Hydrol. Earth Syst. Sci., 29, 6221–6235, 2025
https://doi.org/10.5194/hess-29-6221-2025

Hydrology and Earth System Sciences
Open Access
EGU

How to deal w___ missing input data

Martin Gauch[1], **Frederik Kratzert**[2], **Daniel Klotz**[2,3], **Grey Nearing**[1], **Deborah Cohen**[4], **and Oren Gilon**[4]

[1]Google Research, Zurich, Switzerland
[2]Google Research, Vienna, Austria
[3]IT:U Interdisciplinary Transformation University, Linz, Austria
[4]Google Research, Tel Aviv, Israel

Invited contribution by Martin Gauch, recipient of the EGU Hydrological Sciences Virtual Outstanding Student and PhD candidate Presentation Award 2021.

Correspondence: Martin Gauch (gauch@google.com)

Received: 14 March 2025 – Discussion started: 7 April 2025
Revised: 14 August 2025 – Accepted: 15 October 2025 – Published: 13 November 2025

Abstract. Deep learning hydrologic models have made their way from research to applications. More and more national hydrometeorological agencies, hydro power operators, and engineering consulting companies are building Long Short-Term Memory (LSTM) models for operational use cases. All of these efforts come across similar sets of challenges – challenges that are different from those in controlled scientific studies. In this paper, we tackle one of these issues: how to deal with missing input data? Operational systems depend on the real-time availability of various data products – most notably, meteorological forcings. The more external dependencies a model has, however, the more likely it is to experience an outage in one of them. We introduce and compare three different solutions that can generate predictions even when some of the meteorological input data do not arrive in time, or not arrive at all: First, *input replacing*, which imputes missing values with a fixed number; second, *masked mean*, which averages embeddings of the forcings that are available at a given time step; third, *attention*, a generalization of the masked mean mechanism that dynamically weights the embeddings. We compare the approaches in different missing data scenarios and find that, by a small margin, the masked mean approach tends to perform best.

1 Introduction

Deep learning approaches for hydrologic modeling are now making their way from research settings into real-world operational deployments (e.g., Nearing et al., 2024; Frame et al., 2025; Read et al., 2021; Franken et al., 2022). Unfortunately, the real world is messy and in many ways does not conform to the controlled settings we can assume in research studies (Mitchell and Jolley, 1988). One prime example for such complications is the occurrence of outages with input data products: state-of-the-art operational hydrologic models rely on the real-time availability of several externally provided meteorological forcing products. As an example, the hydrologic model in Google's flood forecasting system uses four different weather data products from four different data providers as inputs (Cohen, 2024). At any point in time, one or more of these providers might experience an outage and not deliver the data in time to make the next prediction. Where the timely arrival of data is usually not an issue in research contexts, not producing forecasts for days or even weeks is not an option for operational systems that are needed for flood forecasts or water management.

Moreover, models that can cope with missing input data are useful in other settings, such as training on data products that are available for different time periods or different spatial extents: the observation that larger and more diverse training sets generally benefit the prediction quality (Kratzert et al., 2024) appears at odds with the fact that local meteorological forcings tend to have higher resolution and be more accurate than global ones (Clerc-Schwarzenbach et al., 2024).

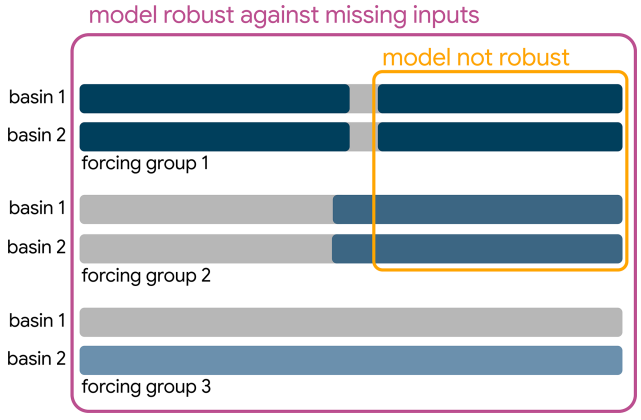

Figure 1. Different scenarios for missing input data (gray bars): outages at individual time steps (top), data products starting at different points in time (middle), and local data products that are not available for all basins (bottom). All of these scenarios reduce the number of training samples for models that are *not* robust, i.e., that cannot cope with missing data (yellow, small box), while the models presented in this paper *are* robust, i.e., they can be trained on all samples with valid targets (purple, large box).

Our proposed methods can mitigate this tension, as they allow us to train a single global model that incorporates local forcings where they are available (Fig. 1). Orthogonally to spatial coverage, our methods further allow us to train models with forcings that have different temporal coverage. This is especially useful for more recent data products based on remote sensing information.

Inevitably, the quality of predictions degrades as fewer input data products are available (Kratzert et al., 2021). Fortunately, deep learning methods are flexible enough to offer solutions that limit this decay while remaining competitively accurate when all data are available. In the following sections, we present three strategies to accomplish this goal:

- First, *input replacing* replaces missing forcing data with a fixed value and adds binary flags to indicate outages.
- Second, *masked mean* embeds each forcing product separately and averages the embeddings of all products that are available at a given time.
- Third, we show how the masked mean strategy is a special case of a theoretically more expressive but practically equally accurate *attention* mechanism (Bahdanau et al., 2015) that can dynamically adjust the weighting of each forcing product, e.g., depending on the static attributes of a basin.

We evaluate these strategies in three settings:

- First, random *time step dropout*. We investigate how accuracy deteriorates as forcings are missing at more and more time steps during training and inference (corresponding to the top row in Fig. 1).
- Second, *sequence dropout*. We investigate how accuracy deteriorates as certain forcings become entirely unavailable during inference (corresponding to the middle row in Fig. 1).
- Third, *regional forcing products*. We investigate how the proposed strategies allow training global models that leverage regional forcing data (corresponding to the bottom row in Fig. 1).

We are not the first to study deep learning models that are robust to missing input data (Afifi and Elashoff, 1966). In fact, today's large language models rely heavily on learning schemes that train the model to predict words given incomplete and masked-out input sentences (e.g., Devlin et al., 2019; Raffel et al., 2020; Brown et al., 2020). These masked language models use special mask tokens to indicate dropped-out data, which – at a high level – are similar to the binary indicators we use in the input replacing strategy. Similar techniques are used in computer vision models, such as Masked Autoencoders (He et al., 2022). Srivastava et al. (2014) highlight an additional benefit of dropping out inputs (or hidden activations) during training: dropout has a regularizing effect on training and therefore reduces overfitting and leads to models that generalize better.

Data-driven methods are also used to explicitly impute missing data (e.g., Schafer, 1997; Wu et al., 2020), including in hydrological and meteorological applications (e.g., Gao et al., 2018; Yozgatligil et al., 2013). Imputation subsequently allows using models that cannot cope with missing data. However, this strategy requires an additional imputation model that needs to be trained separately or jointly with the downstream model, making the setup and training more complex. As we are less focused on the reconstruction of missing data and more focused on maintaining prediction accuracy, we do not consider such approaches in this study.

Our masked mean and attention mechanisms also bear similarity to deep learning approaches that merge multimodal input data, such as LANISTR (Ebrahimi et al., 2023). Their approach merges inputs from different modalities (such as images, text, or structured data) into a joint embedding space, while allowing individual modalities to be missing at training or inference time. Further, the attention mechanism's dynamic weighting of forcing embeddings can be seen as a variant of the conditioning operation described by Perez et al. (2018) and at a higher level by Dumoulin et al. (2018).

2 Data and methods

2.1 Data

We ran all experiments on the 531 basins of the CAMELS dataset (Newman et al., 2015; Addor et al., 2017) that previous studies used, e.g., Kratzert et al. (2021). The CAMELS dataset comes with three sets of daily meteorological forc-

ings: Daymet (Thornton et al., 1997), Maurer (Maurer et al., 2002), and NLDAS (Xia et al., 2012). We consider these forcings the "external dependencies" in this study. All models use all 15 forcing variables (precipitation, solar radiation, min/max temperature, and vapor pressure for each of the three forcing products) and the same set of 26 static attributes as Kratzert et al. (2021)[1]. All models are trained with streamflow as the target variable.

Again following Kratzert et al. (2021), we trained our models on the period 1 October 1999 to 30 September 2008, validated on 1 October 1980 to 30 September 1989, and tested them on 1 October 1989 to 30 September 1999. All results in this paper refer to the test period.

2.2 Methods

The models we train in this paper closely follow the architecture that was used in Kratzert et al. (2021), except that we employ different mechanisms to feed the input data into the LSTM itself. The following paragraphs describe these approaches in more detail.

2.2.1 Input replacing

The first mechanism to cope with missing input data sets any missing values to a fixed value and adds a binary flag to indicate these replacements, before concatenating all input data and flags (Fig. 2; see also Nearing et al., 2024). Optionally, we can embed the concatenated vector (in our case, through a small fully-connected network). This reduces the feature dimensions before the vector is finally used as input to the LSTM. Further, we can concatenate a positional encoding vector to the forcings before the embedding, making the model aware of the current input's position relative to the overall sequence length (not shown in Fig. 2). In initial experiments, we also tried to make the replacement value a learned parameter instead of setting it to a fixed value, but we did not see meaningful improvements when doing so. Hence, in all subsequent experiments, we used zero as the fixed value.

2.2.2 Masked mean

This approach embeds the forcings of each provider through individual embedding networks, each of them yielding an embedding vector of the same size. At every input time step, we average the non-NaN embeddings of that time step (i.e., the embeddings that correspond to providers that were available at that time step; hence the name "masked mean") and pass the resulting joint embedding on to the LSTM (Fig. 3). The inputs to the embedding networks could be extended by additional features, such as the static catchment attributes. However, in our experiments we found that this deteriorated the performance. The flood forecasting system described by Cohen (2024) uses a masked mean approach in the current operational model.

2.2.3 Attention

Readers who are familiar with deep learning might recognize the masked mean architecture as the simplification of a more general attention mechanism (Bahdanau et al., 2015). Attention mechanisms have become ubiquitous in deep learning, as they are the core component of the popular Transformer architecture (Vaswani et al., 2017). The most common realizations of attention allow the model to dynamically adjust its focus on different input time steps. Appendix D provides a brief introduction to the concept of attention for readers who are not familiar with the topic.

In our case, we apply the attention mechanism over the different available providers at each time step. Figure 4 illustrates the process. Similar to the masked mean approach, we embed each forcing with its own embedding network, resulting in vectors that we use as both the keys and values of the attention mechanism. Additionally, we concatenate the static attributes with a positional embedding of the input time step (not shown in Fig. 4 for brevity) and three binary flags that indicate the availability of each forcing product at the given time step. A separate embedding of the resulting concatenated vector acts as the query. Based on the similarity of the query and each of the key vectors, we obtain a weighting by which we average the values, i.e., the embedding vectors of the forcing products. This weighted average is the input to the LSTM. Hence, the attention mechanism, could – at least in theory – learn to dynamically adjust its focus on each forcing product based on the basin it is asked to predict.

2.3 Experiments

We conducted three experiments to test how well each architecture can cope with different scenarios where input data are missing in certain temporal periods or spatial regions. To save computational resources, we performed one hyperparameter tuning and used the resulting best hyperparameters for all further experiments. Appendix A covers our tuning procedure in more detail. In all experiments, we trained each model with three different random seeds.

2.3.1 Experiment 1: Forcings missing at individual time steps

This experiment simulates short-term outages of certain input products. Because the LSTMs used in hydrologic applications typically ingest input data with one year (365 d) of lookback, even an outage for a single time step can cause problems for the next year to come: for the next 365 d, there will be a NaN input time step, which breaks models that cannot deal with missing input data. We trained and evaluated the different models with an increasing probability of ran-

[1] Unlike what is mentioned in Kratzert et al. (2021), p_seasonality was actually not used as a static input, as the experiment configuration files show.

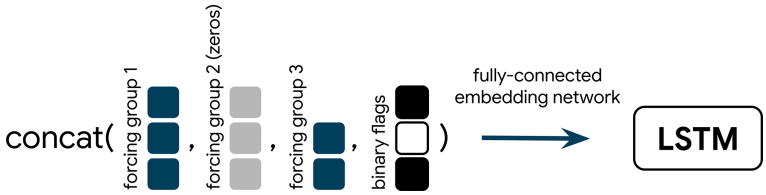

Figure 2. Illustration of the input replacing strategy. Each box represents an input variable (like precipitation, temperature) from one of the forcing groups. NaNs in the input data for a given time step are replaced by zeros (gray boxes for forcing group 2), all forcings are concatenated, together with one binary flag for each forcing group which indicates whether that group was NaN or not. The resulting vector is passed through an embedding network to the LSTM.

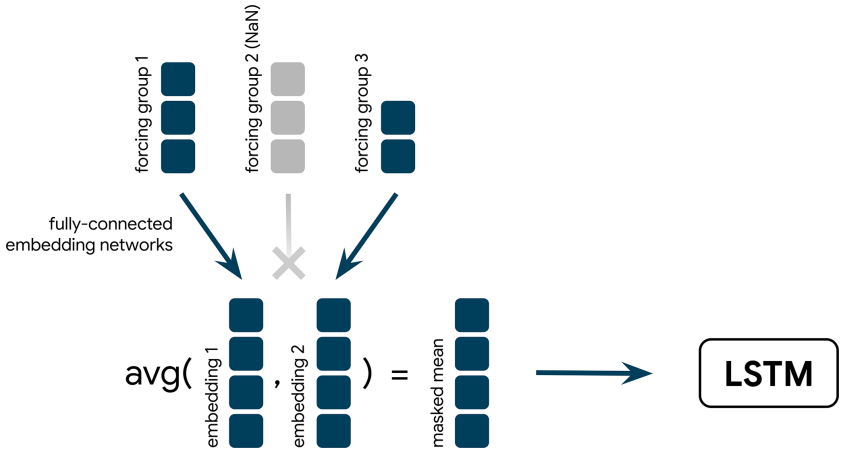

Figure 3. Illustration of the masked mean strategy. Each forcing provider is projected to the same size through its own embedding network. The resulting embeddings of valid providers are averaged and passed on to the LSTM.

domly missing input time steps. The time step dropout is sampled independently at random, i.e., at each input time step, each forcing is missing with probability p_{time}. This means that all products can be missing at once for certain time steps. We sweep p_{time} from 0 to 0.6 in increments of 0.1.

As baselines, we used the three-forcing model from Kratzert et al. (2021). This shows the upper bound of performance we can expect when no data are missing. We also included the worst of the three single-forcing models (based solely on NLDAS) from the same source as a point of reference.

2.3.2 Experiment 2: Forcings missing for the entire time sequence

This experiment simulates extended time periods with missing input data. In practical applications, this may happen when an input product has limited temporal coverage, either because it became available later than other products, or because it went out of service or had an extended outage while the model was still in use. We evaluated this scenario by running inference with samples where all time steps of one or two providers were set to NaN, and we report the results for each combination of one or two missing providers.

To make sure the models can cope with this scenario, we trained the models with samples that contained NaNs of two types: (1) dropout of individual time steps (as in the previous experiment) with $p_{\text{time}} = 0.1$, and (2) dropout of entire input sequences with $p_{\text{sequence}} = 0.1$[2]. We made sure to never drop all three sequences entirely, but allowed the case where all three products are missing at individual time steps.

The natural baselines in these experiments are the corresponding one- and two-forcing models from Kratzert et al. (2021). These baselines are not robust to missing input data, and they simply ingest the concatenated forcing variables from one or two forcing groups.

2.3.3 Experiment 3: Forcings missing for certain spatial regions

Finally, we explored how the different approaches to missing input data fare in settings where an input product is missing for certain regions in space. This is relevant because for many regions there exist local meteorological data products that are of higher quality than globally available ones. At the same

[2]We also performed some preliminary experiments with $p_{\text{time}} = 0.0$, $p_{\text{sequence}} = 0.1$ since this more closely matches the evaluation setup, but saw no meaningful differences in the results.

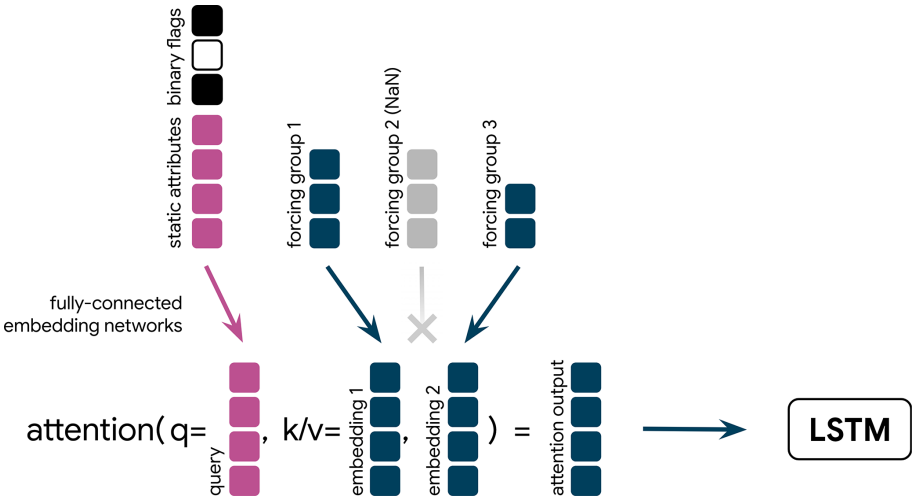

Figure 4. Illustration of the attention embedding strategy. Each forcing provider is projected to the same size through its own embedding network. The resulting embedding vectors become the keys (k) and values (v). The static attributes, together with a binary flag for each provider, serve as the query. The attention-weighted average of embeddings is passed on to the LSTM.

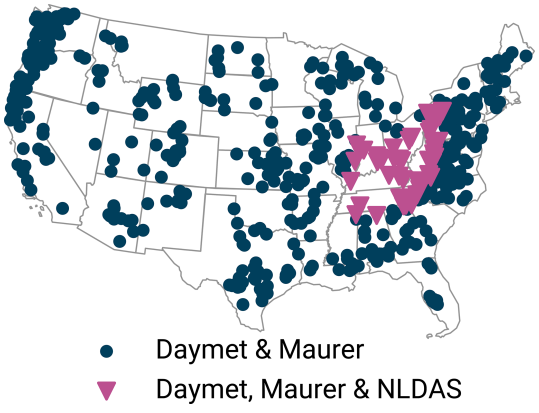

Figure 5. Map of the 531 CAMELS basins used in this study. For the 51 basins in the Ohio, Cumberland, and Tennessee River basins (purple), we assumed all three forcing to be available. For all other basins (blue), we assumed only Daymet and Maurer forcings to be available.

time, training on diverse sets of basins benefits performance (see Kratzert et al., 2024). Hence, being able to merge local high-quality forcing data with global streamflow could – at least in theory – combine the best of two worlds.

We simulated this setting on the CAMELS dataset by training models that received Daymet and Maurer forcings everywhere, but NLDAS forcings only for the 51 basins in the Ohio, Cumberland, and Tennessee River basins (USGS site numbers starting in 03, cf. Wells, 1960, depicted in Fig. 5). As baselines, we trained a model on all three forcings but only the 51 basins, and a model on all 531 basins but only the two forcings that we assumed as available anywhere (Daymet and Maurer).

3 Results

3.1 Experiment 1: Forcings missing at individual time steps

In the first experiment, we trained models at different probabilities p_{time} of input products being NaN at individual time steps. Figure 6 shows the resulting Nash–Sutcliffe efficiency (NSE; Nash and Sutcliffe, 1970) and Kling–Gupta efficiency (KGE; Gupta et al., 2009) values at $p_{\text{time}} = 0.0, 0.1, \ldots, 0.6$, and Appendix C contains plots with additional metrics. As expected, the accuracy of all methods drops with increasing amounts of NaNs. At 0 % NaNs, all methods perform roughly as good as the three-forcing baseline from Kratzert et al. (2021), which cannot cope with missing input data. The models exhibit slightly worse NSE values than the baseline, while masked mean and input replacing are slightly better in KGE. These minor differences arise because our newly trained models were tuned for a setting with moderate amounts of missing input data and therefore use slightly different hyperparameters than the three-forcings baseline.

As p_{time} increases, we see no clear winner in terms of NSE; all methods decay by roughly equal amounts in this metric. For KGE, the masked mean architecture tends to perform better than input replacing and attention: except for $p_{\text{time}} = 0.2$, the masked mean results are significantly better than those of input replacing (one-sided Wilcoxon signed-rank test at $\alpha = 0.05$). The attention mechanism generally performs significantly worse than masked mean and input replacing, except at the highest missing data probabilities. To investigate why attention under-performs at low p_{time}, we plotted the attention weights placed by the model on each set of forcings (Fig. C3 in Appendix C), and found that, apart from a select few basins, the weights fluctuate closely around

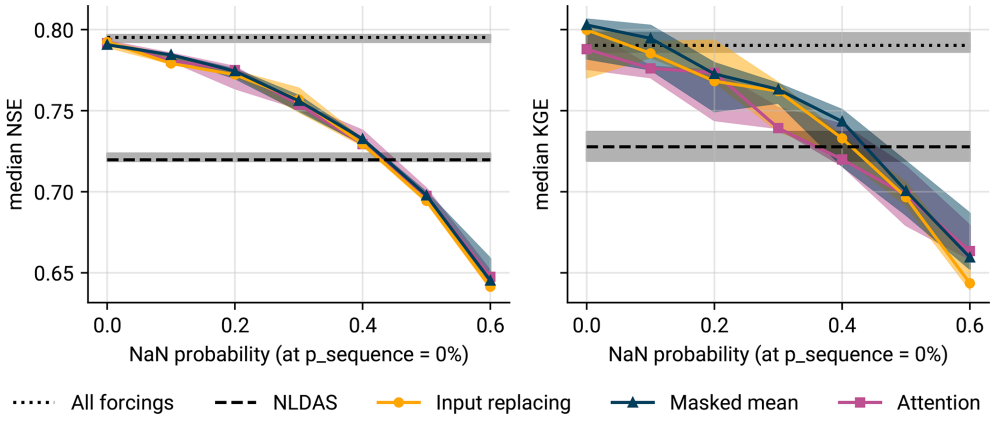

Figure 6. Median NSE and KGE across 531 basins at different amounts of missing input time steps. The dotted horizontal line provides the baseline of a model that cannot deal with missing data but is trained to ingest all three forcing groups at every time step. The dashed line represents the baseline of a model that uses the worst individual set of forcings (NLDAS). Both baselines stem from Kratzert et al. (2021). The shaded areas indicate the spread between minimum and maximum values across three seeds; the solid lines represent the median.

1/3. Hence, the model merely attempted to recover the solution that is hard-coded in the masked mean strategy.

3.2 Experiment 2: Forcings missing for the entire time sequence

In this experiment, we evaluated to what extent the different architectures can maintain their accuracy when one or two sets of forcings are missing entirely at inference time. Figure 7 shows the resulting empirical cumulative distribution functions (CDFs) of NSE values. Kratzert et al. (2021) already provide results which indicate that the availability of fewer forcing products implies worse model performance.

The results from experiment 2 corroborate this finding. In the experiments where one set of forcings is available at inference time (first column in Fig. 7), the baseline trained on that one set of forcings (dashed line) performs significantly better than the missing-inputs architectures, but the effect sizes in the comparison to masked mean and attention are small (Cohen's $d < 0.1$). The only exception to this is the NLDAS-only experiment, where the baseline does not perform significantly better than masked mean and attention. Input replacing tends to perform the worst across all evaluations.

In the experiments where two sets of forcings are available at inference time (second column in Fig. 7), we find similar results as in the experiments with one missing set of forcings. However, the margins in accuracy between the different methods are even smaller and likely not relevant for most practical applications.

3.3 Experiment 3: Forcings missing for certain spatial regions

The last experiment investigated how well the missing-input architectures can incorporate regional input data, i.e., forcings that are available only in a subset of the training basins. Figure 8 shows the resulting empirical CDF curves of NSE and KGE values, and Appendix C provides figures with additional metrics.

Masked mean, attention, and input replacing all improve the predictions when compared to the globally trained two-forcing model. The three-forcing regional model trained only on the 51 basins in the Ohio, Cumberland, and Tennessee River basins is significantly better than input replacing and attention, but not significantly better than masked mean (one-sided Wilcoxon signed-rank test, $\alpha = 0.05$). This pattern is similar for the additional metrics from Appendix C. However, from a practical hydrological perspective, all approaches perform quite similar, despite the statistical significance.

4 Discussion and conclusions

In this study, we presented three different strategies to build models that can provide streamflow predictions when parts of the meteorological input data are missing. *Input replacing* replaces NaNs with a fixed value, concatenates all forcings, and adds binary flags to indicate the missing data. *Masked mean* embeds each forcing product separately and averages the embeddings of available forcings. Finally, *attention* generalizes the masked mean approach and dynamically calculates a weighting of the different embeddings. Across all experiments (missing individual time steps, missing sequences, regional forcings), the masked mean strategy tends to perform best, although the differences are often small and depend on metrics. The fact that the models are unable to outperform the baseline trained on all three forcings but only 51 local basins (experiment 3) lets us conjecture that the high-quality CAMELS forcings may not be the ideal testbed for an evaluation of regional forcings. All three forcings

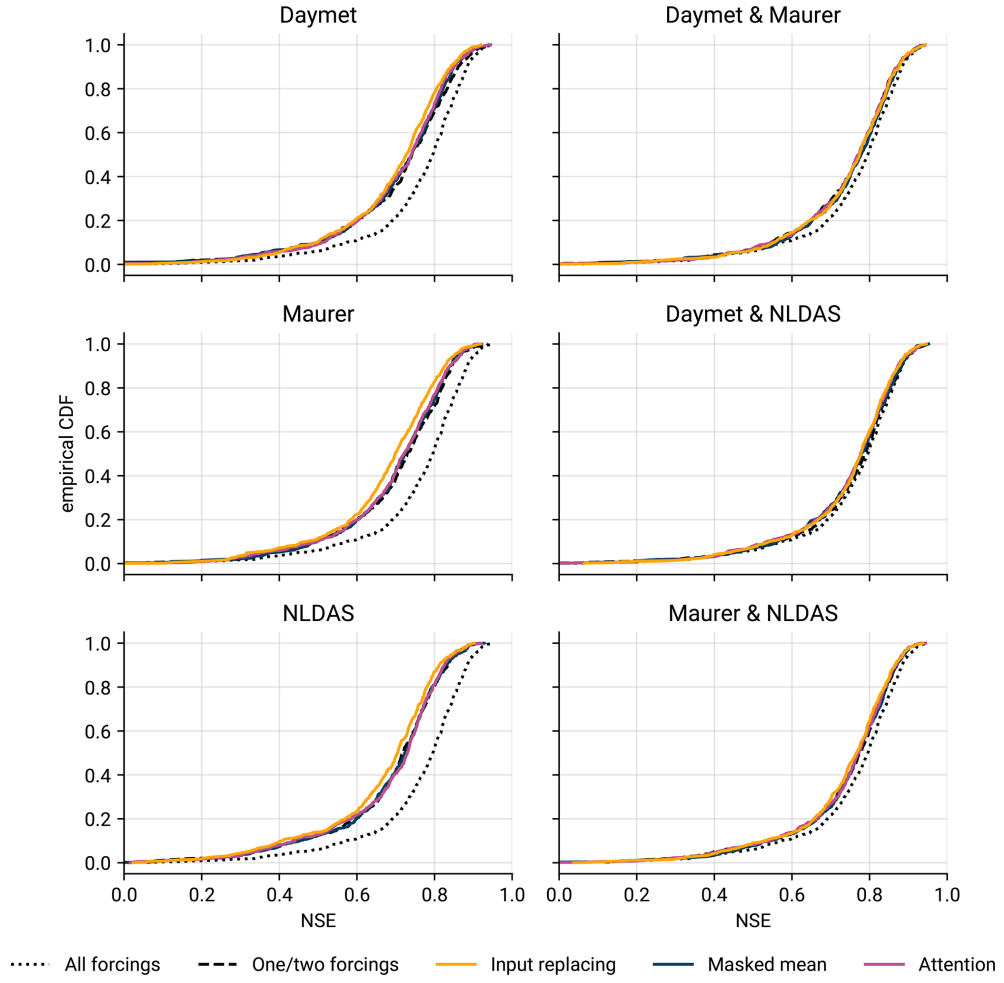

Figure 7. Empirical cumulative distribution functions of NSE values across all 531 basins when two (first column) or one (second column) forcing groups are continuously missing. The subplot titles denote which products we passed to the model during inference. The dotted line represents the upper bound baseline, a model that is trained and evaluated with all three forcings; the dashed line represents the performance of a model trained specifically for the available combination of forcings. All results show the mean performance across three seeds; curves further to the right are better.

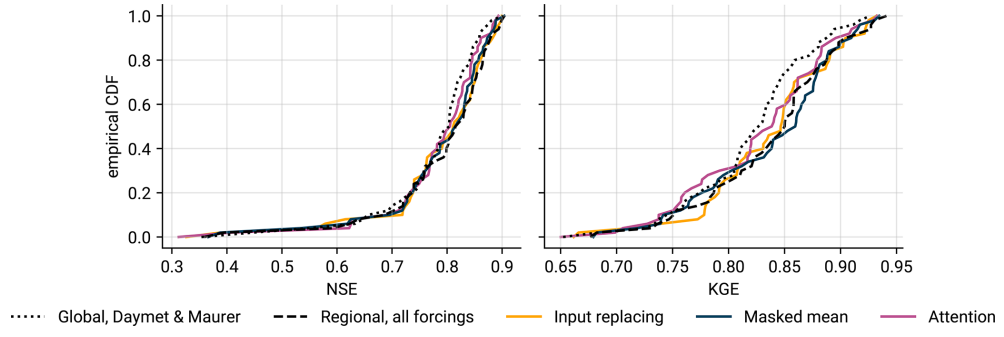

Figure 8. Empirical CDFs of NSE values across the 51 basins of the Ohio, Cumberland, and Tennessee River basins. The dashed line represents the baseline model trained only on those basins but with all forcings. The dotted line is the baseline two-forcing model trained on all 531 basins. The other models are trained on all 531 basins with NLDAS set to NaN outside of the 51 basins. All results show the mean performance across three seeds.

are of similar quality and the basins in the chosen region are comparably similar and easy to predict, hence, a rather small set of training gauges appears to already yield satisfactory predictions and it becomes difficult to discern meaningful differences. We therefore hypothesize that evaluations on larger datasets and with forcings of more varied quality would yield clearer conclusions. Unfortunately, these larger datasets are still missing the type of widely accepted baseline models and state-of-the-art LSTM configurations that exist for CAMELS. Hence, for this study we chose to stick with the CAMELS dataset in order to maintain consistency with Kratzert et al. (2021) and to allow for easy reproduction of experiments with limited resources. We see great potential for future work that extends the experiment to such settings.

Notably, the attention mechanism – despite being strictly more expressive than the masked mean strategy – does not improve upon these results and largely learns to recover the masked mean solution. We also experimented with analyzing the attention weights grouped by time steps with falling/rising streamflow or by the forcing whose precipitation deviated the furthest from the mean, but could not identify any patterns (results not shown). Therefore, in its current form, attention appears unnecessary. Nevertheless, we do encourage further work in this direction as our experiments do not fully exhaust the space of possible attention configurations, and we hypothesize that attention might play to its strengths especially in settings where the quality of inputs varies significantly across forcings, space, or time. Extending the scope beyond established baselines, future work could evaluate this, for example, with the new Caravan MultiMet dataset (Shalev and Kratzert, 2024). Caravan MultiMet provides forcings from seven different providers for all basins in the Caravan dataset and its extensions (Kratzert et al., 2023). There are also many alternative approaches to calculating query, keys, and values: e.g., incorporating the forcing information also into the query vector or incorporating static information into the keys and values.

Lastly, we would like to look at the presented strategies from a different perspective: we can view them as means to *inject* additional data into a model. Such injections can happen already during training (the multiple forcings we use in our experiments are an example for this), but they could also happen after training: for example, hydromet agencies could download a publicly available global model and inject locally available forcings or even lagged observations into the model. We encourage exploring such approaches further, as they could alleviate current trade-offs between training set size and input data resolution.

Appendix A: Hyperparameter tuning

All hyperparameter tuning experiments used $p_{\text{time}} = p_{\text{sequence}} = 0.1$. We chose these values as an intermediate level of missing data to avoid the computational expense of tuning each architecture for each experiment setup separately. As we built upon the established baselines from Kratzert et al. (2021), we did not tune the LSTM architecture itself for the experiments in this paper. Hence, all LSTMs are trained with 365 daily input time steps, a hidden size of 256, batch size 256, dropout fraction of 0.4 on the output head, and an Adam optimizer with initial learning rate of 1×10^{-3}, which we lowered to 5×10^{-4} in epoch 10 and to 1×10^{-4} in epoch 25. We used the NSE* loss function from Kratzert et al. (2019). For a more in-depth description of these settings, we refer to Kratzert et al. (2019).

We did, however, tune the hyperparameters of the missing-inputs mechanisms as well as the number of training epochs. For input replacing configurations, we chose slightly larger embedding sizes, such that the total parameter count in input replacing configurations is roughly equal to the parameter count in masked mean configurations. Attention configurations are marginally larger as they have an additional query embedding network, but we consider this difference irrelevant for the results in our comparisons – especially given that the optimal attention configuration was not the largest one in the hyperparameter grid.

We performed a grid search of the hyperparameter combinations listed in Table A1. As for the main experiments, we trained each combination with three different random seeds. Finally, we chose the best configuration for each architecture as the one with the best median NSE value across all basins in the validation period, averaged across seeds. Table A2 lists the best configuration for each architecture.

Table A1. Hyperparameter tuning grid.

Hyperparameter		Values
Embedding hidden layer sizes (ReLU-activated)	Input replacing	[5], [7, 5], [17, 10], [17, 17, 10], [17, 17, 17, 10]
	Masked mean	[5], [5, 5], [10, 10], [10, 10, 10], [10, 10, 10, 10]
	Attention*	[10, 10], [10, 10, 10], [10, 10, 10, 10]
Positional encoding size		0, 5
Number of attention heads		1, 2, 5
Evaluated epochs		5, 10, 15, 20, 25, 30, 35, 40

* We excluded configurations with hidden size 5, because the final embedding size must be divisible by the number of attention heads.

Table A2. Best hyperparameter configurations based on validation period results.

Architecture	Embedding hidden layer sizes	Positional encoding size	Number of attention heads	Epoch
Input replacing	[17, 10]	5	–	30
Masked mean	[10, 10, 10, 10]	0	–	35
Attention	[10, 10, 10]	5	1	30

Appendix B: Computational resources

We conducted all experiments on Nvidia P100 GPU machines running Python 3.11 and NeuralHydrology 1.11.0 (with local modifications that are part of the 1.12.0 release). In total, including preliminary experiments, hyperparameter tuning, and final experiments, we trained approximately 800 models. This amounts to approximately 286 wall-time computation days (measuring the time from writing the configuration to disk to the last Tensorboard update). We did not spend any effort optimizing the runtime of these jobs; many runs could have been sped up significantly, e.g., through increased parallelism in data loading.

Appendix C: Additional figures

In consideration of the fact that no single metric adequately captures the quality of a model (Gauch et al., 2023), we provide Fig. C1 as an extended version of Fig. 6 (showing the performance with increasing number of NaN inputs for a variety of additional metrics). Further, Fig. C2 extends Fig. 8 and shows empirical CDFs of the experiment with regional forcings for additional metrics. We refer to Gauch et al. (2023) for the definitions of these measures.

Lastly, Fig. C3 shows the fractional attention to each forcing product for three models trained with different random seeds (see experiment 1).

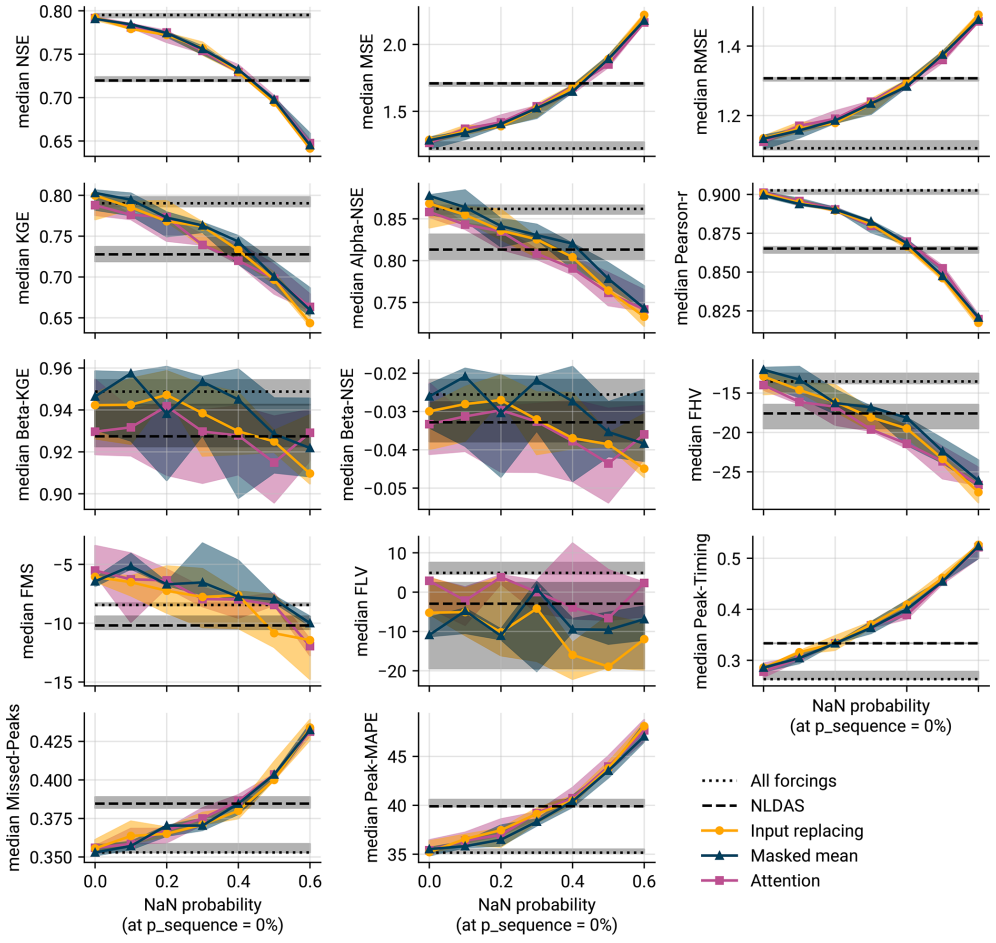

Figure C1. Extended version of Fig. 6, showing additional metrics (see Gauch et al., 2023 for the definitions of these metrics).

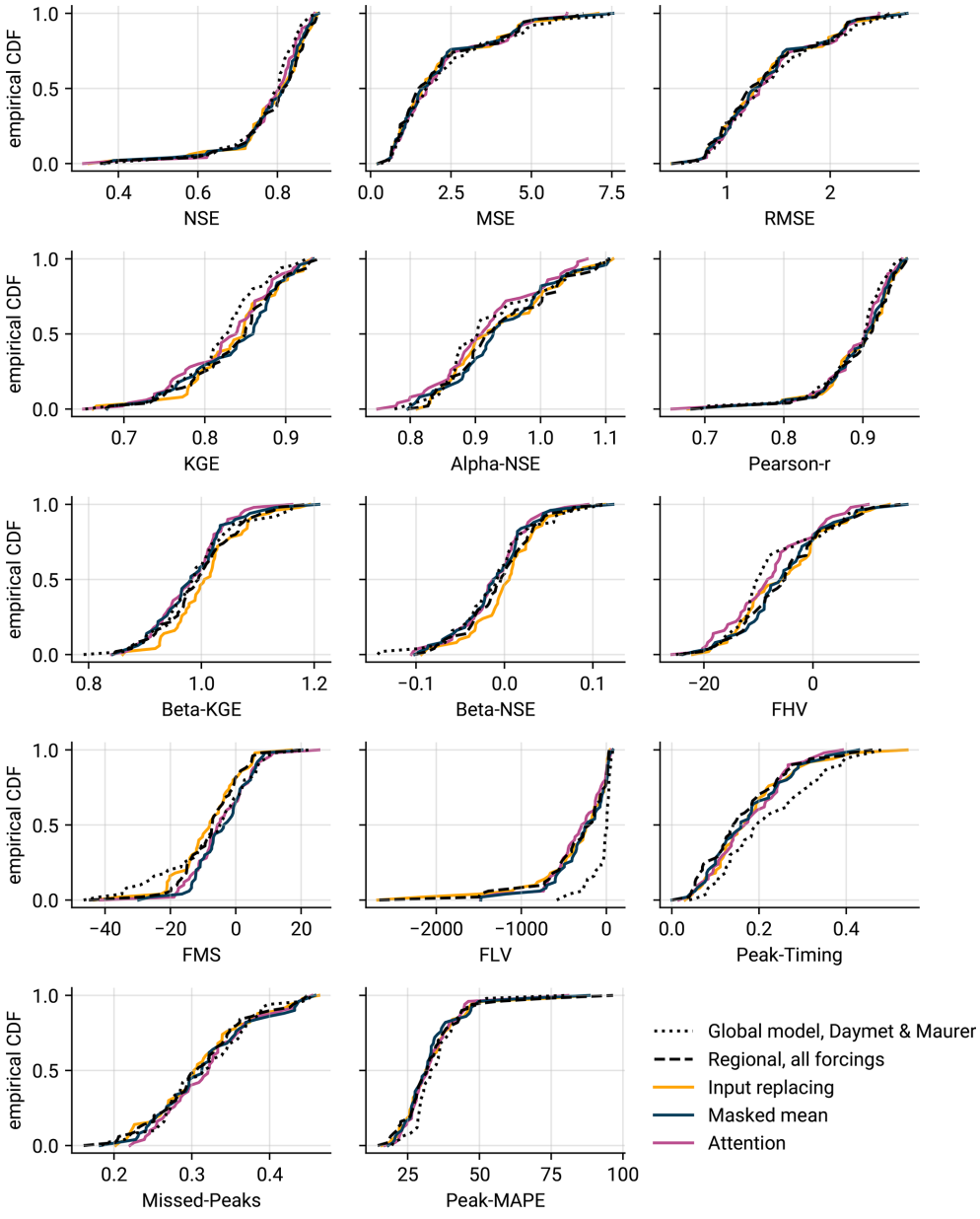

Figure C2. Extended version of Fig. 8, showing additional metrics (see Gauch et al., 2023 for the definitions of these metrics).

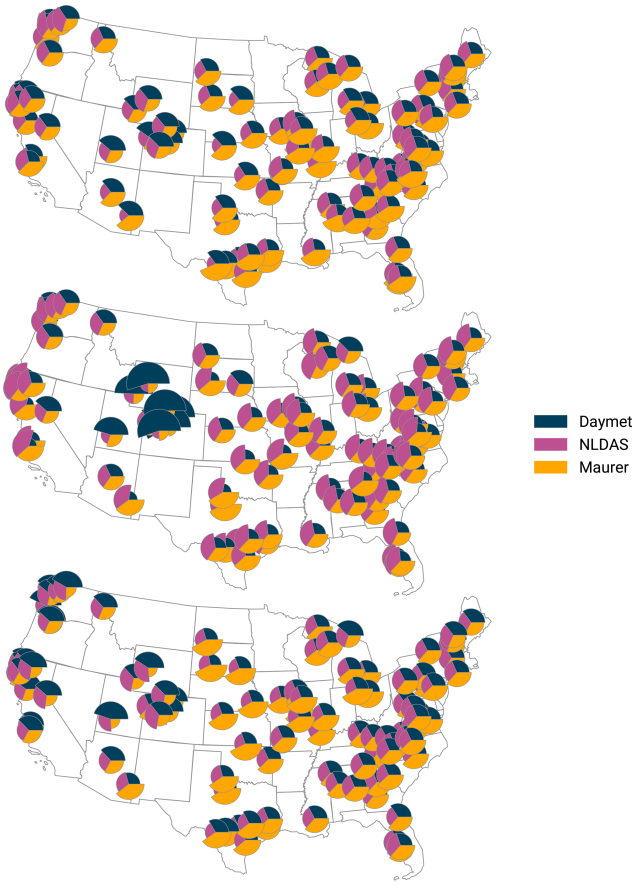

Figure C3. Fraction of attention each product received at each basin, averaged over time. The pie slices are scaled by their fraction to (overly) emphasize differences. Each subplot shows the results for a model trained with a different seed. For better overview, we only plot a random sample of 100 gauges.

Appendix D: A very brief introduction to attention

This section gives a brief high-level introduction to attention, since, as of now, attention is not a widely used concept in hydrologic deep learning applications. As the name suggests, the main idea of "attention" is to provide neural networks with a way to focus on specific parts of their inputs, depending on the current context. Early attention mechanisms come from language applications (Graves, 2013; Bahdanau et al., 2015), where models would focus on relevant words in the source language to produce the corresponding translated words in the target language. With the introduction of the Transformer architecture, attention became one of the most widely used concepts in deep learning (Vaswani et al., 2017). By now, attention and similar approaches have made their way into applications in various fields, including hydrology (e.g., Auer et al., 2024; Rasiya Koya and Roy, 2024).

One way to think about attention – and the origin of today's query/key/value nomenclature – is as a learned similarity-based soft database retrieval (Fig. D1). Let us de-

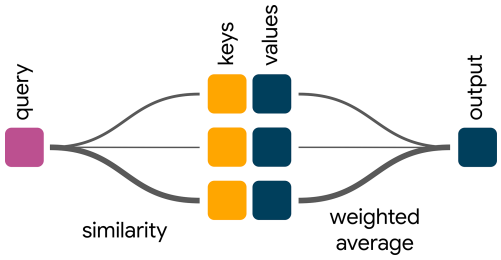

Figure D1. High-level illustration of attention. The query vector (left) is compared to each key vector (middle), and the corresponding value vectors are merged in a weighted average according to the similarity measure, producing the attention output (right).

construct this: by "database", we refer to pairs of so-called *values* and *keys*. That is, each value is an entry in the database that we can retrieve with its associated key. Given a *query*, we calculate a similarity score between the query and each key (this constitutes the "similarity-based" component). All three elements (query/keys/values) are network embeddings, i.e., vectors. For example, one could embed a timeseries of runoff observations as keys, create a one-to-one mapping to the values and then use a given event as the query to search for similar occurrences. The output of the attention operation is a weighted mean of all values, where the weight is higher for values whose keys are more similar to the query (hence "soft" lookup; we do not return a specific value from the database but a weighted average across all values). For example, if we use attention for a translation task, the query would be a learned embedding of the word currently being processed, and keys and values would be embeddings of all source language words[3]. By adjusting the embedding networks, the model can now learn to achieve higher similarity between query and words that are relevant for translating the current word and lower similarity between query and irrelevant words. Finally, we can apply masking (setting the similarity to zero) to disallow attention to certain words.

While the most common application of attention is retrieval along a temporal axis (such as the progression of a sentence), the concept generalizes to retrieval of values from arbitrary sets (Dosovitskiy et al., 2021; Ramsauer et al., 2021). In this paper, we consider the embeddings of meteorological forcings as our key–value database (the embeddings act both as keys and as values), and the static attributes of a basin as our query. Hence, the model can learn to retrieve different forcing combinations in different places.

We conclude this short introduction with the caveat that deep learning is an active field, and at this point there are thousands of publications leveraging, improving, or analyzing attention mechanisms. Therefore, this introduction is by

[3] We ignore some specifics to language modeling here (e.g., positional encoding or tokenization), because they are not immediately relevant to the attention mechanism at the high level of our explanation.

far not exhaustive, nor does it cover any of the formal and mathematical aspects. For a deeper introduction, including the actual equations, we refer to Alammar (2018), Rohrer (2021), and Bishop and Bishop (2023).

Code and data availability. We conducted all experiments with the NeuralHydrology library (Kratzert et al., 2022). The CAMELS dataset necessary to run the experiments is available at https://ral.ucar.edu/solutions/products/camels (last access: 10 November 2025; Newman et al., 2015; Addor et al., 2017). The extended Maurer and NLDAS forcings (which include daily minimum and maximum temperature) are available at https://doi.org/10.4211/hs.17c896843cf940339c3c3496d0c1c077 (Kratzert, 2019a) and https://doi.org/10.4211/hs.0a68bfd7ddf642a8be9041d60f40868c (Kratzert, 2019b). The additional code for analyses and figures presented in this paper are available at https://github.com/gauchm/missing-inputs (https://doi.org/10.5281/zenodo.17362593, Gauch, 2025b). Finally, all trained models and results files are available at https://doi.org/10.5281/zenodo.15008460 (Gauch, 2025a).

Author contributions. MG, FK, and DK developed the idea, conceptualization, and methods of the paper. MG wrote the code and ran the experiments. All authors were involved in the writing of the paper.

Competing interests. At least one of the (co-)authors is a member of the editorial board of *Hydrology and Earth System Sciences*. The peer-review process was guided by an independent editor, and the authors also have no other competing interests to declare.

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.

Review statement. This paper was edited by Albrecht Weerts and reviewed by Juliane Mai and one anonymous referee.

References

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.

Afifi, A. A. and Elashoff, R. M.: Missing Observations in Multivariate Statistics I. Review of the Literature, Journal of the American Statistical Association, 61, 595–604, 1966.

Alammar, J.: Visualizing A Neural Machine Translation Model (Mechanics of Seq2seq Models With Attention), Google Research Blog, https://jalammar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention/ (last access: 10 November 2025), 2018.

Auer, A., Gauch, M., Kratzert, F., Nearing, G., Hochreiter, S., and Klotz, D.: A data-centric perspective on the information needed for hydrological uncertainty predictions, Hydrol. Earth Syst. Sci., 28, 4099–4126, https://doi.org/10.5194/hess-28-4099-2024, 2024.

Bahdanau, D., Cho, K., and Bengio, Y.: Neural Machine Translation by Jointly Learning to Align and Translate, in: 3rd International Conference on Learning Representations (ICLR), arXiv, https://doi.org/10.48550/arXiv.1409.0473, 2015.

Bishop, C. M. and Bishop, H.: Deep learning: Foundations and concepts, Springer Nature, https://doi.org/10.1007/978-3-031-45468-4, 2023.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D.: Language Models are Few-Shot Learners, in: Advances in Neural Information Processing Systems, vol. 33, 1877–1901, Curran Associates, https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html (last access: 10 November 2025), 2020.

Clerc-Schwarzenbach, F., Selleri, G., Neri, M., Toth, E., van Meerveld, I., and Seibert, J.: Large-sample hydrology – a few camels or a whole caravan?, Hydrol. Earth Syst. Sci., 28, 4219–4237, https://doi.org/10.5194/hess-28-4219-2024, 2024.

Cohen, D.: An improved flood forecasting AI model, trained and evaluated globally, Google Research Blog, https://research.google/blog/a-flood-forecasting-ai-model-trained-and-evaluated-globally/ (last access: 10 November 2025), 2024.

Devlin, J., Chang, M., Lee, K., and Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 4171–4186, Association for Computational Linguistics, https://doi.org/10.18653/v1/N19-1423, 2019.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N.: An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, in: 9th International Conference on Learning Representations (ICLR), openreview.net, https://openreview.net/forum?id=YicbFdNTTy (last access: 10 November 2025), 2021.

Dumoulin, V., Perez, E., Schucher, N., Strub, F., Vries, H. d., Courville, A., and Bengio, Y.: Feature-wise transformations, Distill, https://doi.org/10.23915/distill.00011, 2018.

Ebrahimi, S., Arik, S. O., Dong, Y., and Pfister, T.: LANISTR: Multimodal learning from structured and unstructured data, arXiv [preprint], https://doi.org/10.48550/arXiv.2305.16556, 2023.

Frame, J. M., Araki, R., Bhuiyan, S. A., Bindas, T., Rapp, J., Bolotin, L., Deardorff, E., Liu, Q., Haces-Garcia, F., Liao,

M., Frazier, N., and Ogden, F. L.: Machine Learning for a Heterogeneous Water Modeling Framework, JAWRA Journal of the American Water Resources Association, 61, e70000, https://doi.org/10.1111/1752-1688.70000, 2025.

Franken, T., Gullentops, C., Wolfs, V., Defloor, W., Cabus, P., and De Jongh, I.: An operational framework for data driven low flow forecasts in Flanders, EGU General Assembly 2022, Vienna, Austria, 23–27 May 2022, EGU22-6191, https://doi.org/10.5194/egusphere-egu22-6191, 2022.

Gao, Y., Merz, C., Lischeid, G., and Schneider, M.: A review on missing hydrological data processing, Environmental earth sciences, 77, 47, https://doi.org/10.1007/s12665-018-7228-6, 2018.

Gauch, M.: Models and predictions for "How to deal w___ missing input data", Zenodo [data set], https://doi.org/10.5281/zenodo.16983185, 2025a.

Gauch, M.: How to deal w___ missing inputs: Code (v1.0), Zenodo [code, ICE], https://doi.org/10.5281/zenodo.17362593, 2025b.

Gauch, M., Kratzert, F., Gilon, O., Gupta, H., Mai, J., Nearing, G., Tolson, B., Hochreiter, S., and Klotz, D.: In Defense of Metrics: Metrics Sufficiently Encode Typical Human Preferences Regarding Hydrological Model Performance, Water Resources Research, 59, e2022WR033918, https://doi.org/10.1029/2022WR033918, 2023.

Graves, A.: Generating Sequences With Recurrent Neural Networks, arXiv [preprint], https://doi.org/10.48550/arXiv.1308.0850, 2013.

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, 2009.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R.: Masked Autoencoders Are Scalable Vision Learners, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 16000–16009, https://doi.org/10.1109/CVPR52688.2022.01553, 2022.

Kratzert, F.: CAMELS Extended Maurer Forcing Data, HydroShare [data set], https://doi.org/10.4211/hs.17c896843cf940339c3c3496d0c1c077, 2019a.

Kratzert, F.: CAMELS Extended NLDAS Forcing Data, HydroShare [data set], https://doi.org/10.4211/hs.0a68bfd7ddf642a8be9041d60f40868c, 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, 2019.

Kratzert, F., Klotz, D., Hochreiter, S., and Nearing, G. S.: A note on leveraging synergy in multiple meteorological data sets with deep learning for rainfall–runoff modeling, Hydrol. Earth Syst. Sci., 25, 2685–2703, https://doi.org/10.5194/hess-25-2685-2021, 2021.

Kratzert, F., Gauch, M., Nearing, G., and Klotz, D.: NeuralHydrology – A Python library for Deep Learning research in hydrology, Journal of Open Source Software, 7, 4050, https://doi.org/10.21105/joss.04050, 2022.

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, Scientific Data, 10, 61, https://doi.org/10.1038/s41597-023-01975-w, 2023.

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.

Maurer, E. P., Wood, A. W., Adam, J. C., Lettenmaier, D. P., and Nijssen, B.: A Long-Term Hydrologically Based Dataset of Land Surface Fluxes and States for the Conterminous United States, Journal of Climate, 15, 3237–3251, 2002.

Mitchell, M. and Jolley, J.: Research design explained, Holt, Rinehart & Winston Inc., ISBN: 0030040248, https://psycnet.apa.org/record/1987-98845-000 (last access: 10 November 2025), 1988.

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.

Nearing, G., Cohen, D., Dube, V., Gauch, M., Gilon, O., Harrigan, 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, volume 627, 559–563 pp., https://doi.org/10.1038/s41586-024-07145-1, 2024.

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.

Perez, E., Strub, F., de Vries, H., Dumoulin, V., and Courville, A.: FiLM: Visual Reasoning with a General Conditioning Layer, Proceedings of the AAAI Conference on Artificial Intelligence, 32, 3942–3951 pp., https://doi.org/10.1609/aaai.v32i1.11671, 2018.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J.: Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer, Journal of Machine Learning Research, 21, 1–67, 2020.

Ramsauer, H., Schäfl, B., Lehner, J., Seidl, P., Widrich, M., Gruber, L., Holzleitner, M., Adler, T., Kreil, D., Kopp, M. K., Klambauer, G., Brandstetter, J., and Hochreiter, S.: Hopfield Networks is All You Need, in: 9th International Conference on Learning Representations (ICLR), openreview.net, https://openreview.net/forum?id=tL89RnzIiCd (last access: 10 November 2025), 2021.

Rasiya Koya, S. and Roy, T.: Temporal Fusion Transformers for streamflow Prediction: Value of combining attention with recurrence, Journal of Hydrology, 637, 131301, https://doi.org/10.1016/j.jhydrol.2024.131301, 2024.

Read, L., Sampson, A. K., Lambl, D., Butcher, P., Gulland, L., and Elkurdy, M.: Lessons learned applying a machine learning hydrologic forecast model in a live forecasting competition, in: AGU Fall Meeting Abstracts, Vol. 2021, H22A–07, 2021.

Rohrer, B.: Transformers from Scratch, https://e2eml.school/transformers.html (last access: 10 November 2025), 2021.

Schafer, J. L.: Analysis of incomplete multivariate data, CRC press, http://dx.doi.org/10.1201/9781439821862, 1997.

Shalev, G. and Kratzert, F.: Caravan MultiMet: Extending Caravan with Multiple Weather Nowcasts and Forecasts, arXiv [preprint], https://doi.org/10.48550/arXiv.2411.09459, 2024.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res., 15, 1929–1958, 2014.

Thornton, P. E., Running, S. W., and White, M. A.: Generating surfaces of daily meteorological variables over large regions of complex terrain, Journal of Hydrology, 190, 214–251, 1997.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I.: Attention is All you Need, in: Advances in Neural Information Processing Systems, vol. 30, Curran Associates, https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html (last access: 10 November 2025), 2017.

Wells, J.: Compilation of records of surface waters of the United States through September 1950: Part 1-B. North Atlantic slope basins, New York to York River, Tech. rep., US Geological Survey, https://doi.org/10.3133/wsp1302, 1960.

Wu, R., Zhang, A., Ilyas, I., and Rekatsinas, T.: Attention-based Learning for Missing Data Imputation in HoloClean, Proceedings of Machine Learning and Systems, 2, 307–325, 2020.

Xia, Y., Mitchell, K., Ek, M., Sheffield, J., Cosgrove, B., Wood, E., Luo, L., Alonge, C., Wei, H., Meng, J., Livneh, B., Lettenmaier, D., Koren, V., Duan, Q., Mo, K., Fan, Y., and Mocko, D.: Continental-scale water and energy flux analysis and validation for the North American Land Data Assimilation System project phase 2 (NLDAS-2): 1. Intercomparison and application of model products, Journal of Geophysical Research: Atmospheres, 117, https://doi.org/10.1029/2011JD016048, 2012.

Yozgatligil, C., Aslan, S., Iyigun, C., and Batmaz, I.: Comparison of missing value imputation methods in time series: the case of Turkish meteorological data, Theoretical and applied climatology, 112, 143–167, 2013.