The Merit of River Network Topology for Neural Flood Forecasting

Nikolas Kirschstein [*][†] **Yixuan Sun** [*]

Abstract

Climate change exacerbates riverine floods, which occur with higher frequency and intensity than ever. The much-needed forecasting systems typically rely on accurate river discharge predictions. To this end, the SOTA data-driven approaches treat forecasting at spatially distributed gauge stations as isolated problems, even within the same river network. However, incorporating the known topology of the river network into the prediction model has the potential to leverage the adjacency relationship between gauges. Thus, we model river discharge for a network of gauging stations with GNNs and compare the forecasting performance achieved by different adjacency definitions. Our results show that the model fails to benefit from the river network topology information, both on the entire network and small subgraphs. The learned edge weights correlate with neither of the static definitions and exhibit no regular pattern. Furthermore, the GNNs struggle to predict sudden, narrow discharge spikes. Our work hints at a more general underlying phenomenon of neural prediction not always benefitting from graphical structure and may inspire a systematic study of the conditions under which this happens.

1. Introduction

Floods are among the most destructive natural disasters that occur on Earth, causing extensive damage to infrastructure, property, and human life. They are also the most common type of disaster, accounting for almost half of all disaster events recorded (cp. Figure 1). In 2022 alone, floods affected 57.1 million people worldwide, killed almost 8000, and caused 44.9 billion USD in damages (CRED, 2022). With climate change ongoing, floods have become increas-

Figure 1. Historical occurrence of natural disasters by disaster type. The number of events increased over time, with floods being the most common. (Ritchie et al., 2024).

ingly frequent over the last decades and are expected to be even more prevalent in the future (United Nations, 2022). Thus, early warning systems that can help authorities and individuals prepare for and respond to impending floods play a crucial role in mitigating fatalities and economic costs.

Operational forecasting systems such as Google's Flood Forecasting Initiative (Nevo et al., 2022) typically focus on riverine floods, which are responsible for the vast majority of damages. A key component in these systems is the prediction of future river discharge[1] at a gauging station based on environmental indicators such as past discharge and precipitation. The state-of-the-art data-driven approaches are based on Kratzert et al. (2019b) and consist in training an LSTM variant on multiple gauges jointly to exploit the shared underlying physics. However, even when some of those gauges are in the same river network, this topology information is not taken into account. One reason might be that the main benchmarking dataset family CAMELS-x (Addor et al., 2017; Alvarez-Garreton et al., 2018; Coxon et al., 2020; Chagas et al., 2020; Fowler et al., 2021) does not contain such information. Recently, Klingler et al. (2021) published a new benchmarking dataset LamaH-CE that follows the CAMELS-x framework but includes topology data.

---

[*]Mathematical Institute, University of Oxford, UK
[†]Department of Informatics, Technical University of Munich, Germany. Correspondence to: Nikolas Kirschstein <nikolas.kirschstein@maths.ox.ac.uk>.
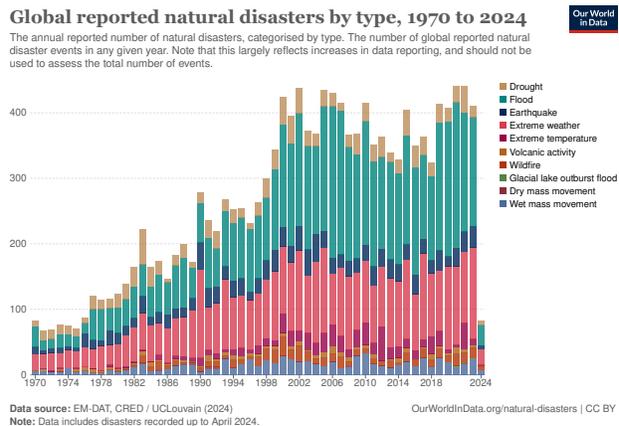
---

[1]amount of water volume passing through a given river section per unit time

In this work, we investigate the effect of river network topology information on discharge predictions by employing a single end-to-end GNN to allow the network structure to be utilised during the prediction process. We train GNNs on LamaH-CE and, to assess the merit of incorporating the graph structure, compare the effect of different adjacency definitions:

(1) no adjacency, which is equivalent to existing approaches with cross-gauge shared parameters but isolated gauges,

(2) binary adjacency of neighbouring gauges in the network,

(3) weighted adjacency according to physical relationships, namely stream length, elevation difference, and average slope between neighbouring gauges, and

(4) learned adjacency by treating edge weights as a model parameter.

We perform this comparison for both the entire dataset as well as four deliberately chosen small-scale subnetworks with different local topologies. Furthermore, we inspect how the learned edge weights in (4) correlate with the static weights in (3). Finally, we analyse the model's behaviour on the worst-performing gauge. Our source code is publicly available at https://github.com/nkirschi/neural-flood-forecasting.

2. Related Work

Classical approaches towards river discharge prediction stem from finite-element solutions to partial differential equations such as the Saint-Venant shallow-water equations (Vreugdenhil, 1994; Wu, 2007). However, these models suffer from scalability issues since they become computationally prohibitive on larger scales, as required in the real world (Nevo et al., 2020). Furthermore, they impose a strong inductive bias by making numerous assumptions about the underlying physics.

On the other hand, data-driven methods and in particular deep learning provide excellent scaling properties and are less inductively biased. They are increasingly being explored for a plethora of hydrological applications, including discharge prediction (see surveys by Mosavi et al., 2018; Chang et al., 2019; Sit et al., 2020), where they tend to achieve higher accuracy than the classical models. The vast majority of studies employ Long Short-Term Memory models (LSTM; Hochreiter & Schmidhuber, 1997) due to their inherent suitability for sequential tasks and reliability in predicting extreme events (Frame et al., 2022). Whereas these studies usually consider forecasting for a single gauging station, Kratzert et al. (2019a;b) demonstrate the generalisation benefit of training a single spatially distributed LSTM
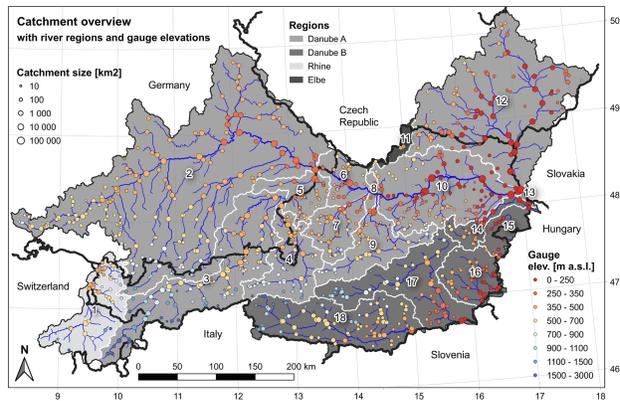

Figure 2. Geographical contextualisation of the LamaH-CE dataset. Circle colour indicates gauge elevation, while circle size indicates catchment size. (Klingler et al., 2021)

model on multiple gauging sites jointly. Their approach exploits the shared underlying physics across gauges but is still agnostic to the relationship between sites.

Incorporating information from neighbouring stations or even an entire river network into a spatially distributed model potentially improves prediction performance. Upstream gauges could "announce" the advent of increased water masses to downstream gauges, which in turn could provide forewarning about already ongoing flooding further downstream. The input then becomes a graph whose vertices represent gauges and edges represent flow between gauges. The corresponding deep learning tool to capture these spatial dependencies is Graph Neural Networks (GNN). Kratzert et al. (2021) employ it as a post-processing step to route the per-gauge discharge predicted by a conventional LSTM along the river network. In contrast, we seek to unify prediction and routing in a single GNN.

3. Methodology

3.1. Data Preprocessing

The LamaH-CE[2] dataset (Klingler et al., 2021) contains historical discharge and meteorological measurements on an hourly resolution for 859 gauges in the broader Danube river network shown in Figure 2. Covering an area of $170\,000\,\mathrm{km}^2$ with diverse environmental conditions, Klingler et al. expect that results from investigations on this dataset carry over to other river networks. Unfortunately, LamaH-CE does not provide any flood event annotations, so that we can only model continuous discharge but not floods as discrete events. Moreover, the dataset does not include average propagation time between gauges, meaning that a predictor needs to implicitly infer the time lag by comparing observations at neighbouring gauges.

---

[2]**LA**rge-Sa**MP**le **DA**ta for **H**ydrology for **C**entral **E**urope

The river network defined by LamaH-CE naturally forms a directed acyclic graph (DAG) $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The nodes $\mathcal{V}$ represent gauges, and the edges $\mathcal{E}$ represent flow between a gauge and the next downstream gauges. Hence, $\mathcal{G}$ is *anti-transitive*, i.e., no skip connections exist. We preprocess $\mathcal{G}$ to distil a connected subgraph with complete data.

Region Selection. Figure 2 shows that $\mathcal{G}$ contains four different connected components, of which we restrict ourselves to the largest one, "Danube A". Its most downstream gauge close to the Austrian-Hungarian border has complete discharge data for the years 2000 through 2017. Starting at this gauge, we determine all connected gauges of the Danube A region by performing an inverse depth-first search given by Algorithm A.1. Overall, 608 out of the original 859 gauges belong to this connected component.

Gauge Filtering. While the meteorological data is complete, the discharge data contains gaps. Klingler et al. have filled any consecutive gaps of at most six hours by linear interpolation and left the remaining longer gaps unaltered. We only want to consider gauges that (a) do not have these longer periods of missing values and (b) provide discharge data for at least the same time frame (2000 to 2017) as the most downstream gauge. To this end, we remove all gauges that violate these requirements from the graph using Algorithm A.2. Predecessors and successors of a deleted node get newly connected with a combined edge weight so that network connectivity is maintained. Note that thanks to antitransitivity, a duplicate check is unnecessary when inserting the new edges. After this preprocessing step, we are left with 358 out of the previously 608 gauges.

Overall, the reduced graph $\mathcal{G}$ consists of $n := |\mathcal{V}| = 358$ gauges with T hours of discharge measurements for the years 2000 to 2017, which we can conceptually represent as a node signal $\mathbf{Q} = \left[\boldsymbol{q}^{(1)} \mid \boldsymbol{q}^{(2)} \mid \ldots \mid \boldsymbol{q}^{(T)} \right] \in \mathbb{R}^{n \times T}$. Equally, we have four node signals of the same shape for each of the meteorological context variables precipitation, topsoil moisture, air temperature, and surface pressure. However, we exclude them from all notation throughout the paper for simplicity of presentation, i.e., drop the implicit third dimension of size 5.

Normalisation. We normalise the data to surrender all gauges to the same scale and accelerate the training process (LeCun et al., 2002). In particular, we normalise per gauge, i.e., element-wise, using the standard score:

$$\boldsymbol{q}^{(t)} \leftarrow \frac{\boldsymbol{q}^{(t)} - \boldsymbol{\mu}}{\boldsymbol{\sigma}} \quad \text{where} \quad \begin{array}{l} \boldsymbol{\mu} = \frac{1}{T} \sum_{t=1}^{T} \boldsymbol{q}^{(t)} \\ \boldsymbol{\sigma}^2 = \frac{1}{T-1} \sum_{i=1}^{T} (\boldsymbol{q}^{(t)} - \boldsymbol{\mu})^2 \end{array}$$

Train-test splits. To robustly assess the performance of a trained model on unseen data via cross-validation, we consider three different train-test splits. The last two years 2016 and 2017 always serve for testing, and from the remainder eight years are chosen for training: once the even years in 2000 to 2015, once the odd years in 2000 to 2015, and once the contiguous years 2008 to 2015. Note the differences to vanilla fold-based cross-validation schemes: (a) we need to ensure the train years temporally precede the test years, and (b) due to the small amount of available years we choose the same test set for all splits.

3.2. The Forecasting Task

We task the model with an instance of supervised node regression. Assume we are given a certain amount of W (*"window size"*) most recent hours of discharge and meteorological measurements, for all gauges. Our goal is to predict the discharge L (*"lead time"*) hours in the future. Again, for simplicity, we restrict all notation to the discharge data in the input since the meteorological data can be trivially added in an extra dimension.

Features & Targets. To conduct supervised learning, we extract input-output pairs from the time series represented by $\mathbf{Q}$ (cp. Section 3.1). For $t = W, W+1 \ldots, T-L$, we define the feature matrix at time step t as

$$\mathbf{X}^{(t)} := \left[\boldsymbol{q}^{(t-W+1)} \mid \ldots \mid \boldsymbol{q}^{(t-1)} \mid \boldsymbol{q}^{(t)} \right] \in \mathbb{R}^{n \times W}$$

and the corresponding target vector as $\boldsymbol{y}^{(t)} := \boldsymbol{q}^{(t+L)} \in \mathbb{R}^n$. We collect all samples into the set $\mathcal{D} = \{(\mathbf{X}^{(t)}, \boldsymbol{y}^{(t)})\}_{t=W}^{T-L}$ and partition it according to a given train-test split into $\mathcal{D} = \mathcal{D}_{\text{train}} \uplus \mathcal{D}_{\text{test}}$. The extraction process can be illustrated as follows:
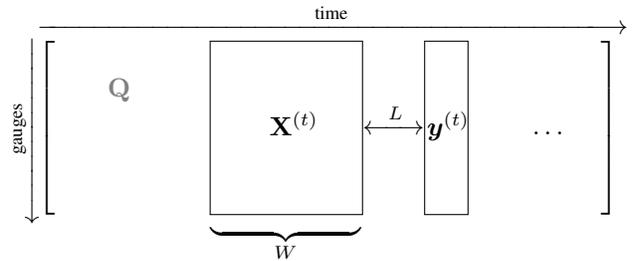
Adjacency. Besides the input and target measurements, we feed the river network topology to the GNN in the form of an adjacency matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$. For the definition of matrix entries corresponding to an edge $(i, j) \in \mathcal{E}$ (the rest being zero), we consider the following choices:

(1) *isolated:* $\mathbf{A}_{i,j} := 0$ equates to removing all edges and results in the augmented normalised adjacency matrix to be a multiple of the identity so that each GNN layer degenerates to a node-wise linear layer.

(2) *binary:* $\mathbf{A}_{i,j} := 1$ corresponds to the unaltered adjacency matrix as it comes with the LamaH-CE dataset.

(3) *weighted:* $\mathbf{A}_{i,j} \coloneqq w_{(i,j)}$ quantifies a physical relationship, for which LamaH-CE provides three alternatives:

- the *stream length* along the river between i and j,
- the *elevation difference* along the river between i and j, and
- the *average slope* of the river between i and j.

(4) *learned:* $\mathbf{A}_{i,j} \coloneqq \omega_{(i,j)}$ where $\boldsymbol{\omega} \in \mathbb{R}^{|\mathcal{E}|}$ is a learnable model parameter.

The first two variants allow us to compare the effect of introducing the river network topology into the model at all. The last two variants enable insights into what kind of relative importance of edges is most helpful. As usual in GNNs, we use the normalised augmented adjacency matrix

$$\bar{\mathbf{A}} \coloneqq (\mathbf{D}_{\text{in}}+\text{diag}(\boldsymbol{\xi}))^{-\frac{1}{2}}(\mathbf{A}+\text{diag}(\boldsymbol{\xi}))(\mathbf{D}_{\text{in}}+\text{diag}(\boldsymbol{\xi}))^{-\frac{1}{2}}$$

where self-loops for node i with weight ξ_i are added and everything is symmetrically normalised based on the diagonal in-degree matrix $\mathbf{D}_{\text{in}}$. We generally set ξ_i as the mean of all incoming edge weights at node i to make self-loops roughly equally important to the other edges. The only exception to this is option (1) above, where that mean would be zero and thus result in no information flow whatsoever, so that in this case, we set the self-loop weights to one instead.

Model. Our desideratum is a GNN $f_\theta : \mathbb{R}^{n \times W} \to \mathbb{R}^n$ parameterised by θ which closely approximates the mapping of windows $\mathbf{X}$ to targets $\boldsymbol{y}$, i.e., $\hat{\boldsymbol{y}} \coloneqq f_\theta(\mathbf{X}) \approx \boldsymbol{y}$. All our models have a sandwich architecture: an affine layer $\text{Encoder}_{\boldsymbol{\Theta}_0} : \mathbb{R}^{n \times W} \to \mathbb{R}^{n \times d}$ embeds the W-dimensional input per gauge into a d-dimensional latent space. On this space, a sequence of N layers $\text{GNNLayer}_{\boldsymbol{\Theta}_i} : \mathbb{R}^{n \times d} \times \mathbb{R}^{n \times n} \to \mathbb{R}^{n \times d}$ with subsequent activation function $\sigma = \text{ReLU}$ are applied. Finally, another affine layer $\text{Decoder}_{\boldsymbol{\Theta}_{N+1}} : \mathbb{R}^{n \times d} \to \mathbb{R}^n$ projects from the latent space to a scalar per gauge. In symbols:

$$\mathbf{H}^{(0)} \coloneqq \text{Encoder}_{\boldsymbol{\Theta}_0}(\mathbf{X})$$
$$\mathbf{H}^{(i)} \coloneqq \sigma(\text{GNNLayer}_{\boldsymbol{\Theta}_i}(\mathbf{H}^{(i-1)}, \bar{\mathbf{A}})) \quad \text{for } i = 1, \ldots, N$$
$$\hat{\boldsymbol{y}} \coloneqq \text{Decoder}_{\boldsymbol{\Theta}_{N+1}}(\mathbf{H}^{(N)}).$$

We consider three choices for GNNLayer: a residual version of the vanilla GCN layer (Kipf & Welling, 2017), the inherently residual GCNII layer (Chen et al., 2020), and a residual version of the attention-based GAT layer (Veličković et al., 2017). Since the GAT layer already contains a learned component, the adjacency case (4) would be redundant for this architecture, so that we replace it with the case of providing all three edge weights in (3) jointly, which is not possible with the other two layer definitions. All three employ residual connections to overcome the phenomenon known as *oversmoothing* (Oono & Suzuki, 2020), where the features of adjacent nodes converge with increasing depth.

Relevancy Score. The dataset contains many periods of almost no discharge activity. To guide the training process and focus on "interesting" discharge windows, we seek to quantify the relevancy of each row $\boldsymbol{x}^{(g)} \in \mathbb{R}^W$ in the feature matrix $\mathbf{X}$. First, we *unnormalise* to recover the original discharge values $\boldsymbol{x}_\star^{(g)} \coloneqq \sigma_g \boldsymbol{x}^{(g)} + \mu_g$. Then, let $\nabla \boldsymbol{x}_\star^{(g)} \in \mathbb{R}^W$ denote the numerical derivative according to the second-order accurate central differences method, and $\int \boldsymbol{x}_\star^{(g)} \in \mathbb{R}$ the numerical integral according to the trapezoidal rule. We define the relevancy as

$$\varrho(\boldsymbol{x}^{(g)}) \coloneqq \text{mean} \left(\frac{\nabla \boldsymbol{x}_\star^{(g)}}{\mu_g} \right)^2 \odot \frac{\int \boldsymbol{x}_\star^{(g)}}{\mu_g} \in \mathbb{R}^W.$$

This heuristic definition captures both the rate of change in a given discharge window and the overall discharge in relation to its mean, while weighting the former twice as strongly. The year-wise maximisers shown in Section 3.2 suggest that this is a reasonable measure of relevancy. Note that it does not depend on the meteorological context variables.

Optimisation Objective. To measure the error between a model prediction $\hat{\boldsymbol{y}}$ for input $\mathbf{X}$ and the target $\boldsymbol{y}$, we weight the standard multi-dimensional regression square loss by the relevancy score:

$$L(\hat{\boldsymbol{y}}, \boldsymbol{y}, \mathbf{X}) \coloneqq \tfrac{1}{n} \|\varrho(\mathbf{X}) \odot (\hat{\boldsymbol{y}} - \boldsymbol{y})\|_2^2.$$

Training is then defined as optimising the expected loss over the empirical distribution of training samples in $\mathcal{D}_{\text{train}}$, regularised by the ℓ^2-norm of the parameters:

$$\min_\theta \mathbb{E}_{(\mathbf{X}, \boldsymbol{y}) \sim \mathcal{D}_{\text{train}}}[L(f_\theta(\mathbf{X}, \bar{\mathbf{A}}), \boldsymbol{y}, \mathbf{X})] + \tfrac{\lambda}{2} \|\theta\|_2^2.$$

Testing Metric. Recall that we perform training on normalised samples. For testing, we must calculate metrics on the unnormalised version of the predictions and targets:

$$\hat{\boldsymbol{y}}_\star \coloneqq \boldsymbol{\sigma} \odot \hat{\boldsymbol{y}} + \boldsymbol{\mu}, \qquad \boldsymbol{y}_\star \coloneqq \boldsymbol{\sigma} \odot \boldsymbol{y} + \boldsymbol{\mu}.$$

The standard metric in hydrology for a single gauge is the *Nash-Sutcliffe Efficiency* (NSE; Nash & Sutcliffe, 1970). It compares the sum of squared errors of the model to the that of the constant mean-predictor and subtracts this value from one to obtain a percentage score in $[0, 1]$. An NSE of zero means that the model's predictive capability is no better than that of the empirical mean, while an NSE of one indicates perfect model predictions. Since we are training the model with a weighted objective, we analogously weight[3] the evaluation metric with the relevancy score:

$$\text{NSE} \coloneqq 1 - \frac{\sum_{i=1}^{|\mathcal{D}_{\text{test}}|} \varrho(\mathbf{X}^{(t)}) \odot (\hat{\boldsymbol{y}}_\star^{(t_i)} - \boldsymbol{y}_\star^{(t_i)})^2}{\sum_{i=1}^{|\mathcal{D}_{\text{test}}|} \varrho(\mathbf{X}^{(t)}) \odot (\boldsymbol{\mu} - \boldsymbol{y}_\star^{(t_i)})^2}.$$

We straightforwardly obtain a summary metric by averaging across gauges: $\overline{\text{NSE}} \coloneqq \tfrac{1}{n} \sum_{g=1}^n \text{NSE}_g$.

---

[3]The unweighted version yields qualitatively similar results but has higher absolute values since it rewards performing well on trivial windows more than the weighted metric.
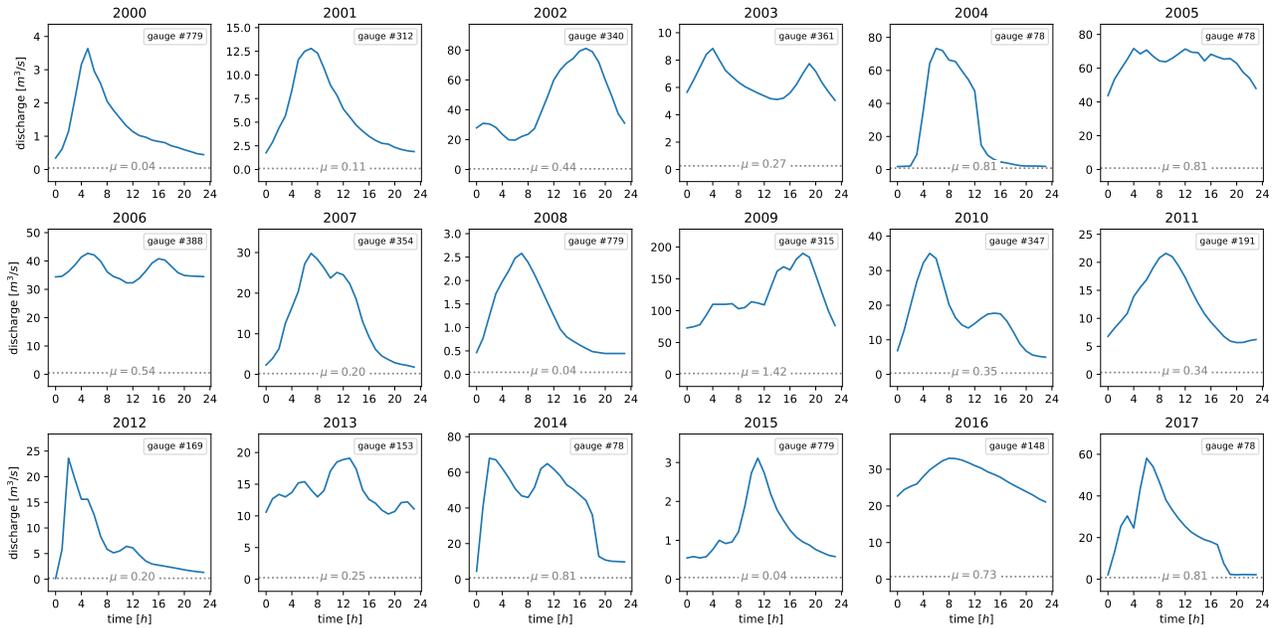
Figure 3. Year-wise maximisers of the relevancy score ϱ. Each maximiser's discharge window (blue) exhibits both high variability as well as an excessive overall discharge level, in relation to the mean discharge of its gauge (gray).

4. Experiments

4.1. Experimental Setup

The code to reproduce our experiments is publicly available[4]. Table 1 lists the relevant hyperparameters we use throughout all experiments unless stated otherwise. On the data side, we set the window size to $W = 24$ and lead time to $L = 6$ hours, which are realistic choices. While, conceptually, larger window sizes would be preferred to provide a longer history to the predictor, they also imply a larger latent space dimensionality d and thus restrict computational feasibility.

Table 1. Default hyperparameters for our experiments.

	HYPERPARAMETER	VALUE
DATA	WINDOW SIZE (W)	24 h
	LEAD TIME (L)	6 h
	NORMALISATION?	Z-SCORE
MODEL	ARCHITECTURE	GCNII
	NETWORK DEPTH (N)	19
	LATENT SPACE DIM (d)	128
	EDGE DIRECTION	BIDIRECTED
	ADJACENCY TYPE	BINARY
TRAINING	INITIALISATION	KAIMING
	OPTIMISER	ADAM
	# EPOCHS	100
	BATCH SIZE	64
	LEARNING RATE	10^{-4}
	REGULARISATION STRENGTH (λ)	10^{-5}

[4] https://github.com/nkirschi/neural-flood-forecasting

On the model side, we consider all three choices of layer definition described in Section 3.2, resulting in three model architectures ResGCN, GCNII, and ResGAT. We choose a depth of $N = 19$ layers to allow information propagation along the entire river graph, since the longest path in the preprocessed graph consists of 19 edges. The latent space dimensionality of $d = 128$ is chosen large enough to allow an injective feature embedding but small enough to avoid memory issues. The edge direction and adjacency type hyperparameters are subject to investigation in Section 4.2.

On the training side, all neural network parameters are randomly initialised using the standard Kaiming initialisation scheme (He et al., 2015) for architectures with ReLU activations. We then perform 100 epochs of stochastic mini-batch gradient descent, which is enough for the process to converge. The descent algorithm is Adaptive Moments (Adam; Kingma & Ba, 2015) with a learning rate of 10^{-4}. To prevent overfitting, besides regularising with a strength of $\lambda = 10^{-5}$, we select the parameters from the epoch with minimal loss on a random holdout set containing $\frac{1}{5}$ of the training data.

4.2. River Topology Comparison

Our main experiment compares the impact of the six different gauge adjacency definitions detailed in Section 3.2 on forecasting performance. In addition, we also consider three alternative edge orientations, which determine the direction of information flow in the GNN, as none of the options is

a priori preferable. The *downstream* orientation is given by the dataset, the *upstream* orientation results from reversing all edges, and the *bidirected* orientation from adding all reverse edges to the forward ones. We cross-validate all 18 topology combinations on the three train-test splits established in Section 3.1 using the summary NSE metric defined in Section 3.2, and report the results in Table 2.

Surprisingly, model performance shows almost no sensitivity to the choice of graph topology. Isolating the gauges does not harm performance beyond the standard deviation, and no combination outperforms a 19-layer MLP baseline. This indicates that the forecasting task for a gauge mainly benefits from the past discharge at that gauge but not from the discharge at neighbouring gauges. The river graph topology makes no difference. Even when the model is allowed to learn an optimal edge weight assignment, it does not manage to outperform the baseline.

4.3. Learning the Weights

The case of learned edge weights is of particular interest. They were initialised by drawing from the uniform distribution in $[0.9, 1.1]$ to arrange them neutrally around one while still introducing sufficient noise to break symmetry. Whenever learned weights get negative during training, we clip them to zero. The distribution of the learned weights (cp. Table A.3) is still centred around one with minima close to zero and maxima below ten.

To see if the learned weights exhibit any similarities with the physical weights, we calculate Pearson correlation coefficients for all topology combinations. Table 3 shows that none of the physical weight assignments correlate much with the learned weights. In multiple instances, the sign even flips when using a different model architecture. For instance, the largest positive correlation occurs with stream length for ResGCN, but in this same case GCNII achieves a negative correlation of the same magnitude. Hence, we conclude that none of the physical edge weights from the datasets are optimal context information for the predictor.

4.4. Small-scale Subnetworks

To exclude the possibility that the considerable depth is causing the GCN to not outperform the baseline MLP due to more general issues with training very deep networks, we repeat the topology comparison from Section 4.2 on four small subgraphs of the river network illustrated in Figure 4. Since the graph rewiring done by Algorithm A.2 can have a strong effect when considering only a handful nodes, we skipped it in the preprocessing for this experiment and chose only subgraphs with full data coverage to begin with. Furthermore, to allow for sufficient model capacity, we increase the latent space dimensionality to 512 for this experiment.

Table 2. Forecasting performance on different river network topologies, given as mean and standard deviation of $\overline{\text{NSE}}$ across folds. A wide 2-layer MLP baseline achieves a result of $85.37\,\% \pm 1.64\,\%$. Bold indicates the best value per column. Note that results for the isolated adjacency type are not affected by the choice of edge orientation due to the absence of edges in this case.

(a) ResGCN

ADJACENCY TYPE	EDGE ORIENTATION		
	DOWNSTREAM	UPSTREAM	BIDIRECTED
ISOLATED	**85.07 %** ±0.66 %	**85.07 %** ±0.66 %	**85.07 %** ±0.66 %
BINARY	82.03 % ±1.97 %	83.90 % ±1.26 %	82.73 % ±2.54 %
STREAM LENGTH	81.64 % ±1.45 %	81.98 % ±3.06 %	83.09 % ±2.37 %
ELEVATION DIFFERENCE	82.16 % ±1.85 %	83.43 % ±0.16 %	83.16 % ±1.76 %
AVERAGE SLOPE	81.93 % ±1.18 %	80.68 % ±1.99 %	81.59 % ±2.21 %
LEARNED	81.34 % ±1.61 %	84.13 % ±0.81 %	83.50 % ±1.59 %

(b) GCNII

ADJACENCY TYPE	EDGE ORIENTATION		
	DOWNSTREAM	UPSTREAM	BIDIRECTED
ISOLATED	84.12 % ±1.88 %	84.12 % ±1.88 %	84.12 % ±1.88 %
BINARY	84.09 % ±1.11 %	**85.16 %** ±1.74 %	84.81 % ±0.53 %
STREAM LENGTH	84.29 % ±1.28 %	85.09 % ±2.11 %	83.90 % ±1.05 %
ELEVATION DIFFERENCE	84.44 % ±0.81 %	84.87 % ±1.78 %	84.06 % ±0.68 %
AVERAGE SLOPE	83.93 % ±1.39 %	84.47 % ±1.11 %	84.68 % ±0.68 %
LEARNED	**84.91 %** ±1.97 %	85.00 % ±2.11 %	**85.56 %** ±1.41 %

(c) ResGAT

ADJACENCY TYPE	EDGE ORIENTATION		
	DOWNSTREAM	UPSTREAM	BIDIRECTED
ISOLATED	83.10 % ±0.88 %	83.10 % ±0.88 %	83.10 % ±0.88 %
BINARY	80.68 % ±4.78 %	82.59 % ±2.01 %	82.77 % ±0.47 %
ALL OF THE BELOW	**83.78 %** ±1.71 %	**83.33 %** ±1.76 %	**82.73 %** ±1.30 %
STREAM LENGTH	80.21 % ±4.85 %	83.28 % ±1.72 %	83.56 % ±1.57 %
ELEVATION DIFFERENCE	80.58 % ±5.00 %	82.88 % ±1.50 %	82.87 % ±1.44 %
AVERAGE SLOPE	81.10 % ±4.67 %	82.81 % ±0.90 %	81.69 % ±0.39 %

Table 3. Pearson correlation between learned and physical edge weights.

| | LEARNED EDGE WEIGHTS | | | | | |
| | DOWNSTREAM | | UPSTREAM | | BIDIRECTED | |
PHYSICAL EDGE WEIGHTS	RESGCN	GCNII	RESGCN	GCNII	RESGCN	GCNII
STREAM LENGTH	-0.375 ±0.012	-0.285 ±0.014	0.012 ±0.056	0.027 ±0.025	0.139 ±0.024	-0.021 ±0.048
ELEVATION DIFFERENCE	-0.148 ±0.006	-0.214 ±0.013	-0.346 ±0.025	-0.325 ±0.030	-0.182 ±0.031	-0.188 ±0.051
AVERAGE SLOPE	0.075 ±0.007	-0.034 ±0.018	-0.325 ±0.014	-0.2955 ±0.051	-0.242 ±0.017	-0.158 ±0.036

The results are consistent with those on the full dataset and hence outsourced into the appendix tables A.4 to A.7. Note that the orders of magnitude of $\overline{\text{NSE}}$ differ as the difficulty of the prediction task naturally changes with the underlying graph. The small-scale experiment confirms the observation that topology context does not benefit prediction.

4.5. Worst Gauge Investigation

The performance on gauge #169 of all trained models is considerably below the mean. For instance, the best overall performing model, bidirected-learned GCNII (third fold), achieves its worst $\overline{\text{NSE}}$ on this outlier gauge. To better understand the scenarios that are challenging for the model, we determine the top disjoint time horizons of 48 hours (24 hours for past and future) in terms of deviation of model prediction from the ground truth. The resulting plots in Figure 5 reveal that the outlier gauge is characterised by sudden and narrow spikes, which are inherently hard to forecast for any predictor. The gauge might be located behind a floodgate. As a result, the forecasting performance is mediocre, with the forecast often missing spikes.

5. Conclusion

In this work, we explored the applicability of GNNs to holistic flood forecasting in a river network graph. Based on the LamaH-CE dataset, we framed a supervised node regression task for predicting future discharge at all gauging stations in the graph given past observations. By modifying the adjacency matrix, we compared the impact of different adjacency definitions on the prediction performance. Our results reveal that the impact of river topology is negligible. The GNN performs equally well even when all edges are removed from the graph, which makes it act like an MLP. It does not benefit from weighted edges that resemble physical relationships between gauges. When the model is allowed to jointly learn the edge weights along with the other parameters, they correlate with neither constant weights nor any of the physical weightings given by the dataset. A small-scale subnetwork study shows that the results are not caused by issues with training deep models but prove consistent
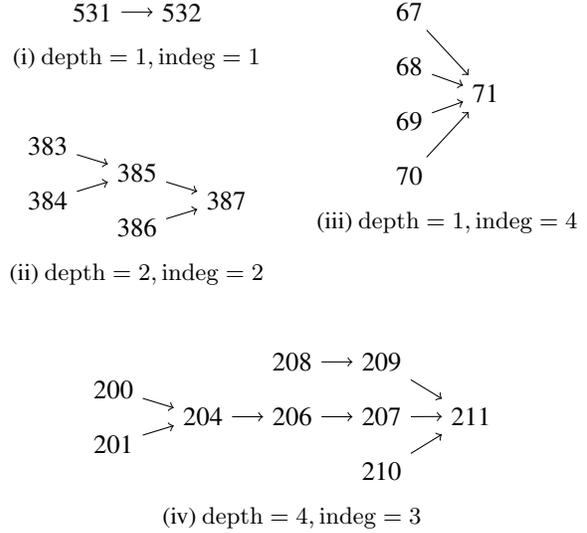

Figure 4. Four subgraphs of the river network with different depth and sink in-degree. The node labels refer to the original gauge IDs from the dataset.
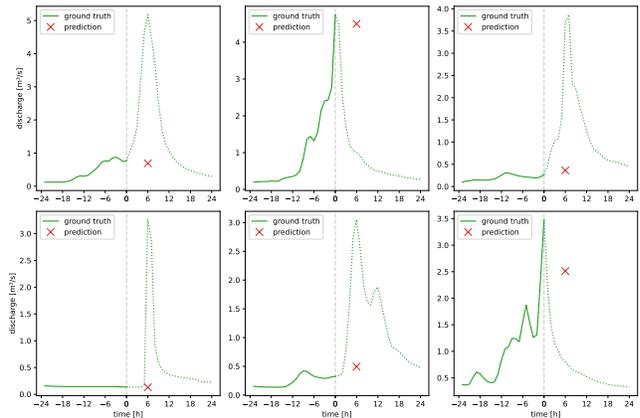

Figure 5. Worst predictions of the bidirected-learned GCNII (third fold) on its overall worst gauge #169. Negative time indicates past, and positive time indicates future discharge.

throughout scales. Investigations on a challenging outlier gauge indicate that the GNNs struggle to predict sudden, narrow discharge spikes.

On a high level, future work is encouraged to investigate under which conditions including graph topology in neural predictors actually helps, which is not clear a priori. While the key could lie in employing more specialised model architectures such as DAGNN (Thost & Chen, 2021) for the dataset at hand, there might be more fundamental limitations to the use of GNNs for large-scale regression problems. Moreover, for the application of flood forecasting, enhancing the dataset with inter-gauge propagation time metadata and reliable flood annotations may allow the predictor to leverage the relational context more effectively. Otherwise, our results suggest that focusing on accurate spike prediction might be more promising than incorporating river network topology information.

Finally, there is a broader issue: we used a river network dataset from central Europe as discharge measurements are readily available there for long time periods. However, the regions most affected by floods happen to be typically located in low-income countries where data is scarce. More gauges need to be installed in those high-risk regions, and large-scale datasets collected to enable more relevant studies and save lives.

Impact Statement

Flood forecasting is a crucial technology to mitigate humanitarian crises in the age of climate change. With floods being the most frequent type of natural disaster, even minor methodological improvements are bound to greatly impact disaster prevention and mitigation. While the results in our work suggest that incorporating river network topology into the forecasting process might not be one such improvement, the momentousness of the topic demands that future research continues to explore the idea as well as conditions under which it potentially can be an improvement.

Acknowledgements

References

Addor, N., Newman, A. J., Mizukami, N., and Clark, M. P. The CAMELS data set: catchment attributes and meteorology for large-sample studies. *Hydrology and Earth System Sciences*, 21(10):5293–5313, October 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. *Hydrology and Earth System Sciences*, 22(11): 5817–5846, November 2018.

Centre for Research on the Epidemiology of Disasters (CRED). Disasters in Numbers 2022. Technical report, 2022.

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 System Science Data*, 12(3):2075–2096, September 2020.

Chang, F.-J., Hsu, K., and Chang, L.-C. *Flood Forecasting Using Machine Learning Methods*. MDPI, February 2019. ISBN 978-3-03897-549-6 978-3-03897-548-9.

Chen, M., Wei, Z., Huang, Z., Ding, B., and Li, Y. Simple and Deep Graph Convolutional Networks. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1725–1735. PMLR, July 2020.

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 System Science Data*, 12(4):2459–2483, October 2020.

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 System Science Data*, 13(8):3847–3867, August 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. *Hydrology and Earth System Sciences*, 26(13): 3377–3392, July 2022.

He, K., Zhang, X., Ren, S., and Sun, J. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In *2015 IEEE International*

Conference on Computer Vision (ICCV), pp. 1026–1034, Santiago, Chile, December 2015. IEEE. ISBN 978-1-4673-8391-2.

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

Kingma, D. P. and Ba, J. Adam: A Method for Stochastic Optimization. In Bengio, Y. and LeCun, Y. (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Kipf, T. N. and Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*, 2017.

Klingler, C., Schulz, K., and Herrnegger, M. LamaH-CE: LArge-SaMple DAta for Hydrology and Environmental Sciences for Central Europe. *Earth System Science Data*, 13(9):4529–4565, September 2021.

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 Resources Research*, 55(12):11344–11354, December 2019a.

Kratzert, F., Klotz, D., Shalev, G., Klambauer, G., Hochreiter, S., and Nearing, G. Towards learning universal, regional, and local hydrological behaviors via machine learning applied to large-sample datasets. *Hydrology and Earth System Sciences*, 23(12):5089–5110, December 2019b.

Kratzert, F., Klotz, D., Gauch, M., Klingler, C., Nearing, G., and Hochreiter, S. Large-scale river network modeling using Graph Neural Networks. Technical report, March 2021.

LeCun, Y., Bottou, L., Orr, G. B., and Müller, K.-R. Efficient backprop. In *Neural networks: Tricks of the trade*, pp. 9–50. Springer, 2002.

Mosavi, A., Ozturk, P., and Chau, K.-w. Flood Prediction Using Machine Learning Models: Literature Review. *Water*, 10(11):1536, October 2018.

Nash, J. and Sutcliffe, J. River flow forecasting through conceptual models part I — A discussion of principles. *Journal of Hydrology*, 10(3):282–290, April 1970.

Nevo, S., Elidan, G., Hassidim, A., Shalev, G., Gilon, O., Nearing, G., and Matias, Y. ML-based Flood Forecasting: Advances in Scale, Accuracy and Reach, 2020. _eprint: 2012.00671.

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., Yuklea, L., Royz, 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. *Hydrology and Earth System Sciences*, 26(15):4013–4032, August 2022.

Oono, K. and Suzuki, T. Graph Neural Networks Exponentially Lose Expressive Power for Node Classification. In *International Conference on Learning Representations*, 2020.

Ritchie, H., Rosado, P., and Roser, M. Natural Disasters. *Our World in Data*, 2024.

Sit, M., Demiray, B. Z., Xiang, Z., Ewing, G. J., Sermet, Y., and Demir, I. A comprehensive review of deep learning applications in hydrology and water resources. *Water Science and Technology*, 82(12):2635–2670, December 2020.

Thost, V. and Chen, J. Directed Acyclic Graph Neural Networks. In *International Conference on Learning Representations*, 2021.

United Nations Office for Disaster Risk Reduction (UN-DRR). Global Assessment Report on Disaster Risk Reduction – Our World at Risk: Transforming Governance for a Resilient Future. Technical report, 2022.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. Graph Attention Networks. 2017. Publisher: arXiv Version Number: 3.

Vreugdenhil, C. B. *Numerical methods for shallow-water flow*, volume 13. Springer Science & Business Media, 1994.

Wu, W. *Computational River Dynamics*. CRC Press, 0 edition, November 2007. ISBN 978-0-203-93848-5.

A. Appendix

A.1. Preprocessing Algorithms

Algorithm A.1: Inverse depth-first search

Input: DAG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, start node $v_0 \in \mathcal{V}$
Output: All direct and indirect predecessors of v_0 in $\mathcal{G}$

$\text{inverseDFS}(\mathcal{G}, v_0)$
1    $\mathcal{V}_{\text{in}} \leftarrow \{v \in \mathcal{V} \mid (v, v_0) \in \mathcal{E}\}$
2   **if** $\mathcal{V}_{\text{in}} = \emptyset$ **then**
3   **return** $\{v_0\}$
4   **else**
5   **return** $\{v_0\} \cup \bigcup\limits_{v \in \mathcal{V}_{\text{in}}} \text{invDFS}(v)$

Algorithm A.2: Rewire-removal of a node

Input: antitransitive weighted DAG $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, moribund node $v_{\text{RIP}} \in \mathcal{V}$
Output: $\mathcal{G}$ without v_{RIP} where its predecessors and successors are rewired

$\text{rewireRemove}(\mathcal{G}, v_{\text{RIP}})$
1    $\mathcal{V}_{\text{in}} \leftarrow \{v \in \mathcal{V} \mid (v, v_{\text{RIP}}) \in \mathcal{E}\}$
2    $\mathcal{V}_{\text{out}} \leftarrow \{v \in \mathcal{V} \mid (v_{\text{RIP}}, v) \in \mathcal{E}\}$
3    $\mathcal{V} \leftarrow \mathcal{V} \setminus \{v_{\text{RIP}}\}$
4    $\mathcal{E} \leftarrow \mathcal{E} \setminus (\mathcal{V}_{\text{in}} \times \{v_{\text{RIP}}\}) \setminus (\{v_{\text{RIP}}\} \times \mathcal{V}_{\text{out}}) \cup (\mathcal{V}_{\text{in}} \times \mathcal{V}_{\text{out}})$
5   **for** $(v_{\text{in}}, v_{\text{out}}) \in \mathcal{V}_{\text{in}} \times \mathcal{V}_{\text{out}}$ **do**
6      $w(v_{\text{in}}, v_{\text{out}}) \leftarrow w(v_{\text{in}}, v_{\text{RIP}}) + w(v_{\text{RIP}}, v_{\text{out}})$

A.2. Learned Edge Weights Statistics

Table A.3. Key statistics of the learned edge weights, accumulated across folds.

STATISTIC	DOWNSTREAM		UPSTREAM		BIDIRECTIONAL	
	RESGCN	GCNII	RESGCN	GCNII	RESGCN	GCNII
MEAN	0.462 ± 0.082	0.263 ± 0.072	0.670 ± 0.039	0.627 ± 0.056	0.789 ± 0.044	0.618 ± 0.062
STD	0.322 ± 0.013	0.281 ± 0.038	0.375 ± 0.004	0.369 ± 0.013	0.329 ± 0.008	0.361 ± 0.005
MIN	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.061 ± 0.045	0.000 ± 0.000
25%	0.191 ± 0.086	0.033 ± 0.033	0.382 ± 0.049	0.345 ± 0.066	0.556 ± 0.039	0.342 ± 0.079
MEDIAN	0.416 ± 0.084	0.158 ± 0.091	0.708 ± 0.064	0.628 ± 0.075	0.802 ± 0.049	0.592 ± 0.066
75 %	0.689 ± 0.102	0.434 ± 0.092	0.959 ± 0.036	0.894 ± 0.078	1.032 ± 0.040	0.901 ± 0.075
MAX %	1.313 ± 0.089	1.376 ± 0.094	1.471 ± 0.052	1.609 ± 0.102	1.565 ± 0.037	1.668 ± 0.083

A.3. Subnetwork Results

Table A.4. Cross-validation $\overline{\text{NSE}}$ on subgraph (i) in Figure 4.

(a) ResGCN

ADJACENCY TYPE	EDGE ORIENTATION		
	DOWNSTREAM	UPSTREAM	BIDIRECTED
ISOLATED	75.07 % ±7.25 %	75.07 % ±7.25 %	75.07 % ±7.25 %
BINARY	82.77 % ±1.26 %	81.20 % ±3.68 %	77.61 % ±4.43 %
STREAM LENGTH	82.77 % ±1.26 %	81.20 % ±3.68 %	77.61 % ±4.43 %
ELEVATION DIFFERENCE	81.76 % ±3.11 %	81.42 % ±3.74 %	77.31 % ±4.18 %
AVERAGE SLOPE	81.76 % ±3.11 %	81.42 % ±3.74 %	77.31 % ±4.18 %
LEARNED	81.82 % ±5.63 %	81.61 % ±2.63 %	75.47 % ±7.02 %

(b) GCNII

ADJACENCY TYPE	EDGE ORIENTATION		
	DOWNSTREAM	UPSTREAM	BIDIRECTED
ISOLATED	80.44 % ±3.16 %	80.44 % ±3.16 %	80.44 % ±3.16 %
BINARY	80.52 % ±4.53 %	74.57 % ±2.27 %	81.16 % ±2.97 %
STREAM LENGTH	80.52 % ±4.53 %	74.57 % ±2.27 %	81.16 % ±2.97 %
ELEVATION DIFFERENCE	78.22 % ±4.74 %	74.42 % ±4.53 %	80.30 % ±4.11 %
AVERAGE SLOPE	78.22 % ±4.74 %	74.42 % ±4.53 %	80.30 % ±4.11 %
LEARNED	80.82 % ±4.05 %	75.67 % ±4.29 %	79.99 % ±7.83 %

(c) ResGAT

ADJACENCY TYPE	EDGE ORIENTATION		
	DOWNSTREAM	UPSTREAM	BIDIRECTED
ISOLATED	76.81 % ±9.40 %	76.81 % ±9.40 %	76.81 % ±9.40 %
BINARY	82.28 % ±6.65 %	83.85 % ±3.56 %	77.47 % ±6.20 %
ALL OF THE BELOW	75.60 % ±7.43 %	76.56 % ±10.01 %	75.88 % ±9.85 %
STREAM LENGTH	82.28 % ±6.65 %	83.85 % ±3.56 %	77.47 % ±6.20 %
ELEVATION DIFFERENCE	82.28 % ±6.65 %	83.85 % ±3.56 %	77.47 % ±6.20 %
AVERAGE SLOPE	82.28 % ±6.65 %	83.85 % ±3.56 %	77.47 % ±6.20 %

Table A.5. Cross-validation $\overline{\text{NSE}}$ on subgraph (ii) in Figure 4.

(a) ResGCN

ADJACENCY TYPE	EDGE ORIENTATION		
	DOWNSTREAM	UPSTREAM	BIDIRECTED
ISOLATED	93.20 % ±0.23 %	93.20 % ±0.23 %	93.20 % ±0.23 %
BINARY	92.61 % ±5.50 %	95.36 % ±1.45 %	95.40 % ±1.80 %
STREAM LENGTH	92.71 % ±5.55 %	94.82 % ±1.80 %	94.30 % ±0.74 %
ELEVATION DIFFERENCE	92.79 % ±5.56 %	94.92 % ±1.49 %	95.03 % ±1.81 %
AVERAGE SLOPE	92.68 % ±5.56 %	95.28 % ±0.94 %	95.25 % ±0.58 %
LEARNED	92.69 % ±5.79 %	96.17 % ±0.25 %	96.12 % ±0.92 %

(b) GCNII

ADJACENCY TYPE	EDGE ORIENTATION		
	DOWNSTREAM	UPSTREAM	BIDIRECTED
ISOLATED	91.95 % ±1.89 %	91.95 % ±1.89 %	91.95 % ±1.89 %
BINARY	95.61 % ±1.18 %	96.01 % ±0.70 %	93.65 % ±3.67 %
STREAM LENGTH	95.50 % ±1.64 %	95.80 % ±0.88 %	93.05 % ±2.57 %
ELEVATION DIFFERENCE	95.06 % ±1.03 %	96.24 % ±0.75 %	92.48 % ±3.43 %
AVERAGE SLOPE	95.58 % ±1.66 %	96.05 % ±0.55 %	95.23 % ±0.52 %
LEARNED	94.78 % ±1.27 %	94.84 % ±1.54 %	94.23 % ±0.33 %

(c) ResGAT

ADJACENCY TYPE	EDGE ORIENTATION		
	DOWNSTREAM	UPSTREAM	BIDIRECTED
ISOLATED	92.36 % ±0.96 %	92.36 % ±0.96 %	92.36 % ±0.96 %
BINARY	93.40 % ±4.02 %	95.08 % ±1.33 %	94.43 % ±2.11 %
ALL OF THE BELOW	94.78 % ±0.81 %	93.08 % ±2.80 %	94.47 % ±3.76 %
STREAM LENGTH	93.40 % ±4.02 %	95.08 % ±1.33 %	94.43 % ±2.11 %
ELEVATION DIFFERENCE	93.40 % ±4.02 %	95.08 % ±1.33 %	94.43 % ±2.11 %
AVERAGE SLOPE	93.40 % ±4.02 %	95.08 % ±1.33 %	94.43 % ±2.11 %

Table A.6. Cross-validation $\overline{\text{NSE}}$ on subgraph (iii) in Figure 4.

(a) ResGCN

ADJACENCY TYPE	EDGE ORIENTATION		
	DOWNSTREAM	UPSTREAM	BIDIRECTED
ISOLATED	78.12 % ±3.80 %	78.12 % ±3.80 %	78.12 % ±3.80 %
BINARY	79.23 % ±3.07 %	81.77 % ±1.30 %	78.10 % ±1.66 %
STREAM LENGTH	79.23 % ±3.07 %	81.66 % ±1.18 %	76.05 % ±2.27 %
ELEVATION DIFFERENCE	79.23 % ±3.07 %	81.81 % ±1.34 %	76.66 % ±2.48 %
AVERAGE SLOPE	79.23 % ±3.07 %	81.62 % ±1.05 %	77.51 % ±2.02 %
LEARNED	77.72 % ±4.22 %	82.39 % ±1.35 %	77.89 % ±3.52 %

(b) GCNII

ADJACENCY TYPE	EDGE ORIENTATION		
	DOWNSTREAM	UPSTREAM	BIDIRECTED
ISOLATED	79.58 % ±1.78 %	79.58 % ±1.78 %	79.58 % ±1.78 %
BINARY	78.64 % ±2.26 %	77.31 % ±0.86 %	77.60 % ±2.24 %
STREAM LENGTH	77.52 % ±4.23 %	75.38 % ±1.89 %	78.92 % ±1.52 %
ELEVATION DIFFERENCE	77.52 % ±4.23 %	76.93 % ±2.07 %	76.93 % ±0.90 %
AVERAGE SLOPE	78.64 % ±2.26 %	76.13 % ±2.35 %	79.46 % ±1.23 %
LEARNED	78.81 % ±2.84 %	76.84 % ±1.38 %	79.04 % ±2.30 %

(c) ResGAT

ADJACENCY TYPE	EDGE ORIENTATION		
	DOWNSTREAM	UPSTREAM	BIDIRECTED
ISOLATED	81.21 % ±2.70 %	81.21 % ±2.70 %	81.21 % ±2.70 %
BINARY	76.35 % ±4.39 %	76.90 % ±1.48 %	77.30 % ±3.02 %
ALL OF THE BELOW	80.69 % ±2.65 %	73.19 % ±11.11 %	75.19 % ±2.83 %
STREAM LENGTH	76.35 % ±4.39 %	76.90 % ±1.48 %	77.30 % ±3.02 %
ELEVATION DIFFERENCE	76.35 % ±4.39 %	76.90 % ±1.48 %	77.30 % ±3.02 %
AVERAGE SLOPE	76.35 % ±4.39 %	76.90 % ±1.48 %	77.30 % ±3.02 %

Table A.7. Cross-validation $\overline{\text{NSE}}$ on subgraph (iv) in Figure 4.

(a) ResGCN

ADJACENCY TYPE	EDGE ORIENTATION		
	DOWNSTREAM	UPSTREAM	BIDIRECTED
ISOLATED	96.00 % ±0.43 %	96.00 % ±0.43 %	96.00 % ±0.43 %
BINARY	94.99 % ±0.25 %	96.20 % ±0.94 %	95.58 % ±0.59 %
STREAM LENGTH	95.17 % ±0.27 %	96.10 % ±1.12 %	95.68 % ±0.54 %
ELEVATION DIFFERENCE	95.01 % ±0.24 %	96.18 % ±0.88 %	95.71 % ±0.87 %
AVERAGE SLOPE	95.14 % ±0.24 %	96.09 % ±1.02 %	95.20 % ±0.85 %
LEARNED	95.11 % ±0.15 %	96.34 % ±0.87 %	95.72 % ±0.70 %

(b) GCNII

ADJACENCY TYPE	EDGE ORIENTATION		
	DOWNSTREAM	UPSTREAM	BIDIRECTED
ISOLATED	95.90 % ±0.09 %	95.90 % ±0.09 %	95.90 % ±0.09 %
BINARY	96.05 % ±0.12 %	96.18 % ±0.45 %	96.02 % ±0.52 %
STREAM LENGTH	95.93 % ±0.13 %	96.26 % ±0.46 %	96.12 % ±0.60 %
ELEVATION DIFFERENCE	96.05 % ±0.08 %	96.09 % ±0.48 %	96.14 % ±0.57 %
AVERAGE SLOPE	96.02 % ±0.06 %	96.15 % ±0.52 %	95.92 % ±0.63 %
LEARNED	95.86 % ±0.25 %	96.22 % ±0.63 %	96.21 % ±0.37 %

(c) ResGAT

ADJACENCY TYPE	EDGE ORIENTATION		
	DOWNSTREAM	UPSTREAM	BIDIRECTED
ISOLATED	96.01 % ±0.34 %	96.01 % ±0.34 %	96.01 % ±0.34 %
BINARY	95.19 % ±0.09 %	96.50 % ±0.06 %	95.90 % ±0.39 %
ALL OF THE BELOW	96.17 % ±0.26 %	96.04 % ±0.26 %	95.89 % ±0.04 %
STREAM LENGTH	95.19 % ±0.09 %	96.50 % ±0.06 %	95.90 % ±0.39 %
ELEVATION DIFFERENCE	95.19 % ±0.09 %	96.50 % ±0.06 %	95.90 % ±0.39 %
AVERAGE SLOPE	95.19 % ±0.09 %	96.50 % ±0.06 %	95.90 % ±0.39 %

A.4. Effect of Window Size and Lead Time

Table A.8. Cross-validation $\overline{\text{NSE}}$ of bidirected-learned GCNII for different window sizes and lead times. Results generally improve with larger window size and smaller lead time.

WINDOW SIZE [h]	LEAD TIME [h]					
	1	2	3	6	9	12
12	98.87 % $\pm0.05\,\%$	96.28 % $\pm0.16\,\%$	92.99 % $\pm0.63\,\%$	82.35 % $\pm1.94\,\%$	72.51 % $\pm2.01\,\%$	63.16 % $\pm9.19\,\%$
24	99.05 % $\pm0.06\,\%$	96.89 % $\pm0.04\,\%$	94.20 % $\pm0.52\,\%$	85.59 % $\pm1.41\,\%$	75.98 % $\pm2.84\,\%$	67.26 % $\pm5.12\,\%$
36	99.04 % $\pm0.04\,\%$	97.03 % $\pm0.15\,\%$	94.53 % $\pm0.24\,\%$	85.51 % $\pm2.84\,\%$	78.54 % $\pm3.65\,\%$	70.52 % $\pm3.01\,\%$
48	99.03 % $\pm0.06\,\%$	97.03 % $\pm0.12\,\%$	94.77 % $\pm0.20\,\%$	87.57 % $\pm1.21\,\%$	79.72 % $\pm2.12\,\%$	74.08 % $\pm2.62\,\%$
60	98.99 % $\pm0.06\,\%$	96.89 % $\pm0.13\,\%$	94.61 % $\pm0.10\,\%$	86.66 % $\pm1.50\,\%$	81.01 % $\pm2.50\,\%$	75.82 % $\pm2.93\,\%$
72	99.02 % $\pm0.03\,\%$	96.97 % $\pm0.02\,\%$	94.65 % $\pm0.31\,\%$	87.59 % $\pm1.12\,\%$	80.25 % $\pm1.85\,\%$	75.52 % $\pm1.73\,\%$