



Spatially resolved rainfall streamflow modeling in central Europe

Marc Aurel Vischer, Noelia Otero, and Jackie Ma

Fraunhofer Heinrich-Hertz Institute, Applied Machine Learning Group, 10587 Berlin, Germany

Correspondence: Marc Aurel Vischer (marc.aurel.vischer@hhi.fraunhofer.de) and Jackie Ma (jackie.ma@hhi.fraunhofer.de)

Received: 26 November 2024 – Discussion started: 21 February 2025

Revised: 5 June 2025 – Accepted: 5 August 2025 – Published: 16 October 2025

Abstract. Climate change increases the risk of disastrous floods and makes intelligent freshwater management an ever more important issue for society. A central prerequisite is the ability to accurately predict the water level in rivers from a range of predictors, mainly meteorological forecasts. The field of rainfall runoff modeling has seen neural network models surge in popularity over the past few years, but a lot of this early research on model design has been conducted on catchments with smaller size and a low degree of human impact to ensure optimal conditions. Here, we present a pipeline that extends the previous neural network approaches in order to better suit the requirements of larger catchments or those characterized by human activity. Unlike previous studies, we do not aggregate the inputs per catchment but train a neural network to predict local runoff spatially resolved on a regular grid. In a second stage, another neural network routes these quantities into and along entire river networks. The whole pipeline is trained end-to-end, exclusively on empirical data. We show that this architecture is able to capture spatial variation and model large catchments accurately, while increasing data efficiency. Furthermore, it offers the possibility of interpreting and influencing internal states due to its simple design. Our contribution helps to make neural networks more operation-ready in this field and opens up new possibilities to more explicitly account for human activity in the water cycle.

flooding risks (Gründemann et al., 2022). Europe is becoming increasingly more vulnerable to flooding due to large-scale atmospheric patterns that lead to widespread precipitation extremes (Bevacqua et al., 2021) and certain landscape properties. This study focuses on river floods in central Europe, where heavy precipitation and snowmelt are driving the expansion of flood-impacted areas (Fang et al., 2024). Accurate prediction of such events is the foundation for creating resilience and preventing material damages, displacement of people, and loss of human lives. The field of hydrological research concerned with predicting river levels from meteorological variables is called rainfall streamflow modeling¹. The aim is to capture the process by which precipitation feeds into rivers and other bodies of water. Predicting runoff, i.e., the amount of excess precipitation being drained away on the surface, requires modeling different processes that take place inside or right above the ground, such as evaporation and seepage. It hinges on keeping some record of the state of the surface, e.g., the amount of precipitation in the last days or how much water is stored as snow during the winter season. These processes are highly localized, and as a next step, the resulting local runoff needs to be converted into streamflow along a network of rivers. This modeling step is called routing. There is a large body of research that employs (conceptually simplified) physical models for both these tasks (Beven, 2012). Furthermore, models based on neural networks have increasingly been proposed in recent years, e.g., Kratzert et al. (2018), Nearing et al. (2024). We build upon this line of work by introducing a neural network that performs both local rainfall runoff modeling *on a regular grid* and routing along the river network to predict

1 Introduction

As one of the most frequent and destructive natural disasters, floods are expected to become more common due to climate change (Bevacqua et al., 2021) and more hazardous as the worldwide population in high-risk areas is likely to increase (Kam et al., 2021). Heavy precipitation is expected to become more frequent, which will increase

¹ Another commonly used term is rainfall runoff modeling. As this paper aims to predict streamflow in rivers, we decided to use the more specific term rainfall streamflow modeling, but we will use rainfall runoff modeling to refer to the general literature.

streamflow time series measured at river gauges. The model is trained to do this in a data-driven, end-to-end fashion. This spatially distributed modeling framework is able to capture spatial covariability of the input features, thereby enhancing prediction accuracy in larger basins (Yu et al., 2024). It also allows for controllability and scientific discovery, and it is ready to scale to higher spatial and temporal resolution.

The following Sect. 2 discusses in detail which types of neural networks have been considered for rainfall runoff prediction and routing and explains our contribution to this ongoing field of research. In Sect. 3, we introduce a novel, publicly available dataset for spatially resolved rainfall streamflow modeling in five river basins in Germany and neighboring countries, and we describe our model architecture in Sect. 4. Section 5 presents the main results from the experiments. Section 6 concludes with a brief outlook on future directions, highlighting the influence of human activity.

2 Related work

We start this section by introducing a classification scheme for rainfall streamflow models. This scheme will provide orientation as we subsequently present previous work on neural networks in rainfall runoff modeling in general and spatially resolved processing and routing in particular. We carve out how our approach is different and end this section with an overview of this paper's contributions.

2.1 Typology of rainfall runoff models

We adapt a classification scheme for rainfall streamflow models originally introduced by Sitterson et al. (2018): Depending on the level of abstraction, models are said to be *empirical* (also called *detailed* or *physical*) if they involve physical equations of the involved processes (Horton et al., 2022). *Conceptual* (or physically inspired) models make some substantial simplifications but still contain (abstract) subsystems or quantities that can be identified with physical entities. Finally, *statistical* models refrain from explicit modeling of anything physical and instead focus exclusively on the statistical relationship between inputs and outputs. Our approach is based on neural networks and falls into the latter category, while most operational models, such as LISFLOOD (Van Der Knijff et al., 2010), fall under the conceptual category.

Another criterion for classifying models is the way space is represented in the model: *Lumped* models aggregate all variables (temporal or static) across a station's catchment area before modeling. Not representing the spatial extension of a catchment can be a reasonable modeling assumption for small catchments, but it implies losing the opportunity to model spatial covariance within the catchment. An example of the importance of spatial covariance is the different effect that heavy rainfall might have over a forested area versus on sealed soil. *Distributed* models, on the other hand, explic-

itly model local processes, usually on a grid and less commonly on a vector-based irregular mesh, e.g., Hitokoto and Sakuraba (2020); Sun et al. (2022). Most physical or conceptual models fall under the latter category, as the underlying formulas are local and it is straightforward to resolve them on a regular grid for computation. Neural networks in this domain, on the other hand, started to be developed as lumped models for a combination of historical and technical reasons, which we discuss in the next Sect. 2.2.1. In between sits a class of models called *semi-distributed*, where some sort of substructure is modeled. Many routing models fall under this category. An example of a neural network-based routing model is Nearing et al. (2024), where a network of gauging stations is modeled with high temporal resolution, but not the processes inside each station's catchment area. As we detail below, our model first predicts runoff fully distributed in space and then maps these runoffs onto the river network in a second stage.

2.2 Neural networks in rainfall runoff modeling

Neural networks have been used for rainfall runoff modeling since the 1990s (Smith and Eli, 1995), but they have surged in popularity since Kratzert et al. (2018), when long short-term memory (LSTM) layers (Hochreiter and Schmidhuber, 1997) were employed for the first time. Kratzert et al. (2019c) then described the beneficial effects of adding static location information to the meteorological inputs, albeit in an aggregated manner. This type of model has since been demonstrated to predict streamflow more accurately than models not based on neural networks, across a variety of locations and experimental setups (Lees et al., 2021; Mai et al., 2022; Clark et al., 2024). It also transfers more readily to ungauged basins (Kratzert et al., 2019b). Calibrating physical or conceptual rainfall runoff models usually requires hand-crafting ancillary input features to support the meteorological forcing variables, such as catchments' climate type or hydrological signature (see Beven, 2012, for an overview). Sometimes, the dataset is first partitioned into hydrologically homogeneous subsets, on which separate parameters are then calibrated (Beven, 2012). Neural networks do not require such human labor and in contrast profit from processing all catchments indiscriminately and with a single model (Kratzert et al., 2024). As we demonstrate in this study, neural networks can be stacked flexibly into a task-specific pipeline and trained end-to-end, without any manual calibration or intermediate steps. They are capable of extracting task-relevant information from a large array of potentially informative, raw static features (Kratzert et al., 2019c). These data sources can include categorical information such as land cover or soil classes, which cannot be readily integrated into physical formulas. Neural networks can also leverage entirely new types of input data, such as large-scale remote sensing data (Zhu et al., 2023), concentrations of isotopes (Smith et al., 2023) and chemical compounds (Sterle et al., 2024). In addition,

research on the explainability of neural networks has been conducted by Kratzert et al. (2019a) and Lees et al. (2022). These studies focused on identifying hydrological quantities and concepts within neural networks. Cheng et al. (2023) used an explainability framework to extract hydrological signatures from networks in a data-driven fashion. Furthermore, Jiang et al. (2022) showed that the use of explainability methods can provide a better understanding of the dominant flooding mechanisms across different catchments. Explainability is crucial for reliable operations in real-life applications because it enables the control of risk. It also enables scientific discovery (Shen et al., 2018). In summary, LSTM-based models have been firmly established as state of the art in rainfall runoff modeling with a combination of consistently superior performance and addressing the most pressing concerns regarding reliability, even though many questions remain to be answered. Due to their flexibility, they are primary candidates for entirely novel approaches that will become more relevant as climate change gives rise to questions of human influence and multi-factor disasters.

2.2.1 Spatially resolved processing

Smith and Eli (1995), the first study on neural networks in rainfall runoff modeling, train a simple, non-recurrent neural network on a five-by-five grid of synthetic data as a proof of concept. Another early example of semi-distributed processing is Hu et al. (2007). The authors evaluate the effect of lumping, but use only five rain gauging stations as input instead of a full grid, and a single catchment as a target. Xiang and Demir (2022), unfortunately not peer-reviewed, presents an architecture closely resembling the first stage of our model, which they call GNRRM-TS: Inputs are processed separately on a regular grid before being aggregated using a manually computed flow direction map. Also here the scope is limited to a single station and the only inputs are precipitation and drainage area of each grid cell. Xie et al. (2022) use LSTMs in a gridded fashion to estimate monthly baseflow instead of daily runoff. They also include static information as inputs, but they train their model on hand-selected subgroups of catchments. Muhebwa et al. (2024) propose a nuanced semi-distributed strategy, which instead of aggregating entire catchments, aggregates regions within a catchment that are similarly far upstream. The resulting set of input features for each region group are concatenated and jointly processed by a LSTM model. Hitokoto and Sakuraba (2020) is an interesting example of using an irregular vector-mesh rather than a regular grid. For each node, a conceptual model provides estimates of local runoff that are then aggregated by iteratively simplifying a mesh using a technique that is inspired by particle filters. Once coarsened to 96 nodes, they use a relatively simple four layer fully connected neural network for routing. This approach can be classified as semi-distributed. When considering only the portion of the pipeline managed by the neural network, i.e., after the

conceptual model's outputs have been coarsened. Sun et al. (2022) also act on an irregular mesh, this time training a graph neural network on the outputs of a conceptual model. After this pre-training, they fine-tune the neural network on streamflow observations – an elegant way to deal with sparse empirical data in this context. However, their study, too, is limited to a single smaller basin in the western United States. Because their model consists of complexly interleaved graph and time convolution layers, they rely on graph coarsening to be able to scale up this approach to another, larger basin. Yu et al. (2024) propose a combination of a LSTM model on the catchment level and a conceptual model for routing.

2.2.2 Routing

Routing refers to modeling the flow of water between gauging stations in a river network. Neural networks have been successfully employed for this task as well. In this context, streamflow at a given station is predicted from the streamflows of upstream stations alone, typically at an hourly resolution. At such high temporal resolution, routing within the river system can ignore slower processes like runoff generation or baseflow and instead focus entirely on the movement of runoff along the river network. Since the stations within a river system can be conceptualized as nodes in a directed acyclic graph, it seems natural to model this data with a graph neural network, although this term is fairly broad (Bronstein et al., 2021). Examples of this approach include Moshe et al. (2020), Sit et al. (2021), Sun et al. (2021), Sun et al. (2022), Nevo et al. (2022), and Nearing et al. (2024), all of which demonstrated excellent performance in this setting. In comparison, the design of the routing stage in our model, as detailed below, is much more parsimonious in order to give the user more fine-grained control and interpretability.

Another line of research investigates models that act on networks of rain gauges instead of a regular grid of inputs. Such models can be considered semi-distributed as well. The general focus here seems to be on finding suitable architectures for this task, combining self-attention, LSTM, convolution, and more complex graph convolution layers. For example, Chen et al. (2023) intricately stack several LSTM layers to represent the river network structure, while Zhou et al. (2023) propose a mixture of self-attention, recurrence, and convolution to build a graph neural network for this routing task. Zhu et al. (2023) aggregate remote-sensing rain data within each catchment in a data-driven fashion by training separate convolutional neural networks for every input product. They then concatenate each sub-basin's lumped information with rain gauge data for further routing in a semi-distributed scheme. This convolutional approach amounts to a more sophisticated form of lumping, as it is concerned with unimodal spatial integration of data that will be integrated and temporally modeled only at a later stage. Hu et al. (2024) partially work on gridded data, namely remote-sensing measurements of rainfall. Each sub-basin is aggregated sepa-

rately using a convolutional LSTM to produce a spatially aggregated time series of rainfall within the sub-basin. Aggregated time series are concatenated with static information and recent runoff and processed further in a semi-distributed fashion. The crucial difference here is that the convolutional LSTM serves as a data-driven aggregation mechanism for the gridded rainfall input data, but hydrological modeling again takes place in the semi-distributed domain.

2.2.3 CAMELS-type datasets

We have just discussed several studies featuring individual or a few select catchments. Yet the bulk of large-scale rainfall runoff modeling with neural networks has extensively featured the CAMELS dataset or one of its derivatives, based on Newman et al. (2015) and extended to its current form by Addor et al. (2017). It contains meteorological time series and ancillary data for 671 catchments located within the contiguous United States, manually selected for minimal human impact. This implies that the catchments are relatively small but, on the other hand, ensure “laboratory conditions” for hydrological modeling. The downside of this is the limited applicability of findings generated with this dataset to areas of the world where human influence contributes significantly to streamflow, such as central Europe. But as the dataset covers the contiguous US homogeneously, spans a large area, contains many catchments with a wide variety of different climates, it offers optimal conditions for training neural networks. And so CAMELS rose to popularity together with the neural network approach in rainfall runoff modeling. Since then, similar datasets were introduced to the public that cover other parts of the world: Chile (Alvarez-Garretón et al., 2018), Great Britain (Coxon et al., 2020), Brazil (Chagas et al., 2020), Australia (Fowler et al., 2021), the Upper Danube Basin (Klingler et al., 2021), France (De-laigue et al., 2022), Switzerland (Höge et al., 2023), Denmark (Liu et al., 2025), and Germany (Loritz et al., 2024).

2.3 Contributions

In this paper, we present three contributions that go beyond what we have discussed so far: we process the data in a spatially resolved manner without prior aggregation, we use a simple routing module that allows for interpretability, and we train this model in an end-to-end fashion on a novel, spatially resolved dataset in central Europe.

Spatially resolved processing takes place in the first or *local* stage of our model, detailed in Sect. 4.1. Its architecture largely follows the one presented in Kratzert et al. (2019b). Instead of using lumped basins as inputs, we apply the same neural network in parallel to each cell of a regular grid of meteorological time series and ancillary inputs. We show that this finer spatial resolution allows us to capture covariances and provide regularization, especially benefiting larger catchments. This approach is natural for physical or conceptual

models that solve local equations at a given rasterization. Yet no one, to our knowledge, has applied a neural network directly to the grid of inputs in a way that scales up to entire river basins. The local stage yields a local runoff quantity for each grid cell, exemplarily visualized in the right panel of Fig. 6.

The second or *routing* stage, detailed in Sect. 4.2, consists of only two simple network layers without any nonlinearity, efficiently mapping these local runoff quantities onto a river network. Both stages are trained jointly in an end-to-end fashion on the entire dataset, rendering any kind of expert knowledge obsolete. This also means that the model is fitted exclusively on empirical data, enabling scientific discovery from raw data. We show that the river network connectivity graph can be used as an inductive bias to constrain the model to reproduce the river’s natural layout. This increases data efficiency and allows for better interpretability. We explain how, in principle – although this has yet to be shown in practice – the model can be controlled interactively: extracting or injecting quantities of water can simulate human influence, such as industrial, agricultural, or hydroelectric energy generation activity, which significantly contributes to streamflow but is independent of the modeled hydrological processes.

Lastly, the lumped datasets discussed above are unsuitable for this spatially resolved modeling approach, since it requires both non-spatially aggregated inputs and streamflow data for entire basins. Hence, for this study, we compiled gridded meteorological and static data as well as river streamflow records for five entire basins in central Europe. These basins are characterized by an overall high level of human activity compared to the CAMELS dataset. The data are publicly available and described in more detail in Vischer et al. (2025b).

3 Data

As discussed above, previously released datasets for rainfall streamflow modeling are unsuitable for spatially resolved processing, so we compiled a new, publicly available dataset, referenced in the data availability statement. We present the data sources, preprocessing steps, and practical aspects in more detail in a separate publication (Vischer et al., 2025b). The river discharge data that we use as targets for training, as well as the catchment metadata from which we derive the river connectivity, are available from the original provider (<https://portal.grdc.bafg.de/applications/public.html>, last access: 5 June 2025). We provide code that processes and combines them with the input data after manual download.

3.1 Study area, study period, and resolution

Our study covers five river basins in Germany and parts of neighboring countries: Elbe, Oder, Weser, Rhine and the upstream part of the Danube River up to Bratislava (see Fig. 1,

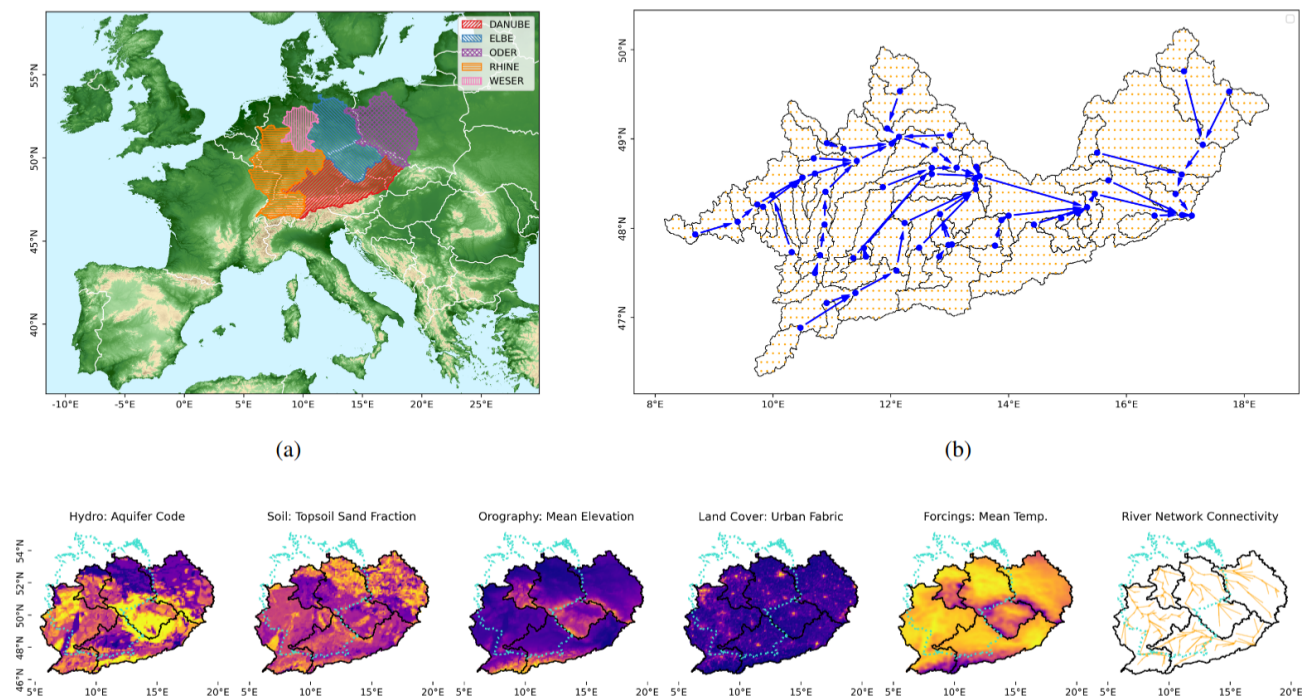


Figure 1. Overview of the study area, input grid, and data types. **(a)** Study area: the study area comprises five basins that cover a contiguous area in central Europe. **(b)** Input grid and station network in the Upper Danube Basin: cells of the input grid (orange) for the Upper Danube basin. Catchment boundaries (black) are overlaid with corresponding stations (blue), and connecting arrows represent the station connectivity network. Cells along catchment boundaries are assigned entirely to the catchment that contains their center point. **(c)** Input types: visualizations for an example feature of each type of input. Basin outlines (black) and borders of Germany (turquoise) are plotted for reference. Orography in panel **(a)** was adapted from the European Space Agency's Copernicus Global 90 m DEM (GLO-90, <https://doi.org/10.5270/ESA-c5d3d65>, European Space Agency, 2021) © EuroGeographics for the administrative boundaries in panels **(a)** and **(c)**. Watershed boundaries in panels **(a)**, **(b)**, and **(c)** were taken from the Global Runoff Data Center (<https://portal.grdc.bafg.de/applications/public.html>, last access: 5 June 2025).

right panel). Due to the sparser coverage of gauging stations in the lower reaches of the Danube basin, we decided to focus on the upper reaches where the station network is more homogeneous. Additionally, the placement of river gauging stations varies across countries, as each follows distinct policies for station location. From a machine learning perspective, this results in diverse sampling strategies across the river network. We decided to limit our study to this region so as not to confound the performance of routing with such different sample distributions. The total study area covers a contiguous 570 581 km² area. Figure 1 also visualizes exemplary features in the study area with boundaries of the river basins and Germany for geographic reference. Based on the consistent availability of streamflow data, we decided to conduct our experiments on the water years 1981–2011. A water year lasts from 1 October of the previous year to 30 September. Due to data availability, we homogenized all input data to daily temporal resolution and a regular grid that is compatible with the ERA5-Land dataset (Muñoz Sabater, 2019) and covers the Earth's surface with a spatial resolution of $0.1^\circ \times 0.1^\circ$ or roughly 9 km $\times$ 9 km. If a grid cell is located along a catchment boundary, we assign it entirely to the catchment that

contains the cell's center point. This avoids having to represent fractional cells in the pipeline and seemed an acceptable trade-off for the sake of simplicity, considering that the area covered by each grid cell is relatively small.

3.2 Dynamic input data

Runoff is primarily driven by precipitation, but temperature and solar radiation need to be taken into account as well to properly capture processes like evaporation or snow dynamics. Thus, our meteorological input variables or *forcings* are daily minimum, average, and maximum temperature; daily sum and standard deviation of precipitation; and average potential evaporation – a score computed from radiation, temperature, air pressure, and humidity. This set of variables has been widely used in previous studies (Kratzert et al., 2018, 2019b) and was retrieved from ERA5-Land (Muñoz Sabater, 2019). The variables, downloaded at 3 h intervals, were aggregated to a daily time step to match the time resolution of the target

time series² (Copernicus Climate Change Service, 2019). We amend these six meteorological input features with two more sine–cosine embeddings of the day of the week and the day of the year, which can be considered a coarse proxy for human activity (Otero et al., 2023).

3.3 Static input data

Following the insights from Kratzert et al. (2019c) and Shalev et al. (2019), we include static data, also called ancillary data, with the aim of training models jointly on all locations and thus increasing transfer performance. Specifically, we include hydrogeological properties, soil class, land cover, and orographic features derived from a digital elevation map for a total of 46 feature dimensions. Please refer to Vischer et al. (2025b) for a detailed description of the origin and pre-processing steps of all input features. The number of inputs could be streamlined in the future, as Cheng et al. (2023) showed how relevance propagation identifies non-task relevant features.

3.4 Target streamflow time series, station information, and river networks

Target time series of streamflow at each station were obtained from the Global Runoff Data Center (GRDC) data portal <https://portal.grdc.bafg.de/applications/public.html> (last access: 5 June 2025). Together with the streamflow data, the GRDC offers a catalog of station information. We considered all stations in the station catalog but excluded stations that had 10 or more values missing in the time series for the selected study period. Furthermore, initial experiments showed that including stations with less than 500 km² drainage area in training decreased performance, even when evaluating exclusively on larger stations. We decided to exclude these small catchments and discuss this decision in Sect. 6. Another natural limitation on the spatial scope of this approach is that it only captures rainfall streamflow dynamics in locations contained in the drainage area of a gauging station. In coastal areas, runoff might directly enter the sea through smaller streams that are not gauged. Hence, our study area usually starts several kilometers inland from the sea. The following number of stations resulted in each basin: 62 in upper Danube, 34 in Elbe, 36 in Oder, 78 in Rhine, and 29 in Weser basins, for a total of 239 stations. For comparison, the CAMELS dataset contains 671 catchments. Further visualizations of the river networks can be inspected in the pre-processing scripts, along with all details on how the metadata were processed.

²The dataset was downloaded from the Copernicus Climate Change Service (2019), <https://doi.org/10.24381/cds.e2161bac>. The results contain modified Copernicus Climate Change Service information 2020. Neither the European Commission nor ECMWF is responsible for any use that may be made of the Copernicus information or data it contains.

4 Methods

This section provides details on our model's local and routing stages. This is followed by an explanation of how the river network connectivity graph is calculated and integrated into the routing stage of the model as an inductive bias. We then describe the end-to-end training process and metrics and baselines used in our experiments.

4.1 Model local stage – modeling spatially resolved runoff

The task of the model's local stage is to integrate all input modalities and predict local runoff quantities on a regular grid. These local quantities are then fed through the routing layer described in the following subsection. As mentioned before, we follow Kratzert et al. (2019b) by adding static information to the meteorological input and adapting their original model design for the local stage. While this model was originally used on aggregated catchment time series, we apply it to all grid locations in parallel. Also, contrary to the original design, we add a simple, fully connected layer that reduces the 46 static input dimensions down to 10 dimensions. They are concatenated with another 10 dimensions of meteorological forcings for a total input dimensionality of the LSTM layer of 20, compared to 32 in the case of Kratzert et al. (2019b). Their LSTM layer consists of 256 units, while ours consists of 250. However, we reduce the 250 output values of the LSTM layer by using two regression layers instead of one. We do not employ a nonlinearity after the second readout layer, meaning that the network's outputs are not confined to a range, e.g., $[-1, 1]$ but rather live in the range of actual, physical quantities. Figure 2 visualizes the simple network architecture. In summary, the routing stage of our model is largely identical to the model used in Kratzert et al. (2019b), with a few small modifications. We want to emphasize here that the model does not have any predictive capabilities in itself. It uses meteorological forecasts to produce a forecast of runoffs.

In an exploratory experiment, we compared concatenating static inputs to dynamic inputs before and after the LSTM layer. Feeding the static inputs through the LSTM together with the dynamic inputs resulted in substantially better performance. This is consistent with the findings of Kratzert et al. (2019b) and can be explained by the static inputs helping the LSTM to better adapt to the hydrological dynamics of a location. It is similar to training separate models for different climatic zones but in a data-driven fashion. Indeed, Cheng et al. (2023) showed that clustering on the relevance values of the different inputs results in hydrologically plausible clusters. Concatenating all features before feeding them through the LSTM layer requires more parameters in this layer, which, due to its intricate inner workings, is particularly expensive to train in terms of data and compute. We also used gated recurrent units (GRUs) (Cho et al., 2014) instead

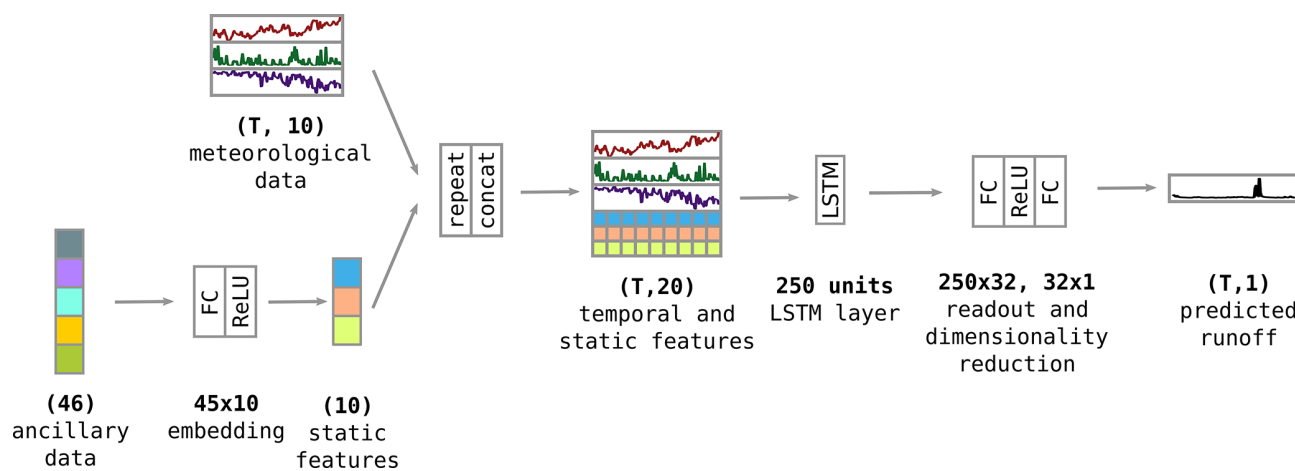


Figure 2. Overview of the local stage of the network architecture, acting in parallel on each of the input grid cells. Static inputs are reduced to 10 features by feeding them through a simple, fully connected embedding layer and applying a nonlinearity function. The resulting feature vector is repeated for every time step T of the meteorological forcing time series and concatenated with the time series. The resulting 20 features are fed through a 250-unit LSTM layer. The recurrent layer's output is then reduced to a 1D output time series by sequentially applying two fully connected readout layers, with a nonlinearity in between. Numbers in parentheses signify feature vectors at a given stage in the model pipeline; numbers without parentheses signify the size of a layer's weight matrix.

of LSTMs as a backend, which mitigates this problem a little because they are computationally more efficient. We found that they do not affect performance but decided to stick with LSTM as our main backbone, as it is more common in the literature. Nevertheless, using GRUs could be another way to further optimize the model.

4.2 Model routing stage – integrating local runoff and routing

The task of the routing stage of our model is to map the locally generated runoff to a station's catchment area and then route the runoff along the river network to predict streamflow time series for every station in the basin. Figure 3 visualizes the layout. Within a given river basin, we concatenate the predicted runoff time series of all grid cells. The network learns a simple, strictly linear mapping consisting of two layers: first, a fully connected layer without nonlinearity that maps all grid cells $\mathbf{G}$ to their respective stations $\mathbf{S}$. Since no nonlinearity is added, this layer can be translated into a weighted, time-dependent average of all grid cells within a catchment. Location information from the station catalog, described below, can be used as an inductive bias to constrain this layer so as to only route water in a physically plausible way. Then, a 1D-convolution layer (Kiranyaz et al., 2021) performs time convolution on each station's time series to combine information inside the river network over the last 9 d. Separate kernel values can be learned for each day, but the same kernel is applied jointly over the entire time series. The kernel length of 9 d was chosen as a conservative estimate of the maximum time it would take water to run along the entirety of any of the river networks considered in this study but of

course awaits empirical validation and further optimization. The kernels in this layer are constrained by the connectivity of the river network in order to be physically plausible, as we explain in the following subsection. Crucially, this stage does not involve any nonlinearity. Hence, both the fully connected and the time convolution layers are linear functions and, as such, can be chained to form yet another linear function. This means that the quantities of water “flowing” through this pipeline are physically interpretable. Put differently, input quantities can be added or subtracted meaningfully to and from the input. A practical application example of this, which we plan to further investigate, is the injection or extraction of water in between two stations to simulate agricultural, industrial, or hydroelectric human activity. This sets our approach apart from previous routing approaches. We also investigated the effect of not constraining the weight matrices with the connectivity matrices, which leads to slightly poorer performance when data are scarce (see Sects. 4.5 and 5.5).

4.3 Structural bias

The station catalog contains polygons of each station's catchment area. From this, we can derive two important kinds of information: first, for every grid cell, we can determine in which station's catchment it lies; second, for each station, we can determine how it is connected to upstream and downstream stations. This subsection explains how this information is extracted and how we use it as structural inductive bias to constrain the routing layer to only consider physically plausible routes of water flow.

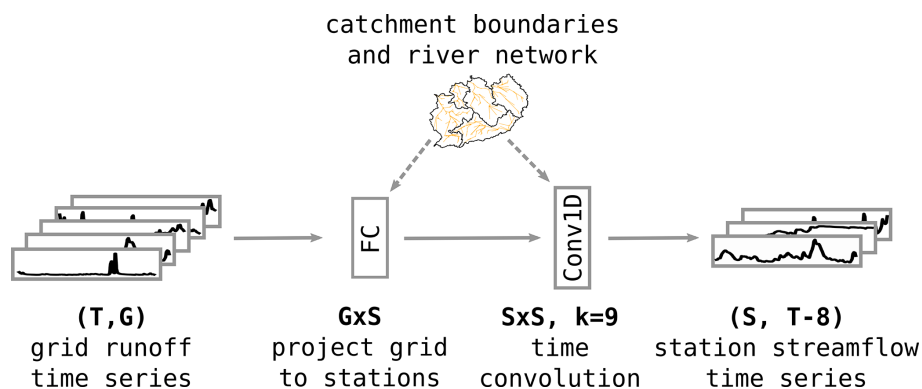


Figure 3. Overview of the routing stage of the network architecture, acting on the output of the local stage. The local runoff time series of all grid cells in a given basin are concatenated. Together, they are fed through a fully connected layer that projects them down onto the number of the basin's stations. The weight matrix in this layer is constrained to be non-zero only if a grid cell lies in the catchment of a given station. The stations' time series are then time-convolved with a kernel length of 9 time steps to model the flow of water between stations. Again, each kernel is constrained to be non-zero only if a given station is directly upstream of another station. The routing stage yields time series for all stations in the basin. It does not include any nonlinearity, so all activation values can be interpreted as streamflow quantities. The basin map icon and dotted arrows indicate that the river connectivity information serves as an inductive bias on these two layers, constraining the activations to replicate the real river network.

4.3.1 Catchment matrices

Mapping grid cells to stations is important to ensure that the runoff predicted at a given location ends up at the only station that is physically plausible. Since a given station's catchment area is contained within all the downstream stations' catchment areas, we need to make sure that we select the one where the generated runoff first enters the river network. To do so, we select the station with the smallest catchment area that contains a given grid cell. For each river network, we represent this information conveniently in a one-hot matrix with grid cells as rows and stations as columns. This matrix is then used to constrain the fully connected layer in the routing stage of the network. This is achieved by multiplying the one-hot matrix pointwise with the freshly initialized weight matrix of this layer before training begins. Weights corresponding to physically impossible connections are thus set to zero from the start. Zero weights cannot contribute to gradients and will remain zero throughout the training. All other weights are free to be optimized.

4.3.2 Connectivity matrices

From the catchment area polygons, a graph representing the connectivity between stations in the river network can be derived. Each node in the graph represents a station; a directed edge exists between station *A* and *B* if *A* is directly upstream of *B*. We determine this by verifying if the catchment area of *A* is contained in the catchment area of *B*, ensuring that there are no intermediary stations in between, i.e., contained by *B* and containing *A*. Note that this automatically leads to a directed acyclic graph. This fits our approach well, as it does not require us to apply any model of routing iteratively in or-

der to capture cyclic movements within the graph. The graph is represented by a connectivity matrix, i.e., a square matrix with rows (input) and columns (output) corresponding to stations, where the entries are 1 if a directed edge exists and 0 otherwise. This matrix is used to constrain the time convolution layer in a manner similar to the catchment matrix. After initialization of the weight matrix, the connectivity matrix is multiplied pointwise, preserving the weights where a connection exists and setting them to zero where no connection is present. The only difference is that the connectivity matrix needs to be repeated by the depth of the temporal convolution, nine times in our case, for the 9 d of past information that we convolve.

4.4 Training and metrics

We split the data into three parts, all containing entire water years: a training set from water years 1981 to 2005, a validation set for model selection from 2005 to 2008, and a test set to report the final performance from 2008 to 2011. We also created two special training datasets to illustrate how the models perform on less training data: a medium length training set ranging from 1991 to 2005 and a short training set from 1999 to 2005. Regardless of the length of the training set, we divide it into chunks of 400 d that partially overlap. The first 30 d are used as a warm-up period for the LSTM. During this time, no gradients are computed, and the LSTM can stabilize into an operating regime before starting the learning process. The value of 400 was chosen to accommodate an entire year plus the warm-up period, while still being able to fit the gradients calculated during training into GPU memory. It would not be detrimental to use even longer time series, and the model is capable of pro-

cessing time series of arbitrary length. In fact, no gradients are computed during inference, which significantly reduces the memory footprint and allows us to calculate all metrics on uninterrupted time series in a single model forward pass. In all experiments, we trained for 2000 epochs on the training data. This number of epochs is generous for all models to converge. Since the purpose of this study is to provide a proof of concept of this type of spatially resolved processing, we decided not to conduct an extensive hyperparameter optimization or perform input feature ablation. Instead, we included all potentially relevant static data and ran hyperparameter tuning experiments only on a limited set of values for a few key hyperparameters, summarized in Table A2. We trained 10 different random seeds for every setting and report the best seed in terms of station-wise median Nash–Sutcliffe Efficiency (NSE) score on the test period.

We use the widespread NSE metric (Nash and Sutcliffe, 1970) both as a loss function in training and as a score to quantify performance. The NSE normalizes the squared loss of each station by the standard deviation of the station's values in the training period, so as to count each station equitably toward the loss or performance, regardless of the magnitude of the river at this point. We do not split training and test partitions geographically, as breaking up basins would make routing impossible. We also refer to Klotz et al. (2024) for a word of caution when combining NSE values that are calculated on partitions of a dataset. Unless noted otherwise, the scores we report were calculated on the test period. The median NSE over all stations serves as a robust point estimate of performance, but for the interested reader, we provide mean NSE scores in Table A1, which we found to correlate strongly with the median NSE for all seeds. Likewise, we report the percentage of stations with an NSE score below zero, which indicates predictive performance worse than simply using the average value of a station's runoff for prediction, i.e., chance level. The popular Kling–Gupta efficiency (KGE) metric (Gupta et al., 2009) was developed in the context of univariate, convex optimization; both these assumptions do not hold in the case of training a neural network. However, for the sake of comparability, we also report the KGE values in the appendix.

As stated before, in this study, we refrained from extensive hyperparameter optimization to maximize the performance. A few exploratory experiments to calibrate our pipeline seemed necessary nonetheless: dropout (0 %, 10 %, 30 %, and 50 % separately in recurrent and readout layers of the routing stage) did not increase performance, so we removed it entirely. In the LSTM layer, 250 units (out of 150, 200, 250, and 300) yielded the best results. We use an automatic learning rate scheduler, the ReduceLROnPlateau scheduler provided by PyTorch (Ansel et al., 2024) with a threshold of 1×10^{-3} and patience of 10, so the pipeline trains robustly with regard to the initial learning rate (1×10^{-4} , 5×10^{-4} , 1×10^{-5}). But as the baselines in the experiments have vastly different numbers of parameters, we decided to

continue the experiments by always trying out both of the lower values. Unless explicitly mentioned, we report the performance resulting from the better value for every condition; Table A2 lists the results in detail.

4.5 Baselines

We introduce two baselines in order to evaluate the performance gains of the two central aspects of our pipeline: spatially resolved processing in the local stage and the inclusion of structural bias in the routing stage. In the first baseline experiment, we aggregate all spatial information within a catchment and feed it through the same architecture as used in the local stage. Normally, this stage of the model processes individual grid cells. But here it processes entire, aggregated catchments, and the output is a prediction of the runoff measured at the corresponding station. The aggregated baseline does not require any further routing. This model and processing pipeline are identical to the one in Kratzert et al. (2019b), apart from the small differences in parameter values listed in Sect. 4.1. As we discussed above, it has been widely used in the literature since its introduction, which allows us to compare our own results from our custom dataset to a wider body of literature. We will call this baseline *aggregated*, whereas our default model will be referred to as *spatially resolved*.

The second baseline leaves the local stage unaltered but does not constrain the weight matrices in the routing stage. Instead, we use two fully connected time convolution layers with a kernel size of 9 and 3 d, respectively, and a nonlinearity in between. The design is intended to be simpler and conventional for neural networks. We will refer to this baseline as *naive routing* and to our default model as *structured routing*.

5 Results and discussion

We start this section by presenting and contextualizing the general performance of our model. We then show that our model excels in modeling large catchments, is less prone to overfitting, and learns from the data more efficiently than the baselines. Inductive bias does not make a big difference in performance. We end this section by showing that our model is, unlike most neural networks, not an entirely black box model and that capturing human influence seems to be the biggest challenge in our study area.

5.1 Model performance

The aggregated baseline achieves a median NSE test performance of 0.69 on our dataset (Table 1). As discussed above, it is virtually identical to the model from Kratzert et al. (2019b), who reported a median NSE of 0.74 on the CAMELS dataset. These results were subsequently confirmed, small modifications of the model parameters or test period notwithstanding, by Shalev et al. (2019) (median

Table 1. Comparison of median catchment NSE scores and fraction of catchments with NSE scores below zero for the spatially resolved pipeline (structured and naive routing) and the aggregated baseline, which corresponds to the model introduced by Kratzert et al. (2019b). Structured routing outperforms naive routing, and spatial processing outperforms aggregated processing. Metrics are reported for the best of 10 random seeds in terms of median NSE score. An extended version of this table, including KGE scores and shorter training periods, can be found in Table A1.

Input Processing	Routing	median NSE	NSE < 0
spatially resolved	structured	0.773	0.000
spatially resolved	naive	0.719	0.000
aggregated	–	0.691	0.013

NSE 0.73), Acuña Espinoza et al. (2025) (median NSE 0.75), and Frame et al. (2022) (median NSE between 0.72 and 0.81). We conjecture that our baseline performance is probably lower due to our dataset being smaller than CAMELS and likely containing greater human activity. Turning to other regions, the same model was used, for example, by Mai et al. (2022) for the Great Lakes Area (median KGE of 0.76), by Lees et al. (2021) for Great Britain (median NSE 0.88), and by Loritz et al. (2024) for Germany (median NSE 0.84). The latter two results go against our conjecture for inferior performance due to human signal. The fact that Loritz et al. (2024) used an ensemble of models might be a factor at play, but in any case, this requires further investigation. These studies provide valuable comparisons with other model types. In all the studies mentioned, the authors compared the neural network model against a variety of physical, conceptual, or hybrid models and consistently found that the alternatives were outperformed by the neural network.

Our model achieves a median NSE performance on the test dataset of 0.77 compared to the baseline of 0.69 and thus appears to be able to compensate somewhat for the more demanding modeling context. Only training and testing our model on a spatially extended version of CAMELS would allow for more direct comparability with other approaches. Unfortunately, this has to be left for future research as it requires creating a spatially resolved version of CAMELS first. In any case, despite the data being different and not allowing for a straightforward comparison, this shows that runoff generation and routing can be learned end-to-end by a single model pipeline and without additional data along the way. Moreover, the simplicity of the routing module’s design, along with the possibilities it offers, does not come at a significant performance cost. An example in this case is that the internal activations inside the network between local and routing stages appear to be hydrologically plausible, as Fig. 6 illustrates. We want to stress that this is an emergent property and was not incentivized during training.

5.2 Modeling large catchments

Do larger catchments benefit more from spatially resolved processing as we expected? Figure 4 shows the effect of catchment size on per-catchment performance, comparing our spatially resolved model to the aggregated baseline for training and testing periods. We focus on the test results first (right panel): For aggregated processing, we observe a negative trend between performance and catchment size. This might be explained by the assumption that underlies spatial aggregation, namely the homogeneity of the aggregated area, does not hold for large catchments. Indeed, the trend is reversed for spatially resolved processing, where larger catchments outperform smaller ones. Our results suggest that spatially resolved processing should be considered particularly when modeling large and heterogeneous catchments.

5.3 Inherent regularization

A comparison between the left and right panels of Fig. 4 reveals a substantial performance drop between the training and test periods. To a certain degree, this is normally to be expected due to overfitting. But the effect is much more pronounced for the aggregated than for the spatially resolved model. It stands to reason that the neural network in the aggregated baseline, with its associated reduction of data, overfits severely. This also becomes apparent when looking at the median NSE values in training and test datasets. While overall performance drops from 0.86 in training to 0.69 in test for aggregated processing, spatially resolved processing deteriorates more gracefully from 0.90 to 0.77. Spatially resolved processing, with its shared local stage and overall much more data, seems to have an intrinsic regularization effect.

5.4 Data efficiency

The positive effect of spatially resolved training, especially for large catchments, becomes even more pronounced when looking at modeling in an environment with limited available training data. Figure 5 visualizes the difference in NSE score between spatial and aggregated processing on a per-catchment level, when training on 25, 15, and 6 years of training data. Differences are positive across all sizes of catchments, meaning spatial processing on average performs better regardless of the catchment size. Yet the positive trend becomes stronger as data become scarcer. Spatially resolved processing utilizes the available data more efficiently.

5.5 Inductive bias

Including inductive bias for what we call structured routing leads to slightly better performance than naive routing without the additional real-world information, resulting in a median NSE of 0.77 compared to 0.72. Figure A1 contains more detailed results but also shows that the performance gain is small. The point we want to make here is that the practical

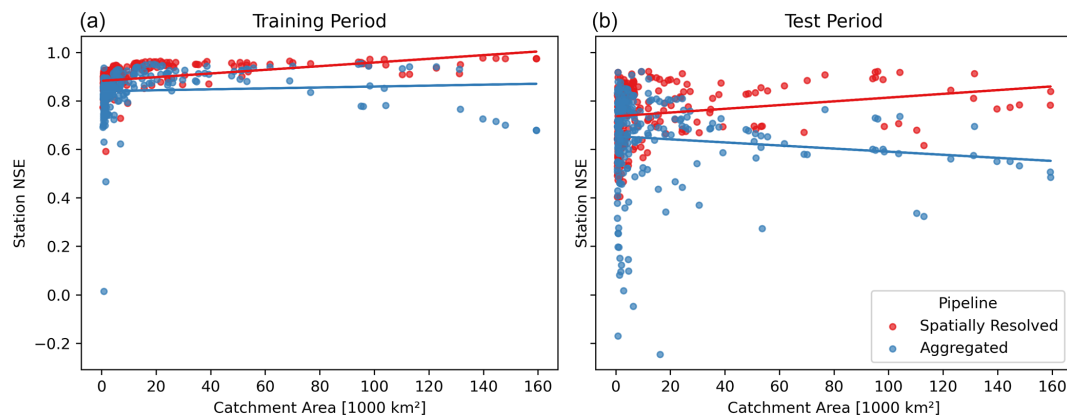


Figure 4. Effect of catchment size: each point corresponds to a station's performance vs. the size of its catchment, for the aggregated baseline (blue) and the spatially resolved pipeline (red) on the training (a) and test (b) periods. The individual data points are fitted linearly to show the trend. Aggregated processing impairs performance, especially in larger catchments. The trend is more pronounced in the test dataset, indicating overfitting in the training dataset. Spatially resolved processing is less prone to overfitting and manages to handle large catchments accurately in a low-data setting.

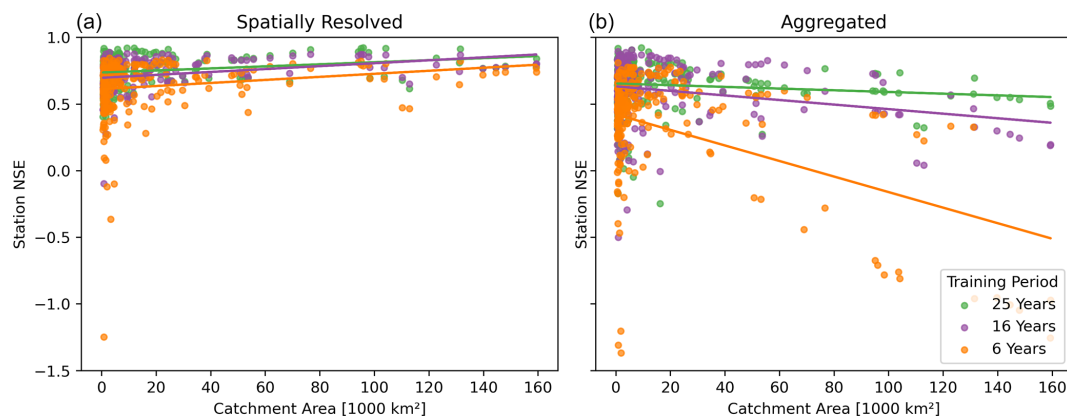


Figure 5. Effect of training period length: each point corresponds to a station's NSE test performance after training for 25 years (green), 15 years (purple), and 6 years (orange). Across all catchment sizes and different quantities of training data, the spatial pipeline (a) outperforms the aggregated baseline (b). The latter's performance deteriorates quickly when training data are scarce. The spatially resolved pipeline is much more robust in this regard, demonstrating its increased data efficiency.

benefit of being able to simulate the injection or extraction of quantities of water in the routing process does not come at the cost of lowering performance. As we mentioned before, naive routing, on the other hand, remains an important tool in modeling basins where catchment delineation information is unavailable or unreliable, or where lateral transport of water inside the bedrock layer across catchment boundaries is suspected. Whether or not we use inductive bias in the routing layer, our networks are extremely simple compared to other networks proposed in the literature for routing that we discussed above. Certainly, we demonstrate that routing modules do not need to be complex, and river network extraction algorithms are not necessary for end-to-end routing, e.g., when no catchment boundary information is available.

5.6 Interpretable internal states

Internal states of neural networks are usually not interpretable. In our case, however, the activations of the final layer of the local stage that would go into the routing module stage of the model appear to be hydrologically plausible. As an illustrative example, Fig. 6 displays these activation values for two exemplary days in spring and summer. The spatio-temporal correlation seems to suggest that on the example day in spring, runoff is primarily driven by snowmelt in low mountain ranges, whereas in summer, it seems to be driven by heavy precipitation events. We want to make the point that we did not enforce this property during the training process, e.g., by providing additional target information or training a special readout layer. This is a purely data-driven, emergent behavior, resulting from both the end-to-end train-

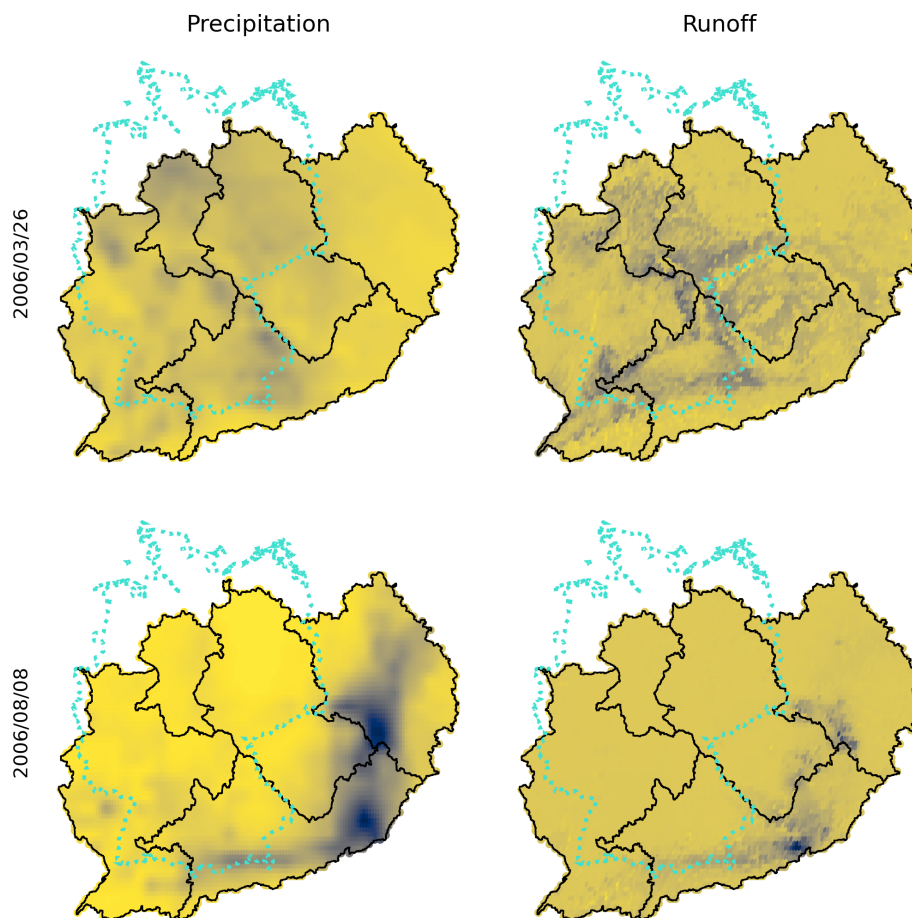


Figure 6. Internal states of the model are hydrologically interpretable. This figure shows precipitation data from the input (left column) and activation values after the local stage of the model (right column) for 2 d in March (top row) and August 2006 (bottom row). The days were manually selected for illustration purposes. Values are displayed in arbitrary units, with blue signifying more runoff or precipitation. We included the outline of Germany (© EuroGeographics for the administrative boundaries) in dotted lines for geographic reference. For the day in spring (top row), a low spatial correlation between precipitation and runoff, together with a pattern of high runoff values in low mountain ranges, suggests that runoff on this day is primarily driven by snowmelt in intermediate altitudes. For the day in summer (bottom row), on the other hand, we see runoff that is driven by two clusters of heavy precipitation in the eastern part of our study area.

ing process and the model's parsimonious design. Particularly, we attribute this phenomenon to the shared recurrent layer in the local stage and the combination of inductive bias and linearity of the routing stage. Unlike more complex neural network designs that are generally considered black box models, this suggests that our model naturally allows for a certain degree of internal control by manipulating these internal states, e.g., by subtracting or adding quantities of water to the natural runoff. It also creates new opportunities for further scientific discovery from large quantities of data.

5.7 On human influence

We conclude this section by discussing a specific negative outlier in terms of station-wise NSE. As explained above, a negative NSE value indicates performance below chance level. The only two stations that yielded negative NSE val-

ues after training – consistently across all 10 random seeds – are Spremberg and Boxberg (GRDC numbers 6340800 and 6340810) located along small rivers next to large brown coal surface mining operations in the Lusatia region. A potential explanation for these extreme outliers is that those mining operations have an influence on the overall water balance that is relatively large compared to the hydrological processes in such catchments. This seems to support our assumption that human influence is one of the main obstacles to be overcome by rainfall streamflow models in the densely populated areas of central Europe. We want to emphasize that the two stations in question only performed very poorly in the validation period but not in the test period. We hypothesize that operations might have changed in the meantime or that the test period simply lacked substantial events of human influence by chance. We plan to investigate this phenomenon more closely and explore potential solutions in future work.

6 Summary and future directions

We have successfully trained a neural network in an end-to-end fashion to capture runoff generation in a spatially resolved manner on a large scale. Training on five entire river basins in central Europe, we have shown that this approach is advantageous especially in large catchments. The parsimonious network design of the module that performs routing within these river basins is also noteworthy. Not only does this approach mitigate overfitting and increase data efficiency, but the simplicity of the design and the ability to integrate inductive bias open new possibilities to control the inner workings of the model. This is not common for neural networks.

Our model reaches a level of performance comparable to that of other benchmark models, both conceptual and statistical. In future work, we plan to compare our model to other state-of-the-art models used in science and operations. The creation of a spatially resolved version of the popular CAMELS dataset would make our model's performance directly comparable to a large portion of the neural network literature. A direct comparison of our overall pipeline to that of an operational system like LISFLOOD would be similarly interesting. Another question that needs to be addressed in the future is how the performance of our model decreases with the forecast horizon. As we discussed before, our model does not have predictive capabilities of its own. Instead, it relies on a suitable meteorological forecast as input to generate a forecast of streamflow quantities, and the quality of the model's predictions thus depends on the quality of the forecast meteorological input. Quantifying this effect is important for real-world applications.

In this study, we excluded catchments where we suspected that human influence was too strong based on a simple catchment area heuristic. Unlike much research in this area, we do not train our model exclusively on catchments with little human influence. However, a more sophisticated strategy – inspired, for example, by Loritz et al. (2024) or Tursun et al. (2024) – is needed to properly quantify human impact and thus be able to disentangle the effects of catchment area and human influence on performance. Likewise, while we included day of the week and day of the year as simple approximations to human influence, the effectiveness of this measure can only be properly evaluated with a suitable ground truth.

Another significant limitation in terms of data availability is the temporal and spatial resolution, as the relatively small size of the recurrent layer enables our model to process time series at a higher temporal resolution than daily. Because the local stage is applied in parallel to all input locations, the number of parameters is independent of the number of inputs. This is where most of the computation happens, and its computational demands grow only linearly with the number of locations. The weight matrices in the routing stage grow quadratically but are much smaller in the first place.

Another important aspect that we will address in future work is discussed in Klotz et al. (2022), where the authors extend a model similar to ours by using the outputs as parameters of a distribution. Such distributional predictions could be obtained from our model in the same way and would be of great interest for many real-world applications. Producing genuinely probabilistic forecasts and warnings in this manner is theoretically more sound than training an ensemble of more or less different models and combining their predictions.

As would be expected from the high degree of human activity in our study area, we found evidence that the effect of human influence was the central obstacle to further improving model performance in such an environment. Yet, a comprehensive investigation into the extent and impact of this phenomenon is still required. Future research could demonstrate that our neural network is capable of incorporating simulated human activity, such as water extraction or diversion, into the modeling of hydrological processes.

Appendix A

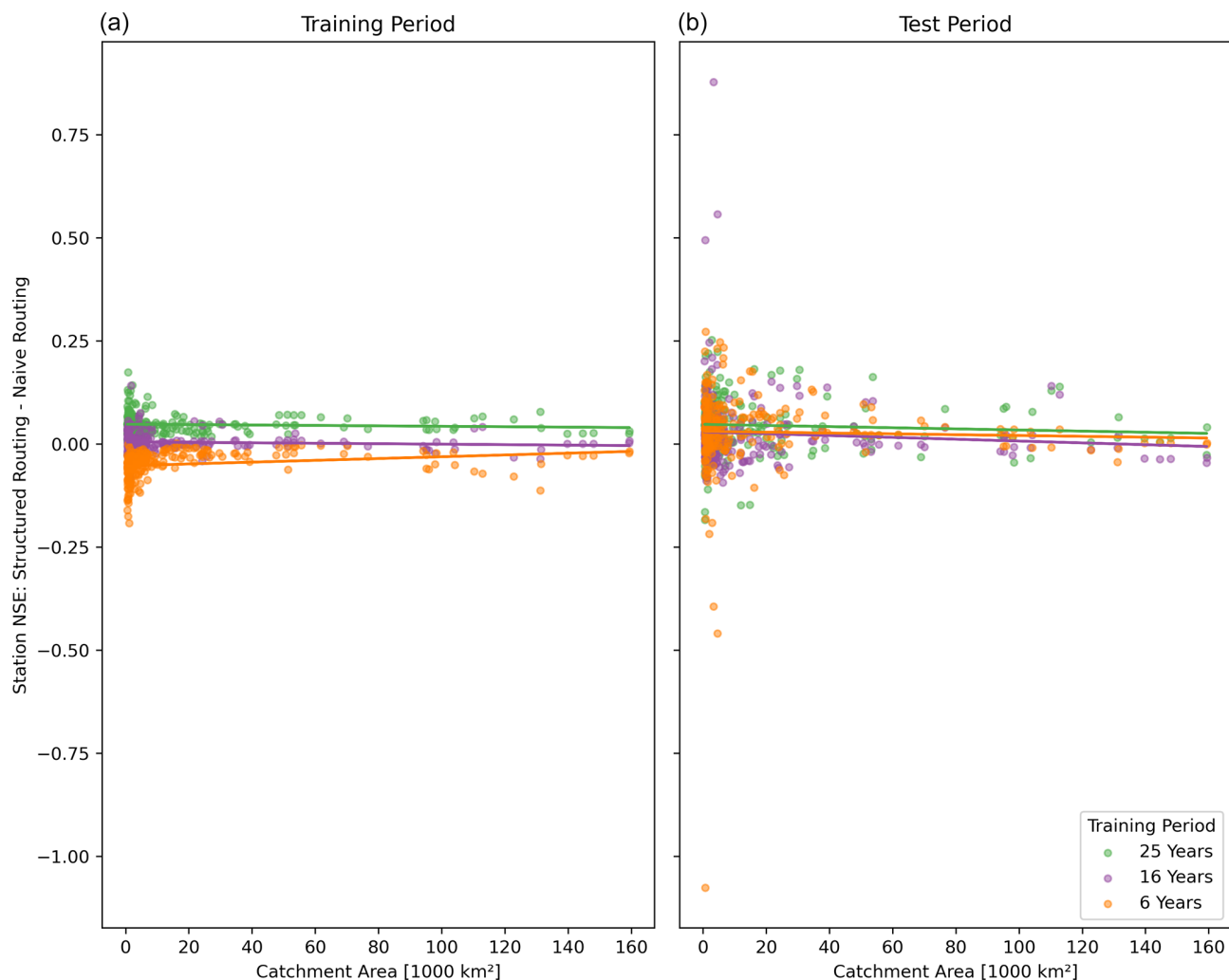


Figure A1. Station-wise differences in NSE performance between structured and naive routing, plotted against the size of the station's catchment in the training (a) and test periods (b). Positive values indicate that structured routing performs better. The individual data points are fitted linearly. Structured routing marginally outperforms naive routing on the test dataset (indicated by values greater than zero). In the training dataset, naive routing performs better when data are scarce (values below zero for the short training period in green), indicating overfitting.

Table A1. Various performance metrics for the experiments presented in Sect. 5. The metrics were calculated after picking the best out of 10 random seeds for each condition based on median NSE.

Input Processing	Routing	Train Period	median NSE	mean NSE	NSE < 0	median KGE	mean KGE
spatially res.	structured	long	0.773	0.751	0.000	0.803	0.779
spatially res.	naive	long	0.719	0.706	0.000	0.791	0.775
aggregated	–	long	0.691	0.643	0.013	0.731	0.702
spatially res.	structured	med.	0.739	0.717	0.004	0.775	0.753
spatially res.	naive	med.	0.735	0.692	0.008	0.776	0.744
aggregated	–	med.	0.642	0.603	0.013	0.708	0.666
spatially res.	structured	short	0.687	0.633	0.017	0.752	0.724
spatially res.	naive	short	0.653	0.605	0.004	0.739	0.709
aggregated	–	short	0.485	0.318	0.126	0.610	0.513

Table A2. Median NSE over 20 seeds for exploratory experiments on the optimal initial learning rate. Performance was evaluated in the *validation* period, as this is part of the model selection process.

Input Processing	Routing	initial LR	median NSE
spatially res.	structured	5×10^{-4}	0.738
spatially res.	naive	5×10^{-4}	0.724
aggregated	–	5×10^{-4}	0.691
spatially res.	structured	1×10^{-3}	0.745
spatially res.	naive	1×10^{-3}	0.721
aggregated	–	1×10^{-3}	0.704

Code and data availability. The data used in this study are publicly available under CC BY-NC-SA license at <https://doi.org/10.4211/hs.05d5633a413b4aec93b08a7e61a2abbb> (Vischer et al., 2025a); the code used to preprocess them is available under Clear BSD license at https://gitlab.hhi.fraunhofer.de/vischer/spatial_streamflow_dataprep (last access: 5 June 2025).

Author contributions. MAV and JM designed the experiments with crucial suggestions from NO. MAV prepared the data, implemented the model, performed experiments, and analyzed the results. MAV prepared the manuscript with significant contributions from JM and NO. All authors reviewed the manuscript.

Competing interests. The contact author has declared that none of the authors has any competing interests.

Disclaimer. Publisher's note: Copernicus Publications remains neutral with regard to jurisdictional claims made in the text, published maps, institutional affiliations, or any other geographical representation in this paper. While Copernicus Publications makes every effort to include appropriate place names, the final responsibility lies with the authors. Views expressed in the text are those of the authors and do not necessarily reflect the views of the publisher.

Acknowledgements. This work was supported by the Federal Ministry for Economic Affairs and Climate Action (BMWK) under grant DAKI-FWS (01MK21009A) and by the European Union's Horizon Europe research and innovation program (EU Horizon Europe) project MedEWSa under grant agreement no. 101121192.

Financial support. This work was supported by the Federal Ministry for Economic Affairs and Climate Action (BMWK) as grant DAKI-FWS (01MK21009A) and by the European Union's Horizon Europe research and innovation program (EU Horizon Europe) project MedEWSa under grant agreement no. 101121192.

Review statement. This paper was edited by Fuqiang Tian and reviewed by two anonymous referees.

References

- Acuña Espinoza, E., Loritz, R., Kratzert, F., Klotz, D., Gauch, M., Álvarez Chaves, M., and Ehret, U.: Analyzing the generalization capabilities of a hybrid hydrological model for extrapolation to extreme events, *Hydrol. Earth Syst. Sci.*, 29, 1277–1294, <https://doi.org/10.5194/hess-29-1277-2025>, 2025.
- Addor, N., Newman, A. J., Mizukami, N., and Clark, M. P.: The CAMELS data set: catchment attributes and meteorology for large-sample studies, *Hydrol. Earth Syst. Sci.*, 21, 5293–5313, <https://doi.org/10.5194/hess-21-5293-2017>, 2017.
- Alvarez-Garreton, C., Mendoza, P. A., Boisier, J. P., Addor, N., Galleguillos, M., Zambrano-Bigiarini, M., Lara, A., Puelma, C., Cortes, G., Garreaud, R., McPhee, J., and Ayala, A.: The CAMELS-CL dataset: catchment attributes and meteorology for large sample studies – Chile dataset, *Hydrol. Earth Syst. Sci.*, 22, 5817–5846, <https://doi.org/10.5194/hess-22-5817-2018>, 2018.
- Ansel, J., Yang, E., He, H., Gimelshein, N., Jain, A., Voznesensky, M., Bao, B., Bell, P., Berard, D., Burovski, E., Chauhan, G., Chourdia, A., Constable, W., Desmaison, A., DeVito, Z., Ellison, E., Feng, W., Gong, J., Gschwind, M., Hirsh, B., Huang, S., Kalambarkar, K., Kirsch, L., Lazos, M., Lezcano, M., Liang, Y., Liang, J., Lu, Y., Luk, C. K., Maher, B., Pan, Y., Puhersch, C., Reso, M., Saroufim, M., Siraiichi, M. Y., Suk, H., Zhang, S., Suo, M., Tillet, P., Zhao, X.,

- Wang, E., Zhou, K., Zou, R., Wang, X., Mathews, A., Wen, W., Chanan, G., Wu, P., and Chintala, S.: PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation, in: Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2, pp. 929–947, ACM, La Jolla CA USA, ISBN 9798400703850, <https://doi.org/10.1145/3620665.3640366>, 2024.
- Bevacqua, E., Shepherd, T. G., Watson, P. A. G., Sparrow, S., Wallom, D., and Mitchell, D.: Larger Spatial Footprint of Wintertime Total Precipitation Extremes in a Warmer Climate, *Geophys. Res. Lett.*, 48, e2020GL091990, <https://doi.org/10.1029/2020GL091990>, 2021.
- Beven, K. J.: *Rainfall-Runoff Modelling: The Primer*, Wiley-Blackwell, Chichester, West Sussex, Hoboken, NJ, 2nd edn., ISBN 978-0-470-71459-1, 2012.
- Bronstein, M. M., Bruna, J., Cohen, T., and Velicković, P.: Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges, *arXiv [preprint]*, [arXiv:2104.13478](https://arxiv.org/abs/2104.13478), 2021.
- Chagas, V. B. P., Chaffe, P. L. B., Addor, N., Fan, F. M., Fleischmann, A. S., Paiva, R. C. D., and Siqueira, V. A.: CAMELS-BR: hydrometeorological time series and landscape attributes for 897 catchments in Brazil, *Earth Syst. Sci. Data*, 12, 2075–2096, <https://doi.org/10.5194/essd-12-2075-2020>, 2020.
- Chen, Z., Lin, H., and Shen, G.: TreeLSTM: A Spatiotemporal Machine Learning Model for Rainfall-Runoff Estimation, *J. Hydrol.: Regional Studies*, 48, 101474, <https://doi.org/10.1016/j.ejrh.2023.101474>, 2023.
- Cheng, X., Vischer, M., Schellin, Z., Arras, L., Kuglitsch, M. M., Samek, W., and Ma, J.: Explainability in GeoAI, in: *Handbook of Geospatial Artificial Intelligence*, CRC Press, ISBN 978-1-00-330842-3, 2023.
- Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y.: Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation, *arXiv [preprint]*, <https://doi.org/10.48550/arXiv.1406.1078>, 2014.
- Clark, S. R., Lerat, J., Perraud, J.-M., and Fitch, P.: Deep learning for monthly rainfall-runoff modelling: a large-sample comparison with conceptual models across Australia, *Hydrol. Earth Syst. Sci.*, 28, 1191–1213, <https://doi.org/10.5194/hess-28-1191-2024>, 2024.
- Copernicus Climate Change Service (C3S): ERA5-Land hourly data from 1950 to present. Copernicus Climate Change Service (C3S) Climate Data Store (CDS) [data set], <https://doi.org/10.24381/cds.e2161bac> (last access: 23 October 2021), 2019.
- Coxon, G., Addor, N., Bloomfield, J. P., Freer, J., Fry, M., Hannaford, J., Howden, N. J. K., Lane, R., Lewis, M., Robinson, E. L., Wagener, T., and Woods, R.: CAMELS-GB: hydrometeorological time series and landscape attributes for 671 catchments in Great Britain, *Earth Syst. Sci. Data*, 12, 2459–2483, <https://doi.org/10.5194/essd-12-2459-2020>, 2020.
- Delaigue, O., Brigode, P., Andréassian, V., Perrin, C., Etchevers, P., Soubeyroux, J.-M., Janet, B., and Addor, N.: CAMELS-FR: A large sample hydroclimatic dataset for France to explore hydrological diversity and support model benchmarking, *IAHS-AISH Scientific Assembly 2022*, Montpellier, France, 29 May–3 June 2022, IAHS2022-521, <https://doi.org/10.5194/iahs2022-521>, 2022.
- European Space Agency: Copernicus DEM – Global and European Digital Elevation Model (GLO90), The ESA itself publishes this on their website, <https://dataspace.copernicus.eu/explore-data/data-collections/copernicus-contributing-missions/collections-description/COP-DEM> (last access: 5 June 2025), 2021.
- Fang, B., Bevacqua, E., Rakovec, O., and Zscheischler, J.: An increase in the spatial extent of European floods over the last 70 years, *Hydrol. Earth Syst. Sci.*, 28, 3755–3775, <https://doi.org/10.5194/hess-28-3755-2024>, 2024.
- Fowler, K. J. A., Acharya, S. C., Addor, N., Chou, C., and Peel, M. C.: CAMELS-AUS: hydrometeorological time series and landscape attributes for 222 catchments in Australia, *Earth Syst. Sci. Data*, 13, 3847–3867, <https://doi.org/10.5194/essd-13-3847-2021>, 2021.
- Frame, J. M., Kratzert, F., Klotz, D., Gauch, M., Shalev, G., Gilon, O., Qualls, L. M., Gupta, H. V., and Nearing, G. S.: Deep learning rainfall-runoff predictions of extreme events, *Hydrol. Earth Syst. Sci.*, 26, 3377–3392, <https://doi.org/10.5194/hess-26-3377-2022>, 2022.
- Gründemann, G. J., van de Giesen, N., Brunner, L., and van der Ent, R.: Rarest Rainfall Events Will See the Greatest Relative Increase in Magnitude under Future Climate Change, *Commun. Earth Environ.*, 3, 1–9, <https://doi.org/10.1038/s43247-022-00558-8>, 2022.
- 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, *J. Hydrol.*, 377, 80–91, <https://doi.org/10.1016/j.jhydrol.2009.08.003>, 2009.
- Hitokoto, M. and Sakuraba, M.: Hybrid Deep Neural Network and Distributed Rainfall-Runoff Model for Real-Time River-Stage Prediction, *Journal of JSCE*, 8, 46–58, https://doi.org/10.2208/journalofjsce.8.1_46, 2020.
- Hochreiter, S. and Schmidhuber, J.: Long Short-term Memory, *Neural Computation MIT-Press*, 9, 1735–1780, 1997.
- Höge, M., Kauzlaric, M., Siber, R., Schönenberger, U., Horton, P., Schwanbeck, J., Florianic, M. G., Viviroli, D., Wilhelm, S., Sikorska-Senoner, A. E., Addor, N., Brunner, M., Pool, S., Zappa, M., and Fenicia, F.: CAMELS-CH: hydro-meteorological time series and landscape attributes for 331 catchments in hydrologic Switzerland, *Earth Syst. Sci. Data*, 15, 5755–5784, <https://doi.org/10.5194/essd-15-5755-2023>, 2023.
- Horton, P., Schaefli, B., and Kauzlaric, M.: Why Do We Have so Many Different Hydrological Models? A Review Based on the Case of Switzerland, *WIREs Water*, 9, e1574, <https://doi.org/10.1002/wat2.1574>, 2022.
- Hu, F., Yang, Q., Yang, J., Luo, Z., Shao, J., and Wang, G.: Incorporating Multiple Grid-Based Data in CNN-LSTM Hybrid Model for Daily Runoff Prediction in the Source Region of the Yellow River Basin, *J. Hydrol.: Regional Studies*, 51, 101652, <https://doi.org/10.1016/j.ejrh.2023.101652>, 2024.
- Hu, T., Wu, F., and Zhang, X.: Rainfall-Runoff Modeling Using Principal Component Analysis and Neural Network, *Hydrol. Res.*, 38, 235–248, <https://doi.org/10.2166/nh.2007.010>, 2007.
- Jiang, S., Bevacqua, E., and Zscheischler, J.: River flooding mechanisms and their changes in Europe revealed by explainable

- machine learning, *Hydrol. Earth Syst. Sci.*, 26, 6339–6359, <https://doi.org/10.5194/hess-26-6339-2022>, 2022.
- Kam, P. M., Aznar-Siguan, G., Schewe, J., Milano, L., Ginnetti, J., Willner, S., McCaughey, J. W., and Bresch, D. N.: Global Warming and Population Change Both Heighten Future Risk of Human Displacement Due to River Floods, *Environ. Res. Lett.*, 16, 044026, <https://doi.org/10.1088/1748-9326/abd26c>, 2021.
- Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., and Inman, D. J.: 1D Convolutional Neural Networks and Applications: A Survey, *Mechanical Systems and Signal Processing*, 151, 107398, <https://doi.org/10.1016/j.ymssp.2020.107398>, 2021.
- Klingler, C., Schulz, K., and Herrnegger, M.: LamaH-CE: LArge-SaMple Data for Hydrology and Environmental Sciences for Central Europe, *Earth Syst. Sci. Data*, 13, 4529–4565, <https://doi.org/10.5194/essd-13-4529-2021>, 2021.
- 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, *Hydrol. Earth Syst. Sci.*, 26, 1673–1693, <https://doi.org/10.5194/hess-26-1673-2022>, 2022.
- Klotz, D., Gauch, M., Kratzert, F., Nearing, G., and Zscheischler, J.: Technical Note: The divide and measure nonconformity – how metrics can mislead when we evaluate on different data partitions, *Hydrol. Earth Syst. Sci.*, 28, 3665–3673, <https://doi.org/10.5194/hess-28-3665-2024>, 2024.
- Kratzert, F., Klotz, D., Brenner, C., Schulz, K., and Herrnegger, M.: Rainfall-runoff modelling using Long Short-Term Memory (LSTM) networks, *Hydrol. Earth Syst. Sci.*, 22, 6005–6022, <https://doi.org/10.5194/hess-22-6005-2018>, 2018.
- Kratzert, F., Herrnegger, M., Klotz, D., Hochreiter, S., and Klambauer, G.: NeuralHydrology – Interpreting LSTMs in Hydrology, in: *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, edited by: Samek, W., Montavon, G., Vedaldi, A., Hansen, L. K., and Müller, K.-R., Lecture Notes in Computer Science, 347–362, Springer International Publishing, Cham, ISBN 978-3-030-28954-6, https://doi.org/10.1007/978-3-030-28954-6_19, 2019a.
- Kratzert, F., Klotz, D., Herrnegger, M., Sampson, A. K., Hochreiter, S., and Nearing, G. S.: Toward Improved Predictions in Ungauged Basins: Exploiting the Power of Machine Learning, *Water Resour. Res.*, 55, 11344–11354, <https://doi.org/10.1029/2019WR026065>, 2019b.
- 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, *Hydrol. Earth Syst. Sci.*, 23, 5089–5110, <https://doi.org/10.5194/hess-23-5089-2019>, 2019c.
- Kratzert, F., Gauch, M., Klotz, D., and Nearing, G.: HESS Opinions: Never train a Long Short-Term Memory (LSTM) network on a single basin, *Hydrol. Earth Syst. Sci.*, 28, 4187–4201, <https://doi.org/10.5194/hess-28-4187-2024>, 2024.
- Lees, T., Buechel, M., Anderson, B., Slater, L., Reece, S., Coxon, G., and Dadson, S. J.: Benchmarking data-driven rainfall-runoff models in Great Britain: a comparison of long short-term memory (LSTM)-based models with four lumped conceptual models, *Hydrol. Earth Syst. Sci.*, 25, 5517–5534, <https://doi.org/10.5194/hess-25-5517-2021>, 2021.
- Lees, T., Reece, S., Kratzert, F., Klotz, D., Gauch, M., De Bruijn, J., Kumar Sahu, R., Greve, P., Slater, L., and Dadson, S. J.: Hydrological concept formation inside long short-term memory (LSTM) networks, *Hydrol. Earth Syst. Sci.*, 26, 3079–3101, <https://doi.org/10.5194/hess-26-3079-2022>, 2022.
- Liu, J., Koch, J., Stisen, S., Trolldborg, L., Højberg, A. L., Thodsen, H., Hansen, M. F. T., and Schneider, R. J. M.: CAMELS-DK: hydrometeorological time series and landscape attributes for 3330 Danish catchments with streamflow observations from 304 gauged stations, *Earth Syst. Sci. Data*, 17, 1551–1572, <https://doi.org/10.5194/essd-17-1551-2025>, 2025.
- Loritz, R., Dolich, A., Acuña Espinoza, E., Ebeling, P., Guse, B., Götze, J., Hassler, S. K., Hauffe, C., Heidbüchel, I., Kiesel, J., Mälicke, M., Müller-Thomy, H., Stölzle, M., and Tarasova, L.: CAMELS-DE: hydro-meteorological time series and attributes for 1582 catchments in Germany, *Earth Syst. Sci. Data*, 16, 5625–5642, <https://doi.org/10.5194/essd-16-5625-2024>, 2024.
- Mai, J., Shen, H., Tolson, B. A., Gaborit, É., Arsenault, R., Craig, J. R., Fortin, V., Fry, L. M., Gauch, M., Klotz, D., Kratzert, F., O'Brien, N., Princz, D. G., Rasiya Koya, S., Roy, T., Seglenieks, F., Shrestha, N. K., Temgoua, A. G. T., Vionnet, V., and Waddell, J. W.: The Great Lakes Runoff Intercomparison Project Phase 4: the Great Lakes (GRIP-GL), *Hydrol. Earth Syst. Sci.*, 26, 3537–3572, <https://doi.org/10.5194/hess-26-3537-2022>, 2022.
- Moshe, Z., Metzger, A., Elidan, G., Kratzert, F., Nevo, S., and El-Yaniv, R.: HydroNets: Leveraging River Structure for Hydrologic Modeling, *arXiv [preprint]*, <https://doi.org/10.48550/arXiv.2007.00595>, 2020.
- Muhebwa, A., Gleason, C. J., Feng, D., and Taneja, J.: Improving Discharge Predictions in Ungauged Basins: Harnessing the Power of Disaggregated Data Modeling and Machine Learning, *Water Resour. Res.*, 60, e2024WR037122, <https://doi.org/10.1029/2024WR037122>, 2024.
- Muñoz Sabater, J.: ERA5-Land Hourly Data from 1950 to Present, Copernicus Climate Change Service (C3S) Climate Data Store (CDS) [data set], <https://doi.org/10.24381/Cds.E2161bac> (last access: 17 September 2024), 2019.
- Nash, J. E. and Sutcliffe, J. V.: River Flow Forecasting through Conceptual Models Part I – A Discussion of Principles, *J. Hydrol.*, 10, 282–290, [https://doi.org/10.1016/0022-1694\(70\)90255-6](https://doi.org/10.1016/0022-1694(70)90255-6), 1970.
- Nearing, G., Cohen, D., Dube, V., Gauch, M., Gilon, O., Harri-gan, S., Hassidim, A., Klotz, D., Kratzert, F., Metzger, A., Nevo, S., Pappenberger, F., Prudhomme, C., Shalev, G., Shenzis, S., Tekalign, T. Y., Weitzner, D., and Matias, Y.: Global Prediction of Extreme Floods in Ungauged Watersheds, *Nature*, 627, 559–563, <https://doi.org/10.1038/s41586-024-07145-1>, 2024.
- Nevo, S., Morin, E., Gerzi Rosenthal, A., Metzger, A., Barshai, C., Weitzner, D., Voloshin, D., Kratzert, F., Elidan, G., Dror, G., Begelman, G., Nearing, G., Shalev, G., Noga, H., Shavitt, I., Yuk-lea, L., Roy, M., Giladi, N., Peled Levi, N., Reich, O., Gilon, O., Maor, R., Timnat, S., Shechter, T., Anisimov, V., Gigi, Y., Levin, Y., Moshe, Z., Ben-Haim, Z., Hassidim, A., and Matias, Y.: Flood forecasting with machine learning models in an operational framework, *Hydrol. Earth Syst. Sci.*, 26, 4013–4032, <https://doi.org/10.5194/hess-26-4013-2022>, 2022.
- Newman, A. J., Clark, M. P., Sampson, K., Wood, A., Hay, L. E., Bock, A., Viger, R. J., Blodgett, D., Brekke, L., Arnold, J. R., Hopson, T., and Duan, Q.: Development of a large-sample

- watershed-scale hydrometeorological data set for the contiguous USA: data set characteristics and assessment of regional variability in hydrologic model performance, *Hydrol. Earth Syst. Sci.*, 19, 209–223, <https://doi.org/10.5194/hess-19-209-2015>, 2015.
- Otero, N., Horton, P., Martius, O., Allen, S., Zappa, M., Wechsler, T., and Schaeffli, B.: Impacts of Hot-Dry Conditions on Hydropower Production in Switzerland, *Environ. Res. Lett.*, 18, 064038, <https://doi.org/10.1088/1748-9326/acd8d7>, 2023.
- Shalev, G., El-Yaniv, R., Klotz, D., Kratzert, F., Metzger, A., and Nevo, S.: Accurate Hydrologic Modeling Using Less Information, *arXiv [preprint]*, <https://doi.org/10.48550/arXiv.1911.09427>, 2019.
- Shen, C., Laloy, E., Elshorbagy, A., Albert, A., Bales, J., Chang, F.-J., Ganguly, S., Hsu, K.-L., Kifer, D., Fang, Z., Fang, K., Li, D., Li, X., and Tsai, W.-P.: HESS Opinions: Incubating deep-learning-powered hydrologic science advances as a community, *Hydrol. Earth Syst. Sci.*, 22, 5639–5656, <https://doi.org/10.5194/hess-22-5639-2018>, 2018.
- Sit, M., Demiray, B., and Demir, I.: Short-Term Hourly Streamflow Prediction with Graph Convolutional GRU Networks, *arXiv [preprint]*, <https://doi.org/10.48550/arXiv.2107.07039>, 2021.
- Sitterson, J., Knightes, C., Parmar, R., Wolfe, K., Avant, B., and Muche, M.: An Overview of Rainfall-Runoff Model Types, 9th International Congress on Environmental Modelling and Software Fort Collins, Colorado, USA, edited by: Arabi, M., David, O., Carlson, J., and Ames, D. P., <https://scholarsarchive.byu.edu/iemssconference/2018/> (last access: 5 June 2025), 2018.
- Smith, A., Tetzlaff, D., Marx, C., and Soulsby, C.: Enhancing Urban Runoff Modelling Using Water Stable Isotopes and Ages in Complex Catchments, *Hydrol. Process.*, 37, e14814, <https://doi.org/10.1002/hyp.14814>, 2023.
- Smith, J. and Eli, R. N.: Neural-Network Models of Rainfall-Runoff Process, *Journal of Water Resources Planning and Management*, 121, 499–508, [https://doi.org/10.1061/\(ASCE\)0733-9496\(1995\)121:6\(499\)](https://doi.org/10.1061/(ASCE)0733-9496(1995)121:6(499)), 1995.
- Sterle, G., Perdrial, J., Kincaid, D. W., Underwood, K. L., Rizzo, D. M., Haq, I. U., Li, L., Lee, B. S., Adler, T., Wen, H., Middleton, H., and Harpold, A. A.: CAMELS-Chem: augmenting CAMELS (Catchment Attributes and Meteorology for Large-sample Studies) with atmospheric and stream water chemistry data, *Hydrol. Earth Syst. Sci.*, 28, 611–630, <https://doi.org/10.5194/hess-28-611-2024>, 2024.
- Sun, A. Y., Jiang, P., Mudunuru, M. K., and Chen, X.: Explore Spatio-Temporal Learning of Large Sample Hydrology Using Graph Neural Networks, *Water Resour. Res.*, 57, e2021WR030394, <https://doi.org/10.1029/2021WR030394>, 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, *Hydrol. Earth Syst. Sci.*, 26, 5163–5184, <https://doi.org/10.5194/hess-26-5163-2022>, 2022.
- Tursun, A., Xie, X., Wang, Y., Peng, D., Liu, Y., Zheng, B., Wu, X., and Nie, C.: Streamflow Prediction in Human-Regulated Catchments Using Multiscale Deep Learning Modeling With Anthropogenic Similarities, *Water Resour. Res.*, 60, e2023WR036853, <https://doi.org/10.1029/2023WR036853>, 2024.
- Van Der Knijff, J. M., Younis, J., and De Roo, A. P. J.: LIS-FLOOD: A GIS-based Distributed Model for River Basin Scale Water Balance and Flood Simulation, *International Journal of Geographical Information Science*, 24, 189–212, <https://doi.org/10.1080/13658810802549154>, 2010.
- Vischer, M., Felipe, N. O., and Ma, J.: Spatially Resolved Meteorological and Ancillary Data in Central Europe for Rainfall Streamflow Modeling, *HydroShare [data set]*, <https://doi.org/10.4211/hs.05d5633a413b4aec93b08a7e61a2abbb>, 2025a.
- Vischer, M. A., Otero, N., and Ma, J.: Spatially Resolved Meteorological and Ancillary Data in Central Europe for Rainfall Streamflow Modeling, *arXiv [preprint]*, <https://doi.org/10.48550/arXiv.2506.03819>, 2025b.
- Xiang, Z. and Demir, I.: Fully Distributed Rainfall-Runoff Modeling Using Spatial-Temporal Graph Neural Network, *Earth ArXiv [data set]*, <https://doi.org/10.31223/X57P74>, 2022.
- Xie, J., Liu, X., Tian, W., Wang, K., Bai, P., and Liu, C.: Estimating Gridded Monthly Baseflow From 1981 to 2020 for the Contiguous US Using Long Short-Term Memory (LSTM) Networks, *Water Resour. Res.*, 58, e2021WR031663, <https://doi.org/10.1029/2021WR031663>, 2022.
- Yu, Q., Tolson, B. A., Shen, H., Han, M., Mai, J., and Lin, J.: Enhancing long short-term memory (LSTM)-based streamflow prediction with a spatially distributed approach, *Hydrol. Earth Syst. Sci.*, 28, 2107–2122, <https://doi.org/10.5194/hess-28-2107-2024>, 2024.
- Zhou, F., Chen, Y., and Liu, J.: Application of a New Hybrid Deep Learning Model That Considers Temporal and Feature Dependencies in Rainfall-Runoff Simulation, *Remote Sens.*, 15, 1395, <https://doi.org/10.3390/rs15051395>, 2023.
- 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, *J. Hydrol.*, 616, 128727, <https://doi.org/10.1016/j.jhydrol.2022.128727>, 2023.