure, and that this effective water balance is in general better than that achieved by traditional conceptual and PB model architectures. Results from (Lees et al., 2022) suggest that LSTM learns to reproduce stores of water, such as soil moisture and snow cover.

A likely reason for this is that that the current body of hydrological theory does not aggregate well to the scale of unorganized complex watershed systems (Nearing et al., 2020c). While it is true that hydrological theory can enable a modeler to "interpret" a watershed response (assuming a proper accounting for uncertainty), such theory does not currently translate into accurate predictions of catchment-scale behaviors using available data. Meanwhile, the most accurate way to generate a predictive model is to impose as few "physical constraints" as possible on its ability to extract information from the available data, and consequently any model that is constrained to obey some "deeper" physical understanding of the system must be less accurate in a predictive sense, unless that physical understanding actually contributes predictively-useful information that cannot be otherwise extracted directly from the data.

Looking forward, a particular application of rainfall-runoff modeling that necessarily requires the imposition of strict mass-balance constraints is “Earth-system-scale ” modeling. In this context, any model that seeks to explain components of long-term climate variability (for instance) cannot allow for any significant amount of residual mass to go unexplained. To use a dramatic example, unaccounted for losses at the catchment-scale could potentially result in the removal of all water mass from the global water cycle, which would render a long-term simulation useless. Global-scale modeling of land-surface dynamics could be a potentially powerful application of the MC-LSTM network approach, and could be implemented by training additional model targets of mass-loss representations of “losses” (transfers) to the sub-surface and “losses” (transfers) to the atmosphere.

4.7 Appendix: LSTM

Long Short Term Memory networks (Hochreiter and Schmidhuber, 1997) are dynamic state-space recurrent neural networks that are well known to be appropriate for rainfall-runoff modeling (Kratzert et al., 2018). The rainfall-runoff process is a time evolving system that is comprised of 1) the state of the watershed, and 2) the dynamic response to inputs. This is the same general principal on which recurrent neural networks operate.

The input to an LSTM, as we use in hydrologic modeling, is a vector of both forcings (values that change with time) and static attributes of the particular system under consideration. Our LSTM is run at a discrete timestep, in this case a daily timestep, making the input vector, $\boldsymbol{x}[t]$, with forcings representing the daily values. This LSTM is trained to represent the dynamics between the input-state-output relationship at this particular timestep. Given a time sequence of inputs ($\boldsymbol{x} = [\boldsymbol{x}[1], ..., \boldsymbol{x}[T]]$), which act on the state of the previous timestep ($\boldsymbol{h}[t-1]$) and we get an output sequence ($\boldsymbol{y} = [\boldsymbol{y}[1], ..., \boldsymbol{y}[T]]$).

The LSTM network as a discrete timestepping model is represented as the following equations:

$$\boldsymbol{i}[t] = \sigma(\boldsymbol{W_i}\boldsymbol{x}[t] + \boldsymbol{U_i}\boldsymbol{h}[t-1] + \boldsymbol{b_i}) \tag{4.10}$$

$$\boldsymbol{f}[t] = \sigma(\boldsymbol{W_f}\boldsymbol{x}[t] + \boldsymbol{U_f}\boldsymbol{h}[t-1] + \boldsymbol{b_f}) \tag{4.11}$$

$$\boldsymbol{g}[t] = \tanh(\boldsymbol{W_g}\boldsymbol{x}[t] + \boldsymbol{U_g}\boldsymbol{h}[t-1] + \boldsymbol{b_g}) \tag{4.12}$$

$$\boldsymbol{o}[t] = \sigma(\boldsymbol{W_o}\boldsymbol{x}[t] + \boldsymbol{U_o}\boldsymbol{h}[t-1] + \boldsymbol{b_o}) \tag{4.13}$$

$$\boldsymbol{c}[t] = \boldsymbol{f}[t] \odot \boldsymbol{c}[t-1] + \boldsymbol{i}[t] \odot \boldsymbol{g}[t] \tag{4.14}$$

$$\boldsymbol{h}[t] = \boldsymbol{o}[t] \odot \tanh(\boldsymbol{c}[t]), \tag{4.15}$$

The symbols $\boldsymbol{i}[t]$, $\boldsymbol{f}[t]$ and $\boldsymbol{o}[t]$ refer to the *input gate*, *forget gate*, and *output gate* of the LSTM respectively, $\boldsymbol{g}[t]$ is the *cell input* and $\boldsymbol{x}[t]$ is the *network input* at time step t, $\boldsymbol{h}[t-1]$ is the LSTM output, which is also called the *recurrent input* because it is used as inputs to all gates in the next timestep, and $\boldsymbol{c}[t-1]$ is the cell state from the previous time step.

Cell states contain information of the system at any point in the discrete time. $\sigma(\cdot)$ are sigmoid activation functions, which return values in $[0, 1]$, which act as an attention of the internal network dynamics. The forget gate controls the memory timescales of each of the cell states, and the input and output gates control flows of information from the input features to the cell states and from the cell states to the outputs (recurrent inputs), respectively. $\boldsymbol{W}$, $\boldsymbol{U}$ and $\boldsymbol{b}$ are parameters that are tuned such that the dynamics of the LSTM produce an output that matches the target output, streamflow in our case. Parameter subscripts (i, f, g & o) indicate which gate the particular parameter matrix/vector is associated with. $\tanh(\cdot)$ is the hyperbolic tangent activation function, which serves to add nonlinearity to the model in the cell input and recurrent input, and $\odot$ indicates element-wise multiplication. For a hydrological interpretation of the LSTM, see Kratzert et al. (2018).

4.8 Appendix: Mass conserving LSTM

The LSTM can be trained to represent system dynamics, but without constraints that are typically used to represent a physical system. Although similar to the type of models that use a physical conceptualization to represent system dynamics, the LSTM lacks a few key ingrediants, including physical units associate with the system inputs and outputs. The MC-LSTM aims to bridge the gap by enforcing Equation 4.1 within the LSTM arcitecture. Using the notation from Appendix 4.7, this is:

$$\boldsymbol{c}^*[t] = \boldsymbol{c}^*[t-1] + \boldsymbol{x}^*[t] - \boldsymbol{h}^*[t], \tag{4.16}$$

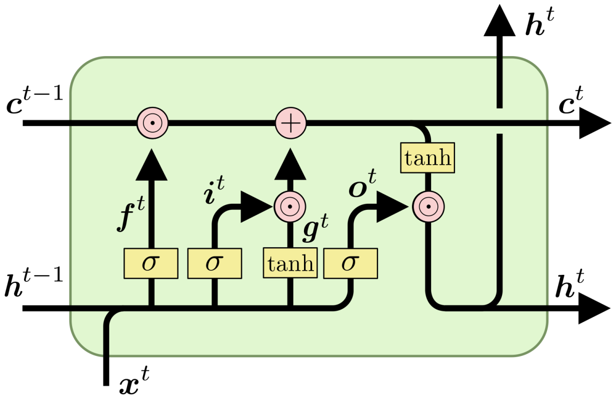

Figure 4.10: A single timestep of a standard LSTM with timesteps marked as superscripts for clarity. $\boldsymbol{x}^t$, $\boldsymbol{c}^t$, and $\boldsymbol{h}^t$ are the input features, cell states, and recurrent inputs at time t, respectively. $\boldsymbol{f}^t$, $\boldsymbol{i}^t$, and $\boldsymbol{o}^t$ are the forget-, input- and output-gate and $\boldsymbol{g}^t$ denotes the cell input. Boxes labeled σ and tanh represent single sigmoid and hyperbolic tangent activation layers with the same number of nodes as cell states. The addition sign represent element-wise addition and $\odot$ represents element-wise multiplication.

where $\boldsymbol{c}^*[t]$, $\boldsymbol{x}^*[t]$ and $\boldsymbol{h}^*[t]$ are components of the cell states, input features, and model outputs (recurrent inputs) that contribute to a particular conservation law.

Two major modifications to the LSTM are required to enforce mass conservation, 1) activation functions that ensure of all elements sum to one (through normalization) on gates where mass passes, and 2) subtracting the mass out of the system from the cell states (4.1) (Hoedt et al., 2021).

The constrained model architecture is illustrated in Fig. 4.11. Note that that the inputs are separated into *mass inputs* $\boldsymbol{x}$ and *auxiliary inputs* a. In our case, the mass input is precipitation and the auxiliary inputs are everything else (e.g. temperature, radiation, catchment attributes). The input gate (sigmoids) and cell input (hyperbolic tangents) in the standard LSTM are (collectively) replaced by one of these normalization layers, while the output gate is a standard sigmoid gate, similar to the standard LSTM. The forget gate is also replaced by a normalization layer, with the important difference that the output of this layer is a square matrix with dimension equal to the size of the cell state. This matrix is used to “reshuffle” the mass between the cell states at each timestep. This *reshuffling matrix* is column-wise normalized so that the dot product with the cell state vector at time t results in a new cell state vector having the same absolute norm (so that no mass is lost or gained).

We call this general architecture a *Mass-Conserving LSTM* (MC-LSTM), even though it works for any type of conservation law (mass, energy, momentum, counts, etc.). The architecture is illustrated in Figure 4.11 and is described formally as follows:

$$\hat{\boldsymbol{c}}[t-1] = \frac{\boldsymbol{c}[t-1]}{||\boldsymbol{c}[t-1]||_1} \tag{4.17}$$

$$\boldsymbol{i}[t] = \widehat{\sigma}(\boldsymbol{W}_i\boldsymbol{x}[t] + \boldsymbol{U}_i\hat{\boldsymbol{c}}[t-1] + \boldsymbol{V}_i\boldsymbol{a}[t] + \boldsymbol{b}_i) \tag{4.18}$$

$$\boldsymbol{o}[t] = \sigma(\boldsymbol{W}_o\boldsymbol{x}[t] + \boldsymbol{U}_o\hat{\boldsymbol{c}}[t-1] + \boldsymbol{V}_o\boldsymbol{a}[t] + \boldsymbol{b_o}) \tag{4.19}$$

$$\boldsymbol{R}[t] = \widehat{\mathrm{ReLU}}(\mathbf{W}_R\boldsymbol{x}[t] + \mathbf{U}_R\hat{\boldsymbol{c}}[t-1] + \mathbf{V}_R\boldsymbol{a}[t] + \boldsymbol{b}_R) \tag{4.20}$$

$$\boldsymbol{m}[t] = \boldsymbol{R}[t]\boldsymbol{c}[t-1] + \boldsymbol{i}[t]\boldsymbol{x}[t] \tag{4.21}$$

$$\boldsymbol{c}[t] = (1 - \boldsymbol{o}[t]) \odot \boldsymbol{m}[t] \tag{4.22}$$

$$\boldsymbol{h}[t] = \boldsymbol{o}[t] \odot \boldsymbol{m}[t] \tag{4.23}$$

Learned parameters are $\boldsymbol{W}$, $\boldsymbol{U}$, $\boldsymbol{V}$, and $\boldsymbol{b}$ for all of the gates. The normalized activation functions are, in this case, $\widehat{\sigma}$ $(\frac{\sigma(s_k)}{\sum_k \sigma(s_k)})$ for the input gate and $\widehat{\mathrm{ReLU}}$ $(\frac{\max(s_k,0)}{\sum_k \max(s_k,0)})$ for the redistribution matrix $\boldsymbol{R}$, as in the hydrology example of Hoedt et al. (2021). The product of $\boldsymbol{i}[t]\boldsymbol{x}[t]$ and $\boldsymbol{o}[t] \odot \boldsymbol{m}[t]$ are input and output fluxes, respectively.

Because this model structure is fundamentally conservative, all cell states and information transfers within the model are associated with physical units. Our objective in this study was to maintain the overall water balance in a catchment – our conserved input feature, $\boldsymbol{x}$, is precipitation in units $[mm/day]$ and our training targets are catchment discharge also in units of $[mm/day]$. Thus, all input fluxes, output fluxes, and cell states in the MC-LSTM have units of $[mm/day]$.

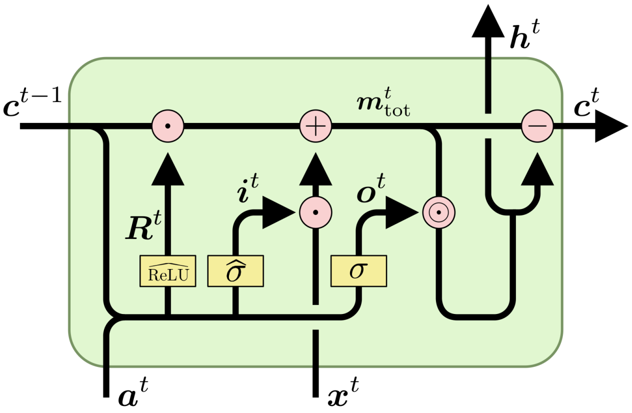

Figure 4.11: A single timestep of a Mass-Conserving LSTM with timesteps marked as superscripts for clarity. As in Figure 4.10, $\boldsymbol{c}^t$, $\boldsymbol{a}^t$, $\boldsymbol{x}^t$, $\boldsymbol{i}^t$, $\boldsymbol{o}^t$, and $\boldsymbol{R}^t$ are the cell states, conserved inputs, input features, input fluxes, output fluxes, and reshuffling matrix at time t, respectively. σ represents a standard sigmoid activation layer, $\widehat{\sigma}$ and $\widehat{\text{ReLU}}$ represent normalized sigmoid activation layers and normalized ReLU activation layer respectively. Addition and subtraction signs represent element-wise addition and subtraction, $\odot$ represents element-wise multiplication and the $\cdot$ sign represents the dot-product.

4.9 Appendix: Comparison with the U.S. National Water Model

The NOAA National Water Model (NWM) retrospective run version 2 (NWM-Rv2) is used as an additional benchmark because of its wide scale use and availability. The NWM is based on WRF-Hydro (Salas et al., 2018), which is a model that includes Noah-MP (Niu et al., 2011) as a land surface component, kinematic wave overland flow, and Muskingum-Cunge channel routing. NWM-Rv2 was previously used as a benchmark for LSTM simulations in CAMELS by Kratzert et al. (2019a), Gauch et al. (2021a) and Frame et al. (2021). Public data from NWM-Rv2 is hourly and CONUS-wide – we pulled hourly flow estimates from the USGS gauges in the CAMELS data set and averaged these hourly data to daily over the time period October 1, 1980 through September 30, 2008. As a point of comparison, Gauch et al. (2021a) compared hourly and daily LSTM predictions against the NWM-Rv2 and found that the NWM-Rv2 was significantly more accurate at the daily timescale than at the hourly timescale, whereas the LSTM did not lose accuracy at the hourly timescale vs. the daily timescale. All experiments in the present study were done at the daily timescale.

The NWM is also susceptible to the kinds of mass bias error propagation from the forcings. We can't, however, test the same hypothesis with the NWM because we do not have the capability to re-calibrate and run the NWM with Daymet forcing, as the complete set of data to run the NWM are not publicly available. The National Oceanic and Atmospheric Association (NOAA) has made publicly available a NWM retrospective run using NLDAS forcing data. This allows us to directly compare the mass balance errors with the LSTM, MC-LSTM and SAC-SMA. The NWM retrospecive run (NWM-Rv2) does not completely overlap with our test period (1989-1999). We performed the same experiment on a test period that can be compared with the NWM-Rv2, which includes training/calibrated the LSTM, MC-LSTM and SAC-SMA. The train/test period split used a test period that aligns with the availability of benchmark data from the US National Water Model. The train period included water years 1981-1995, and the test period included water years 1996-2014 (i.e., from October 1, 1995 through September 30, 2014). This was the same training period used by Newman et al. (2017) and Kratzert et al. (2019a), but with an extended test period. This train/test split was used because the NWM-Rv2 data record is not long enough to accommodate the train/test split used by previous studies (item above in this list).

The NWM-Rv2 was calibrated by NOAA personnel on about 1400 basins with NLDAS forcing data and includes a regionalization strategy that attempts to use the calibrated parameters across basins not included in the calibration set, however most of the CAMELS basins are included in that calibration set. The NWM-Rv2 calibration time period is on water years 2009-2013. Because of the inconsistencies in the time period and basins included in the calibration, we cannot directly compare the NWM-Rv2 to the other models. But we include the NWM-Rv2 here as an appendix because it is relevant to the hydrologic community to see, even if not directly comparable, the results of a physics-based model.

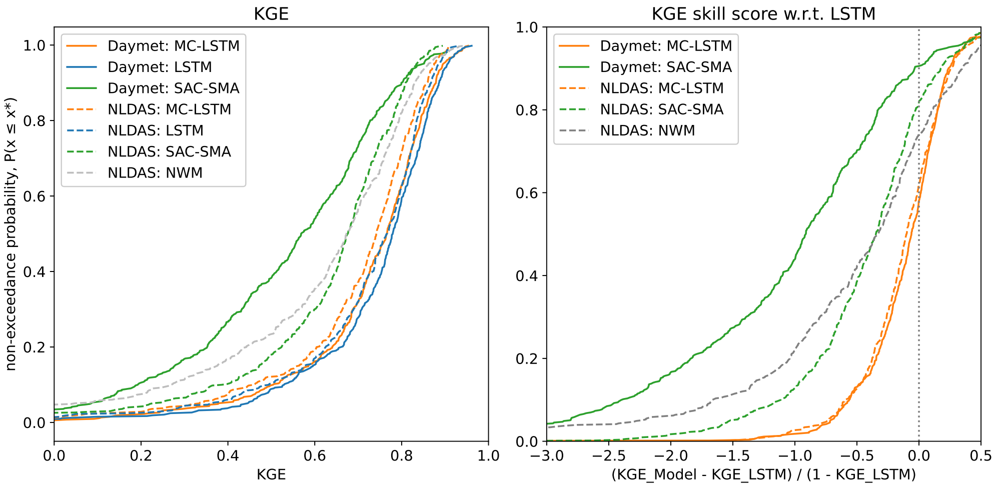

Figure 4.12: Left: Cumulative distribution of the Kling-Gupta Efficiency (KGE) for four models on two different forcing products. Right: KGE skill score with respect to the unconstrained LSTM.

The information loss between the LSTM model and the constrained MC-LSTM and SAC-SMA models for this test period, shown in Figure 4.12 is similar to that of the test period shown in the main text, Figure 4.6. He we also show the information loss from the NWM, which actually shows more information loss than SAC-SMA. This is likely because the NWM has more constraints, that come in the form of a multi-layered modeling chain. The NWM starts with a land surface model, which causes runoff across a terrain routing model, which is also two-way coupled with the land model, and finally the terrain model feeds into the channel routing model, which provides an estimate of streamflow. There are multiple steps along that modeling chain that cause different amounts of information loss.

Figure 4.13 shows the cumulative density functions (CDFs) of long-term cumulative discharge from the 484 CAMELS basins from the models during the 1996-2014 test period. Note that we excluded basins that did not have a complete observation time series throughout the entire test period. The LSTM and MC-LSTM both predicted streamflows that result in more accurate long-term cumulative discharge than the

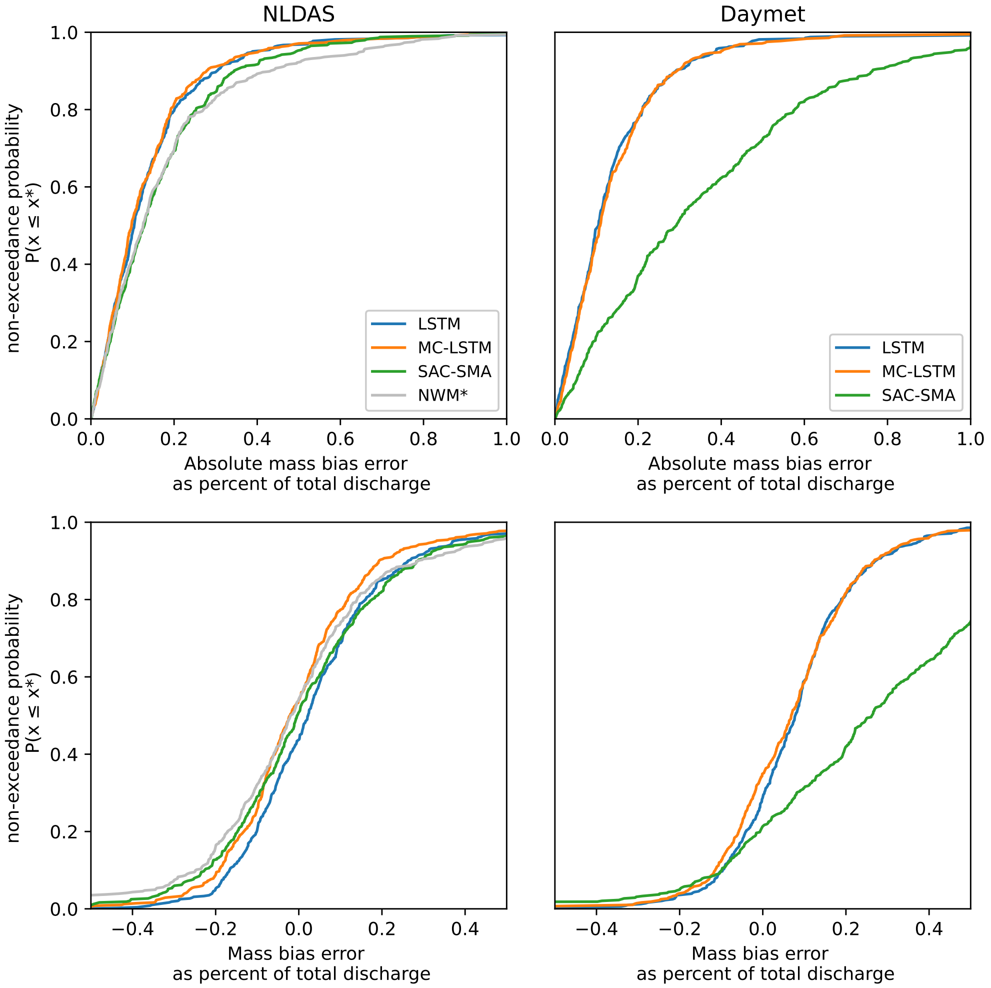

Figure 4.13: Distribution of mass balance error across the 484 basins. Top: Cumulative distribution curves of the absolute mass error form models forced with NLDAS (left) and Daymet (right). Bottom: Cumulative distributions of mass error from models forced with NLDAS (left) and Daymet (right).

calibrated SAC-SMA model. The LSTM and the MC-LSTM performed roughly similarly on both NLDAS and Daymet

Figure 4.14 shows the Mass balance results from for the three models with both Daymet and NLDAS forcings. The result of the SAC-SMA simulation with Daymet forcings shows a clear positive mass bias error in the eastern half of CONUS. The result of the simulation with NLDAS forcings shows a mix of positive and negative mass bias throughout CONUS.

Figure 4.15 shows the mass bias errors for the model runs with NLDAS forcings in box and whisker plots for the U.S. Water Resources Regions. A mass bias error is clearly shows for SAC-SMA in the easter CONUS regions, while the LSTM and MC-LSTM do not express this pattern. There is generally a correlation between the three models, where the regions with high mass bias error are expressed by all three models. For instance, the Upper Colorado region shows low mass bias error for all three models, and the Lower Colorado shows a relatively high mass bias error by all three models.

Figure 4.16 shows the mass bias errors for the model runs with Daymet forcings in box and whisker plots for the U.S. Water Resources Regions. The NWM-Rv2 has high outliers in the Central CONUS regions. There is a correlation of mass bias error across all four models, where when the conceptual model (SAC-SMA), the physics-based model (NWM-Rv2), the physics informed ML Model (MC-LSTM) and the pure data driven model (LSTM) all show relatively small to moderate mass bass bias error in the Northeastern CONUS, high mass bias error in the central CONUS and moderate mass bias in west coast regions. The exception to this trend is that the Great Basin has a low mass bias error from LSTM, MC-LSTM and SAC-SMA, but a high mass bias error from NWM.

4.10 Code and data availability

All LSTMs and MC-LSTMs were trained using the NeuralHydrology Python library available at https://github.com/neuralhydrology/neuralhydrology. A snapshot of the exact version that we used is available at https://github.com/jmframe/mclstm_2021_extrapolate/neuralhydrology and under DOI number 10.5281/zenodo.5051961. Code for calibrating SAC-SMA is from https://github.com/Upstream-Tech/SACSMA-SNOW17, which includes the SpotPy calibration library https://pypi.org/project/spotpy/. Input data for all model runs except the NWM-Rv2 came from the public NCAR CAMLES repository https://ral.ucar.edu/solutions/products/camels and were used according to instructions outlined in the NeuralHydrology readme. NWM-Rv2 data are available publicly from https://registry.opendata.aws/nwm-archive/. All model output data generated by this project is available on the CUAHSI HydroShare platform under a DOI number https://doi.org/10.4211/hs.d750278db868447dbd252a8c5431affd. Interactive Python scripts for all post-hoc analysis reported in this paper, including calculating metrics and generating tables and figures, are available at https://github.com/jmframe/mclstm_2021_mass_balance.

Table 4.6: Median performance metrics (plus or minus the 95% confidence interval) across 484 basins calculated on the test period 1996-2014 with two separate forcing products.

	Daymet forcing			NLDAS forcing			
Metric	LSTM	MC-LSTM	SAC-SMA	LSTM	MC-LSTM	SAC-SMA	NWM*
NSE	0.74 ± -0.02	0.74 ± -0.02	0.59 ± -0.08	0.71 ± -0.05	0.72 ± -0.02	0.63 ± -0.05	0.63 ± -0.05
KGE	0.78 ± -0.02	0.77 ± -0.02	0.56 ± n/a	0.77 ± -0.02	0.74 ± -0.02	0.68 ± -0.02	0.67 ± -0.05
Pearson-r	0.88 ± -0.01	0.88 ± -0.01	0.81 ± n/a	0.86 ± -0.01	0.86 ± -0.01	0.81 ± -0.01	0.82 ± -0.01
Alpha-NSE	0.96 ± -0.02	0.91 ± -0.01	0.88 ± -0.02	0.94 ± -0.02	0.87 ± -0.02	0.83 ± -0.02	0.85 ± -0.03
Beta-NSE	0.03 ± -0.01	0.03 ± -0.01	0.13 ± -0.02	0.01 ± -0.01	-0.01 ± -0.01	-0.01 ± n/a	-0.01 ± n/a
Peak-Timing	0.34 ± − 0.03	0.33 ± − 0.03	0.45 ± − 0.06	0.38 ± -0.03	0.4 ±-0.03	0.53 ± -0.06	0.54 ± -0.05

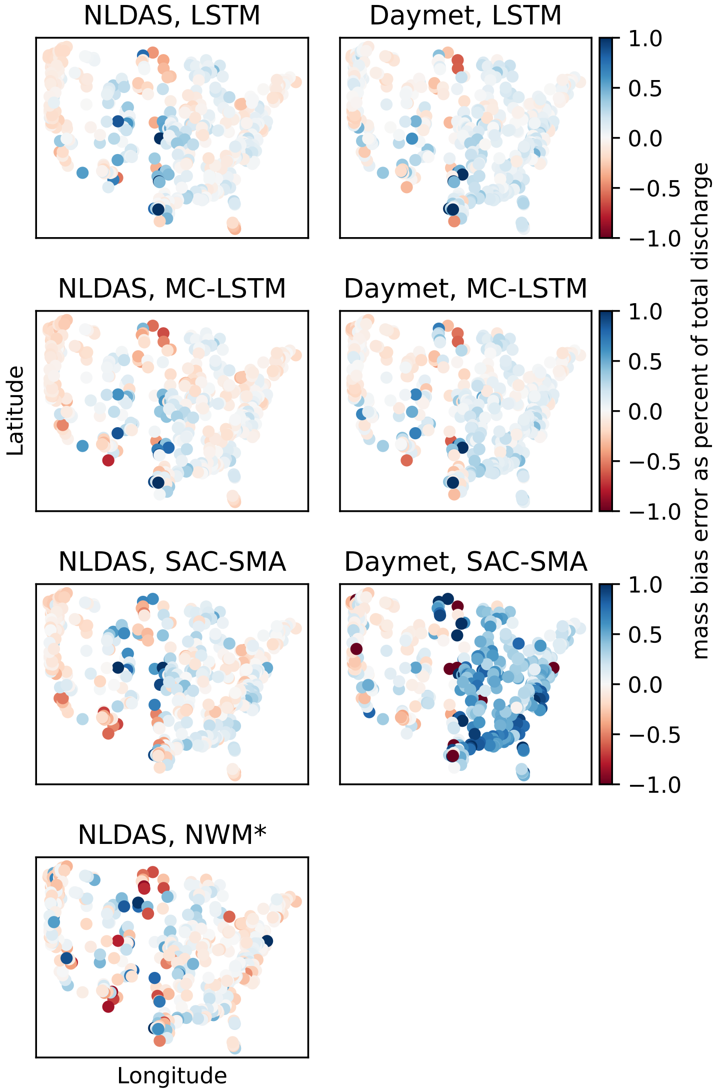

Figure 4.14: Geospatial distribution of long term positive or negative mass bias error. The left and right columns show the results with NLDAS and Daymet meteorological forcing data, respectively. The four rows are associated (from top to bottom) with LSTM, MC-LSTM, SAC-SMA and NWM. The astrisct (*) on the bottom left sub-plot label indicates that the NWM was not calibrated on the same time period as the LSTM, MC-LSTM and SAC-SMA models.

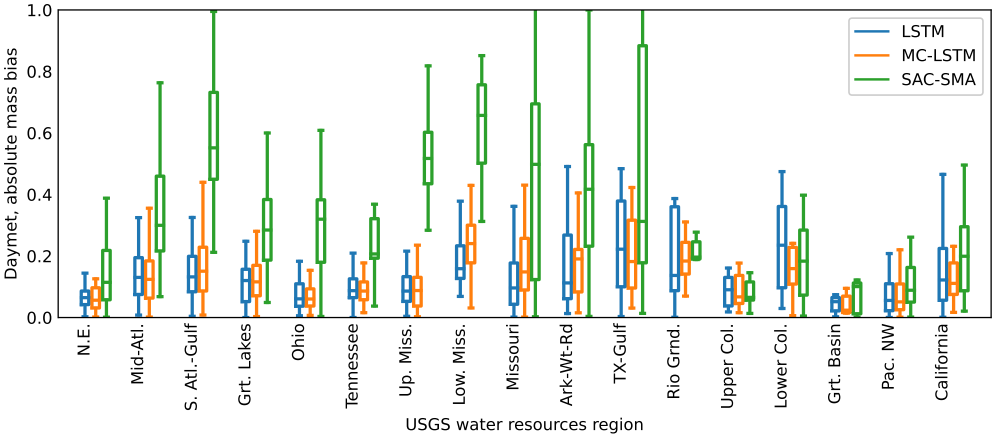

Figure 4.15: Regional mass balance errors from LSTM, MC-LSTM and SAC-SMA with Daymet forcings. Souris-Red-Rainy region (Hydrologic Unit Code 09) is absent due to a lack of sufficient basins.

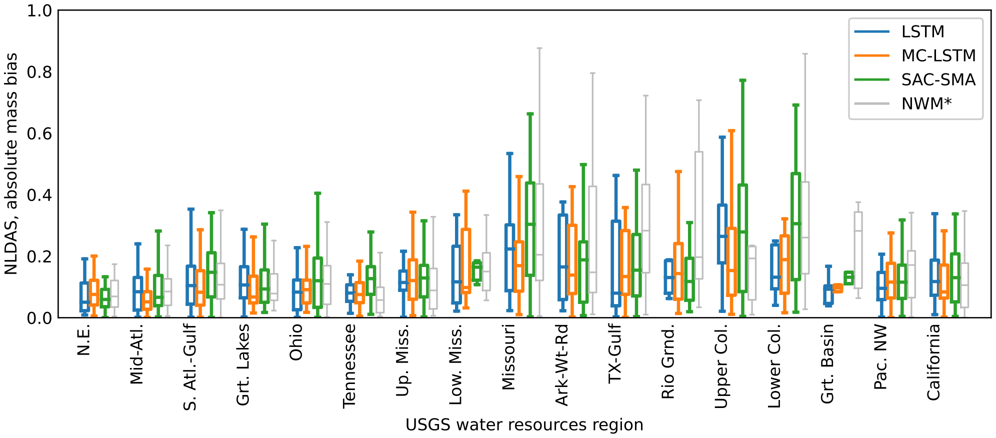

Figure 4.16: Regional mass balance errors from LSTM, MC-LSTM and SAC-SMA with NLDAS forcings. Souris-Red-Rainy region (Hydrologic Unit Code 09) is absent due to a lack of sufficient basins. The astrict on the NWM label indicates that the model was calibrated on a separate time period than the other three models, and is thus not directly comparable.

REFERENCES

Addor, N., Newman, A. J., Mizukami, N., and Clark, M. P. (2017). The camels data set: catchment attributes and meteorology for large-sample studies. *Hydrology and Earth System Sciences (HESS)*, 21(10):5293–5313.

Bennett, A. and Nijssen, B. (2021). Deep Learned Process Parameterizations Provide Better Representations of Turbulent Heat Fluxes in Hydrologic Models. *Water Resources Research*, 57(5):1–14.

Beven, K. (1989). Changing ideas in hydrology — the case of physically-based models. *Journal of Hydrology*, 105(1):157–172.

Beven, K. (2019). Towards a methodology for testing models as hypotheses in the inexact sciences. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 475.

Beven, K. (2020). Deep learning, hydrological processes and the uniqueness of place. *Hydrological Processes*, 34(16):3608–3613.

Beven, K. and Westerberg, I. (2011). On red herrings and real herrings: Disinformation and information in hydrological inference. *Hydrological Processes*, 25(10):1676–1680.

Beven, K. J., Smith, P. J., and Freer, J. E. (2008). So just why would a modeller choose to be incoherent? *Journal of Hydrology*, 354(1-4):15–32.

Chow, V. T., Maidment, D. R., and Mays, L. W. (1988). *Applied Hydrology Chow 1988.pdf*. McGraw-Hill.

Cover, T. M. and Thomas, J. A. (2005). *Elements of Information Theory.*

Daw, A., Thomas, R. Q., Carey, C. C., Read, J. S., Appling, A. P., and Karpatne, A. (2020). Physics-Guided Architecture (PGA) of Neural Networks for Quantifying Uncertainty in Lake Temperature Modeling. *Proceedings of the 2020 SIAM International Conference on Data Mining*, pages 532–540.

Frame, J. M. (2022). Mass balance paper, supplemental figures, https://doi.org/10.4211/hs.03b262396bd54486b7120c37905322e4.

Frame, J. M., Kratzert, F., Klotz, D., Gauch, M., Shelev, G., Gilon, O., Qualls, L. M., Gupta, H. V., and 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., and Nearing, G. S. (2021). Post-Processing the National Water Model with Long Short-Term Memory Networks for Streamflow Predictions and Model Diagnostics. *Journal of the American Water Resources Association*, pages 1–21.

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

Gauch, M., Mai, J., and Lin, J. (2021b). The proper care and feeding of camels: How limited training data affects streamflow prediction. *Environmental Modelling & Software*, 135:104926.

Gupta, H. V., Kling, H., Yilmaz, K. K., and 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. and Nearing, G. S. (2014). Debates—the future of hydrological sciences: A (common) path forward? using models and data to learn: A systems theoretic perspective on the future of hydrological science. *Water Resources Research*, 50(6):5351–5359.

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

Hoedt, P.-J., Kratzert, F., Klotz, D., Halmich, C., Holzleitner, M., Nearing, G. S., Hochreiter, S., and Klambauer, G. (2021). Mc-lstm: Mass-conserving lstm. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 4275–4286. PMLR.

Houska, T., Kraft, P., Chamorro-Chavez, A., and Breuer, L. (2019). Spotpy: A python library for the calibration, sensitivity-and uncertainty analysis of earth system models. In *Geophysical Research Abstracts*, volume 21.

Jia, X., Willard, J., Karpatne, A., Read, J. S., Zwart, J. A., Steinbach, M., and Kumar, V. (2020). Physics-Guided Machine Learning for Scientific Discovery: An Application in Simulating Lake Temperature Profiles. pages 1–25.

Jiang, S., Zheng, Y., and Solomatine, D. (2020). Improving ai system awareness of geoscience knowledge: Symbiotic integration of physical approaches and deep learning. *Geophysical Research Letters*, 47(13):e2020GL088229.

Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., and Yang, L. (2021). Physics-informed machine learning. *Nature Reviews Physics*, (May).

Klotz, D., Kratzert, F., Gauch, M., Keefe Sampson, A., Brandstetter, J., Klambauer, G., Hochreiter, S., and Nearing, G. (2021). Uncertainty estimation with deep learning for rainfall–runoff modelling. *Hydrology and Earth System Sciences Discussions*, pages 1–32.

Knoben, W. J. M., Freer, J. E., and Woods, R. A. (2019). Technical note : Inherent benchmark or not ? Comparing Nash – Sutcliffe and Kling – Gupta efficiency scores. pages 4323–4331.

Kratzert, F., Klotz, D., Brenner, C., Schulz, K., and 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., and Nearing, G. S. (2019a). Toward improved predictions in ungauged basins: Exploiting the power of machine learning. *Water Resources Research*, 55(12):11344–11354.

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

Kratzert, F., Klotz, D., Shalev, G., Klambauer, G., Hochreiter, S., and Nearing, G. (2019b). 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.

Lees, T., Reece, S., Kratzert, F., Klotz, D., Gauch, M., De Bruijn, J., Kumar Sahu, R., Greve, P., Slater, L., and Dadson, S. J. (2022). Hydrological concept formation inside long short-term memory (lstm) networks. *Hydrology and Earth System Sciences*, 26(12):3079–3101.

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

Nearing, G., Research, G., Kratzert, F., Klotz, D., Hoedt, P.-J., Klambauer, G., Hochreiter, S., Gupta, H., Nevo, S., and Matias, Y. (2020a). A Deep Learning Architecture for Conservative Dynamical Systems: Application to Rainfall-Runoff Modeling. *AI for Earth Sciences Workshop at NEURIPS 2020.*

Nearing, G., Sampson, A. K., Kratzert, F., and Frame, J. (2020b). Post-processing a conceptual rainfall-runoff model with an lstm.

Nearing, G. S. and Gupta, H. V. (2015). The quantity and quality of information in hydrologic models. *Water Resources Research*, 51(1):524–538.

Nearing, G. S., Kratzert, F., Sampson, A. K., Pelissier, C. S., Klotz, D., Frame, J. M., Prieto, C., and Gupta, H. V. (2020c). What role does hydrological science play in the age of machine learning? *Water Resources Research*, page e2020WR028091.

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.

Newman, A. J., Mizukami, N., Clark, M. P., Wood, A. W., Nijssen, B., and Nearing, G. (2017). Benchmarking of a physically based hydrologic model. *Journal of Hydrometeorology*, 18(8):2215–2225.

Niu, G.-Y., Yang, Z.-L., Mitchell, K. E., Chen, F., Ek, M. B., Barlage, M., Kumar, A., Manning, K., Niyogi, D., Rosero, E., et al. (2011). The community noah land surface model with multiparameterization options (noah-mp): 1. model description and evaluation with local-scale measurements. *Journal of Geophysical Research: Atmospheres*, 116(D12).

Pelissier, C., Frame, J., and Nearing, G. (2019). Combining Parametric Land Surface Models with Machine Learning.

Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., et al. (2019). Deep learning and process understanding for data-driven earth system science. *Nature*, 566(7743):195–204.

Salas, F. R., Somos-Valenzuela, M. A., Dugger, A., Maidment, D. R., Gochis, D. J., David, C. H., Yu, W., Ding, D., Clark, E. P., and Noman, N. (2018). Towards real-time continental scale streamflow simulation in continuous and discrete space. *JAWRA Journal of the American Water Resources Association*, 54(1):7–27.

Shen, C., Chen, X., and Laloy, E. (2021). Editorial: Broadening the Use of Machine Learning in Hydrology. *Frontiers in Water*, 3(May):1–4.

Sivapalan, M., Takeuchi, K., Franks, S. W., Gupta, V. K., Karambiri, H., Lakshmi, V., Liang, X., McDonnell, J. J., Mendiondo, E. M., O'Connell, P. E., Oki, T., Pomeroy, J. W., Schertzer, D., Uhlenbrook, S., and Zehe, E. (2003). IAHS Decade on Predictions in Ungauged Basins (PUB), 2003-2012: Shaping an exciting future for the hydrological sciences. *Hydrological Sciences Journal*, 48(6):857–880.

Thornton, P., Thornton, M., Mayer, B., Wilhelmi, N., Wei, Y., Devarakonda, R., and Cook, R. (2014). Daymet: Daily surface weather data on a 1-km grid for north america, version 2.

Tsai, W.-P., Pan, M., Lawson, K., Liu, J., Feng, D., and Shen, C. (2020). From parameter calibration to parameter learning: Revolutionizing large-scale geoscientific modeling with big data. *arXiv preprint arXiv:2007.15751*.

USGS (1987). Hydrologic Unit Maps: U.S. Geological Survey Water-Supply Paper 2294.

Willard, J., Jia, X., Xu, S., Steinbach, M., and Kumar, V. (2021). Integrating Scientific Knowledge with Machine Learning for Engineering and Environmental Systems. 1(1):1–35.

Xia, Y., Mitchell, K., Ek, M., Sheffield, J., Cosgrove, B., Wood, E., Luo, L., Alonge, C., Wei, H., Meng, J., Livneh, B., Lettenmaier, D., Koren, V., Duan, Q., Mo, K., Fan, Y., and Mocko, D. (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: Oceans*, 117(3). Copyright: Copyright 2018 Elsevier B.V., All rights reserved.

Xie, K., Liu, P., Zhang, J., Han, D., Wang, G., and Shen, C. (2021). Physics-guided deep learning for rainfall-runoff modeling by considering extreme events and monotonic relationships. *Journal of Hydrology*, 603(PC):127043.

Zhao, W. L., Gentine, P., Reichstein, M., Zhang, Y., Zhou, S., Wen, Y., Lin, C., Li, X., and Qiu, G. Y. (2019). Physics-Constrained Machine Learning of Evapotranspiration. *Geophysical Research Letters*, 46(24):14496–14507.

CHAPTER 5

LSTM AS A RAINFALL-RUNOFF MODULE FOR THE NEXT GENERATION U.S. NATIONAL WATER MODEL FRAMEWORK

Jonathan M. Frame, Fred L. Ogden, Scott D. Peckham, Jessica Garrett, Nels J. Frazier, Rachel McDaniel, Brian Avant, David Blodgett, Edward Clark, Brian Cosgrove, Shengting Cui, Luciana Kindl da Cunha, Trey Flowers, Thomas Graziano, Donald W. Johnson, David Mattern, Kieth Jennings, Matt Williamson, Mehdi Rezaeianzadeh, Andy Wood, Michael J. Johnson

In preparation as a section in an upcoming paper on the Next Generation U.S. National Water Model

5.1 Abstract

Deep learning (DL) has been shown to make extremely accurate predictions in simulating various hydrologic processes. Operational water resources management and prediction need to begin taking advantage of these new modeling techniques, even as the research continues to evolve. The Next Generation U.S. National Water Model Prototype Framework (Nextgen) is an opportunity to combine state-of-the-art DL research with robust operational modeling needs. Nextgen is model agnostic, simulates fluxes of interest across arbitrary scales and control volumes, and uses evidence-based evaluation of different approaches. Two strict requirements are enforced to Nextgen models and modeling components, 1) the function on standardized surface water hydrology conceptualizations, and 2) the modules run from a standardized list of general commands. A long short-term memory (LSTM) network is developed as a rainfall-runoff module, as one of the four core options for large scale deployment. We show the results of a three-year simulation in the

Northeast region of the United States. The qualitative results show that the LSTM matches the flow distribution of observed stream gauges.

5.2 Introduction

The rapid advancement of deep learning (DL) applications for hydrology gives us an opportunity to make forecasts of the hydrosphere with unprecedented accuracy (Kratzert et al., 2019; Nearing et al., 2020). New approaches and applications of DL are being developed to span almost every aspect of the hydrosphere, including groundwater (Tao et al., 2022), soil moisture (O and Orth, 2021; Fang et al., 2017), energy fluxes (Bennett and Nijssen, 2021), reservoir releases (Yang et al., 2019), etc. While the search for hydrologic laws has generated general knowledge valuable for developing water resources management strategies, the effort has fallen short of improving our predictive capabilities to implement well throughout management practices. We are now obligated to exploit the demonstrated power of DL for this purpose (Nearing et al., 2020).

The U.S. National Water Model (NWM) is operational across the Continental United States (CONUS), and is expanding to Hawaii, Alaska and U.S. territories. This model is a specific configuration of the community WRF-Hydro modeling system which provides streamflow predictions for 2.7 million reaches and other hydrologic information on 1km and 250 m grids (Gochis et al., 2019).The Next Generation National Water Model Prototype Framework (Nextgen) is a framework for continental-scale modeling with the ability for easy integration of state-of-the-art research (top performing models and modeling strategies). Let's first be clear that this is not a DL specific framework. In fact, Nextgen is model agnostic. This means that Nextgen can run any model that conforms to a specific standard (described in the next section) for full integration into the NWM. This allows 1) researchers to benchmark their work directly against the operational model, and 2) users to adapt/modify the model to their unique needs.

Modeling frameworks have previously been successful at streamlining development, testing and deployment for local applications (site to watershed scale)

(Watson and Rahman, 2004). No framework of this sort, however, has been designed for continental scale simulation. Nearing et al. (2020), calls for state-of-the-art spatiotemporal DL models in all areas of hydrology. We believe that Nextgen fills this need allowing DL models representing all aspects of surface water hydrology to form the predictive crux of any spatiotemporal distributed hydrological models.

We need this approach because 1) we want to take advantage of the benefits of every/any model component, at any spatiotemporal scale, and 2) keeping up with the rapidly advancing science of hydrologic prediction is beyond the scope of one individual or team. We want this approach because the on-the-fly benchmarking approach incentivizes the research community to develop directly within the NWM, ensuring model compatibility.

5.3 Next generation U.S. National Water Model

The Next Generation National Water Model Prototype Framework (Nextgen) is designed for simulating surface water hydrology across the United States and territories (Ogden et al., 2021). The distinction between Nextgen and the existing operational National Water Model (NWM) is three tenants of modeling philosophy:

- Allow arbitrary methods/models
- Simulate fluxes of interest across arbitrary scales and control volumes
- Evidence based evaluation of different approaches

5.3.1 Hy_features and hydrofabric

Hy_features are a standardized set of surface water hydrology conceptualizations (Blodgett et al., 2021). These are not distinctly different from most general hydrological concepts, but Hy_features provides precise definitions that will prevent misalignment of models, model components, modules, solvers, discretization, etc. For instance, Blodgett et al. (2021) described precisely that drainage basins have one - and only one - headwater source area and a single mainstem that flows to a single outlet. This example might come off as rudimentary, but these definitions are critically important to a community effort to make contributions to a single model.

5.3.2 Basic Model Interface (BMI)

Component-based software engineering enables the integration of plug-and-play components, but significant additional challenges must be addressed in any specific domain in order to produce a usable development and simulation environment that is also going to encourage contributions and adoption by entire communities (Peckham et al., 2013).

5.3.3 Integration

Hy_features standardized hydrologic conceptualizations and component based modeling with the BMI are strict requirements for integration into NextGen. This will constrain AI development to be applicable to a fully coupled hydrological model, instead of a piecemeal approach. For instance, developing a deep learning model that generates forecasts of aquifer recharge from surface water levels is only partially useful if it does not also take into account the vadose zone in between. Although such a model can find a suitable application, it is not directly applicable to a large-scale model of the hydrologic cycle.

We develop the capability to integrate the rapidly advancing machine learning approaches and applications from the research phase through operational deployment. This requires a flexible tiered development environment.

5.4 LSTM for Nextgen

We add a long short-term memory (LSTM) network (Hochreiter and Schmidhuber, 1997) based rainfall-runoff model. This requires the forward pass only, and does not require the ability for backwards propagation. The LSTM is built as a module callable by Nextgen to adhere to all the requirements of Hy_features and BMI. Any predictive functionality of an ML, DL and/or AI (data-driven) based model must share boundaries, states and fluxes with its dynamic neighbors. So, in accordance with BMI, our

LSTM implementation has been defined completely as a method with a particular name, standard arguments and a return value. Our LSTM acts exactly as any other Earth systems model. A set of state values are initialized, the model (module) code is called within a time loop, state values are updated and used to calculate the flux of interest which is then returned. In this way, integrating the LSTM is no different than building any type of model, be it process-based or conceptual. A strength of our approach is the ability to use specific models for specific applications. For instance if we know that a basin contains a series of strategically managed reservoirs, and the runoff is not consistent with a typical rainfall-runoff pattern, we can 'plug and play' the appropriate reservoir model we want to use.

5.5 Example integration of deep learning on a large scale

We applied the LSTM to the HUC 01 (New England) region. Streamflow in HUC 01 includes snowmelt and liquid precipitation runoff. The hydrofabric for this region includes about 10,000 catchments of 3-15 km^2 and has a combined 191,020 km^2 drainage area. The HUC 01 region includes 26 of the CAMELS basins.

Figure 5.1 shows the average flow duration curve (FDC) for the example simulation period across the 1000 randomly chosen HUC 01 subcatchments over the three-year simulation period, as compared to the observed flows in those CAMELS catchments. The average LSTM Nextgen predicted streamflows match up well with the higher flows (lower percent exceedance), but tends to slightly overestimates the lowest flows (highest percent exceedance). The minimum and maximum FDC of the HUC 01 LSTM Nextgen have a much greater variation from the mean than the observed CAMELS FDC. This is not necessarily a fair comparison, though, since the Nextgen subcatchments are much smaller than the CAMELS catchments, so it is reasonable that their variation would be greater, and still contribute to a good streamflow prediction at the gauge location.

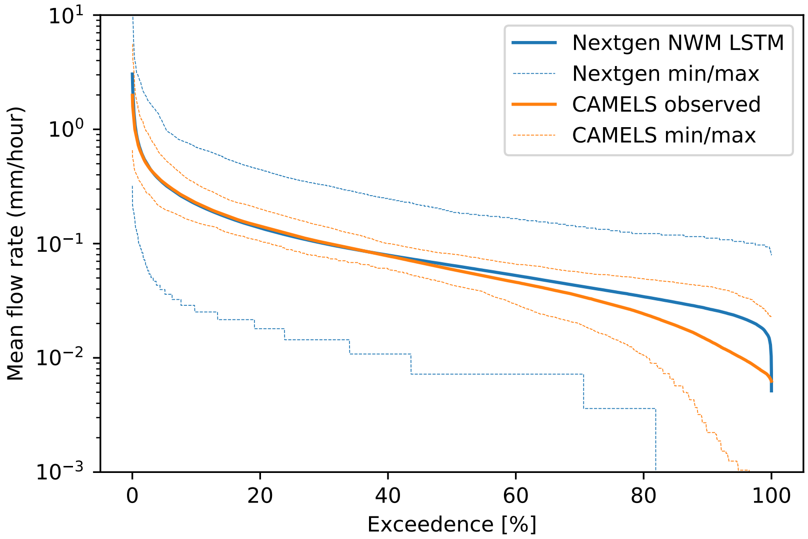

Figure 5.1: Average flow duration curves for HUC 01, New England, from streamflow observation and model predictions using the LSTM module within the Next Generation U.S. National Water Model

Figure 5.2 shows a snapshot of the LSTM module simulating runoff at 10,000 catchments encompassing the HUC 01 (New England) region.

5.6 Discussion

The Next Generation U.S. National Water Model is capable of using LSTM to make runoff predictions at each catchment within the hydrofabric. Development of Nextgen, and the LSTM module, is on-going, and scheduled for operational deployment in 2024. The conceptual and process-based modules within Nextgen need to be calibrated, which will include some sort of rationalization strategy to apply to ungauged basins. The LSTM does not need to be calibrated in the same manner, and no regionalization is required, as LSTM has been shown to make good predictions in ungauged basins (Kratzert et al., 2019).

In principle, any such DL architecture can be used in this module, as long as the forward pass is callable by BMI commands. Training must be done outside of the Nextgen framework. Trained model weights and biases are then loaded in during the initialization of the module instance.

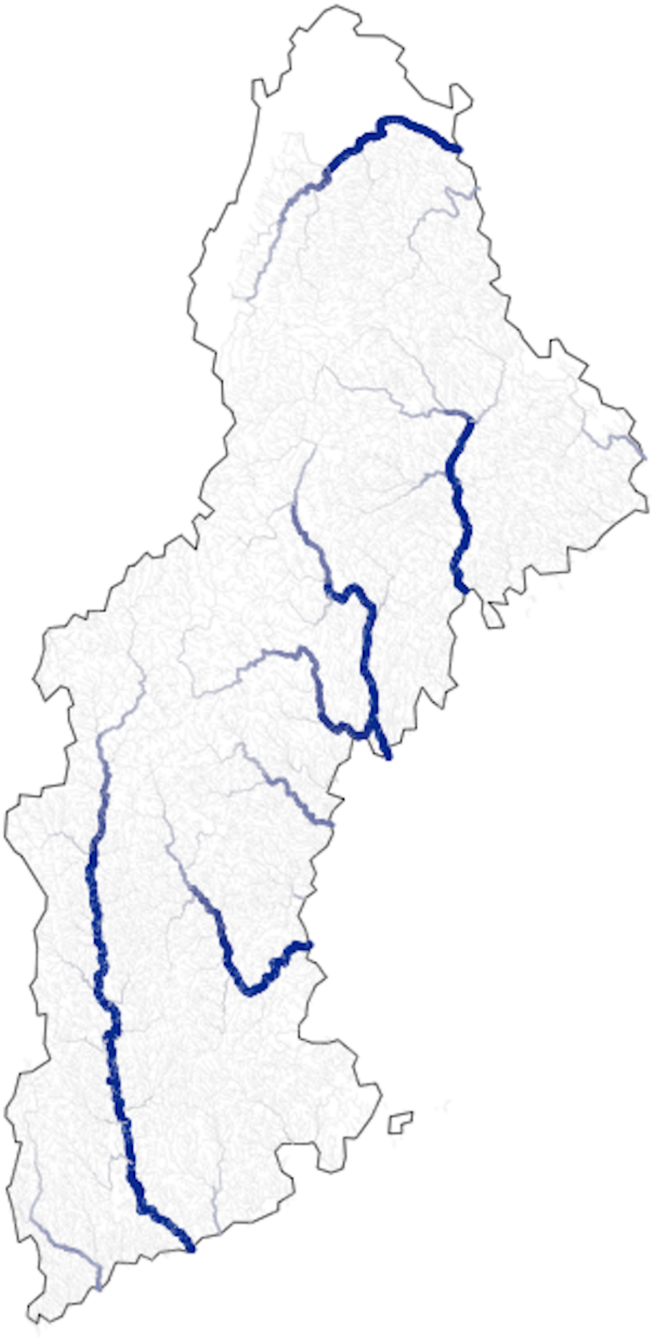

Figure 5.2: Streamflow predictions ($\frac{ft^3}{s}$) in HUC 01, New England, using the LSTM module within the Next Generation U.S. National Water Model.

REFERENCES

Bennett, A. and Nijssen, B. (2021). Deep Learned Process Parameterizations Provide Better Representations of Turbulent Heat Fluxes in Hydrologic Models. *Water Resources Research*, 57(5).

Blodgett, D., Johnson, J. M., Sondheim, M., Wieczorek, M., and Frazier, N. (2021). Mainstems: A logical data model implementing mainstem and drainage basin feature types based on WaterML2 Part 3: HY Features concepts. *Environmental Modelling and Software*, 135(November 2020):104927.

Fang, K., Shen, C., Kifer, D., and Yang, X. (2017). Prolongation of SMAP to Spatiotemporally Seamless Coverage of Continental U.S. Using a Deep Learning Neural Network. *Geophysical Research Letters*, 44(21):11,030–11,039.

Gochis, D., Yates, D., Dugger, A., McCreight, J., Barlage, M., RafieeNasab, A., Karsten, L., Read, L., Zhang, Y., McAllister, M., Cabell, R., and FitGerald, K. (2019). Overview of National Water Model CalibrationGeneral Strategy Optimization.

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

Kratzert, F., Klotz, D., Herrnegger, M., Sampson, A. K., Hochreiter, S., and Nearing, G. S. (2019). Towards Improved Predictions in Ungauged Basins: Exploiting the Power of Machine Learning. *Water Resources Research*, page 2019WR026065.

Nearing, G. S., Kratzert, F., Sampson, A. K., Pelissier, C. S., Klotz, D., Frame, J. M., Prieto, C., and Gupta, H. V. (2020). What Role Does Hydrological Science Play in the Age of Machine Learning? *Water Resources Research*.

O, S. and Orth, R. (2021). Global soil moisture data derived through machine learning trained with in-situ measurements. *Scientific Data*, 8(1):1–14.

Ogden, F. L., Avant, B., Blodgett, D., Clark, E., Coon, E., Cosgrove, B., Cui, S., Kindl da Cunha, L., Farthing, M., Flowers, T., Frame, J. M., Frazier, N. J., Graziano, T., Gutenson, J., Johnson, D. W., Loney, D., Mattern, D., McDaniel, R., Moulton, J., Peckham, S. D., Jennings, K., Savant, G., Tubbs, C., Williamson, M., Garrett, J., Wood, A., and Johnson, J. M. (2021). The next generation water resources modeling framework: Open source, standards based, community accessible, model interoperability for large scale water prediction. american geophysical union, fall meeting 2021.

Peckham, S. D., Hutton, E. W., and Norris, B. (2013). A component-based approach to integrated modeling in the geosciences: The design of CSDMS. *Computers and Geosciences*, 53:3–12.

Tao, H., Hameed, M. M., Marhoon, H. A., Zounemat-Kermani, M., Salim, H., Sungwon, K., Sulaiman, S. O., Tan, M. L., Sa'adi, Z., Mehr, A. D., Allawi, M. F., Abba, S., Zain, J. M., Falah, M. W., Jamei, M., Bokde, N. D., Bayatvarkeshi, M., Al-Mukhtar, M., Bhagat, S. K., Tiyasha, T., Khedher, K. M., Al-Ansari, N., Shahid, S., and Yaseen, Z. M. (2022). Groundwater level prediction using machine learning models: A comprehensive review. *Neurocomputing*.

Watson, F. G. and Rahman, J. M. (2004). Tarsier: A practical software framework for model development, testing and deployment. *Environmental Modelling and Software*, 19(3):245–260.

Yang, S., Yang, D., Chen, J., and Zhao, B. (2019). Real-time reservoir operation using recurrent neural networks and inflow forecast from a distributed hydrological model. *Journal of Hydrology*, 579(May):124229.

CHAPTER 6

CONCLUSION

In this dissertation I provide evidence to suggest that the ”theory” part of the ”theory-guided” machine learning fails to provide value over a data-driven recurrent neural network for making predictions of streamflow (over large spatial scales) from atmospheric forcings. In other words, a hydrologist acting with the goal of making streamflow predictions at the continental scale is better off using a recurrent neural network (pure DL model, probably an LSTM) without introducing any information from the physical or conceptual representations of hydrologic processes. To put it quite bluntly, we are better off (at the moment anyway) letting a neural network learn relationships from data, rather that formulating equations (parameterizations) that describe hydrologic processes.

6.1 Research projects

In Chapter 2 concludes that using the output of a PB model as an input to a DL model (post-processing) can de-stabilize predictive performance, with respect to the DL model without the influence of the PB model. The results do show, however, that post-processing the PB hydrology model with DL improves upon the performance of the physics-based model itself. So, in the absence of sufficient input data for the DL model (for instance, satellite inputs that are not available due to sensor issues), a PB post-processor could still be an improvement. The results also show that we can use the post-processor to diagnose specific problems with the PB model.

In Chapter 3 concludes that the theory of mass conservation in a ”theory-guided deep learning model” (also known as a hybrid, (HB) model) degrades predictive performance during extreme runoff events. This is an important, and perhaps surprising result, as a major criticism of DL models has been the potential for failing to capture these types of extreme events. The results indicate that DL models actually outperform PB and HB models when extrapolating to events outside the training set.

In Chapter 4 concludes that mass conservation itself is not a particularly useful theory for eliminating long term mass biases in a watershed model. We also see that the hydrologic "theory" encoded in hydrology models is a large source of uncertainty than any biases from precipitation or streamflow measurements. It is possible that any conceptualization of a hydrological system that is enforced on a model will degrade performance as compared to DL. We do not yet have the capability to aggregate all the hydrologic sub-processes correctly with PB models.

Chapter 5 describes a research-to-operation type applied project. LSTM is one of the first prototype modules in the Next Generation U.S. National Water Model Prototype Framework. The decision to use the "vanilla" purely DL-based LSTM, instead of one of the "theory-guided" HB versions of LSTM is based on the results discussed above. The results in Chapter 5 show a large scale, three-year, hydrological simulation of the Northeast Hydrology Unit Code 01 (New England).

6.2 Looking forward in combining hydrologic theory with deep learning

There is presently no strong evidence that DL needs to be merged with hydrologic theory. Perhaps, hydrologic theory should be re-written around DL. A watershed, composed of many sub-processes, is a giant combination of phase transitions (i.e., water moves through some control volume as a result of some suddenly occurring hydraulic gradient until the water mass is either depleted, or the hydraulic gradient is zero) that occur at many spatial scales. Neural networks are perfect to represent dynamic movement of water during spontaneous activation of water movement. We could develop hydrologic theory based on non-reciprocal phase transitions (Fruchart et al., 2021). Individual, and combinations of neurons within a neural network, can be expressed as representations of hydrologic sub-processes. We have the ability to analyze a neural network trained to predict streamflow from atmospheric forcings, and decipher which watershed processes are represented by the pieces of the network (Lees et al., 2021). So, it is reasonable to suspect that the sub-network containing physically intuitive representations of hydrologic processes (e.g., snowmelt and soil moisture) is the hydrologic theory.

The DL models presented in this dissertation generalize the hydrologic processes across basins using static catchment attributes, which capture similarities and differences across CONUS (Jehn et al., 2020). We train the DL models to learn these differences in the neural network. This leads to a tough way to distinguish general hydrologic processes from basin specific processes. Reservoir computing provides a method of training a DL model that learns general dynamics applicable to watershed processes, followed by a linear mapping to specific basins (Gauthier, 2021). A formal analysis of this mapping could be a guide for identifying hydrologic theory that is general and learned by the network (e.g., snowmelt and soil moisture) vs basin specific phenomena learned by the linear layer such as compensation for heterogeneities, anthropogenic flow controls or preferential flow paths.

REFERENCES

Fruchart, M., Hanai, R., Littlewood, P. B., Vitelli, V., and Information, S. (2021). Non-reciprocal phase transitions. *Nature*, 592(April 2020):363–369.

Gauthier, D. J. (2021). Next generation reservoir computing. *Nature Communications*, (2021):1–8.

Jehn, F. U., Bestian, K., Breuer, L., Kraft, P., and Houska, T. (2020). Using hydrological and climatic catchment clusters to explore drivers of catchment behavior. *Hydrology and Earth System Sciences*, 24(3):1081–1100.

Lees, T., Reece, S., Kratzert, F., Klotz, D., Gauch, M., De Bruijn, J., Sahu, R. K., Greve, P., Slater, L., and Dadson, S. J. (2021). Hydrological Concept Formation inside Long Short-Term Memory (LSTM) networks. *Hydrology and Earth System Sciences Discussions*, (November):1–37.