Fully differentiable, fully distributed Rainfall-Runoff Modeling

Fedor Scholz[1], Manuel Traub[1], Christiane Zarfl[2], Thomas Scholten[3], and Martin V. Butz[1]

[1]Neuro-Cognitive Modeling Group, University of Tübingen, Tübingen, Germany
[2]Environmental Systems Analysis, University of Tübingen, Tübingen, Germany
[3]Soil Science and Geomorphology, University of Tübingen, Tübingen, Germany

Correspondence: Fedor Scholz (fedor.scholz@uni-tuebingen.de)

Abstract. Traditional hydrological modeling simulates rainfall-runoff process dynamics using process-based models (PBMs)~~, which~~ . PBMs are grounded in physical laws and therefore highly interpretable. ~~Due to environmental systems being~~ As environmental systems are highly complex, though, ~~sub-processes~~ subprocesses are sometimes hard or even impossible to identify and quantify. ~~Alternatively, data-driven~~ Data-driven approaches, like ~~deep~~ artificial neural networks (~~DNNs),~~ ANNs), offer an alternative. Such approaches can automatically discover hidden relationships within the data~~, which often leads to superior performance. Due to DNNs' complexity, however, these~~ . As a result, superior model performance may be achieved. However, the uncovered relationships are hard to ~~investigate~~ analyze within black-box ANNs and often fail to respect physical laws. ~~*Differentiable modeling*~~ Differentiable modeling calls for knowledge discovery by combining both approaches~~to benefit~~ , benefiting from their respective advantages. In this work, we present a physically inspired, fully differentiable, and fully distributed model, which we term DRRAiNN (Distributed Rainfall-Runoff ArtIficial Neural Network)~~, a targeted neural network architecture that successfully~~ . DRRAiNN is a neural network model that estimates river discharge based on meteorological forcings and elevation~~in~~ . Focusing on the Neckar river ~~basin, relying on~~ catchment in Southwest Germany, DRRAiNN is trained to predict daily water discharge measurements ~~from only~~ using data from 17 stations ~~. We evaluate our model against the~~ and from ten meteorological years only. DRRAiNN's performance is compared to the performance of the European Flood Awareness System (EFAS) reanalysis~~on the Neckar river catchment in Southwest Germany, where some~~ . Some instances of our model outperform EFAS at lead times of over ~~100 days . Our model architecture is physically inspired, fully differentiable ,~~ 50 days in terms of the applied metrics for model performance. As DRRAiNN is fully differentiable and fully distributed~~. This combination enables the use of~~ , efficient source allocation algorithms ~~, which help us identify the water~~ can be used to identify the precipitation sources responsible for the water discharge dynamics at specific gauging stations. ~~In the future, this approach~~ Besides DRRAiNN's potential to forecast upcoming water discharge dynamics, its full differentiability could be utilized to ~~, e.g.,~~ infer erosion sites from turbidity data, particularly when integrated with an appropriate erosion model.

1 Introduction

Accurate water flow forecasting plays a critical role in mitigating short-term flood impacts, such as preventing loss of life and reducing economic damage (Pilon, 2002). For example, simulating river discharge ~~empowers us to make informed decisions~~ is a prerequisite for flood inundation modeling (Hunter et al., 2007) and enables informed decision-making in water manage-

ment such as dam operations (Valeriano et al., 2010). Accuracy is not everything though. ~~Good hydrological models should~~ Hydrological models that respect physical laws ~~to be able to~~ are more likely to generalize well to new situations ~~,~~ and to offer insights into the underlying processes that govern water movement. A solid understanding of the dynamics of water systems is necessary to estimate the impacts of environmental planning and to improve infrastructure design (Palmer et al., 2008; Bharati et al., 2011). It also enables a better assessment of how climate change ~~might~~ may alter existing ecosystems in the future (Palmer et al., 2008; Van Vliet et al., 2013; Al Hossain et al., 2015). Additionally, models that respect physical laws can be used to infer the origins of observed discharge, thereby further facilitating the development of policies that mitigate the damages caused by floods. ~~On a practical level~~From a practical perspective, a good model should ~~be easily tuned~~ allow efficient calibration and perform well even if data ~~is~~ are sparse, which is often the case for river discharge.

~~To address some of these challenges , hydrologists traditionally employ~~ Traditionally, these challenges have been addressed using physically based approaches that explicitly encode domain knowledge. These process-based models (PBMs) ~~that~~ describe physical processes with mathematical equations derived from physical laws and observations (Brutsaert, 2023). ~~This renders PBMs inherently interpretable, allowing researchers to ask specific questions by probing them. Some components of PBMs might be inferred from experiments in a laboratory, such as Darcy's law (Darcy, 1856). Others are based on simplifications of more general physics equations.~~

~~To simplify models further, sometimes lumped approaches are employed. Here, meteorological forcings like precipitation are averaged over time and space on the basin scale. Therefore, the outline of the basin must be available a priori for this approach to be feasible. The outline is usually inferred from a digital elevation model, thereby potentially disregarding underground flows and pipes.~~

~~Real-world processes can be very complexwith lots of~~ Environmental hydrological processes are highly complex, involving numerous interacting variables that make the overall process highly heterogeneous (Marçais and de Dreuzy, 2017). Recent advances, such as the Multiscale Parameter Regionalization ~~(MPR)~~ framework (Samaniego et al., 2010) and scalable transfer function approaches (Imhoff et al., 2020) have focused on improving parameterization and capturing spatial heterogeneity in ~~distributed hydrological models~~ PBMs to alleviate these issues. To reduce uncertainty and initialize PBMs adequately, data assimilation incorporates concrete observations into running models (Liu et al., 2012; Camporese and Girotto, 2022; Montzka et al., 2012). ~~Advancements~~ Such advancements in data assimilation can improve performance ~~both in~~ in both lumped (Moradkhani et al., 2005; Liu and Gupta, 2007; Liu et al., 2012) ~~as well as~~ and distributed models (Rakovec et al., 2012). ~~Nevertheless, usually, not all involved sub-processes~~ However, significant challenges remain, as the involved processes and their interactions are ~~known~~ in most cases only partially understood (Hrachowitz et al., 2013), leading to high uncertainty and biases ~~.~~ Even if a ~~sub-process is known , though,~~ process is known well in detail, certain input variables may simply be unobservable, such as underground topography. Additionally, ~~the scales of real-world processes may be very different compared to lab~~ environmental processes often occur at scales that differ substantially from those observed under laboratory conditions (Hrachowitz et al., 2013; Shen, 2018; Nearing et al., 2021).

~~As is the case for many sciences, the amount of available data for hydrology is increasing significantly (Sit et al., 2020). But even with abundant data, PBMs struggle to fully exploit them: Only little data is used to adapt their parameters, which~~

nevertheless is a lengthy and costly process since it is often done manually by experts (Shen et al., 2023). Since the parameters are usually calibrated for a specific basin only, these tuned models do not generalize well to other basins with different properties (Hrachowitz et al., 2013; Nearing et al., 2021). This is especially problematic for countries of the Global South for which no detailed land surface and subsurface measurements are available but where intelligent water management might be even more important.

The discovery of relationships in huge amounts of data is a challenging task for which ANNscan be a remedy. In contrast to PBMs, ANN approaches are mainly data-driven, allowing them to automatically find relationships in the training data.

Complementary to PBMs, data-driven models have gained traction in recent years, driven by the increasing amount of available hydrological data (Sit et al., 2020). Artificial neural networks (ANNs) are data-driven models that automatically learn relationships from large datasets. Given the superior performance of early data-driven approaches in hydrology, there is reason to assume that the it is likely that the full potential of data-driven approaches has by no means been fully exploited, yet remains untapped (Shen, 2018; Nearing et al., 2021). However, the relationships found by ANNs often remain latent due to their high complexity. ANNs contain huge amounts of parameters and states that usually do not directly relate to real-world quantities. Training them is feasible only due to gradient descent via automatic differentiation. This process is lengthy and costly. Once an ANN is trained, though, inference is very fast and cheap. Their high complexity also often leads to neural networks not respecting physical laws despite very good performance, in turn leading to bad generalization capabilitiesdespite achieving strong predictive performance, ANNs often fail to respect physical laws due to their purely data-driven nature. This calls for measures that guide such as hybrid or physics-informed models that bias data-driven approaches toward physical plausibility.

While other Earth sciences started to adopt machine learning (ML) techniques, hydrology seems to lag behind (Shen, 2018). There is a significant amount of distrust in the community toward non-physical models (Blöschl et al., 2019). But this is not the only reason: In line with the above-mentioned lack of coordination, the field is missing benchmarks that are easy to access and enable a fair comparison between models (Hrachowitz et al., 2013; Sit et al., 2020; Nearing et al., 2021), although there are some recent efforts (Kratzert et al., 2023). Nevertheless, the amount of available data is steadily increasing. This is exactly where ML techniques offer great potential. They can be trained on vast amounts of data and infer relationships in the data automatically. It was already shown that ML often outperforms traditional approaches with regards to accuracy (Shen, 2018).

It is often criticized , that ML modelers Furthermore, it is often criticized that developers of machine learning (ML) models do not put enough effort into the interpretation of their developed systems, failing to gain a better understanding of their the system's internal dynamics (Muñoz-Carpena et al., 2023). As mentioned above

One promising avenue to overcome these limitations involves leveraging ML to infer latent variables that are otherwise inaccessible to direct measurement. To give an example, a considerable amount of runoff is situated below the ground and therefore not observableportion of total discharge originates from subsurface flow. It is not yet possible to see through the ground, rendering the underground topography latent directly measure subsurface flow, making underground topography a latent driver of hydrological behavior (Shen, 2018). We believe that these latent variables are the reason that models often generalize poorly to other basins. However, this is another problem where may contribute to poor model generalization across basins. ML and especially ANNs can help since latent variables can be inferred retrospectively from observations

~~(Butz et al., 2019; Otte et al., 2020). One question we will ask~~ support hydrological modeling in such cases, because they allow to infer latent variables retrospectively given observation dynamics (Butz et al., 2019; Otte et al., 2020). This motivates a key question we address in this paper~~is:~~ : *Given ~~the~~ observed dynamics, in which areas did precipitation contribute to the measured discharge?* ~~A similar argument can be made for evapotranspiration, which is not directly observable as well. This kind of model inversion (Sit et al., 2020) therefore renders another possibility~~

Similar to subsurface flow, evapotranspiration cannot be directly measured and must also be inferred indirectly. Model inversions of NNs (Sit et al., 2020) may therefore help to extend our understanding of the water cycle with ML. For a ~~comprehensive review of modern ML~~ broader overview of ML applications in hydrology, we refer the reader to Shen (2018) and Sit et al. (2020) .

A combination of ~~the above-mentioned~~ PBMs and ML-based approaches could leverage the advantages of both worlds. ~~If pursued~~ When combined with the goal of knowledge discovery, this ~~combination was recently coined~~ approach is referred to as "differentiable modeling" (Shen et al., 2023). It could result in well-performing interpretable models that automatically find new relationships in the data, respect physical laws~~and therefore generalize well and need~~ , generalize well across different settings, and require comparatively little data. ~~It offers two different perspectives: Coming from traditional modeling approaches, the incorporation of differentiable model parts allows to automatically close knowledge gaps. This could manifest in various ways, such as estimating unknown parameters or representing entire sub-processes using approaches like artificial neural networks.~~

~~Coming from the ML side, relationships that are known to be true~~ From the ML perspective, known relationships can be incorporated into ~~already~~ differentiable models as constraints ~~,~~ or inductive biases. ~~These inductive biases introduce~~ Inductive biases encode prior assumptions about the data-generating process, effectively constraining the model's solution space. By doing so, they can improve performance, enhance generalization, and make learning more efficient. Furthermore, they ~~help~~ guide the model towards discovering ~~meaningful~~ interpretable structures in the data, aligning its behavior with established principles (Butz et al., 2024).

~~Our work builds on differentiable modeling, presenting DRRAiNN (Distributed Rainfall-Runoff ArtIficial Neural Network), a physics-inspired, fully differentiable, fully distributed rainfall-runoff model. Our targeted spatio-temporal artificial neural network (ANN) architecture estimates river discharge at given measurement stations from gridded precipitation, solar radiation, elevation, and past discharge. This approach poses a severe challenge~~ A crucial challenge for the modeler is to ~~ANN-based learning approaches, though: the targeted river discharge data is very sparse. To avoid overfitting and to improve interpretability and generalization, we had to incorporate several physics-inspired inductive biases into DRRAiNN. One of these biases is the modularization into a spatially fully distributed rainfall-runoff model and a graph-based river discharge model. Another one preconditions DRRAiNN to represent the sub-processes of lateral propagation of water over the landscape and evapotranspiration in certain sub-components. With this, we hope to show that deep learning approaches must not necessarily be perceived as purely black boxes. Instead, they can be designed to contain meaningful components that correspond to sub-processes of the overall real-world process. To a certain extent, their inner workings can be interrogated with interpretability methods to show what their estimations are based on.~~

We thus focus on successfully designing and training such a fully differentiable model. We evaluate the DRRAiNN's estimation abilities, physical plausibility, and the need for our main design choices. We showcase its performance in a real-world setting on the Neckar River in Southwest Germany, comparing it with simulations from the European Flood Awareness System (EFAS, Mazzetti et al. (2023)). DRRAiNN outperforms EFAS with lead times of up to 45 days. Due to DRRAiNN being fully distributed and fully differentiable, our approach allows us to answer 'where' questions, such as : What is the true catchment area, including underground flows? That is, DRRAiNN opens up the possibility for performing source allocations using gradient-based techniques like integrated gradients (Sundararajan et al., 2017). These techniques can help in examining and understanding internal model dynamics, potentially leading to knowledge discovery and thereby further repealing the black box perspective. We show reconstructed catchment areas from observed dynamics, demonstrating the feasibility of attribution methods within DRRAiNNfind and incorporate those biases that restrict the solution space as much as possible without introducing incorrect or unjustified assumptions and without restricting the self-organizing power of NNs.

2 Related Work

In their seminal work, Kratzert et al. successfully use LSTMs have successfully used a long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) for rainfall-runoff modeling on a daily scale (Kratzert et al., 2018)at the basin scale (Kratzert et al., 2018), demonstrating that purely data-driven models can exceed traditional methods. Since then, numerous studies have emerged, applying basically largely the same model to various data sets (Sit et al., 2020). Notably, significant advancements to the model have also been made, including the incorporation of physical constraints (Kratzert et al., 2019; Hoedt et al., 2021), uncertainty estimation (Klotz et al., 2022), and the extension of modeling to multiple timescales (Gauch et al., 2021). Hybrid models such as neural ODEs, where differential equations of conceptual hydrological models are replaced by neural networks, were also applied in this setting (Höge et al., 2022). All of the above-mentioned models have in common that they are lumpedmodels, i.e., forcings aforementioned models are lumped, meaning that inputs are spatially aggregated over the catchmentarea which was inferred from a digital elevation model. This practice is so prevalent that it is often not even mentioned in the papers. In semi-distributed modeling, models make limited use of the river topology (Xiang and Demir, 2020; Moshe et al., 2020; Sit et al., 2021; Kratzert et al., 2021). Here, the catchment area is divided into multiple sub-basins that can communicate with each othereach catchment. These catchments are typically delineated using digital elevation models.

Semi-distributed models partially leverage river network topology, providing a compromise between lumped and fully distributed representations. These include purely data-driven graph-based models (Xiang and Demir, 2020; Moshe et al., 2020; Sit et al., 20 , as well as hybrid approaches that integrate domain knowledge – for example, by using a differentiable Muskingum-Cunge routing model (Bindas et al., 2024; Zhong et al., 2024). These models typically divide the overall catchment into multiple subbasins connected via the river network, enabling limited spatial interaction. Within each sub-basinsubbasin, however, forcing are again spatially aggregated. forcings are still spatially aggregated, similar to lumped models.

In contrast, fully distributed models directly operate on a ~~grid without any spatial aggregation. Even though~~ spatial grid. While there is a call for more fully distributed data-driven models for rainfall-runoff modeling (Nearing et al., 2021), ~~not many approaches exist in the literature.~~

~~The model presented in (Xiang and Demir, 2022) indeed operates~~ most existing approaches remain limited in critical ways. Some hybrid models operate on a grid ~~, but communication between neighboring cells is only possible in~~ but restrict cell-to-cell communication to the direction of ~~the steepest descent~~ steepest descent (Xiang and Demir, 2022; Wang et al., 2024). This strong assumption ~~essentially turns~~ effectively transforms the grid into a ~~sparser graph. The~~ directed graph, excluding physically plausible underground flows in other directions. CNN-LSTMs ~~presented in (Ueda et al., 2024; Pokharel and Roy, 2024a; Li et al., 2022) operate on a grid without any assumption of flow directions~~ process gridded input data without explicit assumptions about flow directions (Ueda et al., 2024; Pokharel and Roy, 2024b; Li et al., 2022). However, ~~the LSTMs are not applied in every grid cell but instead receive the flattened outputs of the CNNs, which renders the modeling of space and time completely separate.~~

~~In (Schmidt et al., 2020), the authors applied~~ these models separate spatial and temporal processing by flattening the convolutional neural network (CNN) outputs before passing them to an LSTM. As a result, spatial dependencies are not maintained across time steps. This limitation is addressed in Oddo et al. (2024), were a ConvLSTM (Shi et al., 2015) ~~and found that it does not make use of spatial patterns, which made it perform worse than a lumped approach. In contrast, (Oddo et al., 2024) used a ConvLSTM to estimate river discharge with a one-hour lead time by flattening~~ is used to jointly model space and time. Yet, before the final discharge prediction, the outputs of all ~~cells before feeding them into a linear layer. Similarly, (Longyang et al., 2024) used a modified ConvLSTM architecture combined~~ grid cells are flattened into a single feature vector and passed through a fully connected layer. Similar global aggregation strategies can be found elsewhere (Zhu et al., 2023; Tyson et al., 202 . Moving a step closer to physical plausibility, Longyang et al. (2024) combined a ConvLSTM with ridge regression to ~~determine from which grid pixels the output should be aggregated to estimate discharge at the station. More examples of this kind can be found in the literature (Zhu et al., 2023; Tyson et al., 2023; Pokharel and Roy, 2024b; Xu et al., 2022)~~learn which grid cells should contribute to discharge estimation at each station. This allowed the reconstruction of plausible underground flow paths between subbasins. Since all of these distributed models aggregate the outputs of the spatial component globally over space, ~~the model lacks the~~ whether weighted or not, they lack the incentive to propagate water across the landscape ~~. We therefore assume that these models behave physically unrealistic and generalize poorly to other basins~~in a physically plausible way.

~~For a comprehensive list of applications and publications regarding machine learning and hydrology in general, we refer the reader to (Sit et al., 2020).~~

2 ~~Methods~~

1.1 ~~Study site~~

~~The Neckar river network in Southwest Germany has a catchment size of 14 000 km^2 and exhibits a heterogeneous landscape: It encompasses narrow and wide valleys, different kinds of rocks like limestone and sandstone, different types of soils like clay and marl, underground topographies like karst, and formations like aquifers. This makes the modeling of the Neckar~~

River network a challenging endeavor. To give a concrete example, there are underground flows south of Pforzheim that route water toward the east, while the elevation modelsuggests a different flow direction. (Ufrecht, 2002). This relationship cannot be inferred from a digital elevation model alone: Latent underground structures route the water in a different direction than the elevation model alone would suggest.

The Neckar and its sub-catchments were subject to multiple distributed hydrological studies in the past (Imhoff et al., 2020; Samaniego et

.

The study area used in this work is the Neckar River catchment in Southwest Germany.

1.1 Data

We use the following data as input for DRRAiNN: radar-based precipitation, elevation for above-ground topography, Our work builds on differentiable modeling to combine both process-based and data-based modeling, and to address the challenges of physical plausibility, interpretability, and latent variable inference. We present DRRAiNN (Distributed Rainfall-Runoff ArtIficial Neural Network), a physics-inspired, fully differentiable, fully distributed rainfall-runoff model. Our spatio-temporal ANN architecture estimates river discharge at gauging stations from gridded precipitation, solar radiation, elevation, and past discharge. DRRAiNN is fully distributed in the sense that it internally operates on a grid. However, its outputs are point-wise river discharge measurements at given gauging station locations. Its full differentiability allows gradients to flow seamlessly through the entire system, enabling end-to-end optimization of all its components with sparse discharge measurements being the only target variable. To avoid overfitting, and to improve interpretability and generalization, we incorporated several physics-inspired inductive biases into DRRAiNN. These include the modularization into a spatially fully distributed rainfall-runoff model and the utilization of a graph-based river discharge data. Preliminary experiments showed no improvement when including temperature, therefore we exclude it following Occam's razor. By restricting the domain to the Neckar River, we end up with an area of size 200 km^2. After the transformations described in the paragraphs below, all gridded data is reduced from a 1 $\mathrm{km} \times 1$ km gridto a 4 $\mathrm{km} \times 4$ km grid by taking the mean. This results in a 50×50 pixel grid. We use the hydrological years 2006 to 2015 for training and 2016 to 2018 for validation.

For precipitation, we use the estimated product RADOLAN by the Deutsche Wetterdienst (RADOLAN, 2016), which is collected from radar stations that are distributed over Germany. The data domain is a $900\mathrm{km} \times 900\mathrm{km}$ pixel grid with a resolution of $1\mathrm{km} \times 1\mathrm{km}$ that covers all of Germany. This grid forms the basis of the grid our model operates on. RADOLAN data is log-standardized before being sent to the model due to its long-tail distribution. This means we add 1, take the logarithm, subtract the mean, and divide by the standard deviation. We replace missing values with 0s, which is the log-standardized mean.

We derive static topography features from the digital elevation model (DEM) EU-DEM v1.1 by the European Union Copernicus Land Monitoring Service European Environment Agency (EU-DEM, 2016). We use rasterio (Gillies and others, 2013) to transform and reproject the data into the RADOLAN coordinate reference system. We also use the DEM to compute the differences in altitudes between adjacent discharge measurement stations. All these variables are standardized before being sent to the model, i.e., we subtract their mean and divide by their standard deviation.

~~For solar radiation, we use surface short-wave (solar) radiation downwards (SSRD) from the ERA5 data set (Hersbach et al., 2018) . We use rasterio (Gillies and others, 2013) to transform and reproject the data into the RADOLAN coordinate reference system. Like the precipitation data, solar radiation data is log-standardized.~~

~~The topography of our river network is determined by the AWGN data set (AWGN, 2023). We use it to compute the adjacency matrix that describes which stations are connected via river segments and the corresponding river segment lengths.~~

~~Finally, we use discharge measurement data to tune in the discharge model~~ model. Additional architectural choices precondition DRRAiNN to encode distinct processes, such as lateral propagation of water across the landscape and local evapotranspiration. As a result, DRRAiNN turns into a gray-box deep learning model. Its model design encourages the development of sub-modules, which model surface and sub-surface water flow, water inflow into a river network, and ~~, more importantly, as the only target variable to train and validate our model. We use data collected and provided by the German Federal Institute of Hydrology via the Global Runoff Data Centre (GRDC, 2024). The data set contains observed daily river discharge from measurement stations across the world. Since the location information of the discharge measurement stations is partially wrong, we corrected them manually. We then try to snap the station locations to the~~ water flow and discharge across the river network. ~~If this correction is larger than a certain threshold, the station is discarded. If two stations are very close to each other, one of them is discarded. Discharge data is log-standardized station-wise before being sent to the model due to its long-tail distribution. This means we add 1, take the logarithm, subtract the station-wise mean, and divide by the station-wise standard deviation. We replace missing values with 0s, which is the log-standardized mean of the corresponding station.~~

1.1 ~~Model~~

Thanks to DRRAiNN's fully distributed and fully differentiable architecture, it is possible to answer spatially resolved questions, such as: Where is the true catchment area, including contributions from underground flows? In other words, DRRAiNN enables source allocations using gradient-based attribution methods like integrated gradients (Sundararajan et al., 2017). These techniques can help to examine and understand internal model dynamics, enabling knowledge discovery.

2 Methods

We present DRRAiNN, a spatio-temporal ~~artificial neural network~~ ANN architecture that estimates river discharge from static attributes and meteorological forcings in a distributed manner. We evaluate DRRAiNN's estimation abilities, physical plausibility, and the necessity of its architectural design choices. We demonstrate its performance in a real-world setting on the Neckar River in Southwest Germany, comparing it to simulations from the European Flood Awareness System (EFAS, Mazzetti et al. (2023)). DRRAiNN achieves higher KGE and NSE values than EFAS for lead times of up to 50 days and provides interpretable source attributions that enable the reconstruction of effective catchment areas from modeled dynamics.

2.1 Model

DRRAiNN's structure is grounded in the following data and structural information sources. The locations $L_i = (x_i, y_i)$ for estimations of discharge in the river network are determined by discharge ~~measurement~~ gauging stations that provide observed discharge $Q_{i,t}$ for time t in ~~24h~~ 24 h periods. The connectivity of stations, determined by the river network, is encoded in an adjacency matrix $A_{i,j}$. Static maps $S_{x,y}$ and meteorological forcings $F_{x,y,t}$ for hourly time points t are encoded on a grid that spans the whole catchment area of the river network. Given static maps $S_{:,:}$, meteorological forcings $F_{:,:,t_0:t_s+T}$ over the whole duration $(t_0 \dots t_s + T)$ in hours, and past discharge $Q_{i,t_0:t_s}$ over the tune-in period $(t_0 \dots t_s)$ in days, DRRAiNN ~~f estimates future~~ estimates discharge $Q_{i,t_s+1:t_s+T}$ over a temporal future horizon of T days via a function f, representing the learned spatio-temporal mapping implemented by the model:

$$\tilde{Q}_{i,t_s+1:t_s+T} = f(S_{:,:}, F_{:,:,t_0:t_s+T}, Q_{i,t_0:t_s}) \tag{1}$$

~~,~~

~~In contrast to most other neural networks in hydrology, DRRAiNN includes a rainfall-runoff model that is fully spatially distributed: We do not lump variables across space over basins, but our model operates on a grid instead. Since surface /subsurface and~~ Since surface and subsurface flow differ from river flow dynamics ~~behave differently~~ as described above, we model these ~~sub-processes~~ subprocesses separately. Therefore, DRRAiNN consists of two components, the rainfall-runoff model and the discharge model. The rainfall-runoff model operates recurrently on a grid, rendering it fully distributed. It is supposed to model surface ~~/subsurface flow~~and subsurface flow, and evapotranspiration. The discharge model operates recurrently on a graph ~~and is supposed~~ to model river flow inside of channels ~~.~~

~~DRRAiNN processes a time series in the following manner: Over the whole sequence , we alternately call~~ and output estimated discharge $\tilde{Q}$ at the station locations. While DRRAiNN is fully distributed in its internal computation over a spatial grid, its outputs are only available at selected gauging stations.

At each time step, DRRAiNN processes the sequence in an auto-regressive loop by first invoking the rainfall-runoff model, ~~which is implemented by a recurrent convolutional ANN, and~~ followed by the discharge model~~, which is implemented by a recurrent graph ANN~~. The rainfall-runoff model receives ~~static landscape features~~ gridded static maps S and meteorological forcings F as input to ~~estimate runoff~~ model the catchment on a grid. It is primed to ~~model two important sub-processes separately~~distinguish between two important subprocesses, namely surface ~~/~~and subsurface flow, which is mainly driven by topography, and evapotranspiration, which is mainly driven by solar radiation. ~~Even though it cannot directly be interpreted as such, we call its output~~ It produces a latent representation, which we term *runoff embedding*~~since this is the main driver for the discharge model: The estimated runoff is collected~~ , extracted at station locations and ~~sent~~ used as input to the discharge model. Despite being the main driver of discharge, it cannot be directly interpreted as runoff due to its self-organizing nature. The discharge model additionally receives an adjacency matrix A that describes the connectivity between stations, static river segment features, and the (potentially estimated) discharge $Q_{:,t-1}$ from the previous time step. It ~~estimates discharge~~ then estimates discharge $\tilde{Q}$ for each station, from which the training loss is computed.

We implement DRRAiNN in pytorch (Paszke et al., 2019). In the following, we provide a more detailed description of DRRAiNN's components. See Fig. 1 for a depiction of the overall model.

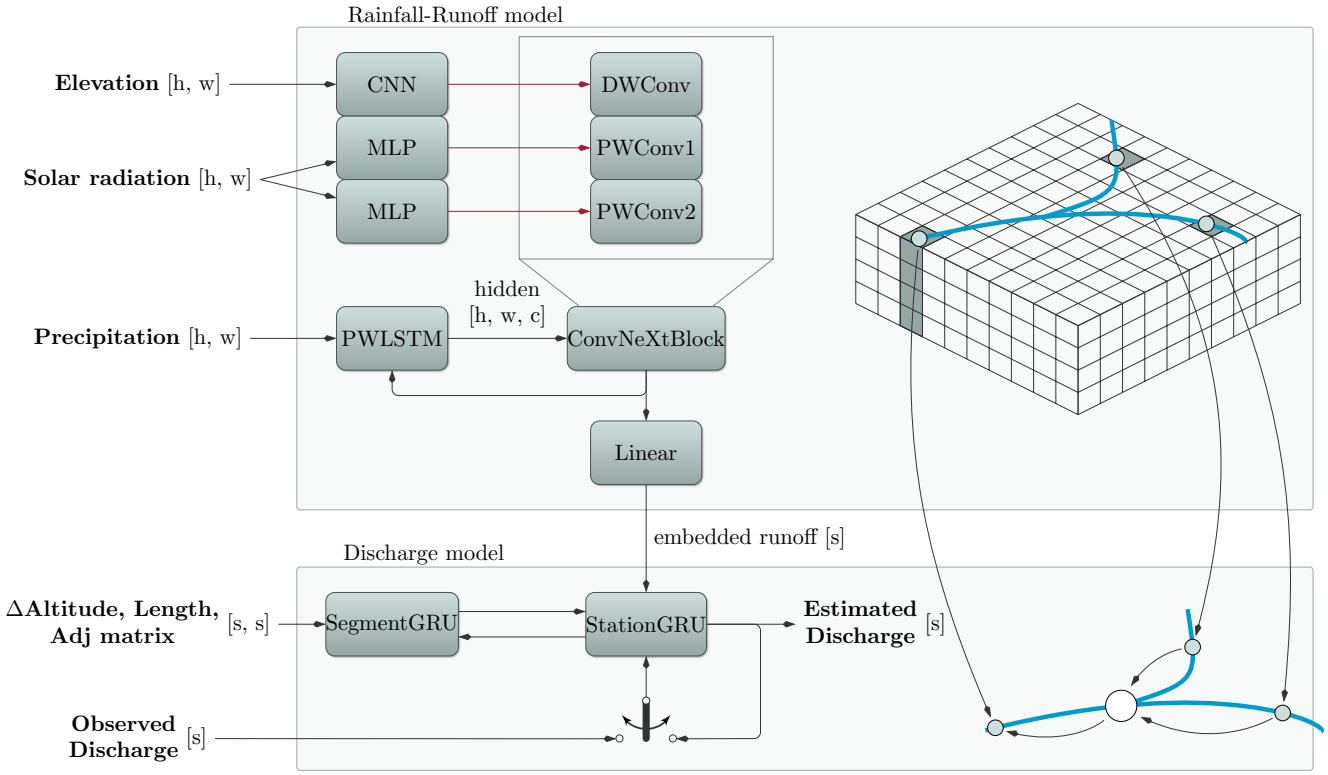

Figure 1. ~~A detailed view~~ Schematic overview of the DRRAiNN architecture. The gridded rainfall-runoff model ~~'s task is~~ has two main tasks: to ~~propagate~~ model the ~~received~~ redistribution of precipitation ~~over~~ across the landscape~~according to the elevation~~ , and to model evapotranspiration based on solar radiation. It receives precipitation as its main input to ~~the~~ a point-wise LSTM, whose hidden ~~state is modified by the~~ states, but not cell states, are updated using a ConvNeXtBlock. The ConvNeXtBlock ~~'s~~ weights are not ~~static~~ fixed but ~~produced~~ dynamically generated by ~~other neural networks~~hypernetworks (indicated by red arrows). The depth-wise convolution ~~'s~~ (DWConv), responsible for lateral water propagation, receives its weights ~~are produced by~~ from a ~~convolutional neural network~~ CNN that ~~has~~ takes elevation as input and shares the same receptive field ~~but receives elevation~~ as ~~input. Its main purpose is to model lateral propagation of water over~~ the ~~landscape~~DWConv. The point-wise convolutions ~~'~~ (PWConv1 and PWConv2), used to model local evapotranspiration processes, receive their weights ~~are produced by a multi-layer perceptron~~ from an MLP that ~~receives~~ takes solar radiation as input. ~~Its main purpose is to model evapotranspiration, a process that is local in space. Before the~~ The LSTM hidden state is ~~sent to the discharge model, it is~~ further processed by a ~~simple~~ linear layer before being passed to the discharge model. ~~The~~ This graph-based discharge model ~~then receives the processed state of the rainfall-runoff model~~ aggregates information at the ~~measurement~~ gauging stations~~' locations and processes it together with~~ , incorporating the last (~~potentially~~ possibly inferred) discharge ~~according to the adjacency of the stations~~values, ~~their~~ elevation differences ~~in altitudes~~between stations, and ~~the~~ river segment lengths~~between the stations~~. Its output is the estimated discharge at each station.

2.1.1 Rainfall-Runoff Model

The rainfall-runoff model consists of a position-wise ~~long short-term memory (LSTM) and a convolutional neural network (CNN)~~ LSTM and a CNN that are called in ~~alternation~~each time step. This renders the rainfall-runoff model local in space and time~~: Only neighboring and past~~ . Only spatially local and temporally previous information is used to update internal states.

2.1.2 ~~Modeling temporal dynamics~~

The position-wise LSTM (PWLSTM) is responsible for modeling the temporal relationships in the data and therefore maintains a hidden and a cell state for each grid cell. The gating mechanism ~~of the LSTM can shield the cell states from unwanted updates. It thus allows to maintain information over long~~ regulates when and how the cell state is updated, allowing the model to retain information over extended time periods. This can be particularly useful ~~to implicitly model, e.g.,~~ for implicitly modeling slow hydrological processes such as soil moisture or groundwater levels, which ~~exhibit slower dynamics~~ evolve more gradually than overland flow. The LSTM receives precipitation as input to update its hidden and cell states. It has a hidden size of 4 (see Appendix B for hidden sizes 2 and 6). Importantly, the weights of the LSTM are shared throughout the gridded area. As a result, while the LSTM at each grid cell maintains individual hidden and cell state values, the temporal processing principle is identical everywhere. The assumption is that the unfolding physics is the same everywhere, although they may be locally parameterized.

2.1.3 ~~Modeling spatial dynamics~~

~~The CNN is responsible for modeling spatial relationships , i.e.,~~ The CNN models spatial relationships such as the propagation of water flow ~~over~~ across the landscape and evapotranspiration. It receives and updates the hidden state h of the PWLSTM ~~,~~ to model spatial interactions, while leaving the PWLSTM's cell states untouched to preserve temporal memory. Surface and subsurface flow are spatially extended processes, whereas evapotranspiration is primarily a local phenomenon, occurring independently at each grid cell. To reflect this distinction, we separate the CNN's treatment of these processes using different convolution types and input sources, introducing an inductive bias into the architecture. ~~Note that surface/subsurface flow is a spatially extended process, while evapotranspiration is a local (though vertical) process that happens mostly independently of neighboring cells. We incorporate this into our architecture as an additional inductive bias .~~

More precisely, the CNN is ~~given by~~ based on a modified ConvNeXt block (Liu et al., 2022). A ConvNeXt block consists of three layers, namely a depth-wise convolutional layer (DWConv) with kernel size 7×7 followed by a position-wise inverted bottleneck given by two linear layers (PWConv1 and PWConv2). This way, ConvNeXt ~~disentangles horizontal and vertical~~ decouples spatial and channel-wise information flow. We ~~use~~ apply the SiLU activation function ~~between all~~ after the convolutional and between the linear layers (Hendrycks and Gimpel, 2016). In contrast to its original formulation, the weights of our ConvNeXt block are not static ~~. Rather, they~~ but location-dependent. They are parameterized by other neural networks, turning this network component into a *hypernetwork* (Traub et al., 2024a). This means that the ConvNeXt block can behave

differently at each location on the grid. Calling DWConv results in the following operation:

$$y_{i,j,c} = \sum_{m=-3}^{3} \sum_{n=-3}^{3} w_{i,j,m,n,c} \cdot x_{i+m,j+n,c}, \tag{2}$$

where y is the output, x the input, w are the weights produced by the hypernetwork, c is the considered channel, and i and j are coordinates. ~~Note that we~~ We can still call this operation a convolution if we regard the input variables together with the weight-generating networks as the kernel. Calling PWConv1 and PWConv2 results in the following operation:

$$y_{i,j,c_{\text{out}}} = \sum_{c_{\text{in}}} w_{i,j,c_{\text{out}},c_{\text{in}}} \cdot x_{i,j,c_{\text{in}}}, \tag{3}$$

~~We parameterize the different layers~~ Each layer of the ConvNeXt block ~~with different weight-generating networks that receive different inputs~~is parameterized by a distinct hypernetwork, tailored to the type of process it represents. The weights of DWConv are produced by a ~~simple~~ CNN that has the same kernel size as DWConv itself. The weights for PWConv1 and PWConv2 are produced by ~~simple~~ position-wise ~~MLPs~~multi-layer perceptions (MLPs). By using different input variables for the different hypernetworks, we can distinguish between local and spatially extended processes. How water propagates ~~over~~ across the landscape depends mainly on the topography, which is why we generate the weights of DWConv from elevation. Before feeding the elevation into the hypernetwork, we subtract the elevation of the center cell from the elevations of all other cells within each receptive field ~~, since we are interested in slopes and not~~ as relative elevation is more informative for flow direction than absolute elevation. Evapotranspiration, on the other hand, is a ~~very~~ local process and ~~should therefore be modeled~~ is therefore best captured by the position-wise components. This is why we generate the weights for PWConv1 and PWConv2 from solar radiation. See Fig. 2 for an illustration.

2.1.4 ~~Adapter~~

Lastly, the ~~hidden states~~ runoff embeddings are extracted at the station locations~~are collected~~, fed through a single linear layer, and sent ~~the the~~ to the river discharge model. ~~Collecting and summing up~~ Aggregating the hidden states of all cells on the corresponding upstream river segment showed a tendency to ~~overfitting~~ overfit in preliminary experiments.

2.1.5 Discharge Model

Our discharge model is a recurrent graph neural network called DISTANA (Karlbauer et al., 2019), with the graph structure ~~determined~~ defined by the actual river network and the stations. ~~It~~ DISTANA maintains two types of ~~kernels,~~ recurrent units: station and segment kernels, both ~~of which are~~ implemented as Gated Recurrent Units (GRUs~~) (Cho et al., 2014)~~ , Cho et al. (2014)) with a hidden size of 8 (see Appendix B for hidden sizes 4 and 16, and a version in which the GRUs are replaced with LSTMs). Station kernels ~~sit on the discharge measurement stations, segment kernels sit on the segments between those stations. They~~ are placed at the gauging stations, while segment kernels are located on segments between stations. These kernels communicate with each other via lateral connections ~~that have~~ with 4 channels (Fig. 1). ~~The station kernel additionally receives the output of the rainfall-runoff model, the last (potentially inferred) discharge, and, based on that,~~

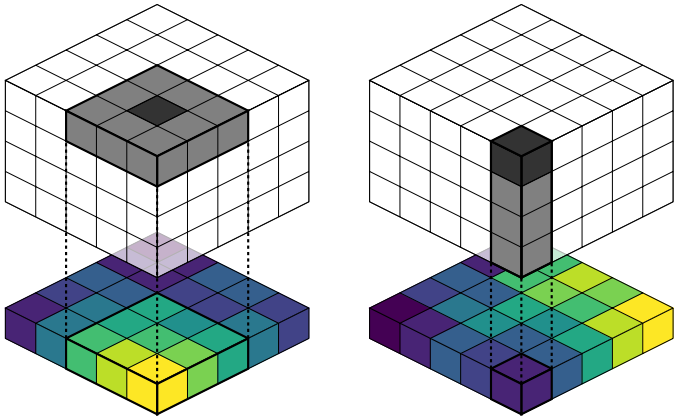

Figure 2. Illustration of the hypernetworks used in DRRAiNN. ~~The~~ In both panels, the dark gray cells represent locations whose hidden states are updated based on information from the ~~values of the~~ light gray cells. The weights for these updates are ~~produced~~ generated by ~~other~~ separate neural networks that ~~have~~ share the same receptive field but ~~look at another type~~ receive different types of input data. Left: ~~The~~ A CNN takes elevation as input and produces the weights for the depth-wise convolution~~are produced by a convolutional neural network that receives elevation as input~~, which models lateral water propagation. Right: ~~The~~ An MLP takes solar radiation as input and produces the weights for the point-wise convolution~~are produced by a multi-layer perceptron that receives solar radiation as input~~, which models localized evapotranspiration.

~~estimates the discharge at that station~~In each time step, the segment kernels are updated first, followed by the station kernels, which then estimate the discharge $\tilde{Q}$ at their respective locations. The segment ~~kernel additionally receives the difference in altitude between the corresponding stations as well as the length of the~~ kernels first concatenate the previous output of the upstream station kernels with static river segment attributes – specifically the altitude difference and segment length. After applying the GRU, the output is multiplied by the adjacency matrix, which is derived from the river ~~segment that it models. In general, both types of kernels work similarly except that the transition kernel receives an adjacency matrixaccording to which it aggregates data from upstream stations. The adjacency matrix is determined by the station locations and the river~~ network topology ~~. In a single time step, first, the transition kernels and subsequently~~ and station positions. The segment kernels thereby sum up information from upstream station kernels. The output of the segment kernels serves as input for the station kernels~~are called. Each kernel first concatenates its static, dynamic, and lateral inputs and then applies the GRU. In the case of the transition kernel, the~~ . The station kernels work similarly. They first concatenate the last output of the segment kernels with the last (potentially inferred) discharge and the output of the ~~GRU is multiplied by the adjacency matrix, thereby summing up incoming information from upstream station kernels. Afterward, the tensor~~ rainfall-runoff model. After applying the GRU, the output is split into ~~dynamic and lateral outputs~~the estimated discharge $\tilde{Q}$ and the input for the segment kernels in the next time step.

~~Even though we feed~~ Although DRRAiNN receives hourly meteorological forcings ~~into DRRAiNN, we only produce daily discharge estimates~~ F, it produces discharge estimates at a daily resolution. During the initial 10 ~~days~~ day tune-in phase of

each sequence, we ~~therefore~~ feed the same observed discharge value ~~into DRRAiNN over one day. Additionally, we feed a daily marker into the station kernel (not depicted in the figure), which informs DRRAiNN about when a new day begins~~Q into DRRAiNN for each hourly step within the day.

2.2 Data

The input data for DRRAiNN consists of radar-based precipitation, elevation for above-ground topography, solar radiation, and river discharge data. Preliminary experiments showed no improvement when including temperature; therefore, we exclude it following Occam's razor.

For precipitation, we use the radar-based precipitation product RADOLAN provided by the Deutsche Wetterdienst (RADOLAN, 2016). The data domain is a $900\ \mathrm{km} \times 900\ \mathrm{km}$ pixel grid with a resolution of $1\ \mathrm{km} \times 1\ \mathrm{km}$ that covers all of Germany and a temporal resolution of $1\ \mathrm{h}$. This grid defines the spatial resolution at which our model operates. RADOLAN data is log-standardized before being sent to the model due to its long-tail distribution. Specifically, we add 1 and take the logarithm, then compute the mean and standard deviation of the transformed data to standardize it. We replace missing values with zeros, which is the standardized mean.

For static topography information we use the digital elevation model (DEM) EU-DEM v1.1 provided by the Copernicus Land Monitoring Service of the European Environment Agency (EU-DEM, 2016). We also use the DEM to compute the differences in altitudes between adjacent discharge gauging stations. Elevation values and derived difference are standardized before being sent to the model, i.e., we subtract their mean and divide by their standard deviation.

For solar radiation, we use surface short-wave downward radiation (SSRD) from the ERA5 data set (Hersbach et al., 2018). It comes with a temporal resolution of $1\ \mathrm{h}$ and a relatively coarse spatial resolution of $0.25° \times 0.25°$. Like the precipitation data, solar radiation data is log-standardized. We use rasterio (Gillies and others, 2013) to transform and reproject the DEM and solar radiation data to match the RADOLAN coordinate reference system.

The topography of our river network is determined by the AWGN data set (AWGN, 2023). We use it to compute the adjacency matrix that describes which stations are connected via river segments and the corresponding river segment lengths.

Finally, we use discharge measurement data to tune in the discharge model and, more importantly, as the only target variable to train, validate, and test our model. We use data collected and provided by the German Federal Institute of Hydrology via the Global Runoff Data Centre (GRDC, 2024). The data set contains observed daily river discharge from gauging stations worldwide, including those in Germany. Since the location information of the discharge gauging stations is partially wrong, we corrected them manually. We then align the station locations to the nearest river segment (snapping). If the correction exceeds a predefined threshold, the station is excluded. If two stations are very close to each other, one of them is discarded. Due to its long-tail distribution, discharge data is log-standardized on a per-station basis before being sent to the model. We add 1 and take the logarithm, then standardize the data using station-wise means and standard deviations. We replace missing values with zeros, which is the standardized mean of the corresponding station.

Our choice of input datasets was guided by temporal resolution, data provenance, and practical availability. Although the European Flood Awareness System (EFAS) employs EMO-1 for precipitation input, we opted for RADOLAN due to important

differences: EMO-1 offers a coarser 6 hresolution and is interpolated from sparse station data, in contrast to RADOLAN's direct radar-based observations. Although we expect only minor differences in performance in some settings, radar-derived datasets like RADOLAN provide finer spatial and temporal resolution, which is advantageous for distributed models. Similarly, we chose ERA5 for solar radiation data due to its gridded format and hourly resolution. Alternative datasets, such as those provided by DWD, are either available only as station-wise hourly data, which lack the required grid format, or as gridded data aggregated monthly, which does not meet our temporal requirements. Daily datasets like EOBS may suffice if subdaily temporal patterns are encoded separately, but this would require additional preprocessing. A transition toward operation flood forecast would place increased importance on the choice of precipitation forecast products (Imhoff et al., 2022). Ultimately, all data products entail inherent uncertainties and errors, and our choices reflect a balance between data availability, temporal resolution, and the specific requirements of our model.

2.3 Study site

The Neckar river network in Southwest Germany spans a catchment area of $14\,000\ \mathrm{km}^2$ with a mean elevation of $460\ \mathrm{m}$. According to ERA5, temperatures in this region ranged from $-25\ ^\circ\mathrm{C}$ to $40\ ^\circ\mathrm{C}$ during our training period. Our dataset includes measurements from 17 gauging stations distributed across the river network (see Fig. 3). At the most downstream station in Rockenau, discharge during the training period ranged from $29.5\ \mathrm{m}^3/\mathrm{s}$ to $1690\ \mathrm{m}^3/\mathrm{s}$ with a mean of $133.3\ \mathrm{m}^3/\mathrm{s}$.

The catchment features a highly heterogeneous landscape, including narrow and wide valleys, diverse geology (e.g., limestone, sandstone), different soil textures (e.g., clay, marl), and subsurface structures such as karst systems and pore water aquifers. This makes the modeling of the Neckar River network a challenging endeavor. To give a concrete example, there are underground flows south of Pforzheim that route water toward the east, while the elevation model suggests a different flow direction. (Ufrecht, 2002). This relationship cannot be inferred from a digital elevation model alone. Latent underground structures route the water in a different direction than the elevation model alone would suggest.

By restricting the domain to the Neckar river network, we end up with an area of size $200\ \mathrm{km} \times 200\ \mathrm{km}$. Following the transformations described above, all gridded data is reduced from a $1\ \mathrm{km} \times 1\ \mathrm{km}$ grid to a $4\ \mathrm{km} \times 4\ \mathrm{km}$ grid by taking the mean. This results in a 50×50 grid covering the study area. We train our model on hydrological years $2006 - 2015$, validate on $2016 - 2018$, and test on 2019. Forcings F are provided at hourly resolution, while discharge is provided at daily resolution.

2.4 Experimental setup

We train DRRAiNN on sequences of 20 days ~~, with~~ (480 hourly steps), using the first 10 days ~~serving~~ as a warm-up phase. During this phase, we feed the model observed discharge values ~~are assimilated into the model to initialize~~ to initialize and align its hidden states ~~and align them with the system's~~ with the true system dynamics. This ~~approach is akin to~~ procedure resembles data assimilation in traditional hydrological models, where observations are used to update model states and reduce uncertainty. In ~~machine learning, this closed-loop setup is called~~ ML terms, this corresponds to teacher forcing. The warm-up phase allows the rainfall-runoff component of DRRAiNN to ~~potentially estimate quantities like~~ infer latent hydrological states,

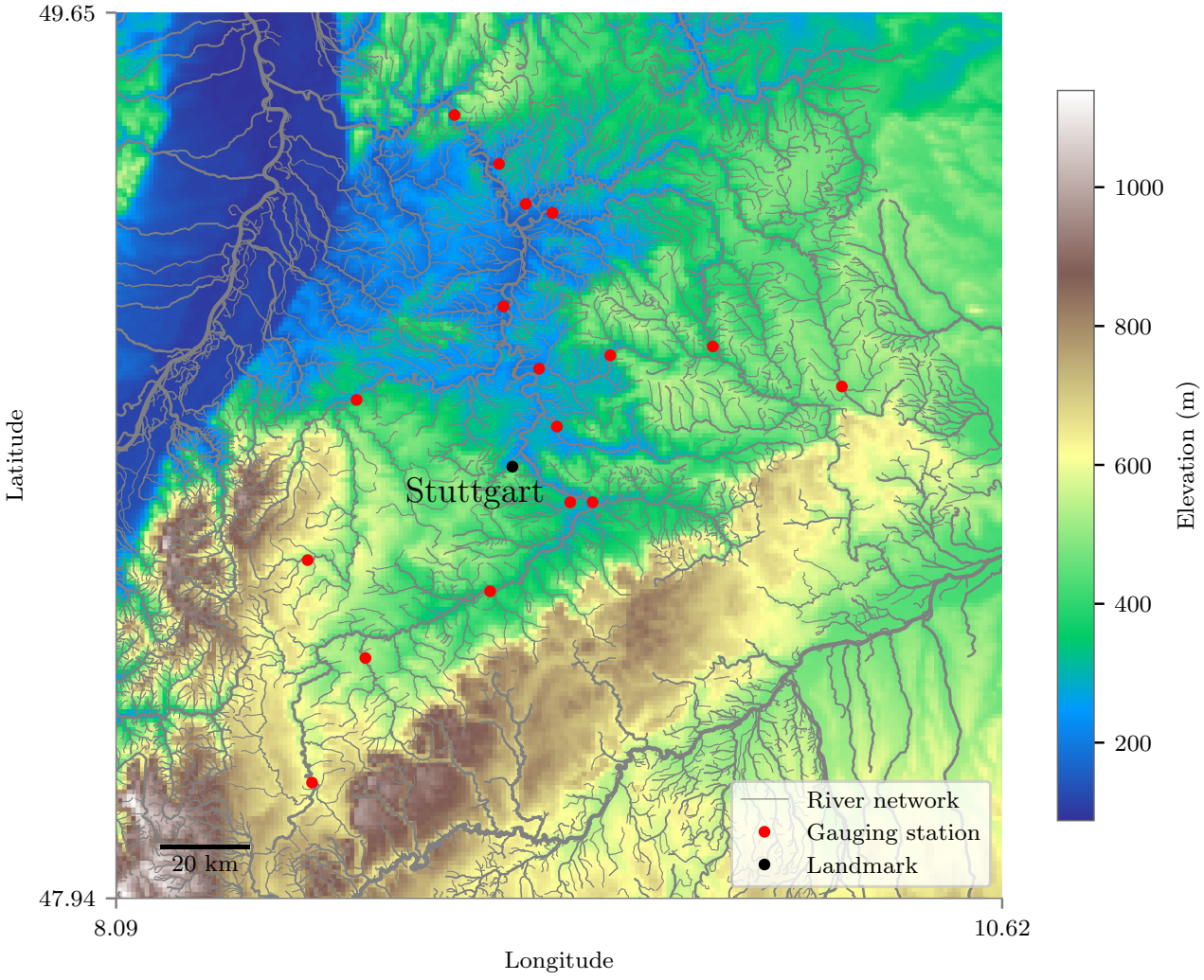

Figure 3. The study area used in this work is the Neckar River catchment in Southwest Germany.

such as soil moisture or aquifer rechargewithin its hidden states. It therefore allows the model to align the hidden states with the physical state of the system before transitioning to , through its hidden state representations. This alignment helps the model transition smoothly to predictive, open-loop mode, where future discharge is estimated without access to ground-truth values.

Following After the warm-up phase, DRRAiNN transitions into an open-loop mode for the remaining 10 days of the each sequence. In this predictive mode, the discharge model feeds its own discharge estimations into previous discharge estimations as inputs for subsequent time steps. The rainfall-runoff modelon the other hand continues to be provided with historical , in contrast, continues to receive observed precipitation and solar radiation . This is not a realistic setting for operational as inputs throughout the sequence. While informative, this setup does not reflect realistic operational conditions for discharge forecasting. Especially precipitation forecastingis a hard problem and currently no algorithm exists that could Precipitation forecasting, in particular, remains a major challenge. Currently no algorithm can accurately predict precipitation 10 days into the future on a ahead at a spatial resolution of $4\ \mathrm{km} \times 4\ \mathrm{km}$scale. However, this setup is useful well suited for knowledge discovery concerning hydrologic processes, which is the main focus of this paperprimary focus in this work. We leave the operational evaluation of our model on historical precipitation forecasts evaluation of DRRAiNN under realistic, forecast-based conditions for future work.

We use the mean squared error (MSE) on the computed on station-wise standardized discharge data as both the training and validation loss. This ensures that the training process does not disproportionately favor stations with higher discharge magnitudesStandardization ensures that stations with larger discharge values do not dominate the loss, promoting a balanced learning across all stations. Training is conducted with performed using truncated backpropagation through time (TBPTT), employing a scheduled truncation length . At the beginning of training, we let DRRAiNN learn relationships in the data that are more local in time by backpropagating the loss over subsequences of where the truncation length increases progressively over the course of training. Initially, we backpropagate the loss over 1 day . Note that our model operates on an hourly time scale, which means that a 1 day sequence consists of 24 time steps. Throughout sequences (24 time steps) to help DRRAiNN focus on short-term temporal relationships and stabilize learning. Over the course of training, we increase the truncation length, thereby allowing enabling the model to learn relationships that are increasingly distant in timelonger-term dependencies. The truncation length schedule , which is shown in 1, was determined empirically. We adjust Table 1. We adapt the batch size such that training can take place on a single to fit the model within the memory constraints of a single NVIDIA A100 graphics card. GPU, with total training time remaining under 8 h. A forward simulation of a 20 day sequence takes approximately 4 s.

To improve generalization and account for model variability due to random initialization, we train five independent instances of DRRAiNN for each per experiment, each initialized with a different seedfor the random number generator. Results are reported . We report test results based on the three seeds runs with the lowest validation loss , a practice we consistently apply out of five seeds. This selection procedure is applied consistently to both the primary model and its ablationsall ablation variants. We use Ranger (Wright, 2019) with the learning rate set to the Ranger optimizer (Wright, 2019) with a learning rate of 0.0025 to optimize the 30.600 30 600 parameters in DRRAiNN, which takes about 7 hours. We . To stabilize training, we clip the gradient if its norm exceeds 1to avoid large jumps at steep regions in , thereby preventing large parameter updates in steep regions of the loss surface. We use hydra to manage our experiment configurations (Yadan, 2019).

Table 1. Truncation length schedule in days for TBPTT

#Epochs	Truncation length	Batch size
10	1	256
4	2	128
2	4	64
1	10	32
1	20	32

To increase the size of the training data set and improve generalization, we ~~perform~~ apply data augmentation. The symmetry group of the square contains eight ~~symmetries, namely: identity~~elements: the identity, rotations by 90°, 180°, and 270°, ~~rotation by 90, 180, and 270 degrees,~~ and reflection in the x, y, and ~~the two~~ both diagonal axes. For each training sequence, we apply a uniformly sampled symmetry to the ~~following~~ spatial variables in each time step~~: elevation, precipitation, solar radiation, and the mask that is used to translate from grid to graph~~ . We ensure physical consistency by tapping into the runoff embeddings at the transformed station locations. The river discharge model's graph structure remains unchanged by this augmentation.

2.5 Benchmark model: European Flood Awareness System

To provide context for ~~DRRAiNNs~~ DRRAiNN's performance, we compare it to the European Flood Awareness System (EFAS), an established and operational distributed process-based model. ~~Since~~ We use publicly available EFAS reanalysis data~~is readily available for download, we do not have~~ , which eliminates the need to tune EFAS ourselves. This avoids potential biases ~~arising from unequal effort in tuning~~ that could arise from allocating unequal tuning effort to the benchmark model versus ~~the self-developed~~ our own model. While DRRAiNN achieves higher performance than EFAS in many scenarios, our ~~focus is not solely on outperforming EFAS but on demonstrating~~ primary aim is to demonstrate the potential of distributed neural networks for river discharge estimation, rather than merely outperforming EFAS.

EFAS simulates runoff on an approximately $1.5\ \mathrm{km} \times 1.5\ \mathrm{km}$ grid with a temporal resolution of $6\ \mathrm{h}$~~, which is similar to our setup~~. It receives as inputs static maps describing topography, river networks, soil, and vegetation, as well as meteorological forcings such as precipitation, temperature, and potential evaporation.

While EFAS serves as a useful benchmark, the comparison to DRRAiNN is not perfectly fair due to fundamental differences in the input and output variables. Both models receive gridded meteorological forcings, but DRRAiNN additionally receives discharge measurements during the tune-in period. In contrast, EFAS does not use discharge measurements as input ~~. Instead, these are used exclusively for~~ but relies on them for offline model calibration. Furthermore, DRRAiNN ~~estimates discharge only at station locations where observations are available, while EFAS estimates discharge in all gridcells~~produces discharge estimates only at gauging station locations, whereas EFAS generates discharge predictions across the entire spatial grid. EFAS also relies on additional input variables not used by DRRAiNN, such as soil type, vegetation, temperature, and potential evapotranspiration. ~~This makes EFAS particularly powerful but also less transferable to regions where~~ While this makes EFAS

a powerful tool, it also limits its applicability in regions lacking such detailed input data~~of this kind might be unavailable~~.
Another difference lies in the precipitation data used: EFAS ~~uses~~ relies on EMO-1, a ~~6-hourly~~ 6 mathrmh product interpolated from weather station data, whereas DRRAiNN uses RADOLAN, a ~~radar-derived dataset with finer~~ radar-based dataset offering higher spatial and temporal resolution. As a result, a direct comparison between EFAS and DRRAiNN is not valid. Nonetheless, ~~the EFAS data can serve~~ EFAS serves as a baseline ~~and an orientation for the performance regime we should be able to match~~to contextualize the expected performance range of DRRAiNN. We thus emphasize that our goal is not to directly compare performance but to provide a baseline that allows us to place the principled quality of DRRAiNN's performance with respect to alternative state-of-the-art forecasting approaches.

2.6 Evaluation

Besides ~~the depiction of hydrographs at some of the modeled~~ visualizing hydrographs for selected gauging stations, we ~~employ the following evaluation metrics to assess the performance of DRRAiNN~~ evaluate DRRAiNN using four standard metrics in hydrology: Kling-Gupta efficiency (KGE, (Gupta et al., 2009)), Nash-Sutcliffe efficiency (NSE, (Nash and Sutcliffe, 1970)), Pearson's correlation coefficient (PCC), and the mean absolute error (MAE). We report all ~~of these metrics because they are widely used in the hydrological sciences and because there is~~ four metrics because each highlights different aspects of model performance, and no single metric ~~that does not have any disadvantages (Gupta et al., 2009). One advantage of the MAE is that it provides a direct and intuitivemeasure that shows to which extent the models' estimations are off as it has~~ is free from limitations (Gupta et al., 2009). MAE is particularly intuitive, as it is expressed in the same unit as ~~the measured quantity. As no normalizationtakes place in its computation, though, this metric is disproportionately influenced by~~ discharge and directly quantities the average deviation between predictions and observations. However, because it lacks normalization, stations with larger ~~discharges. The PCC shows how much variation is shared~~ discharge magnitudes contribute disproportionately to the overall MAE. PCC quantities the strength of linear association between the observed and estimated discharges~~, however, it does not account for~~. While it captures shared variability, it is insensitive to systematic differences in scale or bias. To also capture the scale, the NSE was developed, which can be seen as a mean squared error that is weighted by the variance of the observed discharge. The NSE also does not account for bias, though, which is why the KGE was ~~introduced to capture~~ developed to jointly evaluate correlation, bias, and ~~variance~~variability. When computing KGE and NSE values, we use station-wise means and variances calculated from the training data set~~as done in (Kratzert et al., 2019)~~, following the approach in Kratzert et al. (2019). For KGE, NSE, and PCC, higher values ~~are better with~~ indicate better performance, with a maximum of 1 ~~corresponding to a perfect fit. For MAE~~representing a perfect match. In contrast, lower values ~~are better~~ of MAE are better, with 0 ~~corresponding to~~ indicating a perfect fit.

~~When performing~~ During open-loop inference, we ~~also~~ evaluate metrics separately for ~~the different number of~~ each open-loop ~~steps performed (~~step, where the first ~~one should be similar to~~ step resembles closed-loop estimation~~). This way we can see to which extent performance drops~~. This allows us to assess how model performance degrades with increasing lead times. ~~Even though~~ Although DRRAiNN was only trained on sequences that span 20 days, we ~~always evaluate on 100~~ evaluate it on 50 day sequences to ~~see whether our model can generalize with regards to lead time~~investigate its ability to generalize beyond the

training horizon. Additionally, we will plot the performance of the models against the mean discharge of the different stations to ~~see whether we find systematic relationships between these quantities~~identify potential systematic dependencies between flow magnitude and model accuracy. In all cases, we ~~remove~~ exclude the initial 10 days tune-in period before calculating metrics and producing plots.

~~As discussed above, we are interested in more than just good performance in terms of matching hydrographs and good metrics.~~ With knowledge discovery being the main motivation of this work, we ~~will~~ also test DRRAiNN ~~on~~ for physical plausibility. A physically implausible model might learn spurious relationships in the data. It could, for example, exploit the DEM to encode local biases that ~~let water spawn or disappear without this process being driven by the~~ lead to gains or losses of water not driven by meteorological forcings. By retrospectively inferring catchment areas from observed dynamics, we assess whether the rainfall-runoff model successfully propagates water ~~over~~ across the landscape. The procedure is as follows: After a forward pass, we compute saliency maps by taking the gradient of the ~~last time step output~~ final discharge estimate with respect to the precipitation ~~input. The result is a so-called saliency map which tells~~ inputs. These maps tell us to which extent the model's output depends on the precipitation in each grid cell and time step. We multiply this gradient by the precipitation itself to focus the analysis on cells in which precipitation occurred. ~~By doing~~ To examine how the attributions change over time, we split the sequence into subsequences of 5 days over which we take the mean. We do this for each station separately and ~~visualizing~~ visualize the resulting attributions ~~, we can see which areas on the map contribute to the~~ to identify which areas contribute most to discharge estimation at ~~the corresponding~~ each station. To reduce noise, we ~~do this for every sequence in our validation data set~~ repeat this process across all test sequences and average the ~~outcomes~~resulting attribution maps.

We compare the resulting attributions with catchment areas delineated from elevation data ~~, as those are commonly used in hydrology~~using standard hydrological techniques, which are widely used in the field. To evaluate their agreement quantitatively, we employ the following measure when comparing DRRAiNN to the ablated models: For each station, the attributions are ~~first~~ standardized to lie between 0 and ~~1.~~ 1 using min-max scaling. We then compute the Wasserstein distance between the attributions ~~within~~ values inside the delineated catchment area and those outside ~~of~~ it. A higher Wasserstein distance indicates better alignment between the attributions and the catchment areas delineated from elevation data. This quantitative measure complements the qualitative comparison, providing stronger evidence for our model's ability to propagate water ~~over~~ across the landscape in a physically plausible way. Specifically, it ~~suggests~~ indicates that the model has ~~learned from the observed dynamics alone that water flows downward~~implicitly learned the topographic structure of flow direction – i.e., that water generally flows downhill – solely from observed discharge dynamics.

3 Results

~~For evaluating~~ To evaluate DRRAiNN, we first ~~provide~~ present hydrographs and compare performance with EFAS to contextualize DRRAiNN's results. We furthermore show that DRRAiNN ~~has the potential to infer catchment areas, thus highlighting the system's potential due to its full differentiability~~ can retrospectively infer catchment-like structures, thus demonstrating how full differentiability supports physical interpretability.

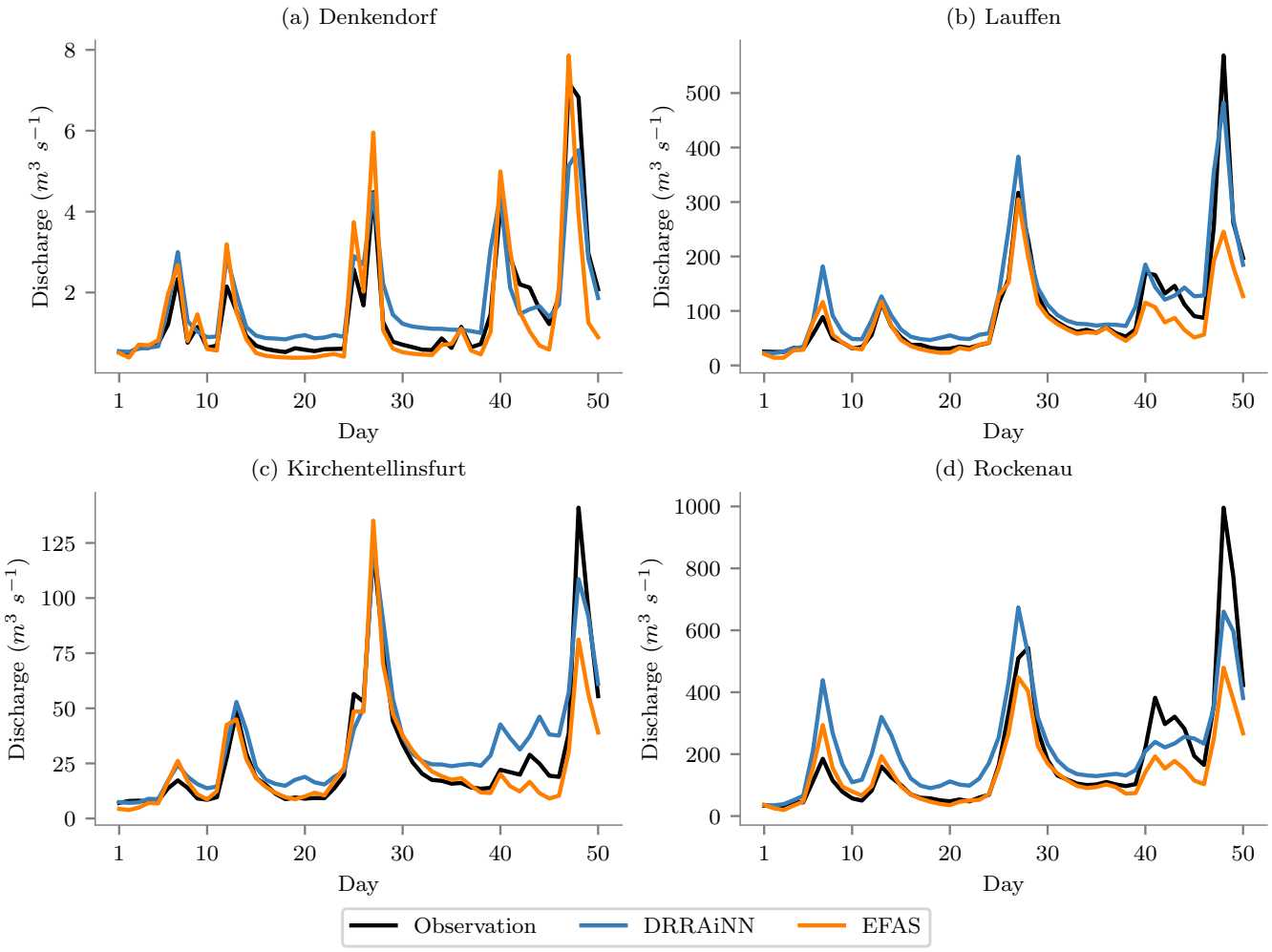

Figure 4. Hydrographs ~~of~~ showing observed discharge, ~~discharge simulated by~~ EFAS simulations, and ~~discharge inferred by~~ predictions from one of ~~our~~ five DRRAiNN model instances ~~out of five with~~ for lead times of up to ~~100~~ 50 days. ~~They correspond to~~ The four panels show the stations with the lowest (a) ~~,~~ and highest (d) mean discharge, as well as ~~to those stations with~~ the ~~best KGE performance of~~ stations where EFAS (b) ~~,~~ and ~~our model on average~~ DRRAiNN (c) ~~, respectively~~ achieve the best KGE performance on average on the validation set. ~~We chose those sequences~~ For each station, we selected the sequence from ~~our validation data~~ the test set ~~that have~~ with the ~~largest variance in~~ highest discharge variance, as variance likely ~~acts~~ serves as a proxy for prediction difficulty.

3.1 Hydrographs

~~First, EFAS produces well-matching and plausible hydrographs~~ EFAS produces hydrographs that match both the shape and magnitude of observed discharge, rendering it a strong contestant (Fig. 4). As EFAS produces gridded outputs, it is necessary to ~~pick the correct grid cells to compare the model outputs at the specific stations. We likely chose the correct cells, since the~~

reanalysis hydrographs produced by EFAS match the historical observations well for all considered stations. Within the low flow regime (Fig. 4a), it seems that EFAS tends to underestimate dischargeextract outputs from EFAS grid cells that correspond to the station locations in order to make meaningful comparisons.

Second, the results show that DRRAiNN can produce DRRAiNN also produces plausible hydrographs that closely match the observed dischargeswell, too, especially during the first few days of the estimation. This includes both low flows (Fig. 4a) as well as and high flows (Fig. 4d)with no apparent . No systematic difference in performance . Throughout the 100 days, however, the hydrographs tend to match the observed discharge less and less, which is expected as is observed across flow regimes. Since DRRAiNN operates autoregressively : After the closed-loop tune-in phase (which is not shown here), DRRAiNN receives its last inferred discharge – using its own discharge estimates as input in the next time step . Therefore, the error accumulates over time. However, considering that DRRAiNN was trained on 20 day sequences only, it is surprising to see that – error can accumulate over time, leading to gradual decline in accuracy. Nonetheless, it is notable that the model is in general able to hit peaks even after 80 days. The large peak on day 80 in Lauffen and Rockenau (Fig. 4b and 4d) is underestimated by both models, indicating a bias towards lower values. almost 50 days, despite being trained only on 20 day sequences.

3.2 Performance

Overall, DRRAiNN can outperform EFAS in the initial days of the estimation horizon in outperforms EFAS in all considered metrics (Fig. 5). Please note that, since discharge values are never fed into EFAS but only used for calibration, the performance of EFAS is Since EFAS does not incorporate discharge values during inference, we report its mean performance over lead times as constant. As expected, the performance of DRRAiNNdecreases described above, DRRAiNN's autoregressive nature causes errors to accumulate over time, since its autoregressive nature leads to error accumulation as described aboveleading to a gradual decline in performance at longer lead times.

The KGE plot (Fig. 5a) shows that DRRAiNN produces significantly better results during the initial daysindicates that DRRAiNN is able to maintain strong performance over time. Averaged over the seeds, starting with a KGE of about 0.76, it takes about 48 days before 0.71, our model's estimations become worse than stay above those of EFAS on average, even though DRRAiNN was only ever trained during the entire estimation horizon of 50 days, despite having been trained only on 20 day sequences. Two instances of our model can keep up with EFAS even after 100 days. The In contrast, the NSE plot (Fig. 5b) shows our model starting with a value of about 0.81, again with EFAS beating it after about 45 dayson average. The fact that one of DRRAiNN's instances intersects EFAS' line earlier in the KGE plot than in the NSE plot points to a larger systematic bias in this instance compared to EFAS, as this is the main difference between those two metricsgradual decline in performance over time with a decrease from 0.72 to 0.62 over the estimation horizon. Regardless, even after 50 days, all seeds show higher NSE values than EFAS. The PCC plot (Fig. 5c) shows a strong linear relationship between the observed and inferred discharge values with a observed and estimated discharges, with an average value of about 0.9 on average at the beginning. Here, at the start. DRRAiNN captures this relationship better than EFAS during the first 40 daysover the entire estimation horizon. Note that the linear correlation is also part of KGE and NSE. As the MAE allows direct interpretation, its plot (Fig. 5d) shows that EFAS is off by about $5.7 m^3 s^{-1}$ $6.5\ m^3\ s^{-1}$ on average, while DRRAiNN with $3.3 m^3 s^{-1}$ $3.9\ m^3\ s^{-1}$ on average on the

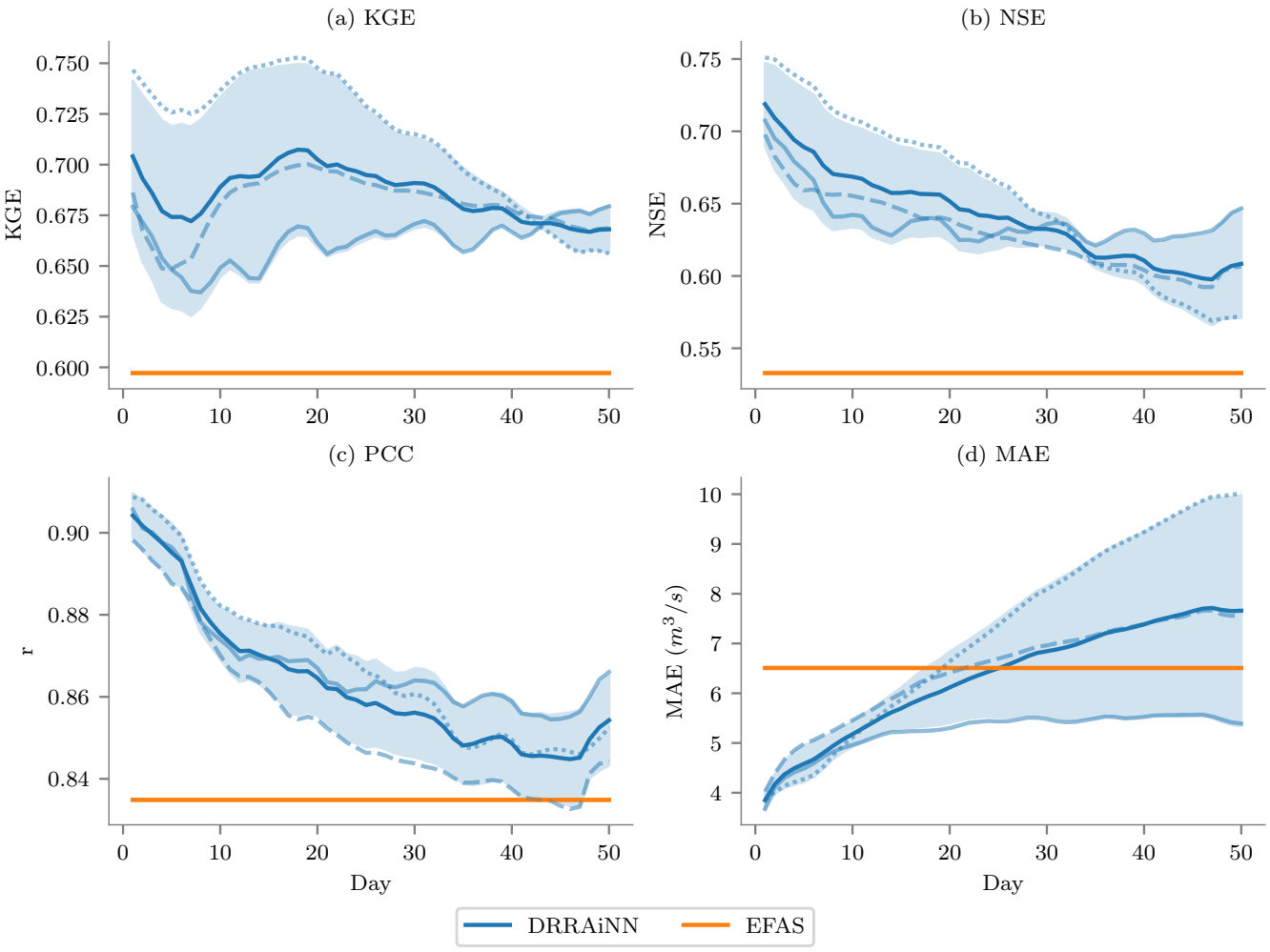

Figure 5. Performances ~~based on different metrics~~ of the best three out of five DRRAiNN model instances, compared to EFAS across different metrics and ~~DRRAiNN with~~ lead times up to ~~100~~ 50 days. ~~The results~~ Results are averaged ~~over the~~ across all stations~~, and the different seeds of DRRAiNN are depicted with different~~ . Each line ~~styles~~style corresponds to a distinct DRRAiNN instance.

first day produces a considerable smaller error. After about ~~40~~ 25 days, EFAS ~~produces better results~~ yields a lower MAE on average.

All ~~considered metrics show that the individual model instances perform differently. The order, however, is not fixed but depends on the considered metric and even more so on the considered~~ metrics reveal differences in performance across the model instances trained with different random seeds. However, the relative ranking of model instances varies depending on the specific metric and lead time. Some seeds perform better during the initial days, while others are better with greater lead times: ~~The best two instances, for example, switch after about 40 daysin KGE. The seeds differ in weight initialization only, meaning that some instances~~ For example, in the KGE plot (Fig. 5a), the ranking changes after about 42 days. The difference between instances are due to random weight initialization and the order of batches only. These stochastic factors may lead some instances to start the training with a larger bias towards capturing short-term~~and others~~ , while others start with a larger bias towards capturing long-term relationships in the data.

The ~~erratic lines~~ plots in Fig. 6 show that ~~stations vary in difficulty: Discharge of some stations is harder to estimate than that of others, regardless of which metric is considered.~~ some stations consistently yield more accurate discharge estimates than others. This observation holds across all evaluation metrics. Which stations are harder to estimate, however, is different across the metrics~~since the metrics focus on different aspects as discussed above. The different seeds of DRRAiNN , and more interestingly, also EFAS , agree~~ , reflecting the distinct sensitivities each metric has, as discussed previously. Interestingly, both the different DRRAiNN instance and EFAS show partial agreement on which stations are ~~harder to estimate to some extent: The~~ more difficult to model. For example, the KGE values in Fig. 6a ~~, e.g.,~~ show that Altensteig ~~, Rottweil, and Kirchentellinsfurt consistently belong to the easier ones~~and Stein are consistently easier to estimate, while Oppenweiler, Bad Imnau, and Murr ~~belong to the harder ones. We assume that this is related to unobservable underground flows and pipes, however, this could be further~~ are among the most challenging. The reasons for this discrepancy – such as differences in catchment size, land cover, or upstream complexity – could be analyzed in future work.

The regression lines ~~help us to see whether there is a systematic relationship between a station's mean discharge and its predictability~~indicate whether model performance correlates with average discharge levels across stations. We performed linear regression~~here, and ~~ ; the regression lines ~~are only~~ appear exponential due to the logarithmic scaling of the x-axis. ~~The KGE plot (Fig. 6a) shows~~ All metrics, except MAE, show that both models tend to perform better at stations with higher ~~mean~~ discharges. This effect is more pronounced in EFAS, while our model exhibits a more balanced behavior. ~~This is even more the case if we consider the NSE (Fig. 6b) and the PCC (Fig. 6c). Here, DRRAiNN's performance barely depends on the station's mean discharge at all, which cannot be said about EFAS. Differences in the patterns of the KGE~~ The differences between KGE and NSE patterns (Fig. 6a ~~) and NSE plots (Fig. 6~~and b) show that the models have different biases for the different stations, since ~~this is the main difference between KGE and NSE, as discussed above. Both ,~~ KGE accounts for both bias and variability, while NSE only captures variance. Both DRRAiNN and EFAS ~~,~~ produce significantly larger MAEs with increased mean discharge (Fig. 6d). This is expected ~~, though, as~~ since MAE does not account for the stations' mean discharges or their variability in discharge, unlike the other metrics.

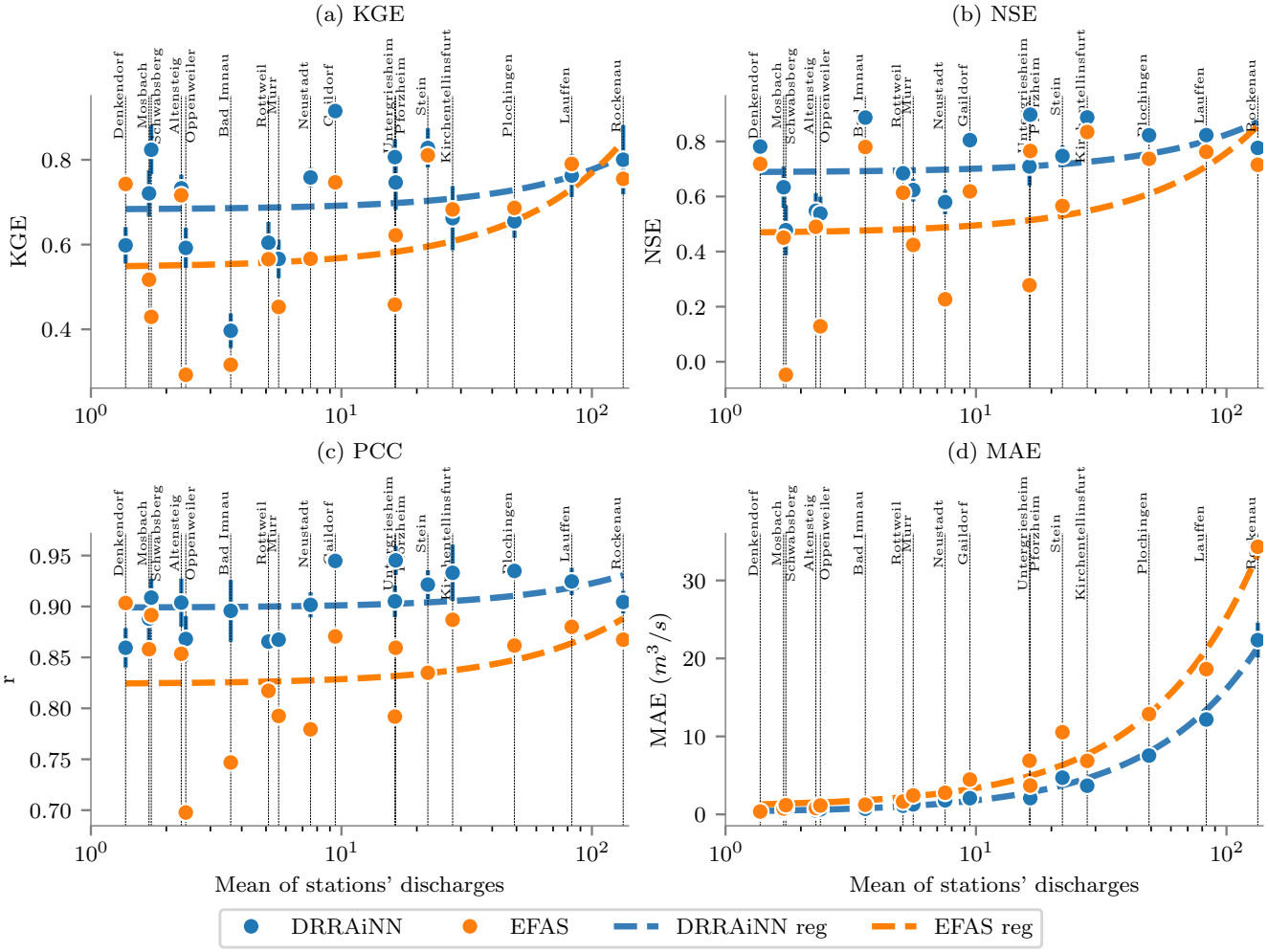

Figure 6. Performances of the best three out of five DRRAiNN model instances and EFAS ~~on~~ at a 1 day lead time ~~based on~~ across different metrics ~~at the different~~ and stations. The x-axis ~~denotes~~ shows the logarithmic ~~means of the stations' discharges~~mean discharge at each station. ~~The blue shadow depicts~~ Blue vertical lines depict the standard deviation ~~over the different~~ across DRRAiNN seeds. ~~The dashed~~ Dashed lines represent ~~a~~ linear ~~regression on~~ regressions between the ~~logarithmic stations' means~~ log-mean discharge and ~~the~~ corresponding metric.

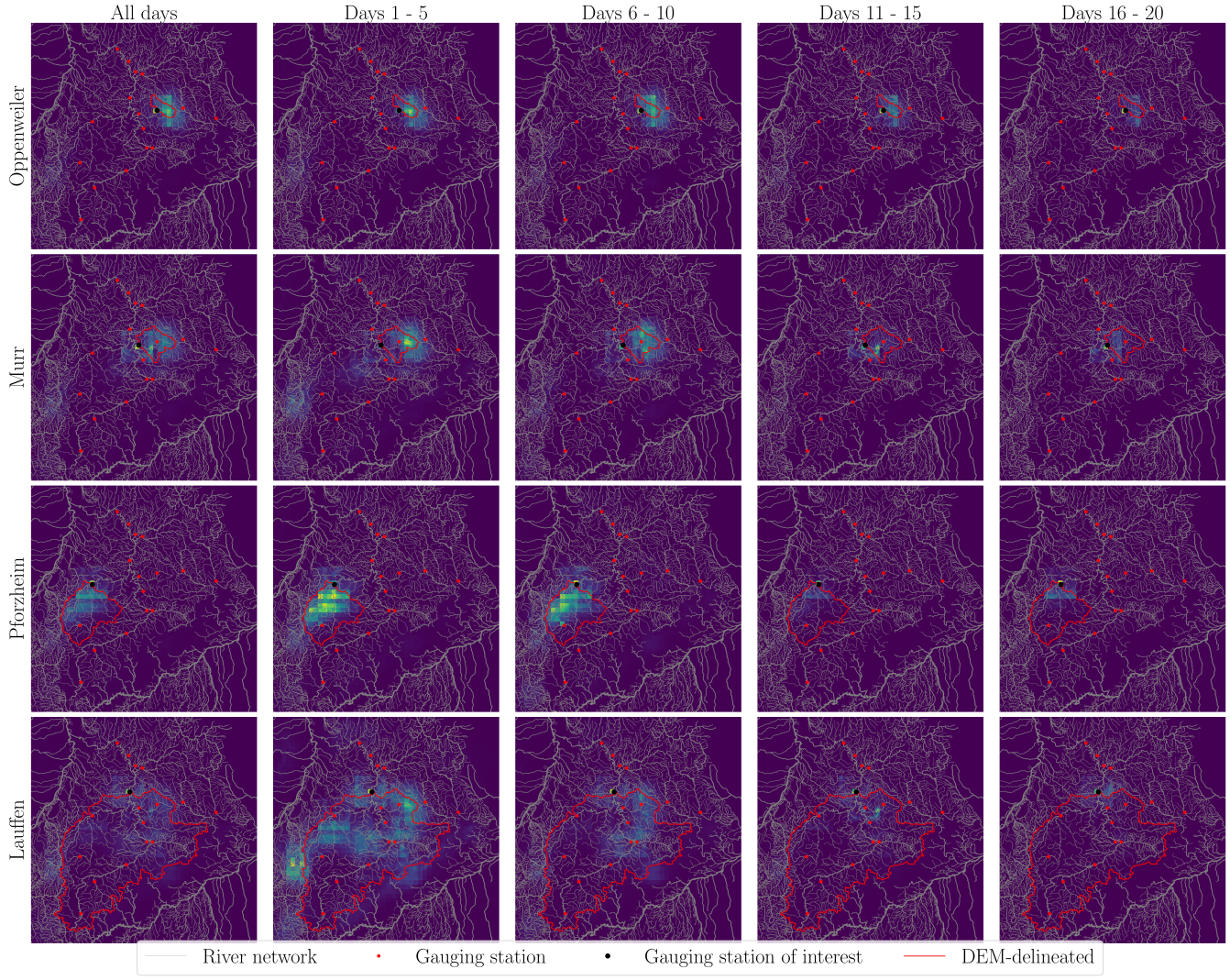

Figure 7. Attribution maps of precipitation for discharge estimation at selected stations and time intervals, averaged over all ~~validation~~ test set sequences. ~~Brighter~~ Darker colors indicate grid cells where precipitation has a stronger influence on the estimated discharge at the corresponding station. For comparison, ~~the~~ traditional catchment areas delineated from elevation data are outlined in red. This juxtaposition highlights the agreement between data-driven attributions and physically derived catchment boundaries. The attribution method ~~used to compute these attributions~~ is described in detail in ~~Subsection~~ Subsect. 2.6 of the main text.

3.3 Catchment area inference

We ~~can successfully reconstruct~~ observe that DRRAiNN implicitly infers physically plausible catchment areas~~that DRRAiNN must have inferred implicitly (~~, as shown in Fig. 7~~)~~. Lighter areas ~~show higher importance~~ indicate regions with higher importance of precipitation for estimating discharge at the corresponding station. These ~~areas correlate~~ attribution patterns

spatially overlap with the catchment areas ~~depicted in red, which are~~ delineated from elevation alone ~~. The results are not perfect. One should keep in mind, however, that DRRAiNN is trained on daily discharge measurements. This means for sharp delineations, the training data set ideally needs to contain sequences in which it rained within the area, but not outside, over the extent of a 24 h period. As precipitation is very dynamic on this time scale, the chances for this are relatively low. In the future, we expect much better results if we go from daily to hourly discharge data.~~ (depicted in red). The first four columns visualize attributions for subsequences of 5 days length to illustrate temporal changes in spatial influence. There is a tendency of the area of influence to increases in size the further we look into the past. This suggests that DRRAiNN propagates encoded water quantities along the landscape in a manner that aligns, at least to some extent, with physical flow processes. The last column shows attributions averaged over the whole 20 day sequences.

In the case of Pforzheim~~(Fig. 7b), DRRAiNN missed~~ , DRRAiNN assigns low importance to an area in the lower right part~~that is considered part of~~ , despite its inclusion in the delineated catchment area. This ~~could be explained by underground flows that were found in previous studies near Pforzheim(Ufrecht, 2002). Water that would~~ discrepancy could be related to known underground flows near Pforzheim, as reported in Ufrecht (2002). In the absence of subsurface flows, water would be expected to pass through Pforzheim~~if no underground flows existed, instead flows~~ ; however, due to the presence of underground flow paths, it instead moves towards the southeast, entering the Neckar River network ~~in a different channel~~via an alternative route. Our results ~~might be evidence that DRRAiNN~~ suggests that DRRAiNN may have detected these unobservable underground flows from precipitation and discharge dynamics~~, however,~~ . However, this hypothesis arguably needs more investigation in the future.

Note ~~,~~ that these results ~~mainly~~ primarily serve as a proof of principle: We ~~only present the results of the best seed here. This is valid as we performed temporal validation on all seeds beforehand to check for their temporal generalization capabilities. This could also be done if one was interested in building an operational model.~~ present results from the seed producing the clearest attributions; others yielded qualitatively worse results. However, it is important to keep in mind that DRRAiNN is trained on daily discharge measurements. Learning sharp catchment delineations would require the training data set to contain sequences in which it rained within the area, but not outside of it, over the extent of a 24 h period. As precipitation is very dynamic on this time scale, the chances for this are relatively low. In the future, we expect sharper results if we go from daily to hourly discharge data.

3.4 Ablations

To assess both the physical plausibility and contributions of specific architectural components, we conducted a series of ablations on DRRAiNN (Appendix A). First, we showed that DRRAiNN can exploit the DEM as a positional encoding by training, validating, and testing it on a rotated DEM. However, it did result in slightly worse performance and less physically plausible behavior (Appendix A1). Next, we evaluated the model's inductive bias in distinguishing between spatially extended and local processes (Appendix A2). Last, we removed the hypernetworks to examine their impact (Appendix A3). Both ablations led to performance degradation across most metrics and lead times. However, the differences were not always

significant. Importantly, neither ablated model was able to produce physically realistic catchment areas, as demonstrated both qualitatively and quantitatively.

4 Discussion

~~In this work, we present~~ We introduce DRRAiNN, a fully differentiable, fully distributed neural network architecture ~~that successfully estimates~~ for estimating river discharge from past discharge, ~~an elevation map, gridded precipitation, and gridded~~ gridded elevation maps, and gridded precipitation and solar radiation. ~~Individual instances of DRRAiNN can produce better KGE~~ DRRAiNN demonstrates better performance than EFAS ~~with~~ on lead times of up to ~~100~~ 50 days. This ~~shows~~ indicates that DRRAiNN can produce ~~reasonable~~ valid estimations far into the future ~~even though it was only trained on~~ despite it being trained on sequences of only 20 ~~day sequences~~ days, including a warm-up period of 10 days.

Our analysis reveals that ~~discharge estimation at various stationsexhibits differing levels of difficulty. Notably,~~ the difficulty of discharge estimation varies across gauging stations. Interestingly, both DRRAiNN and EFAS ~~tend to identify~~ consistently struggle with the same stations~~as challenging to predict~~, suggesting that the difficulty is intrinsic to the stations and their associated data rather than specific to the model architecture. ~~This variability in prediction difficulty likely arises from several factors . Stations influenced~~ Several factors likely contribute to this variability. For example, stations affected by unobserved variables ~~like~~ such as complex subsurface topography, land cover heterogeneity, or anthropogenic factors (e.g., dam operations) may ~~pose greater challenges for both models~~be inherently harder to model. Furthermore, spatial variations in the quality of input data could contribute to discrepancies in performance. Future investigations ~~employing~~ using attribution techniques could offer deeper insights into these station-specific variations and guide the development of architectural modifications or ~~regularizations~~ regularization to address these challenges effectively.

~~We performed several ablations on our model (Appendix A). First, we showed that DRRAiNN can exploit the DEM as a positional encoding by training and validating it on a rotated DEM. While this did not lead to worse performance, it resulted in significantly less physically plausible behavior (Appendix A1). Then, we checked whether the inductive bias that makes the model distinguish between spatially extended and local processes is useful (Appendix A2). Last, we removed the hypernetworks to examine their impact (Appendix A3). For some combinations of performed ablation, metric, and lead time, there is no significant differences in terms of performance compared to the original DRRAiNN model. However, none of the ablated models is able to produce physically realistic catchment areas, which we showed qualitatively and underlined quantitatively.~~

Our ablation studies ~~highlight the importance~~ show the benefits of distinguishing between spatially extended and local processes, ~~as well as the incorporation of~~ and of incorporating hypernetworks. The ~~inability of~~ reduced performance and failure of the ablated models to produce realistic catchment areas suggests that these components encode crucial hydrological processes, such as water movement ~~over~~ across complex topographies. This ~~finding indicates that certain inductive biases not only~~ suggests that incorporating appropriate inductive biases can both improve model interpretability ~~but also prevent~~ and reduce the risk of learning spurious correlations.

Interestingly, the model instance ~~achieving the best attribution maps does not correspond to the one achieving the best performance metrics. This highlights~~ that produces the most physically plausible attribution maps is not the one with the best predictive performance. This points to a trade-off between optimizing for predictive accuracy and ~~ensuring the model behaves in a physically meaningful way. It suggests that while the metricsmeasure how well the model captures patterns in the training data, they~~ encouraging physically realistic model behavior. This suggests that conventional performance metrics, while effective at evaluating predictive accuracy, may not fully ~~capture the alignment with~~ reflect whether the model adheres to underlying physical principles.

~~Our choice of input datasets was guided by considerations of temporal resolution, data sources, and practical availability, all of which impact model performance. Although EFAS uses EMO-1 for precipitation data, we opted for RADOLAN due to key differences: EMO-1 provides 6-hourly resolution and is interpolated from station data rather than derived directly from radar observations. While we expect minor differences in performance between RADOLAN and EMO-1, radar-derived datasets like RADOLAN generally offer finer spatial and temporal detail, which is advantageous for distributed models. Similarly, for solar radiation data, ERA5 was chosen due to its raster format and hourly resolution. Alternative datasets, such as those provided by DWD, are either available only as station-wise hourly data, which lack the required raster format, or as raster data aggregated monthly, which does not meet our temporal requirements. Daily datasets like EOBS could suffice if temporal patterns are encoded separately, but this would require additional preprocessing steps. If one aims to transition toward operational flood forecasting in the future, the choice of precipitation forecast will become critically important (Imhoff et al., 2022). Ultimately, all data products come with inherent uncertainties and errors, and our choices reflect a balance between data availability, temporal resolution, and model needs.~~

~~An increase in~~ Increasing the amount of training data ~~is always beneficial in machine learning~~generally enhances performance in ML. Currently, ~~DDRAiNN~~ DRRAiNN is not designed for scalability, as its application is expected to require retraining ~~for~~ in each specific context. A ~~first step towards improving its~~ natural step toward improving adaptability would be ~~to train DDRAiNN~~ training DRRAiNN on hourly discharge data. ~~We anticipate this would yield performance improvements and qualitatively better attributions, potentially even capturing the~~ This could improve performance and attribution quality, potentially enabling the model to trace the origins of individual ~~peaks in the hydrographs. To explore the model's spatial generalization capabilities, we aim to apply DDRAiNN to diverse catchments across Germany, Europe, or even globally. By validating it on catchments that are not part of the training data, we can systematically assess its ability to generalize to unseen regions. Improving this spatial generalization remains a key challenge and likely requires additional constraints or inductive biases in the model. Promising candidates are the incorporation of physical constraints like mass conservation (Hoedt et al., 2021; Harder et al., 2023; Wi and Steinschneider, 2023) or semantically splitting the hidden state of the rainfall-runoff model into surface and subsurface components. These enhancements could pave the way for future scalability and broader applicability.~~

~~As traditional process-based models make use of many more~~ discharge peaks. Since traditional PBMs rely on a wider range of input variables, feeding them as additional inputs could also lead to performance improvements in DRRAiNN~~as well~~. This includes land cover, ~~geology, soil~~ parent material, soil texture, vegetation, temperature, and potential evapotranspiration among

others. Interpretability methods can then be used to perform a sensitivity analysis, revealing which input variables are important when and, due to our model being fully distributed, where. These methods may also provide insights into the model's internal representations, potentially uncovering links to real-world hydrological variables.

~~Currently, DRRAiNN uses a warm-up period of 10 days for the hidden states to tune into the dynamics. The rainfall-runoff modelpotentially captures precipitation during this time to estimate soil moisture, which has a huge impact on infiltration. Therefore, soil moisture as an additional input variable is of special interest as it might allow us to get rid of the initial 10 days, thereby reducing training costs. An alternative would be to feed in the compressed precipitation history of the days or even weeks before (Traub et al., 2024b).~~ Several strategies can be employed to investigate DRRAiNN's spatial generalization capabilities. One approach is to leave out individual stations within a river network during training to evaluate generalization within hydrologically connected regions. A more demanding test of generalization would involve training and testing on different river networks. By testing it on catchments that are not part of the training data, we can systematically assess its ability to generalize to unseen regions. Ultimately, we aim to apply DRRAiNN to diverse catchments across Germany, Europe, or globally. Due to DRRAiNN's data-driven nature, discharge measurements will always be needed for training. However, recent advances in remote sensing may enable the application of DRRAiNN to ungauged river networks (Gigi et al., 2019).

~~Concerning output variables, DRRAiNN could also be used to estimate quantities other than discharge. For some measurement stations, additional water-related information, like turbidity, is available. To estimate turbidity, information about potential erosion can be helpful, as is provided by the RUSLE model (Renard et al., 1994), for example. Similarly to the catchment area inference performed in this study, a trained instance of such a model could be interrogated to infer the origins of measured turbidity, potentially informing us about sites of actual erosion. This information could be used to create policies for soil protection. Other variables of interest include the concentration of toxins and oxygen for similar applications.~~

~~Operational flood forecasting would be a safety-critical application of DRRAiNN. Therefore, it is important to quantify uncertainties as was suggested elsewhere before (Hrachowitz et al., 2013; Nearing et al., 2021). Equipping our model with this ability would allow us to provide confidence intervals when reporting inferred discharge values. In this regard, distributional parameter estimation is a technique where our architecture would produce an additional output that is interpreted as the standard deviation in a negative log-likelihood loss. Other techniques include Bayesian neural networks (Neal, 2012), Monte-Carlo dropouts (Gal and Ghahramani, 2015), and variational methods (Graves, 2011).~~

~~Another hurdle for operational flood forecasting is the inherent difficulty of obtaining sufficiently accurate, high-resolution precipitation forecasts with lead times of several days. Even though numerical weather prediction models, such as those provided by the DWD, are readily available, they are limited in predicting localized extreme precipitation events and reducing forecast uncertainty. In this work, we always assumed perfect forecasts by using historical observational precipitation data since we focused on the dynamics of water once it reaches the earth's surface. Therefore, further examination of DRRAiNN 's predictive abilities when provided with precipitation forecasts is needed to see whether it would be suitable for operational flood forecasting.~~

5 Conclusions

In this ~~study~~paper, we introduced DRRAiNN, a fully distributed neural network architecture that estimates river discharge from precipitation, solar radiation, elevation maps, and past discharge measurements from gauging stations. Despite being trained on sparse target data ~~, –~~ namely daily discharge observations from ~~only 17 stations ,~~ 17 stations over ten years – DRRAiNN outperforms the operational benchmark model EFAS in terms of KGE and NSE across various lead times. Beyond its predictive accuracy, DRRAiNN provides physically interpretable attributions, enabling the identification of precipitation sources contributing to discharge at specific stations. Our analyses highlight the importance of incorporating hydrologically meaningful constraints, or inductive biases. These biases not only enhance interpretability but also ~~ensure the model adheres to~~ help the model align more closely with physical principles, as evidenced by its ability to delineate realistic catchment areas. With its predictive performance, interpretability, and physical consistency, DRRAiNN represents a promising step forward in the application of neural networks to distributed hydrological modeling.

Code and data availability. The preprocessed data sets can be found at Scholz et al. (2024a). The code can be found at Scholz et al. (2024b).

Appendix A: Ablations

A1 Rotated elevation map

We ~~want to check whether DRRAiNN makes plausible use of~~ aim to assess whether DRRAiNN utilizes the elevation map in a physically plausible way – specifically, to propagate water ~~downwards over~~ downhill across the landscape. An alternative would be that DRRAiNN ~~uses the elevation~~ leverages the elevation map primarily as a positional encoding, allowing it to orient itself ~~in the landscape , exploiting it as a positional encoding. This way DRRAiNN can learn local biasesat the different positions in the map. Most likely, one will always observe a combination of both effects~~within the landscape and learning location-specific biases. In practice, both mechanisms are likely at play to some degree.

To examine this, we ~~here trainand validateDRRAiNN with~~ train, validate, and test DRRAiNN using the same elevation map as before, but rotated by ~~180 degrees. This has the advantage~~ 180°. This setup preserves the statistics of the elevation map~~having the same statistics as before, making this comparison fair~~ , ensuring a fair comparison.

For most metrics and lead times, ~~we do not find a significantly better performance of DRRAiNN if trained and validated~~ DRRAiNN performs better when trained and tested on the original elevation map ~~in contrast~~ compared to the rotated one (Fig. A1). ~~This confirms our suspicion that the model can exploit the elevation map~~ Nonetheless, its continued superior performance relative to EFAS – even with the rotated DEM – supports the hypothesis that DRRAiNN leverages elevation as a positional encoding. ~~However, in this case, we are not able~~ Remarkably, this still enables it to reconstruct plausible catchment areas to some extent (Fig. A2)~~, which is underlined by our quantitative measure~~ . However, our quantitative analysis (Fig. A3) ~~. This is evidence~~ shows that catchment areas are more accurately reconstructed when DRRAiNN is executed on the original DEM.

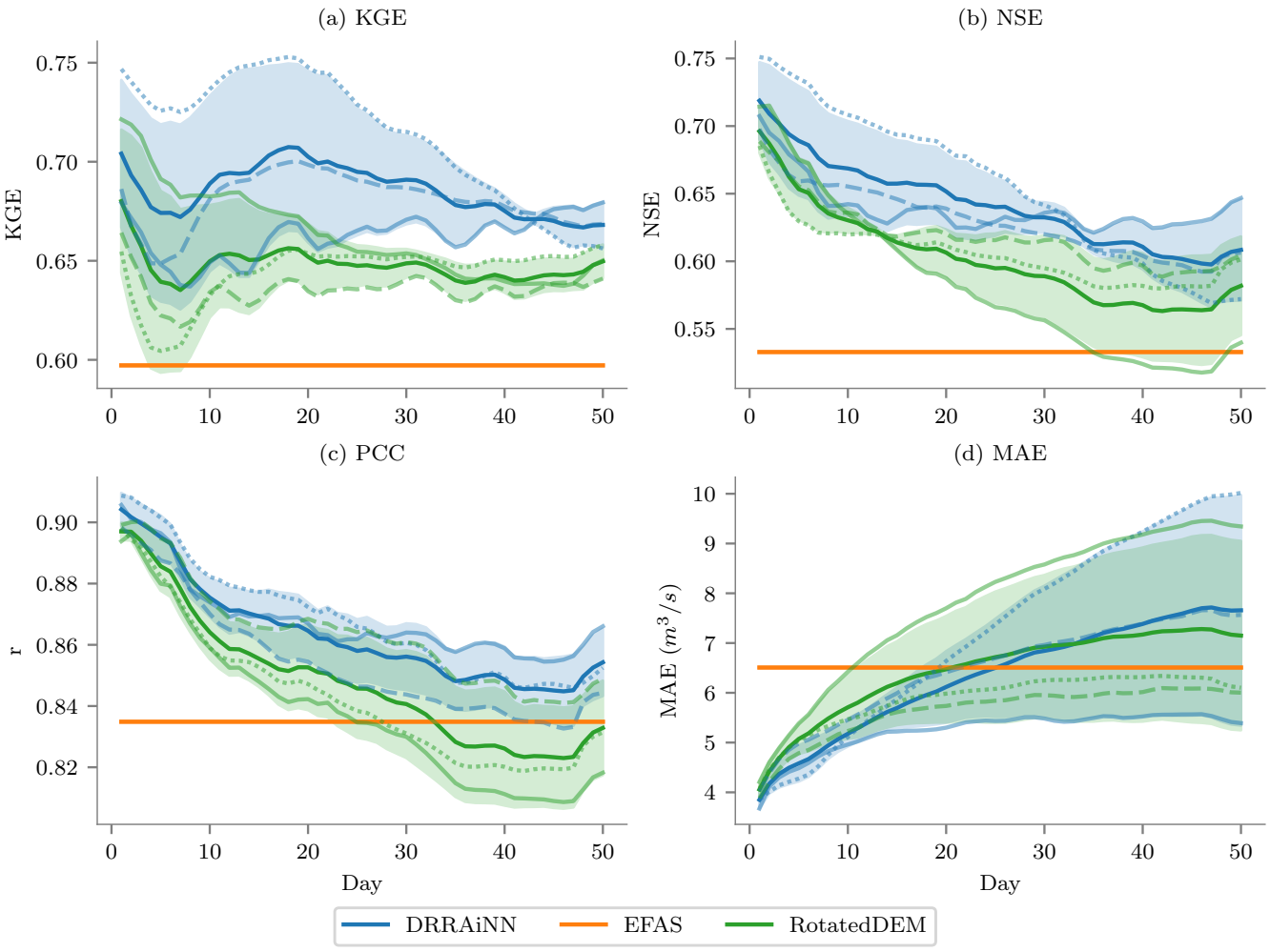

Figure A1. Performances ~~based on different metrics~~ of ~~EFAS,~~ the best three out of five DRRAiNN ~~,~~ model instances and DRRAiNN model instances on a rotated elevation map~~with~~ , compared to EFAS across different metrics and lead ~~time~~ times up to ~~100~~ 50 days. ~~The results~~ Results are averaged ~~over the~~ across all stations~~, and the different seeds of our model are depicted with different~~ . Each line ~~styles~~style corresponds to a distinct DRRAiNN instance.

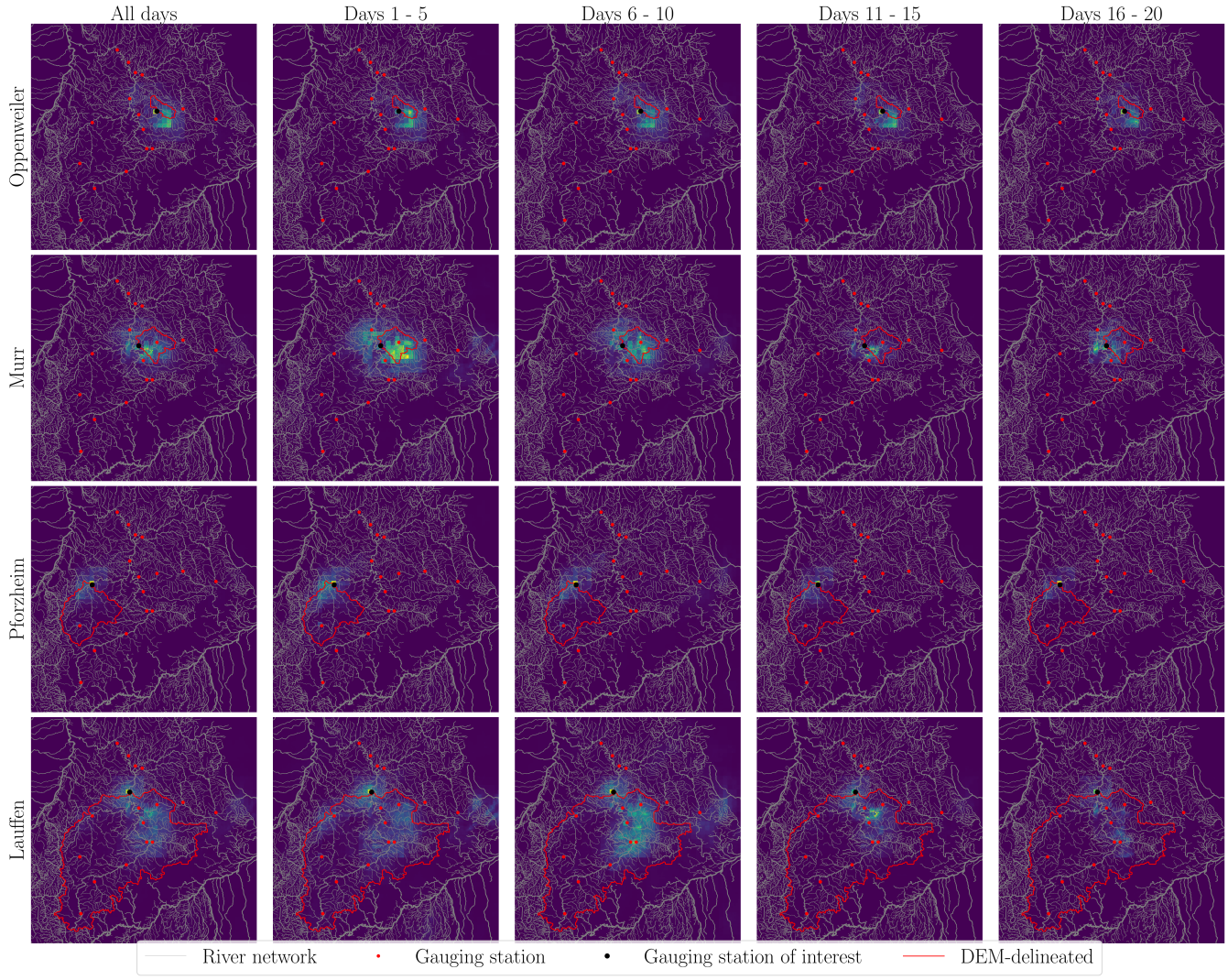

Figure A2. Attribution maps of precipitation for discharge estimation at selected stations ~~aggregated~~ and time intervals, averaged over all ~~validation~~ test set sequences with a rotated elevation map. ~~The brighter the color of~~ Brighter colors indicate grid cells where precipitation has a ~~pixel, stronger~~ influence on the ~~more important is precipitation in that grid cell for~~ estimated discharge ~~estimation~~ at the corresponding station. ~~The~~ For comparison, traditional catchment areas ~~inferred~~ delineated from elevation ~~alone~~ data are ~~shown~~ outlined in red.

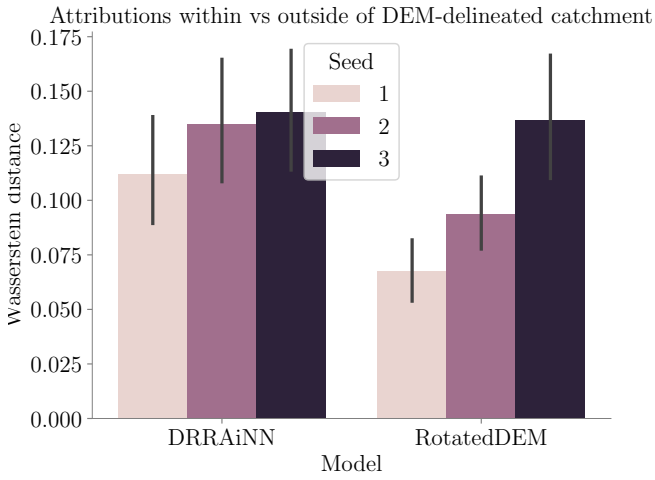

Figure A3. Wasserstein ~~distance~~ distances between normalized attributions ~~within~~ inside and outside ~~of~~ the catchment areas delineated from the digital elevation model. A higher distance ~~points towards a~~ indicates better agreement between inferred and delineated catchment areas, ~~and therefore a~~ suggesting more physically realistic ~~behavior of the~~ model behavior. ~~The depicted standard~~ Standard deviations are computed ~~over~~ across the different gauging stations.

This suggests that our original model~~makes suitable~~ 's use of the elevation map ~~that goes beyond positional encoding~~goes beyond mere positional encoding, incorporating hydrologically meaningful information.

A2 All LSTM

~~One~~ A key inductive bias in DRRAiNN is the explicit ~~distinction~~ separation between spatially extended processes and local processes. ~~The lateral propagation of water over~~ Lateral water movement across the landscape is a spatially extended process ~~that is mainly~~ primarily driven by elevation. Evapotranspiration, on the other hand, is a local process that is ~~mainly driven~~ largely influenced by solar radiation. We ~~incorporate this~~ encode this distinction into DRRAiNN by ~~mapping these processes on the DWConv and the PWConv components within~~ assigning these processes to different components of the ConvNeXt block~~.~~: the DWConv is parameterized by a CNN that receives elevation as input, while PWConv1 and PWConv2 are parameterized by an MLP that receives solar radiation~~as input~~. In this ablation, we discard this bias by feeding the elevation and solar radiation ~~together with the precipitation~~ – together with precipitation – directly into the PWLSTM. ~~Therefore~~Consequently, the relativity bias, realized by subtracting the elevation of the center cell from the elevations of all other cells within each receptive field of the hypernetwork, is ~~discarded here as well~~also removed.

We ~~find~~ observe a significant performance drop ~~in earlier lead times~~ for all metrics except MAE (Fig. A4). ~~Furthermore~~In addition, the inferred catchment areas ~~do not look physically plausible~~ appear less plausible compared to those produced by DRRAiNN (Fig. A5), ~~which is underlined~~ a finding that is supported quantitatively (Fig. A6). ~~This shows that the explicit distinction between these sub-processes is advantageous for DRRAiNN , both~~ These results demonstrate that explicitly distinguishing

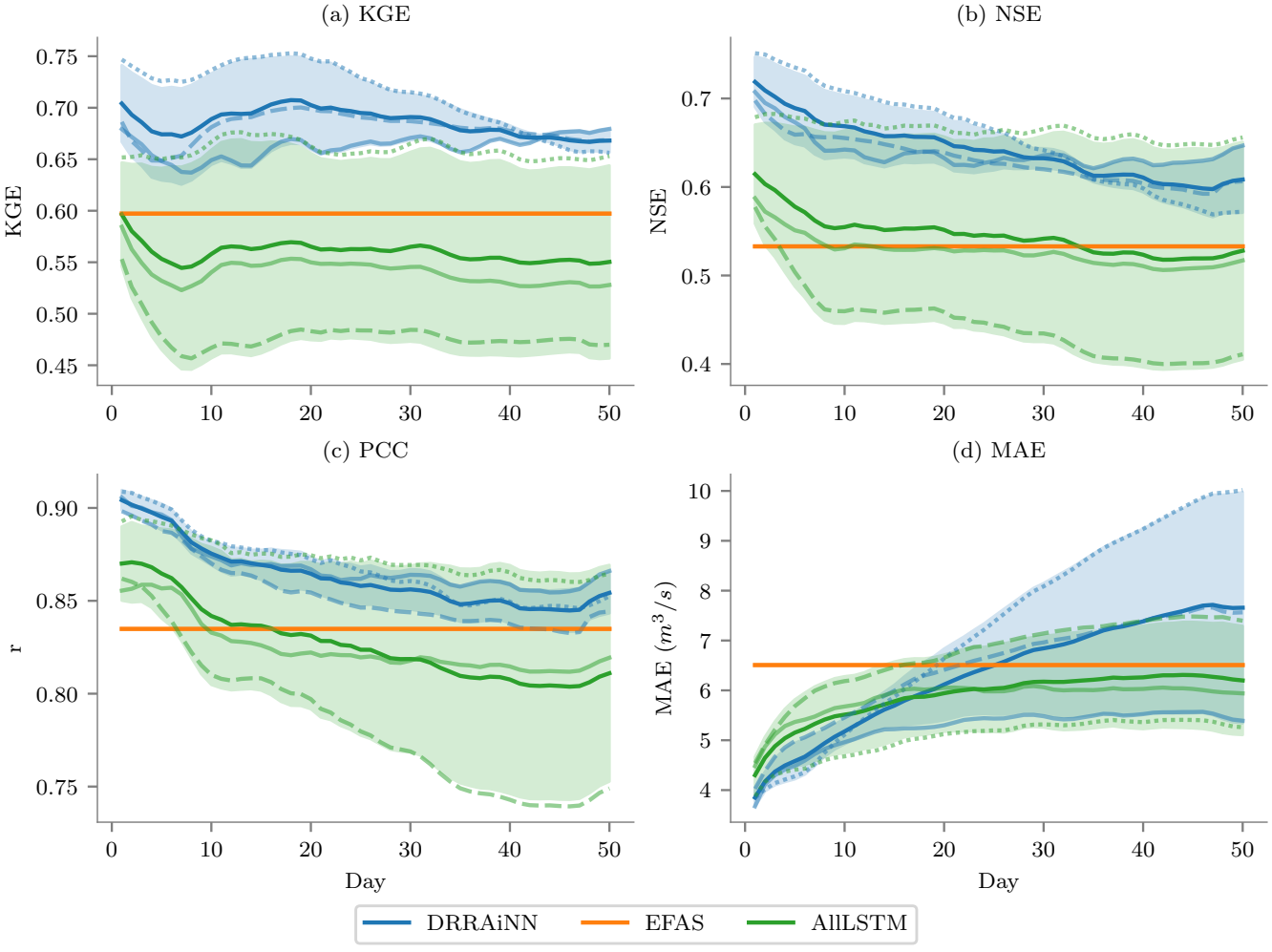

Figure A4. Performances ~~based on different metrics~~ of ~~EFAS,~~ the best three out of five DRRAiNN ~~,~~ model instances and ~~ablated~~ DRRAiNN model instances where all forcings are fed into the PWLSTM~~with~~ , compared to EFAS across different metrics and lead times up to ~~100~~ 50 days. ~~The results~~ Results are averaged ~~over the~~ across all stations~~, and the different seeds of our model are depicted with different~~ . Each line ~~styles~~style corresponds to a distinct DRRAiNN instance.

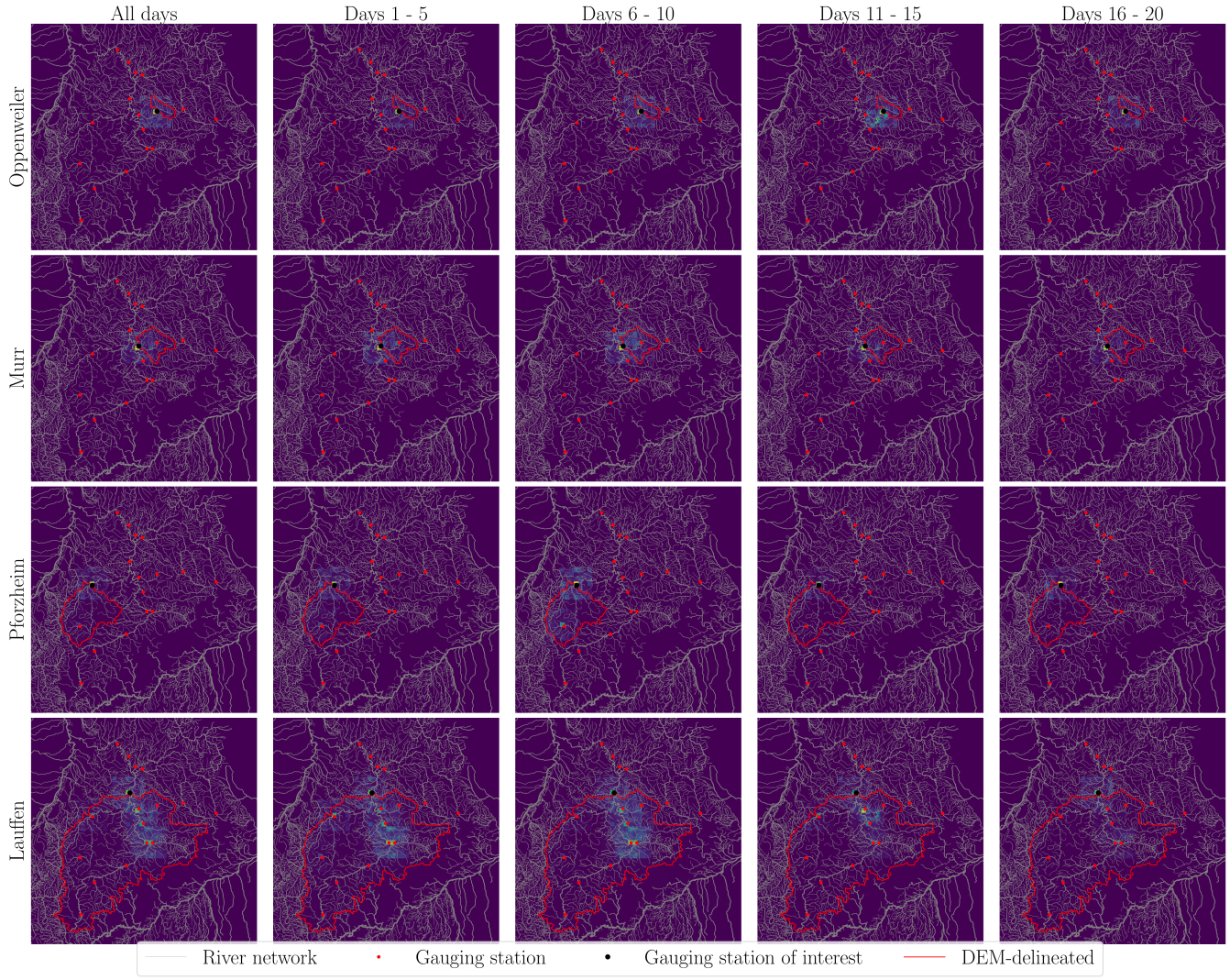

Figure A5. Attribution maps of precipitation for discharge estimation at selected stations ~~aggregated~~ and time intervals, averaged over all ~~validation~~ test set sequences when all forcings are fed into the PWLSTM. ~~The brighter the color of~~ Brighter colors indicate grid cells where precipitation has a ~~pixel,~~ stronger influence on the ~~more important is precipitation in that grid cell for~~ estimated discharge ~~estimation~~ at the corresponding station. ~~The~~ For comparison, traditional catchment areas ~~inferred~~ delineated from elevation ~~alone~~ data are ~~shown~~ outlined in red.

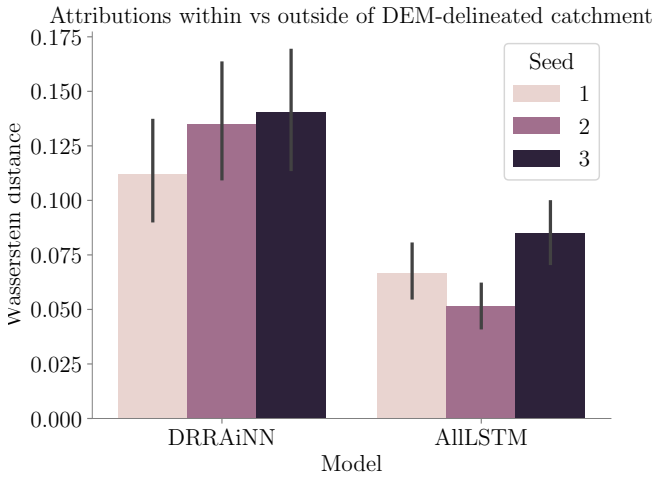

Figure A6. Wasserstein ~~distance~~ distances between normalized attributions ~~within~~ inside and outside ~~of~~ the catchment areas delineated from the digital elevation model. A higher distance ~~points towards a~~ indicates better agreement between inferred and delineated catchment areas, ~~and therefore a~~ suggesting more physically realistic ~~behavior of the~~ model behavior. ~~The depicted standard~~ Standard deviations are computed ~~over~~ across the different gauging stations.

between spatially extended and local processes benefits DRRAiNN in terms of ~~accuracy and~~ both predictive accuracy and physical plausibility.

A3 No hypernetworks

Here, we train DRRAiNN without hypernetworks to ~~examine their usefulness~~assess their contribution. To stay close to the original architecture, we ~~want to maintain the~~ preserve inductive bias that distinguishes between the spatially extended process of ~~propagating water over the landscape~~ water propagation and the local process of evapotranspiration. ~~Therefore~~Specifically, the elevation map is concatenated with the hidden state, ~~fed into~~ passed through a position-wise linear layer, and ~~only~~ then fed into the DWConv. This ~~is necessary as~~ step is necessary because DWConv requires the ~~number of~~ input and output channels to be ~~the same. Therefore~~of equal size. As a result, the relativity bias, realized by subtracting the elevation of the center cell from the elevations of all other cells within each receptive field of the hypernetwork, is ~~discarded here as well. Solar radiation, on the other hand, is concatenated~~ also removed. For solar radiation, we concatenate it with the hidden state and ~~directly fed~~ feed the result directly into PWConv1.

Removing the hypernetworks from DRRAiNN ~~leads to a significant decrease in~~ results in decreased performance for KGE ~~, especially during the first days~~ and NSE (Fig. A7a ~~). For NSE this effect is less pronounced (Fig.~~ and A7b)~~, while~~ . For PCC and MAE, we do not observe a systematic difference ~~in PCC and MAE~~ (Fig. A7c and A7d). The ablated model ~~does not produce plausible attribution maps~~ produces less plausible attributions maps compared to DRRAiNN (Fig. A8), ~~which is underlined~~ a finding that is supported quantitatively (Fig. A9).

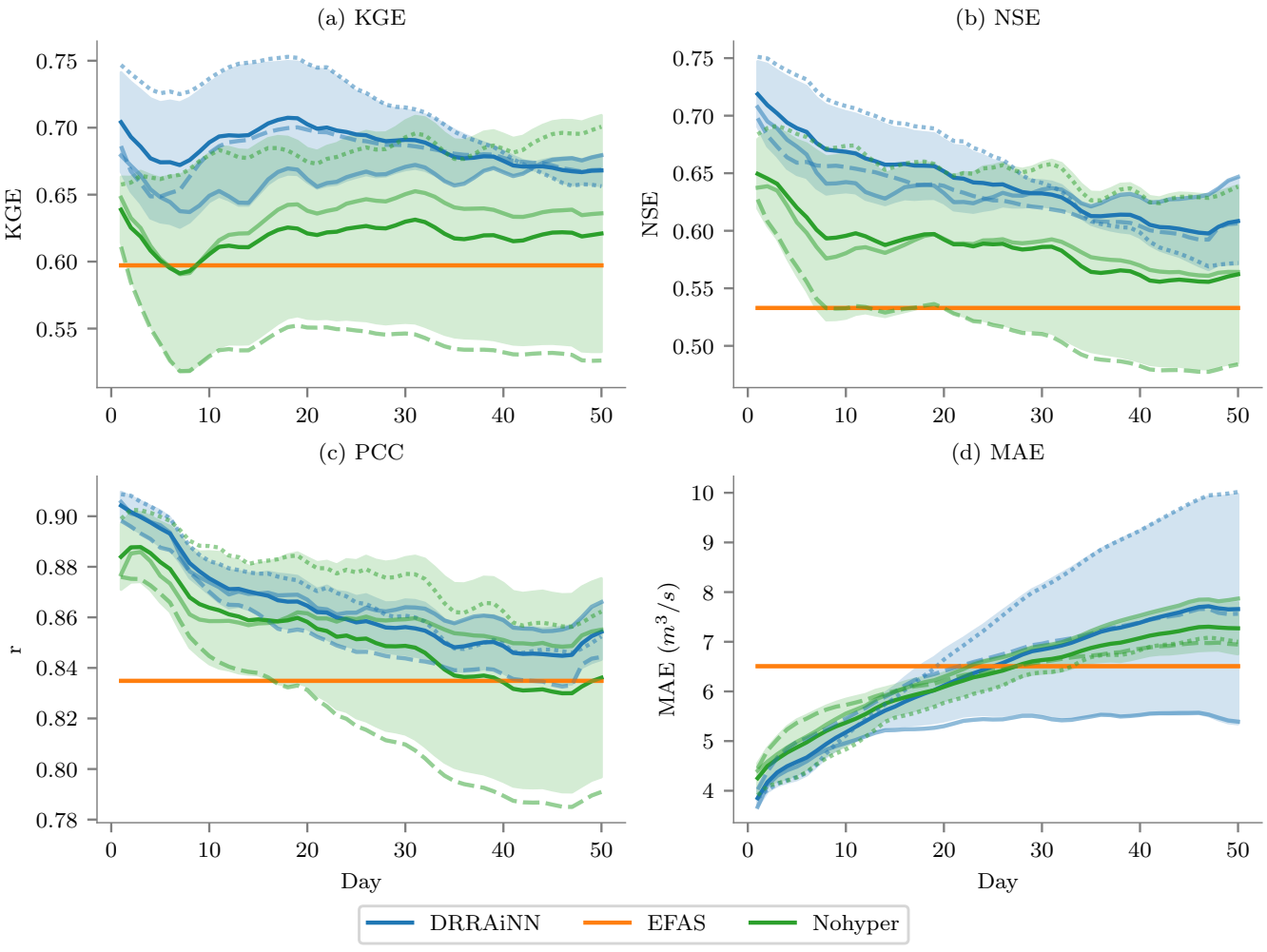

Figure A7. Performances ~~based on different metrics~~ of ~~EFAS,~~ the best three out of five original DRRAiNN ~~,~~ model instances and DRRAiNN model instances without ~~the~~ hypernetworks~~with~~ , compared to EFAS across different metrics and lead ~~time~~ times up to ~~100~~ 50 days. ~~The results~~ Results are averaged ~~over the~~ across all stations~~, and the different seeds of our model are depicted with different~~ . Each line ~~styles~~style corresponds to a distinct DRRAiNN instance.

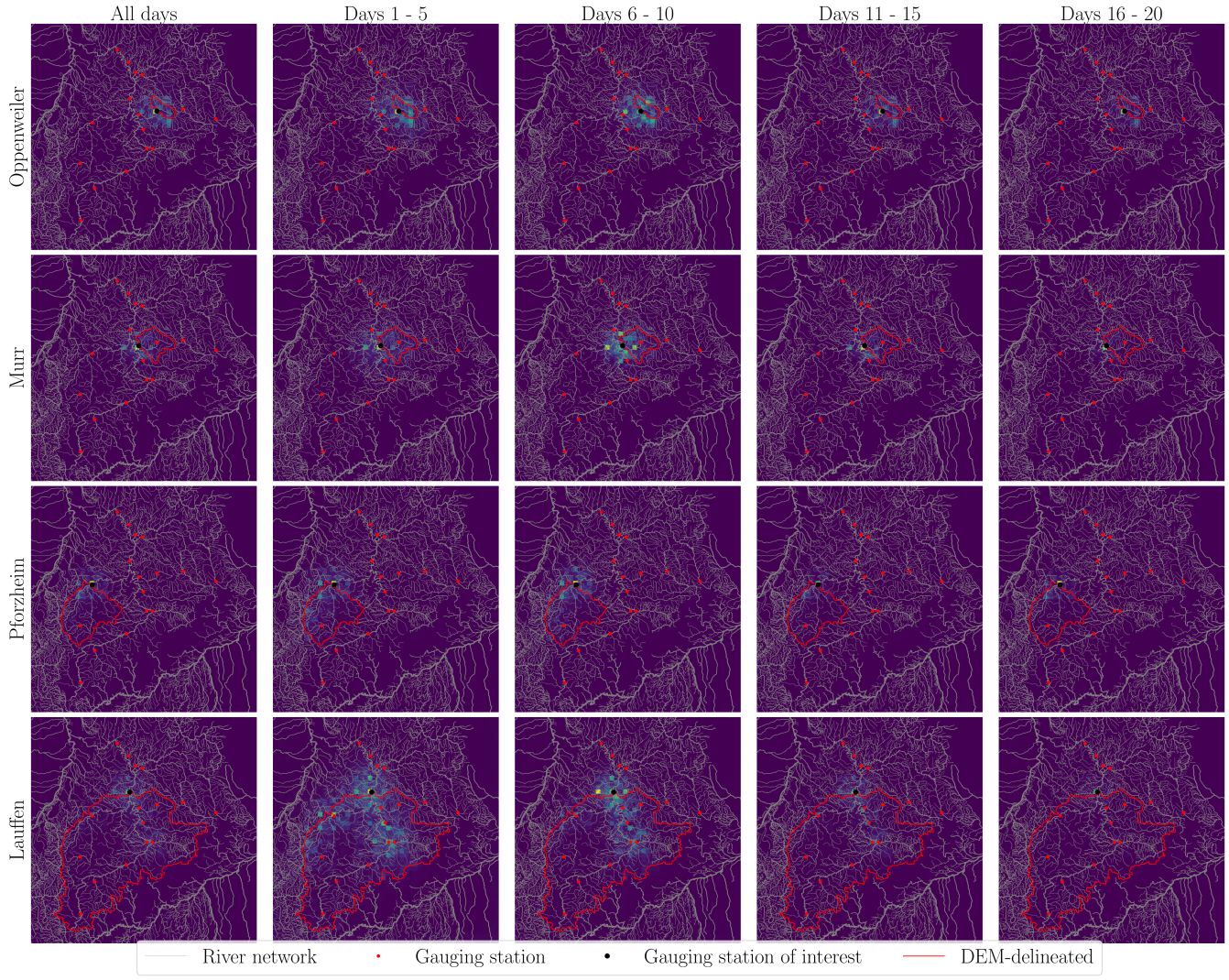

Figure A8. Attribution maps of precipitation for discharge estimation at selected stations ~~aggregated~~ and time intervals, averaged over all ~~validation~~ test set sequences without hypernetworks. ~~The brighter the color of~~ Brighter colors indicate grid cells where precipitation has a ~~pixel,~~ stronger influence on the ~~more important is precipitation in that grid cell for~~ estimated discharge ~~estimation~~ at the corresponding station. ~~The~~ For comparison, traditional catchment areas ~~inferred~~ delineated from elevation ~~alone~~ data are ~~shown~~ outlined in red.

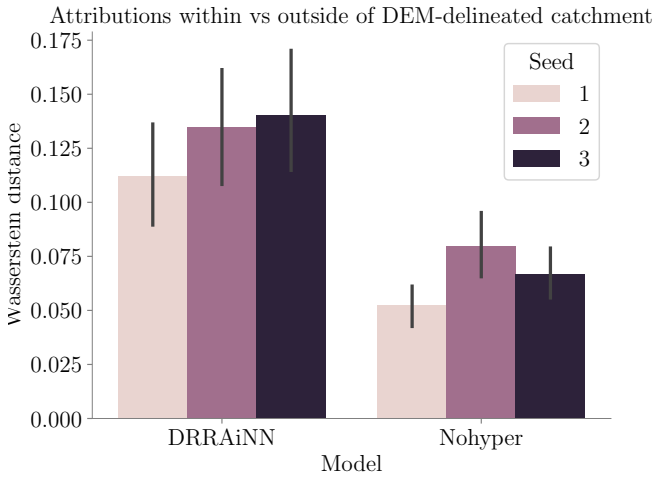

Figure A9. Wasserstein ~~distance~~ distances between normalized attributions ~~within~~ inside and outside ~~of~~ the catchment areas delineated from the digital elevation model. A higher distance ~~points towards a~~ indicates better agreement between inferred and delineated catchment areas, ~~and therefore a~~ suggesting more physically realistic ~~behavior of the~~ model behavior. ~~The depicted standard~~ Standard deviations are computed ~~over~~ across the different gauging stations.

Appendix B: Alternative hyperparameters

In this appendix, we report the performance of DRRAiNN under alternative hyperparameters settings. In the default configuration, the LSTM in the rainfall-runoff model has a hidden size of 4, and the GRU in the discharge model has a hidden size of 8. Here, we examine DRRAiNN's performance using both smaller and larger hidden sizes. Additionally, we assess the impact of replacing the GRUs in the discharge model with LSTMs.

B1 Rainfall-runoff model with hidden size 2

Figure B1 shows that reducing the hidden size of the rainfall-runoff model from 4 to 2 still yields a competitive model. On average, it performs slightly worse during the initial days. However, due to the variance in performance across different seeds, additional experiments are required to draw a more definitive conclusion.

B2 Rainfall-runoff model with hidden size 6

Figure B2 shows that increasing the hidden size of the rainfall-runoff model from 4 to 6 slightly decreases performance on the NSE and PCC metrics, while KGE remains largely unaffected. Since no significant improvement is observed, we argue that the smaller model should be preferred, following Occam's razor.

B3 Discharge model with hidden size 4

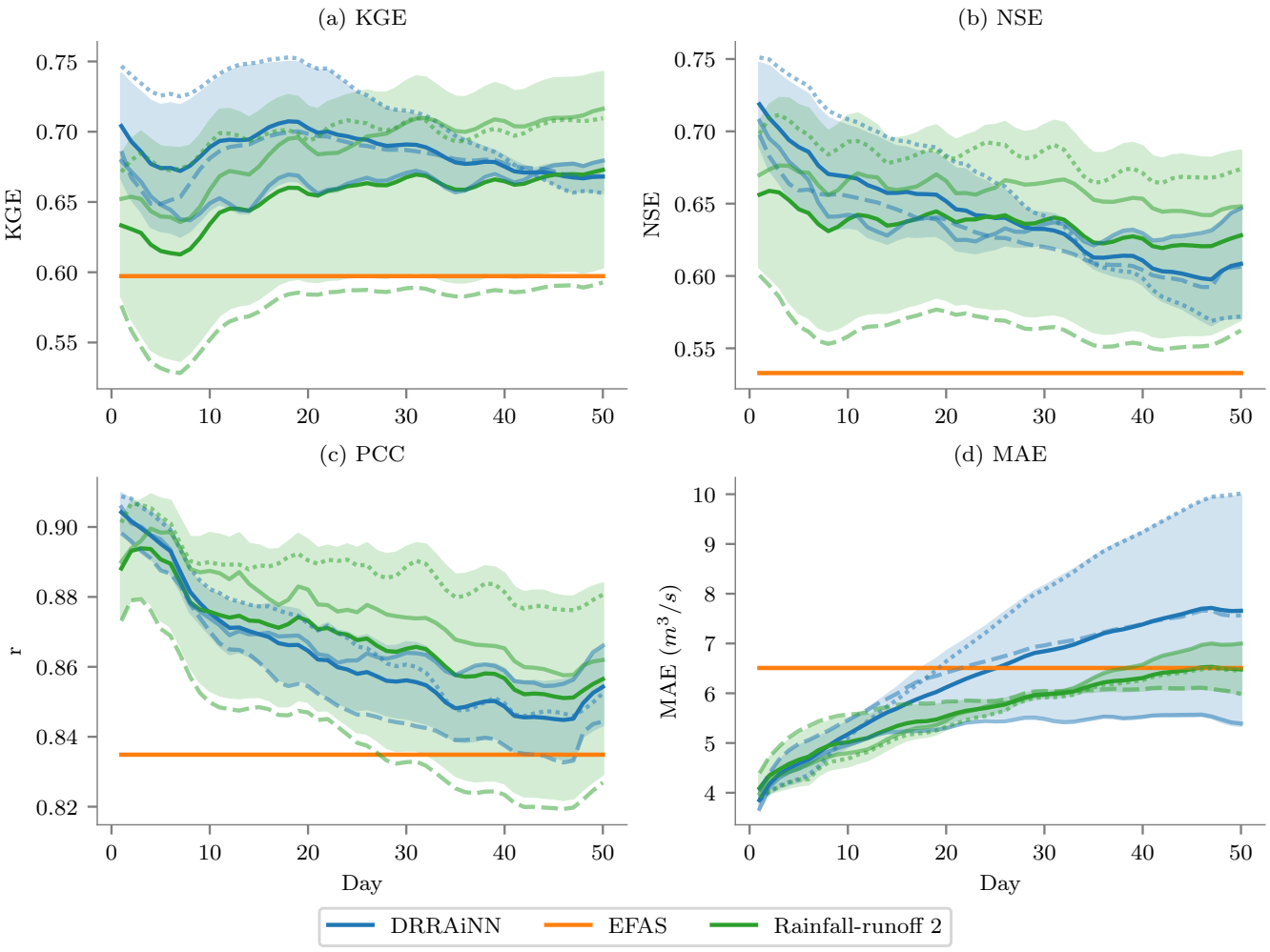

Figure B1. Performances of the best three out of five original DRRAiNN model instances and DRRAiNN model instances with a hidden size of 2 in the rainfall-runoff model, compared to EFAS across different metrics and lead times up to 50 days. Results are averaged across all stations. Each line style corresponds to a distinct DRRAiNN instance.

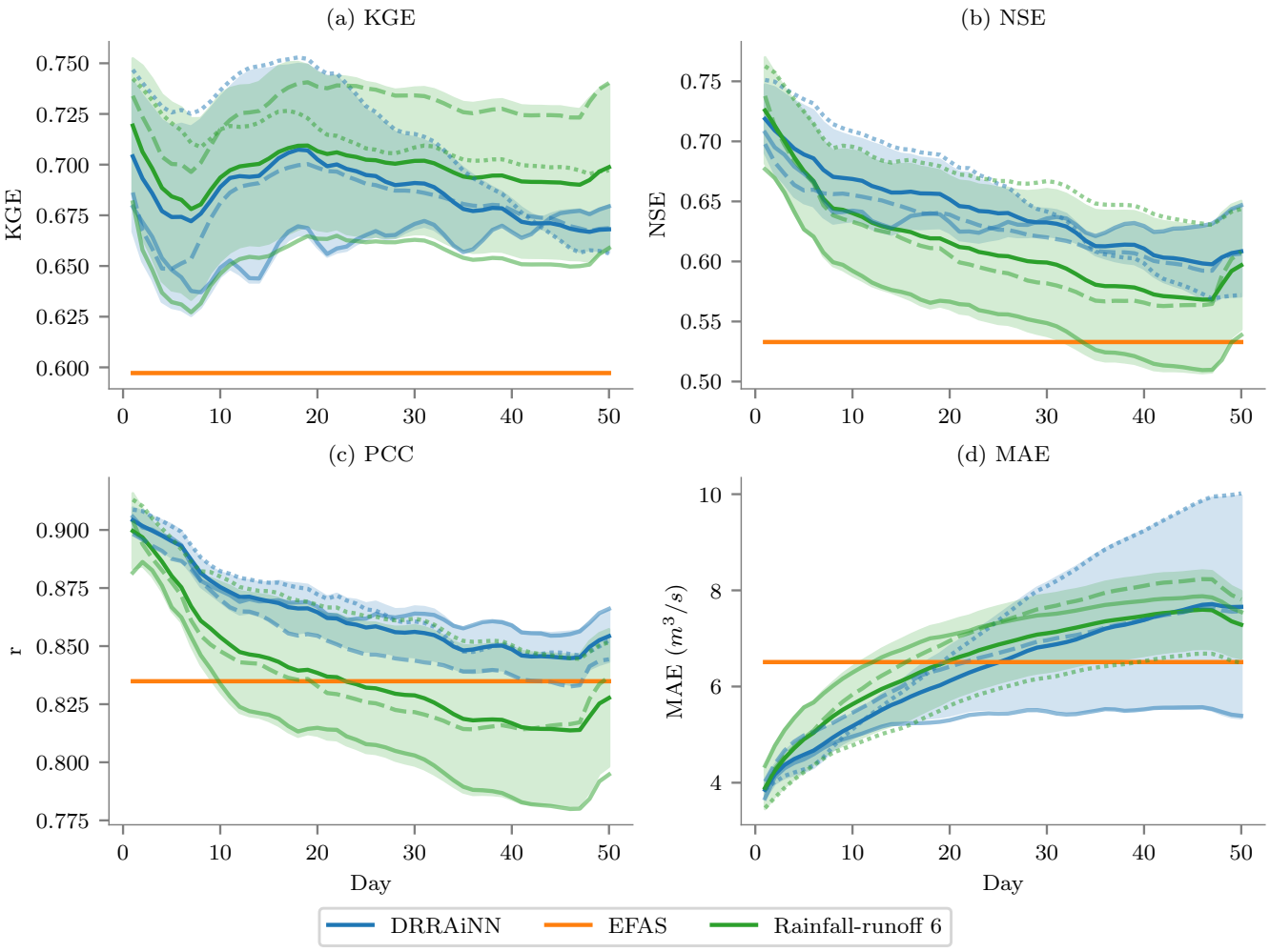

Figure B2. Performances of the best three out of five original DRRAiNN model instances and DRRAiNN model instances with a hidden size of 6 in the rainfall-runoff model, compared to EFAS across different metrics and lead times up to 50 days. Results are averaged across all stations. Each line style corresponds to a distinct DRRAiNN instance.

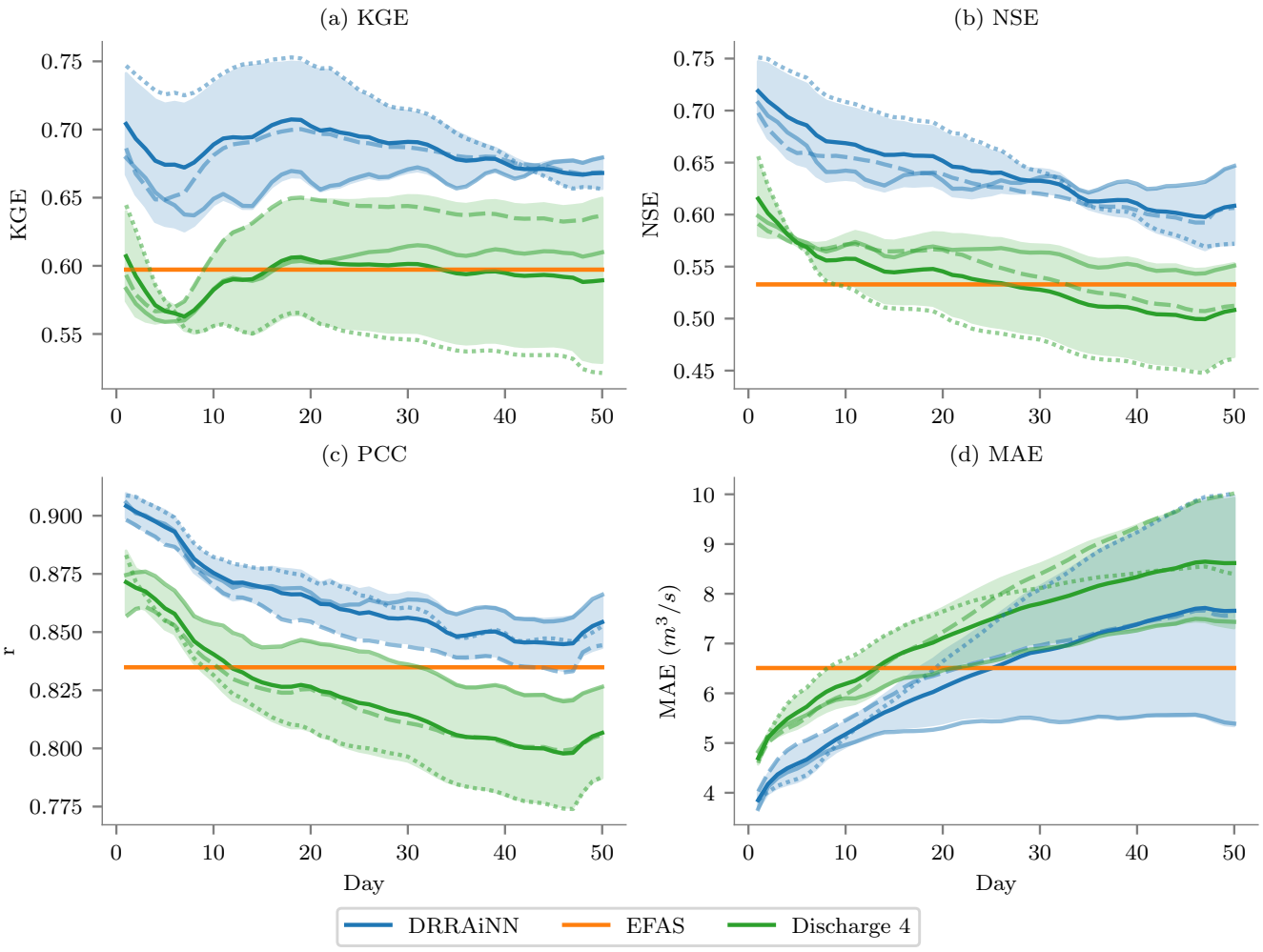

Figure B3. Performances of the best three out of five original DRRAiNN model instances and DRRAiNN model instances with a hidden size of 4 in the discharge model, compared to EFAS across different metrics and lead times up to 50 days. Results are averaged across all stations. Each line style corresponds to a distinct DRRAiNN instance.

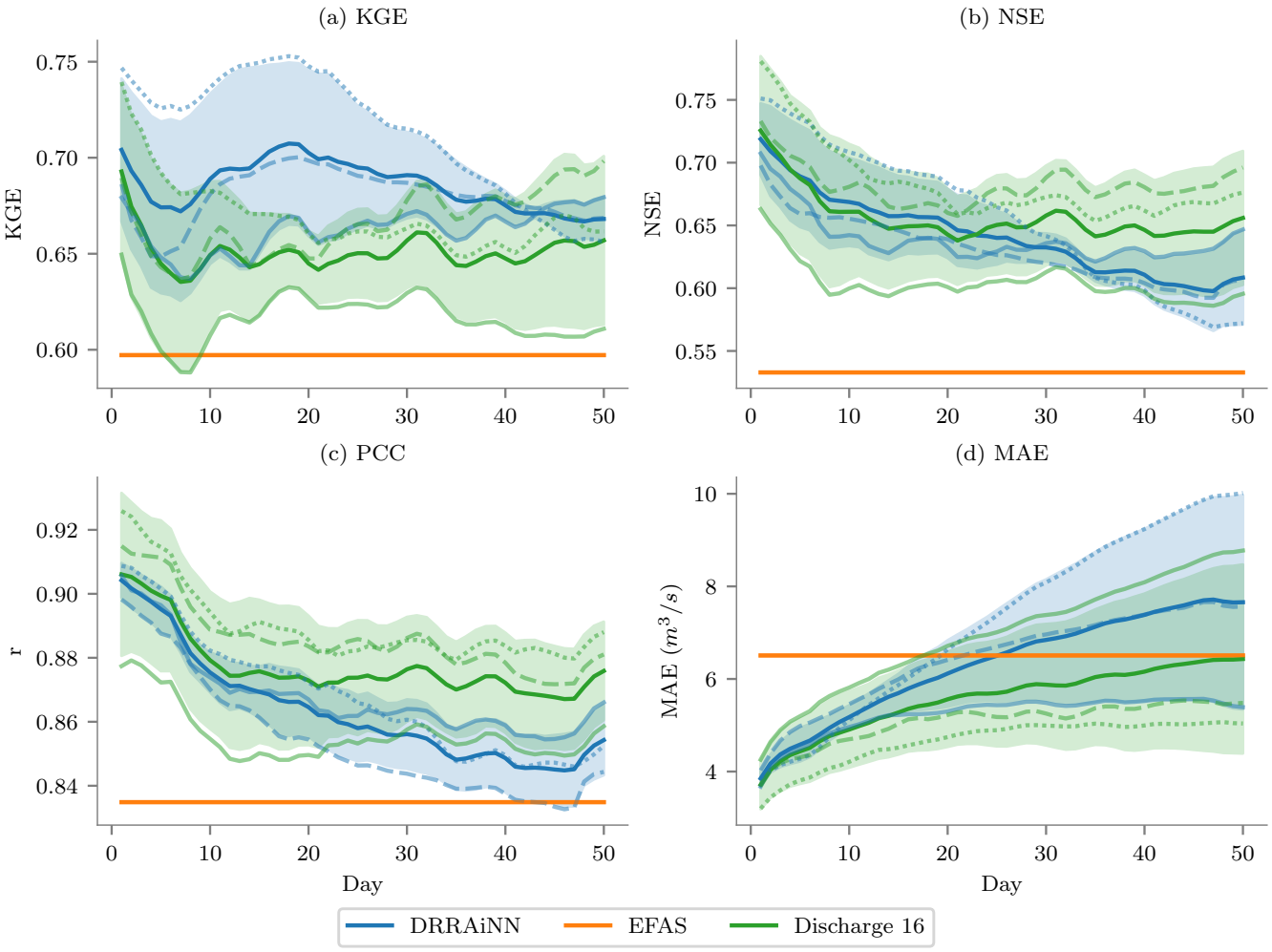

Figure B4. Performances of the best three out of five original DRRAiNN model instances and DRRAiNN model instances with a hidden size of 16 in the discharge model, compared to EFAS across different metrics and lead times up to 50 days. Results are averaged across all stations. Each line style corresponds to a distinct DRRAiNN instance.

Figure B3 shows that reducing the hidden size of the discharge model from 8 to 4 significantly reduces performance across all metrics and lead times.

B4 Discharge model with hidden size 16

Figure B4 shows that increasing the hidden size of the discharge model from 8 to 16 leads to mixed results. While KGE appears to deteriorate, NSE and PCC show slight improvements, particularly at longer lead times. Since no significant improvement can be observed, we argue that opting for the smaller model align better with Occam's razor.

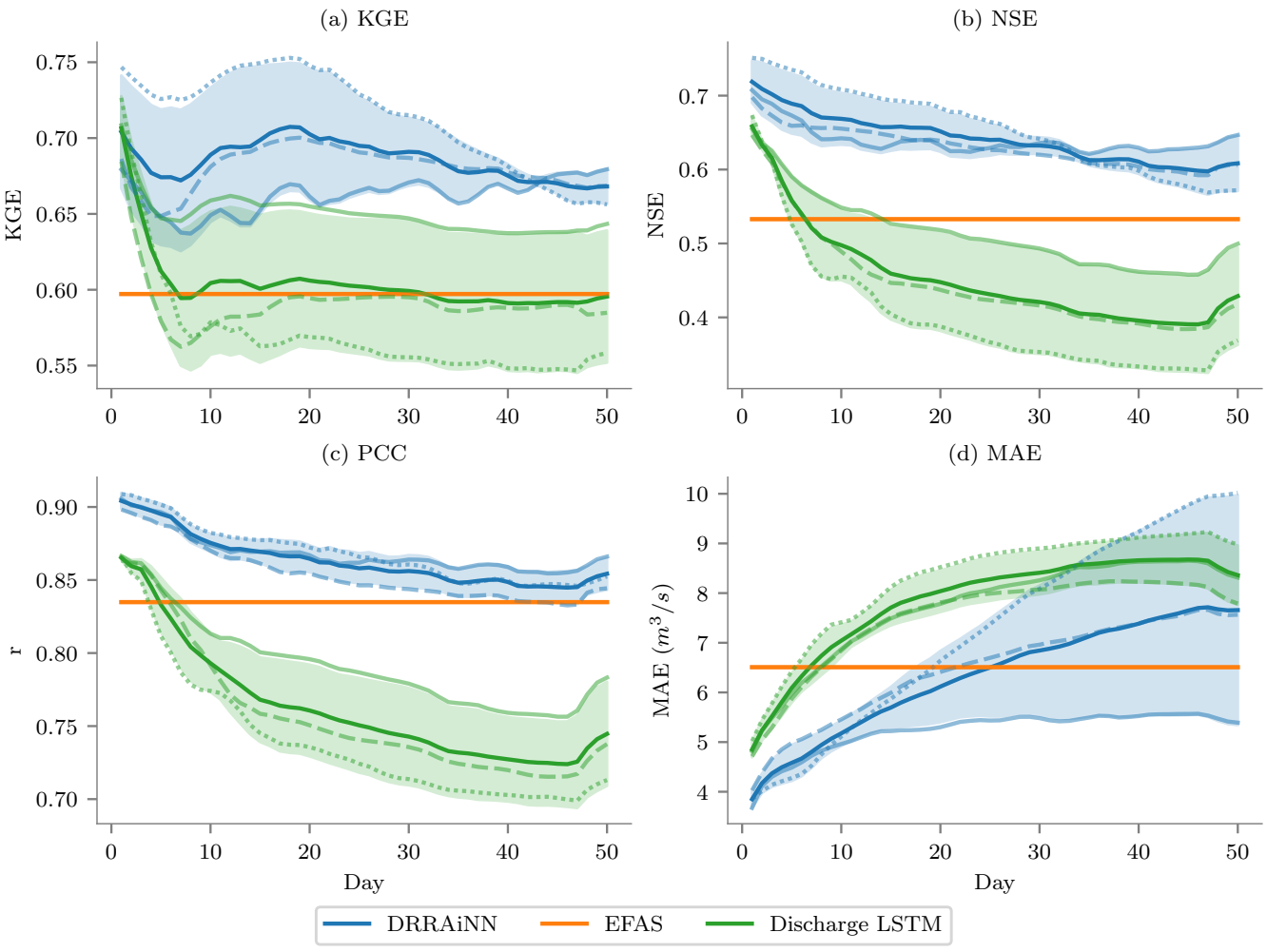

Figure B5. Performances of the best three out of five original DRRAiNN model instances and DRRAiNN model instances with LSTMs instead of GRUs in the discharge model, compared to EFAS across different metrics and lead times up to 50 days. Results are averaged across all stations. Each line style corresponds to a distinct DRRAiNN instance.

B5 Discharge model with LSTM

Figure B5 shows that replacing the GRUs in the discharge model with LSTMs significantly reduces performance across all metrics and almost all lead times. This suggests that model complexity should reflect the complexity of the underlying dynamics: river flow tends to follow simpler dynamics than surface and subsurface flow, which we model with an LSTM. Moreover, water typically resides in channels for shorter periods compared to its residence time below ground. This may explain the superior performance of GRUs in the discharge model, though further investigation is warranted.

Author contributions. All authors contributed to the conceptualization of the paper. FS, MT, and MB designed the model architecture. FS developed the code and performed the experiments. FS prepared the manuscript with contributions from all co-authors.

Competing interests. The authors declare that they have no conflict of interest.

Acknowledgements. We thank the reviewers for their very constructive criticism, feedback, and suggestions. This work received funding from the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC number 2064/1 – Project number ~~390727645.~~ 390727645 as well as from the Cyber Valley in Tübingen, CyVy-RF-2020-15. Additional support came from the Open Access Publishing Fund of the University of Tübingen. The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Fedor Scholz and Manuel Traub. ChatGPT was partially used to improve the writing style of the manuscript.

References

Al Hossain, B. M. T., Ahmed, T., Aktar, M. N., Fida, M., Khan, A., Islam, A., Yazdan, M. M. S., Noor, F., and Rahaman, A. Z.: Climate Change Impacts on Water Availability in the Meghna Basin, in: Proceedings of the 5th International Conference on Water and Flood Management (ICWFM-2015), Dhaka, Bangladesh, pp. 6–8, 2015.

AWGN, 2023: Amtliches Digitales Wasserwirtschaftliches Gewässernetz (AWGN), https://www.lubw.baden-wuerttemberg.de/wasser/awgn, 2023.

Bharati, L., Lacombe, G., Gurung, P., Jayakody, P., Hoanh, C. T., and Smakhtin, V.: The impacts of water infrastructure and climate change on the hydrology of the Upper Ganges River Basin, vol. 142, IWMI, 2011.

Bindas, T., Tsai, W.-P., Liu, J., Rahmani, F., Feng, D., Bian, Y., Lawson, K., and Shen, C.: Improving River Routing Using a Differentiable Muskingum-Cunge Model and Physics-Informed Machine Learning, Water Resources Research, 60, e2023WR035 337, https://doi.org/https://doi.org/10.1029/2023WR035337, e2023WR035337 2023WR035337, 2024.

Blöschl, G., Bierkens, M. F., Chambel, A., Cudennec, C., Destouni, G., Fiori, A., Kirchner, J. W., McDonnell, J. J., Savenije, H. H., Sivapalan, M., Stumpp, C., Toth, E., Volpi, E., Carr, G., Lupton, C., Salinas, J., Széles, B., Viglione, A., Aksoy, H., Allen, S. T., Amin, A., Andréassian, V., Arheimer, B., Aryal, S. K., Baker, V., Bardsley, E., Barendrecht, M. H., Bartosova, A., Batelaan, O., Berghuijs, W. R., Beven, K., Blume, T., Bogaard, T., de Amorim, P. B., Böttcher, M. E., Boulet, G., Breinl, K., Brilly, M., Brocca, L., Buytaert, W., Castellarin, A., Castelletti, A., Chen, X., Chen, Y., Chen, Y., Chifflard, P., Claps, P., Clark, M. P., Collins, A. L., Croke, B., Dathe, A., David, P. C., de Barros, F. P. J., de Rooij, G., Baldassarre, G. D., Driscoll, J. M., Duethmann, D., Dwivedi, R., Eris, E., Farmer, W. H., Feiccabrino, J., Ferguson, G., Ferrari, E., Ferraris, S., Fersch, B., Finger, D., Foglia, L., Fowler, K., Gartsman, B., Gascoin, S., Gaume, E., Gelfan, A., Geris, J., Gharari, S., Gleeson, T., Glendell, M., Bevacqua, A. G., González-Dugo, M. P., Grimaldi, S., Gupta, A. B., Guse, B., Han, D., Hannah, D., Harpold, A., Haun, S., Heal, K., Helfricht, K., Herrnegger, M., Hipsey, M., Hlaváčiková, H., Hohmann, C., Holko, L., Hopkinson, C., Hrachowitz, M., Illangasekare, T. H., Inam, A., Innocente, C., Istanbulluoglu, E., Jarihani, B., Kalantari, Z., Kalvans, A., Khanal, S., Khatami, S., Kiesel, J., Kirkby, M., Knoben, W., Kochanek, K., Kohnová, S., Kolechkina, A., Krause, S., Kreamer, D., Kreibich, H., Kunstmann, H., Lange, H., Liberato, M. L. R., Lindquist, E., Link, T., Liu, J., Loucks, D. P., Luce, C., Mahé, G., Makarieva, O., Malard, J., Mashtayeva, S., Maskey, S., Mas-Pla, J., Mavrova-Guirguinova, M., Mazzoleni, M., Mernild, S., Misstear, B. D., Montanari, A., Müller-Thomy, H., Nabizadeh, A., Nardi, F., Neale, C., Nesterova, N., Nurtaev, B., Odongo, V. O., Panda, S., Pande, S., Pang, Z., Papacharalampous, G., Perrin, C., Pfister, L., Pimentel, R., Polo, M. J., Post, D., Sierra, C. P., Ramos, M.-H., Renner, M., Reynolds, J. E., Ridolfi, E., Rigon, R., Riva, M., Robertson, D. E., Rosso, R., Roy, T., Sá, J. H., Salvadori, G., Sandells, M., Schaefli, B., Schumann, A., Scolobig, A., Seibert, J., Servat, E., Shafiei, M., Sharma, A., Sidibe, M., Sidle, R. C., Skaugen, T., Smith, H., Spiessl, S. M., Stein, L., Steinsland, I., Strasser, U., Su, B., Szolgay, J., Tarboton, D., Tauro, F., Thirel, G., Tian, F., Tong, R., Tussupova, K., Tyralis, H., Uijlenhoet, R., van Beek, R., van der Ent, R. J., van der Ploeg, M., Loon, A. F. V., van Meerveld, I., van Nooijen, R., van Oel, P. R., Vidal, J.-P., von Freyberg, J., Vorogushyn, S., Wachniew, P., Wade, A. J., Ward, P., Westerberg, I. K., White, C., Wood, E. F., Woods, R., Xu, Z., Yilmaz, K. K., and Zhang, Y.: Twenty-three unsolved problems in hydrology (UPH) –a community perspective, Hydrological Sciences Journal, 64, 1141–1158, https://doi.org/10.1080/02626667.2019.1620507, 2019.

Börgel, F., Karsten, S., Rummel, K., and Gräwe, U.: From weather data to river runoff: using spatiotemporal convolutional networks for discharge forecasting, Geoscientific Model Development, 18, 2005–2019, https://doi.org/10.5194/gmd-18-2005-2025, 2025.

Brutsaert, W.: Hydrology, Cambridge university press, 2023.

Butz, M. V., Bilkey, D., Humaidan, D., Knott, A., and Otte, S.: Learning, planning, and control in a monolithic neural event inference architecture, Neural Networks, 117, 135–144, https://doi.org/10.1016/j.neunet.2019.05.001, arXiv: 1809.07412, 2019.

Butz, M. V., Mittenbühler, M., Schwöbel, S., Achimova, A., Gumbsch, C., Otte, S., and Kiebel, S.: Contextualizing predictive minds, Neuroscience \& Biobehavioral Reviews, p. 105948, 2024.

Camporese, M. and Girotto, M.: Recent advances and opportunities in data assimilation for physics-based hydrological modeling, Frontiers in Water, 4, 948 832, 2022.

Chen, S., Zwart, J. A., and Jia, X.: Physics-Guided Graph Meta Learning for Predicting Water Temperature and Streamflow in Stream Networks, in: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD '22, p. 2752–2761, Association for Computing Machinery, New York, NY, USA, ISBN 9781450393850, https://doi.org/10.1145/3534678.3539115, 2022.

Cho, K., van Merrienboer, B., Bahdanau, D., and Bengio, Y.: On the Properties of Neural Machine Translation: Encoder–Decoder Approaches, in: Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, Association for Computational Linguistics, https://doi.org/10.3115/v1/w14-4012, 2014.

Darcy, H.: Les fontaines publiques de la ville de Dijon: exposition et application des principes à suivre et des formules à employer dans les questions de distribution d'eau, vol. 1, Victor dalmont, 1856.

EU-DEM, 2016: EU-DEM v1.1, Dataset, https://land.copernicus.eu/imagery-in-situ/eu-dem/eu-dem-v1.1, 2016.

Gal, Y. and Ghahramani, Z.: Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning, http://arxiv.org/abs/1506.02142v6, 2015.

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

Gigi, Y., Elidan, G., Hassidim, A., Matias, Y., Moshe, Z., Nevo, S., Shalev, G., and Wiesel, A.: Towards global remote discharge estimation: Using the few to estimate the many, arXiv preprint arXiv:1901.00786, http://arxiv.org/abs/1901.00786v1, 2019.

Gillies, S. and others: Rasterio: geospatial raster I/O for Python programmers, https://github.com/rasterio/rasterio, 2013.

Graves, A.: Practical variational inference for neural networks, in: Advances in neural information processing systems, pp. 2348–2356, 2011.

GRDC, 2024: Global Runoff Data Centre, https://grdc.bafg.de/, 2024.

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

Harder, P., Ramesh, V., Hernandez-Garcia, A., Yang, Q., Sattigeri, P., Szwarcman, D., Watson, C., and Rolnick, D.: Physics-Constrained Deep Learning for Downscaling, Tech. rep., Copernicus Meetings, 2023.

Hendrycks, D. and Gimpel, K.: Gaussian Error Linear Units (GELUs), arXiv preprint arXiv:1606.08415, http://arxiv.org/abs/1606.08415v5, 2016.

Hersbach, H., Bell, B., Berrisford, P., Biavati, G., Horányi, A., Muñoz Sabater, J., Nicolas, J., Peubey, C., Radu, R., Rozum, I., and others: ERA5 hourly data on single levels from 1940 to present, 2018.

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

Hoedt, P.-J., Kratzert, F., Klotz, D., Halmich, C., Holzleitner, M., Nearing, G., Hochreiter, S., and Klambauer, G.: MC-LSTM: Mass-Conserving LSTM, Proceedings of Machine Learning Research, http://arxiv.org/abs/2101.05186v3, 2021.

Höge, M., Scheidegger, A., Baity-Jesi, M., Albert, C., and Fenicia, F.: Improving hydrologic models for predictions and process understanding using neural ODEs, Hydrology and Earth System Sciences, 26, 5085–5102, https://doi.org/10.5194/hess-26-5085-2022, 2022.

Hrachowitz, M., Savenije, H., Blöschl, G., McDonnell, J., Sivapalan, M., Pomeroy, J., Arheimer, B., Blume, T., Clark, M., Ehret, U., Fenicia, F., Freer, J., Gelfan, A., Gupta, H., Hughes, D., Hut, R., Montanari, A., Pande, S., Tetzlaff, D., Troch, P., Uhlenbrook, S., Wagener, T., Winsemius, H., Woods, R., Zehe, E., and Cudennec, C.: A decade of Predictions in Ungauged Basins (PUB)—a review, Hydrological Sciences Journal, 58, 1198–1255, https://doi.org/10.1080/02626667.2013.803183, 2013.

Hunter, N. M., Bates, P. D., Horritt, M. S., and Wilson, M. D.: Simple spatially-distributed models for predicting flood inundation: A review, Geomorphology, 90, 208–225, 2007.

Imhoff, R., Van Verseveld, W., Van Osnabrugge, B., and Weerts, A.: Scaling point-scale (pedo) transfer functions to seamless large-domain parameter estimates for high-resolution distributed hydrologic modeling: An example for the Rhine River, Water Resources Research, 56, e2019WR026 807, 2020.

Imhoff, R. O., Brauer, C. C., van Heeringen, K.-J., Uijlenhoet, R., and Weerts, A. H.: Large-sample evaluation of radar rainfall nowcasting for flood early warning, Water Resources Research, 58, e2021WR031 591, 2022.

Karlbauer, M., Otte, S., Lensch, H., Scholten, T., Wulfmeyer, V., and Butz, M. V.: A distributed neural network architecture for robust non-linear spatio-temporal prediction, arXiv preprint arXiv:1912.11141, http://arxiv.org/abs/1912.11141v1, 2019.

Klotz, D., Kratzert, F., Gauch, M., Keefe Sampson, A., Brandstetter, J., Klambauer, G., Hochreiter, S., and Nearing, G.: Uncertainty estimation with deep learning for rainfall–runoff modeling, Hydrology and Earth System Sciences, 26, 1673–1693, 2022.

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

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

Kratzert, F., Klotz, D., Gauch, M., Klingler, C., Nearing, G., and Hochreiter, S.: Large-scale river network modeling using Graph Neural Networks, in: EGU General Assembly Conference Abstracts, pp. EGU21–13 375, 2021.

Kratzert, F., Nearing, G., Addor, N., Erickson, T., Gauch, M., Gilon, O., Gudmundsson, L., Hassidim, A., Klotz, D., Nevo, S., Shalev, G., and Matias, Y.: Caravan - A global community dataset for large-sample hydrology, https://doi.org/10.5194/egusphere-egu23-5256, 2023.

Li, P., Zhang, J., and Krebs, P.: Prediction of flow based on a CNN-LSTM combined deep learning approach, Water, 14, 993, 2022.

Liu, Y. and Gupta, H. V.: Uncertainty in hydrologic modeling: Toward an integrated data assimilation framework, Water resources research, 43, 2007.

Liu, Y., Weerts, A., Clark, M., Hendricks Franssen, H.-J., Kumar, S., Moradkhani, H., Seo, D.-J., Schwanenberg, D., Smith, P., Van Dijk, A., Van Velzen, N., He, M., Lee, H., Noh, S., Rakovec, O., and Restrepo, P.: Advancing data assimilation in operational hydrologic forecasting: progresses, challenges, and emerging opportunities, Hydrology and earth system sciences, 16, 3863–3887, 2012.

Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., and Xie, S.: A ConvNet for the 2020s, in: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, https://doi.org/10.1109/cvpr52688.2022.01167, 2022.

Longyang, Q., Choi, S., Tennant, H., Hill, D., Ashmead, N., Neilson, B. T., Newell, D. L., McNamara, J. P., and Xu, T.: Explainable Spatially Distributed Hydrologic Modeling of a Snow Dominated Mountainous Karst Watershed Using Attention, Authorea Preprints, 2024.

Marçais, J. and de Dreuzy, J.-R.: Prospective interest of deep learning for hydrological inference, Groundwater, 55, 688–692, 2017.

Mazzetti, C., Carton de Wiart, C., Gomes, G., Russo, C., Decremer D Ramos, A., Grimaldi, S., Disperati, J., Ziese, M., Schweim, C., Sanchez Garcia, R., Jacobson, T., Salamon, P., and Prudhomme, C.: River discharge and related historical data from the European Flood Awareness System, v5.0, European Commission, Joint Research Centre (JRC), https://cds.climate.copernicus.eu/cdsapp#!/dataset/efas-historical, 2023.

Montzka, C., Pauwels, V. R., Franssen, H.-J. H., Han, X., and Vereecken, H.: Multivariate and multiscale data assimilation in terrestrial systems: A review, Sensors, 12, 16 291–16 333, 2012.

Moradkhani, H., Hsu, K.-L., Gupta, H., and Sorooshian, S.: Uncertainty assessment of hydrologic model states and parameters: Sequential data assimilation using the particle filter, Water resources research, 41, 2005.

Moshe, Z., Metzger, A., Kratzert, F., Morin, E., Nevo, S., Elidan, G., and Elyaniv, R.: HydroNets: Leveraging River Network Structure and Deep Neural Networks for Hydrologic Modeling, https://doi.org/10.5194/egusphere-egu2020-4135, 2020.

Muñoz-Carpena, R., Carmona-Cabrero, A., Yu, Z., Fox, G., and Batelaan, O.: Convergence of mechanistic modeling and artificial intelligence in hydrologic science and engineering, PLOS Water, 2, e0000 059, 2023.

Nash, J. E. and Sutcliffe, J. V.: River flow forecasting through conceptual models part I—A discussion of principles, Journal of hydrology, 10, 282–290, 1970.

Neal, R. M.: Bayesian learning for neural networks, vol. 118, Springer Science \& Business Media, 2012.

Nearing, G. S., Kratzert, F., Sampson, A. K., Pelissier, C. S., Klotz, D., Frame, J. M., Prieto, C., and Gupta, H. V.: What role does hydrological science play in the age of machine learning?, Water Resources Research, 57, e2020WR028 091, 2021.

Oddo, P. C., Bolten, J. D., Kumar, S. V., and Cleary, B.: Deep Convolutional LSTM for improved flash flood prediction, Frontiers in Water, 6, 1346 104, 2024.

Otte, S., Karlbauer, M., and Butz, M. V.: Active Tuning, arXiv:2010.03958 [cs], http://arxiv.org/abs/2010.03958, arXiv: 2010.03958, 2020.

Palmer, M. A., Reidy Liermann, C. A., Nilsson, C., Flörke, M., Alcamo, J., Lake, P. S., and Bond, N.: Climate change and the world's river basins: anticipating management options, Frontiers in Ecology and the Environment, 6, 81–89, 2008.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S.: PyTorch: An Imperative Style, High-Performance Deep Learning Library, p. 12, 2019.

Pilon, P. J.: Guidelines for reducing flood losses, Tech. rep., United Nations International Strategy for Disaster Reduction (UNISDR), 2002.

Pokharel, S. and Roy, T.: A Parsimonious Setup for Streamflow Forecasting using CNN-LSTM, arXiv preprint arXiv:2404.07924, 2024a.

Pokharel, S. and Roy, T.: A parsimonious setup for streamflow forecasting using CNN-LSTM, Journal of Hydroinformatics, p. jh2024114, 2024b.

RADOLAN, 2016: RADOLAN/RADVOR, https://opendata.dwd.de/climate_environment/CDC/grids_germany/hourly/radolan/, 2016.

Rakovec, O., Weerts, A., Hazenberg, P., Torfs, P., and Uijlenhoet, R.: State updating of a distributed hydrological model with Ensemble Kalman Filtering: effects of updating frequency and observation network density on forecast accuracy, Hydrology and Earth System Sciences, 16, 3435–3449, 2012.

Renard, K. G., Laflen, J., Foster, G., and McCool, D.: The revised universal soil loss equation, Routledge, 1994.

Samaniego, L., Kumar, R., and Attinger, S.: Multiscale parameter regionalization of a grid-based hydrologic model at the mesoscale, Water Resources Research, 46, 2010.

Schalge, B., Baroni, G., Haese, B., Erdal, D., Geppert, G., Saavedra, P., Haefliger, V., Vereecken, H., Attinger, S., Kunstmann, H., Cirpka, O. A., Ament, F., Kollet, S., Neuweiler, I., Hendricks Franssen, H.-J., and Simmer, C.: Presentation and discussion of the high-resolution

atmosphere–land-surface–subsurface simulation dataset of the simulated Neckar catchment for the period 2007–2015, Earth System Science Data, 13, 4437–4464, https://doi.org/10.5194/essd-13-4437-2021, 2021.

Schmidt, L., Gusho, E., de Back, W., Vinogradova, K., Kumar, R., Rakovec, O., Attinger, S., and Bumberger, J.: Spatially-distributed Deep Learning for rainfall-runoff modelling and system understanding, in: EGU General Assembly Conference Abstracts, p. 20736, 2020.

Scholz, F., Traub, M., Zarfl, C., Scholten, T., and Butz, M. V.: Fully differentiable, fully distributed River Discharge Prediction: data sets, https://doi.org/10.5281/zenodo.13970576, 2024a.

Scholz, F., Traub, M., Zarfl, C., Scholten, T., and Butz, M. V.: Fully differentiable, fully distributed River Discharge Prediction: code, https://doi.org/10.5281/zenodo.13992584, 2024b.

Shen, C.: A transdisciplinary review of deep learning research and its relevance for water resources scientists, Water Resources Research, 54, 8558–8593, 2018.

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

Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-k., and Woo, W.-c.: Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting, Advances in neural information processing systems, http://arxiv.org/abs/1506.04214v2, 2015.

Sit, M., Demiray, B., Xiang, Z., Ewing, G., Sermet, Y., and Demir, I.: A Comprehensive Review of Deep Learning Applications in Hydrology and Water Resources, https://doi.org/10.31223/osf.io/xs36g, 2020.

Sit, M., Demiray, B., and Demir, I.: Short-term Hourly Streamflow Prediction with Graph Convolutional GRU Networks, arXiv preprint arXiv:2107.07039, http://arxiv.org/abs/2107.07039v1, 2021.

Sun, A. Y., Jiang, P., Yang, Z.-L., Xie, Y., and Chen, X.: A graph neural network (GNN) approach to basin-scale river network learning: the role of physics-based connectivity and data fusion, Hydrology and Earth System Sciences, 26, 5163–5184, https://doi.org/10.5194/hess-26-5163-2022, 2022.

Sundararajan, M., Taly, A., and Yan, Q.: Axiomatic attribution for deep networks, in: International conference on machine learning, pp. 3319–3328, PMLR, 2017.

Traub, M., Becker, F., Sauter, A., Otte, S., and Butz, M. V.: Loci-segmented: improving scene segmentation learning, in: International Conference on Artificial Neural Networks, pp. 45–61, Springer, 2024a.

Traub, M., Scholz, F., Scholten, T., Zarfl, C., and Butz, M. V.: High-Efficiency Rainfall Data Compression Using Binarized Convolutional Autoencoder, Tech. rep., Copernicus Meetings, 2024b.

Tyson, C., Longyang, Q., Neilson, B. T., Zeng, R., and Xu, T.: Effects of meteorological forcing uncertainty on high-resolution snow modeling and streamflow prediction in a mountainous karst watershed, Journal of Hydrology, 619, 129 304, 2023.

Ueda, F., Tanouchi, H., Egusa, N., and Yoshihiro, T.: A Transfer Learning Approach Based on Radar Rainfall for River Water-Level Prediction, Water, 16, 607, 2024.

Ufrecht, W.: Ein Hydrogeologisches Modell für den Karst-und Mineralwasseraquifer Muschelkalk im Großraum Stuttgart, Hydrogeologische Modelle–ein Leitfaden mit Fallbeispielen, Schriftenreihe der Deutschen Geologischen Gesellschaft, 24, 2002.

Valeriano, O. C. S., Koike, T., Yang, K., and Yang, D.: Optimal dam operation during flood season using a distributed hydrological model and a heuristic algorithm, Journal of Hydrologic Engineering, 15, 580–586, 2010.

Van Vliet, M. T., Franssen, W. H., Yearsley, J. R., Ludwig, F., Haddeland, I., Lettenmaier, D. P., and Kabat, P.: Global river discharge and water temperature under climate change, Global Environmental Change, 23, 450–464, 2013.

Wang, C., Jiang, S., Zheng, Y., Han, F., Kumar, R., Rakovec, O., and Li, S.: Distributed Hydrological Modeling With Physics-Encoded Deep Learning: A General Framework and Its Application in the Amazon, Water Resources Research, 60, e2023WR036 170, https://doi.org/https://doi.org/10.1029/2023WR036170, e2023WR036170 2023WR036170, 2024.

Wi, S. and Steinschneider, S.: On the need for physical constraints in deep learning rainfall-runoff projections under climate change, EGUsphere, 2023, 1–46, 2023.

Wright, L.: Ranger - a synergistic optimizer., \urlhttps://github.com/lessw2020/Ranger-Deep-Learning-Optimizer, 2019.

Xiang, Z. and Demir, I.: Distributed long-term hourly streamflow predictions using deep learning –A case study for State of Iowa, Environmental Modelling & Software, 131, 104 761, https://doi.org/10.1016/j.envsoft.2020.104761, 2020.

Xiang, Z. and Demir, I.: Fully distributed rainfall-runoff modeling using spatial-temporal graph neural network, 2022.

Xu, T., Longyang, Q., Tyson, C., Zeng, R., and Neilson, B. T.: Hybrid physically based and deep learning modeling of a snow dominated, mountainous, karst watershed, Water Resources Research, 58, e2021WR030 993, 2022.

Yadan, O.: Hydra - A framework for elegantly configuring complex applications, Github, https://github.com/facebookresearch/hydra, 2019.

Zhong, L., Lei, H., Li, Z., and Jiang, S.: Advancing streamflow prediction in data-scarce regions through vegetation-constrained distributed hybrid ecohydrological models, Journal of Hydrology, 645, 132 165, https://doi.org/https://doi.org/10.1016/j.jhydrol.2024.132165, 2024.

Zhu, S., Wei, J., Zhang, H., Xu, Y., and Qin, H.: Spatiotemporal deep learning rainfall-runoff forecasting combined with remote sensing precipitation products in large scale basins, Journal of Hydrology, 616, 128 727, 2023.