Hydrol. Earth Syst. Sci., 26, 5163–5184, 2022
https://doi.org/10.5194/hess-26-5163-2022

Hydrology and Earth System Sciences
Open Access

A graph neural network (GNN) approach to basin-scale river network learning: the role of physics-based connectivity and data fusion

Alexander Y. Sun[1], **Peishi Jiang**[2], **Zong-Liang Yang**[3], **Yangxinyu Xie**[4,a], **and Xingyuan Chen**[2]

[1]Bureau of Economic Geology, The University of Texas at Austin, Austin, TX, USA
[2]Pacific Northwest National Laboratory, Richland, WA, USA
[3]Department of Geological Sciences, The University of Texas at Austin, Austin, TX, USA
[4]Department of Computer Science, The University of Texas at Austin, Austin, TX, USA
[a]now at: The Department of Statistics and Data Science at The University of Pennsylvania, Pennsylvania, USA

Correspondence: Alexander Y. Sun (alex.sun@beg.utexas.edu)

Received: 25 March 2022 – Discussion started: 25 April 2022
Revised: 6 September 2022 – Accepted: 13 September 2022 – Published: 14 October 2022

Abstract. Rivers and river habitats around the world are under sustained pressure from human activities and the changing global environment. Our ability to quantify and manage the river states in a timely manner is critical for protecting the public safety and natural resources. In recent years, vector-based river network models have enabled modeling of large river basins at increasingly fine resolutions, but are computationally demanding. This work presents a multistage, physics-guided, graph neural network (GNN) approach for basin-scale river network learning and streamflow forecasting. During training, we train a GNN model to approximate outputs of a high-resolution vector-based river network model; we then fine-tune the pretrained GNN model with streamflow observations. We further apply a graph-based, data-fusion step to correct prediction biases. The GNN-based framework is first demonstrated over a snow-dominated watershed in the western United States. A series of experiments are performed to test different training and imputation strategies. Results show that the trained GNN model can effectively serve as a surrogate of the process-based model with high accuracy, with median Kling–Gupta efficiency (KGE) greater than 0.97. Application of the graph-based data fusion further reduces mismatch between the GNN model and observations, with as much as 50 % KGE improvement over some cross-validation gages. To improve scalability, a graph-coarsening procedure is introduced and is demonstrated over a much larger basin. Results show that graph coarsening achieves comparable prediction skills at only a fraction of training cost, thus providing important insights into the degree of physical realism needed for developing large-scale GNN-based river network models.

1 Introduction

Rivers play a critical role in the hydrosphere, enabling and regulating hydrological, geomorphic and ecological processes along and adjacent to the riverine environment (De Groot et al., 2002; Dai and Trenberth, 2002). Rivers around the world are also under sustained pressure from human activities, losing free-flowing connectivity over time due to the construction of dams, levees and other hydro-infrastructures for providing societal goods and services (e.g., hydropower generation) (Best, 2019). About half of the global river reaches now show signs of diminished connectivity (Grill et al., 2019). The changing climate and population growth further exacerbate the existing stress on river systems by modulating the spatial and temporal patterns of floods and droughts, including their frequency, magnitude and timing (Winsemius et al., 2016; Blöschl et al., 2017; Dottori et al., 2018), and by reducing the rivers' natural ability to absorb disturbances and buffer the ecosystem (Palmer et al., 2008). Thus, our ability to quantify and manage the river states and fluxes in a timely manner has become more important than ever for protecting the public safety and adapting to the changing environment. Stream gauges can provide direct measures of river discharges, but the utility of which is hindered by the poor coverage of gauge networks in many parts of the world. Instead, hydrological models are broadly used to predict river discharges at ungauged locations (Hrachowitz et al., 2013; Beck et al., 2015).

The fidelity of a hydrological model hinges on a number of factors, including the accuracy of forcing data, soundness of process parameterization, and the realism of river network geometry used in the model (Bierkens et al., 2015). In many distributed and semi-distributed models, river networks are delineated on the same grid as the underlying hydrological models or land surface models (Döll et al., 2003; Van Beek et al., 2011; Pokhrel et al., 2012; Alfieri et al., 2013). Prescribing river networks over coarse resolution grids may introduce misrepresentation in river-routing models, leading to inaccurate river discharge estimates and misalignment with local application needs (Zhao et al., 2017; Mizukami et al., 2021). While representation of river networks can always be improved by refining the grid resolution, in reality, such effort is often constrained by computational resources, especially for large-scale simulations (Yamazaki et al., 2013; Bierkens et al., 2015).

In the past decade, vector hydrography has (re)emerged as an alternative to the conventional grid-based approaches for large river basin simulations (David et al., 2011; Lehner and Grill, 2013; Yamazaki et al., 2013; Lin et al., 2018). In a vector network representation, the land surface of a river basin is discretized into unit catchments (polygons) and the river reaches connecting them by using a high-resolution digital elevation model (Saunders, 2000). Unlike the grid-based representation, in a vector-based river model, the unit catchments serve as calculation units, and the time evolution of state variables is solved by calculating the flux exchange between each unit catchment and the next downstream unit catchment along a prescribed river network (Yamazaki et al., 2013). Direct benefits of such a discretization scheme include the following: (a) river geometries are realistically represented; (b) river reach distances are relatively evenly distributed, allowing the use of greater time steps and thus gaining more computational efficiency; (c) the unit catchment resolution can easily be improved by applying more detailed polygon outlines; and (d) hydrologic features such as lakes and irrigation lines can be modeled as additional vector features attached to the catchments and river reaches (Mizukami et al., 2021).

The vector-based river representation is behind the National Water Model (NWM), which operationally generates short- and long-range river discharge forecasts for more than 2.7 million reaches in the United States (Lin et al., 2018; Salas et al., 2018). Vector-based river network simulations were also used to create validation datasets for the upcoming Surface Water and Ocean Topography (SWOT) satellite mission, designed to provide global observations of changing water levels in large rivers, lakes and floodplains (Biancamaria et al., 2016; Lin et al., 2019). Despite their improved efficiency, vector-based river-routing models are still computationally demanding, requiring special domain decomposition techniques for parallel computation (David et al., 2011; Mizukami et al., 2021).

The renewed interests in vector hydrography have been driven by an increased demand for hyperresolution terrestrial hydrology models on the one hand (Bierkens et al., 2015), and the deluge of high-resolution Earth observation data on the other (Sun and Scanlon, 2019; Reichstein et al., 2019). Ultimately, the Earth science community envisions the development of Earth system digital twins, aiming to provide digital replica of the real world through high-fidelity simulations and observations (Bauer et al., 2021). Toward that vision, two active veins of research are currently underway. One is continuously improving the physical realism of process representation within the current Earth system models at all scales and across all subsystem interfaces, which is a daunting task even with today's extreme-scale, high-performance computing power (Schulthess et al., 2018). The other is augmenting process-based models with artificial intelligence/machine learning (AI/ML) techniques, which has attracted significant attention in the past several years (Shen, 2018; Sun and Scanlon, 2019; Reichstein et al., 2019; Camps-Valls et al., 2021; Kashinath et al., 2021; Pathak et al., 2022). Existing AI/ML works related to hydroclimate modeling may be classified as (a) data-driven techniques (Kratzert et al., 2019b; Le et al., 2021; Sun et al., 2014, 2021; Feng et al., 2021); (b) hybrid process-based/ML models (Rasp et al., 2018; Yuval and O'Gorman, 2020; Nonnenmacher and Greenberg, 2021); and (c) physics-guided, post-processing (PGPP) techniques (Ham et al., 2019; Sun et al., 2019; Yang et al., 2019; Feng et al., 2020; Willard et al., 2020; Lu et al., 2021; Kashinath et al., 2021; Pathak et al., 2022). Although the latter two categories are sometimes combined in the literature, in the context of this discussion, the former category is seen as providing ML-based parameterization schemes (e.g., subgrid processes) within a process-based model (Rasp et al., 2018; Reichstein et al., 2019), while the latter category mainly leverages outputs of process-based models and first-order physics principles in an ad hoc manner. The ML-based PGPP methods can provide added values to existing process models, such as improved predictability, reduced bias, and/or computational efficiency, without requiring significant modifications to the existing scientific computing workflows and codes. This largely explains the popularity of the PGPP paradigm in the Earth science community.

So far, only a few studies have exploited the use of ML in vector hydrography. This is partly because many of the deep-learning techniques that are in use today originated from the computer vision and natural language processing literature, dealing mostly with gridded or sequence data. On the other hand, many types of data in natural and social sciences, such as weather stations, river networks, and social networks, are characterized by graph-like data structures (i.e., nodes and node links) that do not conform to the

Euclidean geometry. Various graph neural network (GNN) models have been developed (Bruna et al., 2013; Kipf and Welling, 2016; Bronstein et al., 2017; Zhou et al., 2018) to learn the graph-structured data. Like their counterpart for learning image-like data (e.g., convolutional neural networks or CNNs), GNNs are designed to extract high-level features from input data through the so-called neural message passing and aggregation process, consisting of a series of algebraic operations to progressively encode the features of nodes and their local structures (e.g., the number of neighbors) as latent representations (i.e., low-dimensional embeddings) (Kipf and Welling, 2016; Hamilton et al., 2017). Parameters of the embeddings are then learned from the training data through back propagation. The GNNs, when combined with temporal learning algorithms (e.g., recurrent neural networks or RNNs), comprise powerful tools for disentangling highly complex, spatial and temporal relationships. We point out that the graph theory, which is at the foundation of GNNs, has been widely applied in geosciences to model geological processes, e.g., formation of river deltas (Phillips et al., 2015; Tejedor et al., 2018). However, GNNs focus more on learning the latent-space representations for downstream tasks (e.g., classification and regression) than abstracting graph-level statistics, as commonly done under most graph theoretic studies (Hamilton et al., 2017).

Jia et al. (2021) used a physics-guided, recurrent graph convolution network (GCN) model to predict streamflow and water temperature in a catchment of the Delaware River basin in the United States. In their work, physics constraints are imposed in multiple ways: the actual river reach lengths and upstream/downstream connections are used to construct a weighted adjacency matrix; a process-based streamflow and water temperature model is used to generate synthetic training samples for ungauged reaches; and finally, data collected from gauged locations are used in an extra loss term to enforce physical consistency with observations. The authors reported that their recurrent GCN model generally gives better performance than a baseline RNN model, but may yield large errors at unobserved river reaches and reaches with extremely low flows. Sun et al. (2021) adapted and compared the performance of several recurrent GNN architectures for predicting streamflows of basins in the Catchment Attributes and Meteorology for Large-Sample Studies (CAMELS) dataset, which includes meteorological forcing, basin static attributes, and observed streamflow time series for 671 basins in the conterminous United States (Newman et al., 2015). The GNN algorithms that they investigated include several basic GNNs, such as the GCN (Kipf and Welling, 2016) and ChebNet (Defferrard et al., 2016), as well as a more complex GNN model, the GraphWavenet (GWN for short) (Wu et al., 2019). They used hydro-similarity as a distance measure to connect the spatially scattered basins, treating each basin and its static attributes as node and node features. Their results show the GWN gives the best overall performance, while models constructed using basic GNN layers perform worse than a baseline model trained using the long short-term memory (LSTM) network (Hochreiter and Schmidhuber, 1997). Chen et al. (2021) adopted a heterogeneous recurrent graph model to predict stream temperatures, in which river reaches and dams are represented as separate graphs. For each river reach, the authors used gating variables to control the information flow from upstream river reaches and reservoirs, in addition to the antecedent states of the river reach itself. Similar to (Jia et al., 2021), observations are incorporated directly in an extra loss term during training. These recent studies have demonstrated the potential of GNNs for vector-based river modeling. Specifically, GNNs may allow fine-grained control of information exchanges at the node and edge levels for incorporating the physical realism, an aspect that is missing in many other ML algorithms commonly used to model spatiotemporal datasets (e.g., random forests and RNNs). In the remainder of this discussion, we shall use node and reach interchangeably when streamflow in the reach is implied.

Despite their potential promise, remaining questions pertaining to GNN applications in the vector hydrography are (a) the degree of physical realism that is needed for a GNN model to learn river network representations, (b) the generalization and scalability of GNN models and (c) data fusion. To understand these research needs, we highlight the main differences between river networks and many other types of networks. River networks are hierarchical, with downstream discharges reflecting the integrated hydrologic contributions from all upstream reaches and catchments (Weiler et al., 2003). Reaches adjacent to each other tend to share similar runoff generation processes because of similarities in catchment physiographic properties and meteorological forcing. These unique aspects of river networks imply that information passing in a river network should be multi-directional, rather than strictly following the network topography as prescribed through the physics-based connectivity. Identifying and modeling the multi-directional and heterogeneous information-transfer mechanisms using GNNs is an active research topic (Zhou et al., 2018). Perhaps a more practical question is related to graph generalization and scalability. The river networks considered in Jia et al. (2021) and Chen et al. (2021) include 42 and 56 river reaches, respectively, which are relatively small. In comparison, a typical basin at the US Geological Survey's (USGS) 8-digit hydrological unit code (HUC-8) level contains $\mathcal{O}(10^2 - 10^3)$ river reaches, and at the largest HUC-2 regional basin level, the river network may include $\mathcal{O}(10^4 - 10^5)$ reaches (Simley and Carswell Jr., 2009). State-of-the-art GNN models have demonstrated capabilities to handle large-scale graphs containing $\mathcal{O}(10^8)$ nodes on classification problems (Hu et al., 2020; Wang et al., 2021). Learning large-scale, vector river networks, however, is still a challenging topic because of the dynamic nature of the problem and computing memory requirements. A fruitful research direction may be exploring the graph sparsification or coarsening in a way that preserves the spatiotemporal structure of the river system. Finally, data fusion on graph-based, river network models has not been systematically studied, but represents an important class of post-processing techniques for watershed modeling. In general, post-processing techniques, as their names suggest, attempt to refine model outputs using new observations obtained after model simulations. In streamflow forecasting, post-processors have been used to correct biases and dispersion errors in raw forecasts, downscale forecasts to the scale of applications, and generate forecast ensembles that preserve the spatiotemporal structure of river discharges (Todini, 2008; Weerts et al., 2011; Li et al., 2017). Unlike data assimilation, data fusion as defined and used in the context of this work is generally decoupled from the original model.

In light of the aforementioned challenges, this study was conducted with the following two objectives in mind, namely, (a) to evaluate the role of physics-based connectivity in the GNN river network surrogate modeling, and (b) to adapt and investigate the efficacy of a graph-based data-fusion technique. Main contributions of this work are that we have developed a methodology consisting of pretraining, fine-tuning and data-fusion steps to significantly improve the performance of GNN models; we show that the degree of realism required for a GNN surrogate model to catch spatiotemporal basin flow dynamics largely depends on the parameter structure of the underlying physics-based model. The remainder of this work is organized as follows. In Sect. 2, we describe data and data-processing techniques used in this study. Section 3 focuses on the theoretical background of GNN and data-fusion algorithms. Section 4 describes the demonstration study area and experimental design, which are followed by results and scaling up analysis in Sects. 5 and 6, and conclusions in the last section.

2 Data and data processing

2.1 National water model (NWM)

The NWM is a continental-scale, distributed, hydrological modeling framework implemented and operated by the US National Weather Service for providing short-range (18 h), medium-range (10 d) and long-range (30 d) streamflow forecasts in the United States (Cosgrove et al., 2016). It is based on the WRF-Hydro community model, which is both a standalone model and a coupling architecture to facilitate the exchange between the Weather Research and Forecasting (WRF) atmospheric model and components of a land surface model (e.g., surface runoff, channel flow, lake/reservoir flow, subsurface flow, and land–atmosphere exchanges) (Gochis et al., 2018). The WRF-Hydro model supports surface runoff routing over vector-based river networks. The network topology used in the NWM is derived from the US National Hydrography Dataset Plus (NHDPlus), a georeferenced, hydrologic dataset incorporating a national stream network with a scale of 1 : 100 000 and a 30 m national digital elevation dataset, in addition to a large number of river and catchment attributes for enhancing network analyses (McKay et al., 2012; Moore and Dewald, 2016). We primarily used NWM v2.0 retrospective simulation data that were available when this study was initiated. The NWM v2.0 contains NWM retrospective simulation outputs in hourly time steps at 2 729 076 NHDPlus reaches for the period 1 January 1993–31 December 2018. As part of the sensitivity analysis, we also tested our approaches on NWM v2.1 retrospective data, which cover a longer 42-year period (1 February 1979–31 December 2020). A subset of NWM parameters are calibrated by model developers using historical streamflow data at limited basins, but no nudging is applied on the retrospective runs (Cosgrove et al., 2016). The NWM v2.0 streamflow data are downloaded from a data server hosted by Hydroshare (Johnson and Blodgett, 2020), while NWM 2.1 streamflow data are downloaded from the National Oceanic and Atmospheric Administration's (NOAA) Amazon Web Services data repository (NOAA, 2022). Python scripts are used to automate the remote subsetting and downloading of all NWM streamflow data for any US basin of interest. All data are aggregated into daily steps.

2.2 Meteorological forcing and streamflow

The NWM v2.0 is driven by forcing data resampled from the North American Land Data Assimilation System (NLDAS), which is originally available on a 1/8° grid ($\sim$ 14 km at the Equator) (Xia et al., 2012), while NWM v2.1 is driven by the 1 km Analysis of Record for Calibration (AORC) dataset that was not publicly available at the time of this writing (Kitzmiller et al., 2018). In this study, we used Daymet, which provides gauge-based, gridded estimates of daily weather and climatology variables over the continental North America, including daily minimum and maximum temperature, precipitation, vapor pressure, shortwave radiation, snow water equivalent (Thornton et al., 2012). The spatial resolution of Daymet is 1 km × 1 km and the temporal coverage is from 1950 through the end of the most recent full calendar year. Although built upon similar gauge data, Daymet data are likely different from the meteorological forcing data used in the NWM because of the different interpolation and extrapolation schemes used to create them. Combining Daymet with antecedent NWM outputs as predictors may thus indirectly achieve the effect of using multiple forcing data, which have been shown to improve the generalization skill of Earth science ML models (Sun et al., 2019; Kratzert et al., 2021). The Daymet data are downloaded by programmatically calling the Daymet web services (ORNL, 2022) and getting data closest to the centroid of each reach in a river network. Streamflow gage data are downloaded from USGS' National Water Information System by using the USGS Python package for water data retrieval (USGS, 2022a).

2.3 River network construction

River network for a basin under study may be extracted from the NHDPlus database by performing the following steps. First, the NHDPlus v2.1 geodatabase covering the basin is downloaded from the US Environmental Protection Agency (EPA) data server (EPA, 2022). The basin mask is used to crop the NHDFlow shape layer included in the NHDPlus geodatabase, which is then joined with the PlusFlow table, also from NHDPlus. After this step, we have all reach attributes, including the identification number of each river reach (referred to as COMID in NHDPlus), the upstream/-downstream reaches of each reach, the reach type (e.g., stream or artificial path) and reach length, that are necessary for building a river network and populating the node features. We used a Python script to recursively traverse all river reaches to gather reach attributes and build the network (i.e., in terms of graph adjacency matrix). The reach COMIDs corresponding to USGS gauge locations are also obtained for mapping purposes. All watershed boundaries used in this study are extracted from the Watershed Boundary Dataset, which includes basin boundaries at various HUC levels (USGS, 2022b). For the graph-coarsening demonstration, we used the pour points corresponding to the HUC-12 basins (Price, 2022).

3 Methodology

We start by introducing some notations. A graph is represented by $\mathcal{G}(\mathcal{V},\mathcal{E})$, where $\mathcal{V}=\{v_i\}_{i=1}^{N}$ is a set of N nodes and $\mathcal{E}=\{e_{ij}\}$ is a set of edges connecting node pairs (v_i, v_j) for $v_i \in \mathcal{V}$ and $v_j \in \mathcal{V}$. The neighborhood of a node is a subset of nodes connected to it, $\mathcal{N}(v)=\{u \in \mathcal{V} \mid (u,v) \in \mathcal{E}\}$. Node connections are specified by the adjacency matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$, of which an element a_{ij} is equal to 1 if nodes v_i and v_j are connected and 0 otherwise. A graph can be either undirected (edge is bidirectional) or directed (edge direction matters). The adjacency matrix may also be weighted, in which case elements of $\mathbf{A}$ would be decimal numbers describing the affinity or similarity between two nodes. The graph feature matrix is denoted as $\mathbf{X} \in \mathbb{R}^{N\times D}$, with its rows representing node feature vectors $\boldsymbol{x}_i \in \mathbb{R}^D, i=1,\ldots,N$. In the dynamic setting considered in this work, the node features vary with time and the graph feature matrix is denoted by $\mathbf{X}^t$.

To develop a GNN-based, end-to-end framework for vector-based river network modeling, we propose a three-stage workflow as shown in Fig. 1. In Stage I, a GNN-based model is trained through supervised learning by using the meteorological forcing data and NWM outputs. This pretraining step is essentially a surrogate modeling process that learns the spatiotemporal rainfall–runoff patterns governed by the NWM and meteorological forcing. The predictors include the six meteorological forcing variables from Daymet and the NWM-simulated streamflow for a look-back period of T_B, as well as the adjacency matrix $\mathbf{A}$ describing the river network connectivity. A neural network is trained to approximate the mapping $\mathcal{F}(\mathcal{X}^t, \mathbf{y}^t)$, where $\mathcal{X}^t$ is a collection of predictors $\{\mathbf{X}^i\}_{i=t-1}^{t-T_\mathrm{B}}$, $\mathbf{y}^t \in \mathbb{R}^N$ is the NWM outputs for the entire river network, and the forecast can be done for T_F steps in the future. In Stage II, a fine-tuning step is used to correct the pretrained GNN model by utilizing historical streamflow data available in the training period. The trained model can then be deployed for online prediction in Stage III, which uses data fusion to further correct GNN predictions based on residuals between predictions and observations. We describe these steps in detail below.

3.1 Graph-based surrogate modeling for river networks

The GNN surrogate modeling framework used in this work is adapted from GraphWaveNet (GWN) (Wu et al., 2019), which consists of two types of interleaved layers, namely, GNN layers for spatial learning and temporal convolution network (TCN) layers for temporal learning. A schematic plot of the adapted GWN design is provided in Fig. 2.

In general, the graph-based learning seeks to learn a low-dimensional representation of input data through recurrent information aggregation and propagation steps. At the node level, a basic GNN layer may be written as (Bronstein et al., 2021)

$$\mathbf{h}_v^{(k)} = f\left(\mathbf{h}_v^{(k-1)}, \bigoplus_{u\in\mathcal{N}_v} \left(\psi(\mathbf{h}_u, \mathbf{h}_v)\right)\right), \tag{1}$$

where $\mathbf{h}_v^{(k)} \in \mathbb{R}^{D_k}$ denotes the embedding or hidden state of node v at the kth layer, D_k is the output dimension of the kth hidden layer and $\mathbf{h}_v^{(0)} = \boldsymbol{x}_v$; $\oplus$ is an aggregation operator; ψ is a learnable function parameterizing the messaging passing between node v and its neighbors $u \in \mathcal{N}_v$; and $f(\cdot)$ is an activation function (e.g., the rectified linear unit function or ReLU). Most GNNs differ by how $\oplus$ and ψ are chosen. For example, in the GCN layer that is used within GWN, the following aggregation scheme is used: (Kipf and Welling, 2016)

$$\mathbf{h}_v^{(k)} = f\left(\mathbf{W}_1^{(k)}\mathbf{h}_v^{(k-1)} + \mathbf{W}_2^{(k)} \sum_{u\in\mathcal{N}_v} \mathbf{h}_u^{(k-1)} + \mathbf{b}^{(k)}\right), \tag{2}$$

where the weighted sum of the hidden states of neighboring nodes (2nd term in parentheses) is added to the node's embedding from the previous layer (first term in parentheses). Here $\mathbf{W}_1^{(k)}$ and $\mathbf{W}_2^{(k)}$, both $\in \mathbb{R}^{D_k\times D_{k-1}}$, are learnable weight matrices that determine the influences of node features and node neighbors, respectively; and $\mathbf{b}$ is a learnable bias term added to improve training. At the graph level, the graph convolution operation in Eq. (2) may be written in a simplified matrix form as (Kipf and Welling, 2016)

$$\mathbf{H}^{(k)} = f\left(\tilde{\mathbf{A}}\mathbf{H}^{(k-1)}\mathbf{W}^{(k)}\right), \tag{3}$$

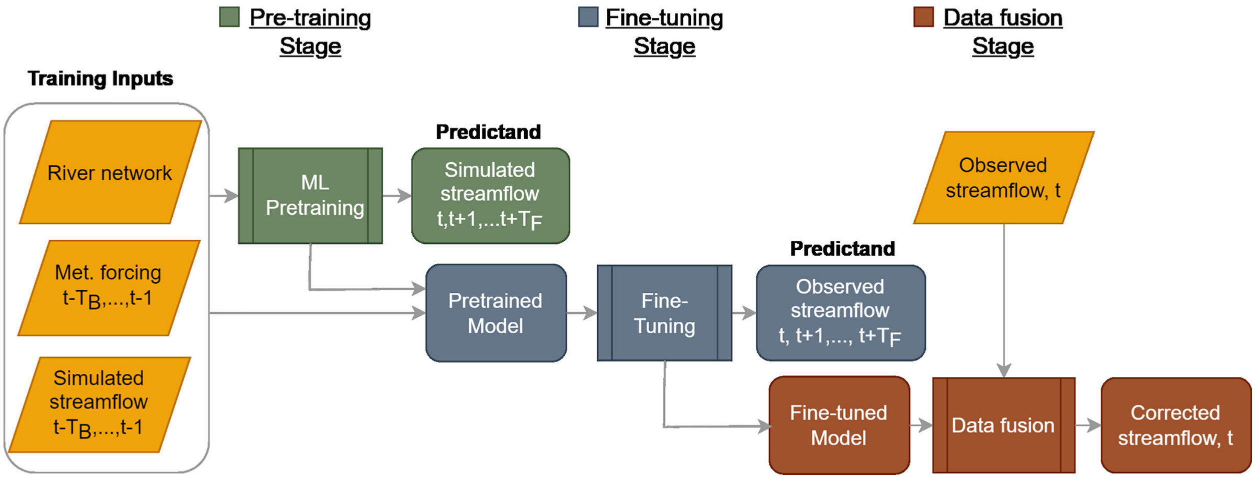

Figure 1. An ML-based workflow for basin-scale streamflow forecasting. The workflow consists of three stages: pretraining, fine-tuning and data fusion. During pretraining, an ML model is trained to learn the input–output mapping as governed by the physics-based NWM. Inputs include river network attributes and connectivity, meteorological forcing, and antecedent simulated streamflow. The predictand is simulated streamflow, where T_B and T_F denote the lengths of look-back and forecast periods, respectively. In the fine-tuning step, the ML model is trained to minimize mismatch with historical streamflow observations. During data fusion, ML streamflow predictions are adjusted through graph-based residual propagation. Both pretraining and fine-tuning are performed offline, but data fusion can be done in real time.

where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ is the adjacency matrix with self-loop (i.e., self-pointing links are added), $\mathbf{I}$ is the identity matrix, $\mathbf{H}^{(k)} \in \mathbb{R}^{N \times D_k}$ includes all hidden-state node vectors in the kth layer, and $\mathbf{W}^{(k)} \in \mathbb{R}^{D_{k-1} \times D_k}$ is a learnable weight matrix. In practice, a normalized form of adjacency matrix is often used in lieu of Eq. (3) to improve numerical stability (Kipf and Welling, 2016),

$$\hat{\mathbf{A}} = \tilde{\mathbf{D}}^{(-1/2)} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{(-1/2)}, \tag{4}$$

where $\tilde{\mathbf{D}}$ is a diagonal matrix containing the node degrees of $\tilde{\mathbf{A}}$.

For temporal learning, GWN adopts TCN, which uses a dilation factor to exponentially increase the receptive field of a CNN filter, thus allowing the capture of long-range dependencies using CNN (Wu et al., 2019; Zhang et al., 2020). Two TCNs (TCN-a and TCN-b in Fig. 2) are used in parallel to form a gated TCN block, as proposed by Dauphin et al. (2017):

$$\mathcal{X}^{(k)} = g(\Theta_1^{(k-1)} \mathcal{X}^{(k-1)} + \mathbf{b}_1^{(k-1)}) \odot \sigma(\Theta_2^{(k-1)} \mathcal{X}^{(k-1)} + \mathbf{b}_2^{(k-1)}), \tag{5}$$

where the $g(\cdot)$ function updates the hidden state using outputs from the previous layer $\mathcal{X}^{(k-1)} \in \mathbb{R}^{N \times D_{k-1} \times T_\mathrm{B}}$, $\sigma(\cdot)$ is a gate function that regulates information flow from one layer to the next, Θ_i and $\mathbf{b}_i$ are learnable weight matrices and bias terms, respectively, and $\odot$ is the element-wise multiplication operator. In GWN, `tanh` is used for $g(\cdot)$ and `sigmoid` is chosen for $\sigma(\cdot)$. Multiple TCN–GCN modules are then stacked to learn spatial and temporal embeddings progressively. To improve ML training, the input and output of each TCN–GCN module are connected for residual learning, and the outputs of all TCN–GCN modules are skip-connected to the final output layer (see Fig. 2). Finally, 1×1 kernel CNN layers are used to condense the tensor dimensions and generate the desired outputs.

A well-known issue with the standard GNNs is oversmoothing, which happens when node-specific information is smoothed out after several rounds of message passing (Li et al., 2018). This can be especially problematic when features from a node's neighboring nodes become dominant, overshadowing the features of the node itself (Hamilton, 2020). Oversmoothing is the main reason behind the shallow design of many GNNs. In the literature, two heuristic strategies have been proposed to mitigate oversmoothing, namely, node dropping and edge dropping. The former strategy randomly masks out a number of nodes in the adjacency matrix during each iteration of training, while the latter strategy randomly drops a fraction of node edges. The DropEdge scheme, originally proposed by Rong et al. (2019), produces varying perturbations of the graph connections, and thus can be seen as a data-augmentation technique for training GNNs. In contrast, DropNode may be seen as both a training technique and a data-imputation strategy – a GNN trained only on a subset of nodes can be used to predict nodes not seen during training. Previously, node masking was used for solving the Prediction at Ungauged Basins (PUB) problem (Sun et al., 2021). In this work, we further demonstrated its use for solving the Prediction at Unmodeled Nodes (PUN) problem at the basin scale. We implemented both of these features as options within the GWN (see Fig. 2).

As mentioned before, the GWN model is first trained using NWM outputs (pretraining) and then the network weights are fine-tuned using observation data falling in the training pe-

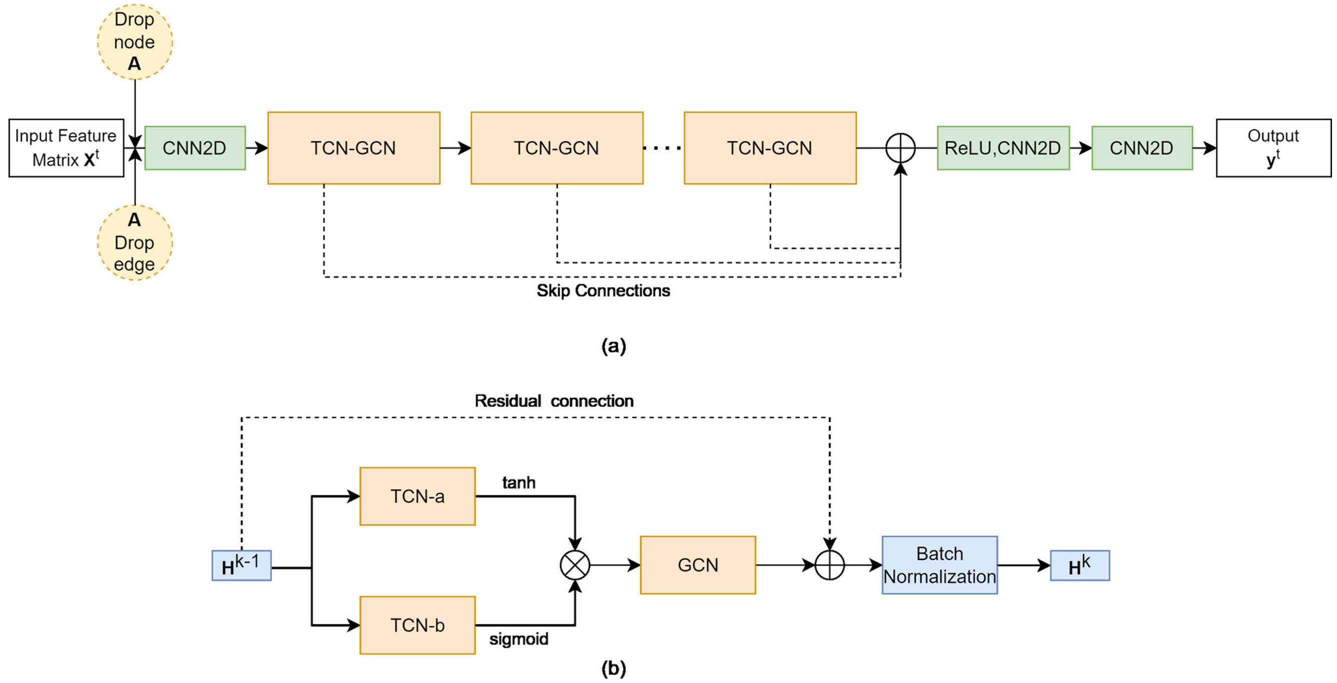

Figure 2. (a) GraphWavenet used in this work consists of a number of temporal convolution network and graph convolution network (TCN–GCN) modules for progressively encoding input data $\boldsymbol{x}^t$; the outputs of each TCN–GCN are skip-connected to improve learning; 1 × 1 kernel convolutional neural network (CNN) layers are used as linear transformation layers; optionally, nodes and edges in the adjacency matrix $\mathbf{A}$ are randomly masked for improving training and for graph coarsening (i.e., DropNode and DropEdge); **(b)** the internal design of each TCN–GCN module, which consists of interleaved TCN and GCN layers to process information from one hidden layer to the next. The layer input and output are connected via residual connections to improve training.

riod. The fine-tuning step is often adopted in physics-guided ML studies to reduce biases of process-based models (Ham et al., 2019; Andersson et al., 2021).

3.2 Data fusion

After fine-tuning, the surrogate model may still be subject to errors resulting from ML approximations and from the process-based NWM. New observation data, when available, can be used to further reduce the surrogate model prediction error through a post-processing step. Our goal is closely related to that of the hydrologic post-processing, which seeks to establish a statistical relationship between model outputs and observations (Li et al., 2017). In computer science, such post-processing is also related to label propagation, referring to assigning class labels to unclassified data using known data labels (Zhu and Ghahramani, 2002).

Let $\mathbf{y}$ and $\hat{\mathbf{y}}$ denote the true and predicted values. In reality, $\mathbf{y}$ can only be accessed at a limited number of observation nodes. Thus, $\mathbf{y}$ is partitioned into two parts corresponding to observed (L) and unobserved (U) values, $\mathbf{y} \doteq [\mathbf{y}_L^T, \mathbf{y}_U^T]^T$. Further, assume $\mathbf{y}$ is multi-Gaussian, $\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. Streamflow values, which typically follow non-Gaussian distributions, can be projected into the normal space via a normal transform technique (Li et al., 2017). As a matter of fact, this normal transform generally improves ML training, and should thus be done as part of the data preprocessing before ML training starts.

The joint distribution of $\mathbf{y}$ in terms of its subsets $\mathbf{y}_L$ and $\mathbf{y}_U$ may be written as follows (Bishop, 2006):

$$p\left(\mathbf{y}_L, \mathbf{y}_U\right) = \mathcal{N}\left(\begin{pmatrix} \boldsymbol{\mu}_L \\ \boldsymbol{\mu}_U \end{pmatrix}, \begin{pmatrix} \boldsymbol{\Sigma}_{LL} & \boldsymbol{\Sigma}_{LU} \\ \boldsymbol{\Sigma}_{UL}, & \boldsymbol{\Sigma}_{UU} \end{pmatrix}\right). \tag{6}$$

It can be shown that the conditional probability distribution of $\mathbf{y}_U$ given $\mathbf{y}_L$, namely, $p\left(\mathbf{y}_U \mid \mathbf{y}_L\right)$, is also multi-Gaussian, for which the conditional mean $\boldsymbol{\mu}_{U|L}$ and covariance $\boldsymbol{\Sigma}_{U|L}$ are (Bishop, 2006):

$$\begin{aligned} \boldsymbol{\mu}_{U|L} &= \boldsymbol{\mu}_U + \boldsymbol{\Sigma}_{UL} \boldsymbol{\Sigma}_{LL}^{-1} \left(\mathbf{y}_L - \boldsymbol{\mu}_L\right) = \boldsymbol{\mu}_U \\ &\quad - \boldsymbol{\Gamma}_{UU}^{-1} \boldsymbol{\Gamma}_{UL} \left(\mathbf{y}_L - \boldsymbol{\mu}_L\right), \\ \boldsymbol{\Sigma}_{U|L} &= \boldsymbol{\Sigma}_{UU} - \boldsymbol{\Sigma}_{UL} \boldsymbol{\Sigma}_{LL}^{-1} \boldsymbol{\Sigma}_{LU} = \boldsymbol{\Gamma}_{UU}^{-1}, \end{aligned} \tag{7}$$

where $\boldsymbol{\Gamma}$, known as the precision matrix, is the inverse of the covariance matrix. In the regression problem considered here, $\boldsymbol{\mu}_U$ may represent the ML estimates at unobserved locations, $\hat{\mathbf{y}}_U$, and $\mathbf{y}_L$ may represent gage observations, then Eq. (7) forms the basis of updating the ML predictions $\hat{\mathbf{y}}_U$ through observations $\mathbf{y}_L$. To adapt the residual propagation for GNNs, Jia and Benson (2020) proposed the following parameterization of the precision matrix:

$$\mathbf{\Gamma} = \beta(\mathbf{I} - \alpha\mathbf{S}), \tag{8}$$

where $\mathbf{S} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{1/2}$ is similar to the normalized adjacency matrix defined in Eq. (4), but with self-loops removed; β and α are learnable shape parameters. The former controls the residual magnitude and the latter reflects the correlation structure. Later, the same authors proposed a residual propagation form involving a single hyperparameter ω (Jia and Benson, 2021):

$$\begin{aligned}\hat{\mathbf{y}}_{U|L} &= \hat{\mathbf{y}}_U - (\mathbf{I}+\omega\mathbf{N})^{-1}_{UU}(\mathbf{I}+\omega\mathbf{N})_{UL}\boldsymbol{r}, \\ \boldsymbol{r} &\doteq \left(\mathbf{y}_L - \hat{\mathbf{y}}_L\right) \text{ and } \mathbf{N} \doteq \mathbf{I} - \mathbf{S},\end{aligned} \tag{9}$$

where the labeled and unlabeled parts of $\mathbf{I}+\omega\mathbf{N}$ are extracted by using mask matrices, $\boldsymbol{r}$ is defined as the residual vector between observations and ML predictions. Equation (9) is the form of data fusion used in this study, which is model agnostic. The hyperparameter ω may be obtained by cross validation. The only time-varying part in Eq. (9) is $\boldsymbol{r}$, and other matrix terms can be calculated offline; thus the update can be applied efficiently in real time.

We remark that the conditional mean and covariance shown in Eq. (7) are generally related to the Gaussian process regression (Rasmussen and Williams, 2006) and has been used in the hydrologic post-processing literature, e.g., in the General Linear Model Post-Processor in Ye et al. (2014). However, the main difference is that the data fusion is extended to operate on graphs via Eq. (9). An advantage of the residual propagation approach taken here is that it allows consideration of spatial correlation among node prediction errors while respecting the graph topology. Intuitively, we expect that unobserved nodes adjacent to a gauged node should share similar spatial and temporal patterns. The data-fusion approach adopted here is different from the data-assimilation method in Jia et al. (2021), in which a prediction is made by using the features of a node and its neighboring nodes without directly considering predictions of the neighboring nodes, thus limiting the use of spatial information (Jia and Benson, 2020).

4 Study area and experimental design

4.1 Description of the study area

The algorithms and workflow described in Sect. 3 are general. For demonstration purposes, we first consider the East–Taylor watershed (ETW), a HUC-8 watershed (drainage area 1984.7 km^2) that lies within the HUC-4 Gunnison River basin (GRB) in the southern Rocky Mountains of Colorado, United States (Fig. 3). Later in Sect. 6, we apply the framework to the entire GRB. The ETW is representative of the high-altitude river basins in the upper Colorado River basin. Elevation of the watershed ranges approximately from 2440 to 4335 m (McKay et al., 2012). Climate of the watershed is defined as continental and subarctic with long, cold winters and short, cool summers (Hubbard et al., 2018). Annual precipitation ranges from 1350 mm yr^{-1} in the high-elevation headwater region of the East River to about 400 mm yr^{-1} near the basin's outlet, and the majority of precipitation falls as snow (see Fig. S1a in the Supplement); the annual temperature is in the range of -3 to 1 °C in the area (Fig. S1b) (PRISM Climate Group, 2022).

The ETW encompasses two alpine rivers, the East River in the west and the Taylor River in the east, both flowing into the Gunnison River in the south which, in turn, serves as a main tributary of the Colorado River, contributing about 40 % of its streamflow (Battaglin et al., 2011). The East River watershed is mostly undeveloped, other than the city of Crested Butte (population 1339) and the ski-resort area that are located near the middle course of the river (Bryant et al., 2020). The Taylor River is dammed by the Taylor Park Dam (storage capacity about 0.13 km^3) in the middle (Bureau of Reclamation, 2022).

Snowmelt provides the main source of runoff in the watershed, with peak discharge occurring between May and July; baseflow conditions prevail from August until the winter freeze (Bryant et al., 2020). Like many other snow-dominated systems in the western United States, the streamflow pattern in the ETW is influenced by frequent droughts and heat waves in recent years (Winnick et al., 2017), and is likely to undergo further changes with the projected lower snowfall and earlier snowmelt under future climate conditions (Davenport et al., 2020). Globally, tremendous interests exist in the hydroclimate modeling community to understand and predict streamflows in snow-dominated regions under global environment change (Barnett et al., 2005; Qin et al., 2020).

Five USGS gages in the watershed are identified to have continuous records (open circles in Fig. 3) and are used as the source for fine-tuning and data fusion in this work. Notably, Gage 09107000 and Gage 09109000 are located immediately upstream and downstream of the Taylor Park Dam. The mean annual flow at the outlet of the East River (Gage 09112500) is 9.3 m^3 s^{-1}, and at the outlet of the Taylor River (Gage 09110000) it is 9.2 m^3 s^{-1}, both are estimated based on 110 years of data from 1911 to 2021 (USGS, 2022a). An extra USGS gage (red circle) with an incomplete record is also identified for validation purposes.

The ETW contains 23 subbasins (dark lines in Fig. 3) at the HUC-12 level and a total of 552 NHDPlus flowlines or reaches (dark blue polylines in Fig. 3). The NHDPlus reach lengths range from 0.047 to 11.48 km, the stream order ranges from 1 (the headwater tributaries) to 5 (Taylor River downstream of the reservoir), and the drainage area of reaches ranges from 0.027–24.99 km^2. Thus, the ETW represents an interesting study area from the perspective of river network modeling: it encompasses two contrasting flow regimes, the East River that is under natural flow conditions and the Taylor River that is subject to human intervention. It also includes a meaningful degree of spatial heterogeneity that is representative of a snow-dominated, mountainous watershed.

4.2 Experimental design and model training

We performed a series of ML experiments to address the two objectives of this study, namely, evaluating the role of physical realism in river network representation and the efficacy of graph-based data fusion. The study period is set to January 1993–December 2018, the same as the retrospective simulation period of NWM v2.0. All scripts are written in Python and, in particular, PyTorch (Paszke et al., 2019) is used to develop all GNN and data-fusion codes. The open-source GWN code (Wu et al., 2019) and GNN data-fusion code (Jia and Benson, 2020) are adapted for the problem at hand. The training/validation/testing split used is 0.7, 0.15, 0.15, respectively. Streamflow data are projected to normal space by using a power transformer scaler proposed in Yeo and Johnson (2000) and available from $\texttt{scikit}-\texttt{learn}$ Python library (Pedregosa et al., 2011).

The node-based ETW river network is constructed based on the NHDPlus flowlines and associated reach IDs (COMIDs) by following the procedures described under Sect. 2.3. A cutoff threshold is applied to extract a subset of river reaches. Thresholding is a common practice in climate networks to help reveal dominant spatiotemporal structures (Donges et al., 2009a, b; Malik et al., 2012). Here, all reaches with a medium flow value less than 0.01 $\mathrm{m}^3\,\mathrm{s}^{-1}$ (approximately at the 30th percentile of median flow distribution of all nodes) are trimmed from the network, keeping 295 out of a total of 552 nodes. In the resulting adjacency matrix, two nodes are connected if an NHDPlus flowline of “stream-river” type exists between them. Reaches falling on the water bodies (light blue polygons in Fig. 3) are not included in the river network. We used undirected and unweighted adjacency matrices. As an ablation study, we also considered NWM v2.1 data, which resulted in a 348-node network after trimming. The different numbers of node probably reflect differences in model parameterization and forcing data between the two NWM versions (see Sect. 2). The HUC-12 subbasin-based ETW river network is constructed by using the information in the NHDPlus pour-point layer (Price, 2022) for connecting each subbasin to its neighbors.

Unless otherwise noted, we train the model with a look-back period of 30 d, which is sufficient for the current case when simulated antecedent flows are also used as predictors to drive streamflow predictions. The forecast horizon is 1 d ahead. The number of TCN–GCN blocks used is 7, the kernel dimensions used in TCN layers are 1×4, the number of filters used in dilation and residual blocks are both 32. We use AdamW, a modified version of the Adam gradient-based optimizer, for training the network (Loshchilov and Hutter, 2018). The mini-batch size is 30.

When using the DropEdge option, we randomly enable 80 % of the edges in the adjacency matrix at the beginning of each epoch. This is done by flattening the adjacency matrix into a vector, performing random permutation, and taking the first 80 % of connection. When the DropNode option is on, we use a mask matrix to mask the indices of nodes to be dropped. The training parameter selection is largely based on our previous experience (Sun et al., 2021). For each GWN configuration, 10 different models are trained by initializing with different random seeds. For pretraining, the loss function used is the mean absolute error (MAE) or L_1 norm between ML predictions and NWM outputs, while for fine-tuning, the loss function is the L_1 norm between ML predictions and observed streamflow at the five gage locations. Pretraining is done for 60 epochs with a learning rate of 5×10^{-4}. Starting with the weights of a pretrained model, fine-tuning is done for another 15 epochs but with a smaller learning rate of 1×10^{-5}. Training time is around 1.3 min wall-clock time per epoch for the 295-node models, and 14 s per epoch for the 23-node models, on the same Nvidia V100 GPU.

We quantify the performance of trained models on test data using three metrics, namely, normalized root mean square error (NRMSE), Kling–Gupta efficiency (KGE) (Gupta et al., 2009), and the Pearson correlation (R). Definitions of the metrics and L_1 norm are given in the Appendix. The total running time is 5.4 s wall-clock time on the test data. The hyperparameter ω in Eq. (9) is selected through leave-one-out cross validation (LOOCV). Specifically, the ω value is varied in the range (50, 5000) with a step size of 50. For each ω, we perform LOOCV by using four of the five gages for residual propagation, and calculating the MAE on the holdout gage. The ω that gives the minimum mean MAE across all gages is selected for data fusion.

5 Results

5.1 Pretraining

We first demonstrate the efficacy of GWN and its variants for approximating the NWM (i.e., the pretraining stage). Three sets of GWN models are trained. The first set uses the original GWN configuration with the self-loop adjacency matrix $\tilde{\mathbf{A}}$ (defined in Eq. 3) corresponding to the actual ETW network (GWN-O). The second set uses the same $\tilde{\mathbf{A}}$ as input but is trained by activating the DropEdge feature to randomly remove node links (GWN-DropEdge). The third set is trained using only a subset of nodes in $\tilde{\mathbf{A}}$ (GWN-Impute). Thus, physical realism is gradually reduced across the three sets of experiments. The $\tilde{\mathbf{A}}$ used in the first two sets corresponds to the trimmed ETW network containing 295 nodes, while the third set uses the HUC-12 pour-point set as nodes (see Fig. 3).

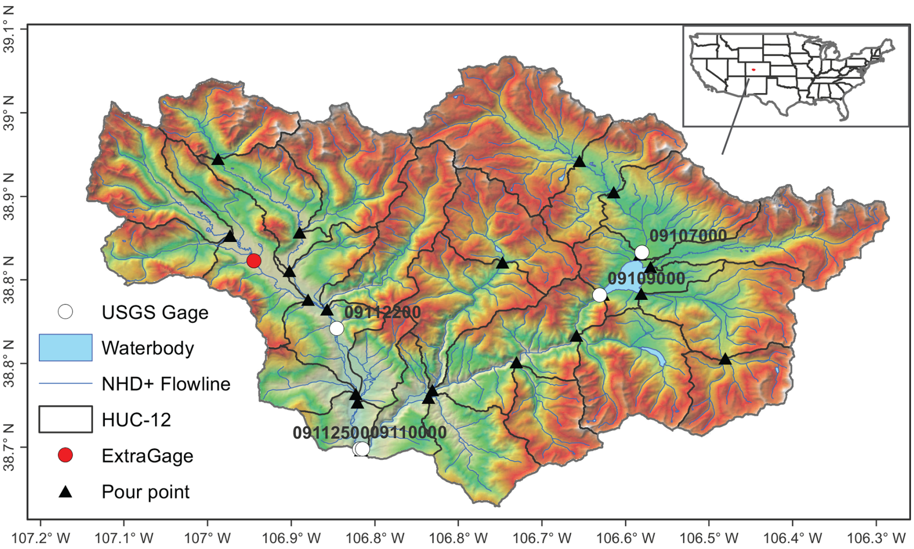

Figure 3. Shaded relief map of the East–Taylor watershed (HUC-8 code 14020001), overlaid by the NHDPlus flowline (solid blue lines), five USGS gage locations (open circles) used for fine-tuning and data fusion, the HUC-12 subbasin boundaries (solid dark lines) and pour points (triangles), and the surface waterbody (light blue) layer. An extra gage (red circle) is held for additional validation.

Table 1 summarizes the performance metrics of all three models in KGE, NRMSE and R. Results suggest that the GWN-O achieves high scores under all 3 metrics. The GWN-DropEdge gives almost the same performance and the GWN-Impute, which is trained using less than one-tenth of the nodes used in the other two models, also achieves a reasonable performance. Figure 4 shows the empirical cumulative distribution function (ECDF) and node-level maps of KGE and NRMSE that are obtained by the GWN-O surrogate model. The KGE is close to 1.0 (NRMSE close to 0.0) on the mainstems of the East and Taylor rivers, but drops (increases) slightly in several tributaries of the Taylor River in the middle and southeastern part of the watershed. Performance at subbasins disconnected by the Taylor Park Dam does not appear to be strongly affected. All reaches adjacent to the reservoir show high KGEs. This is interesting, suggesting that the antecedent NWM outputs provide sufficient node-level information for the GNN to learn.

As an ablation study, we trained a GWN-O model using only Daymet forcing data while keeping all other configurations unchanged. Results, shown in Fig. S2, suggest that the performance of the data-driven surrogate model has deteriorated, especially in the mid-range of both rivers. One possible explanation is that without using the antecedent NWM outputs as predictors, the data-driven models may either require a much longer look-back period to achieve meaningful results (Kratzert et al., 2019a), or simply cannot explain all variations in the NWM outputs. The latter reason points to the inherent difficulty in learning the data-driven, input–output mapping in this snow-dominated watershed. Previously, Ma et al. (2017) evaluated the performance of NOAH-MP, which is the land model in WRF-Hydro, in modeling the snow cover function (SCF). By definition, SCF is the fraction of a grid cell covered by snow, and provides an indirect measure of snow mass and snow depth. They found that the modeled SCF agrees well with the gridded SCF product derived from the Moderate Resolution Imaging Spectroradiometer (MODIS), with relative biases varying from 4 % in the snow-accumulation phase to 14 % in the melting phase. The authors attributed the good performance to the SCF scheme, the use of a vegetation canopy snow interception module, and the multilayer snowpack representation implemented in NOAH-MP (Ma et al., 2017). Thus, these sophisticated parameter schemes in the NWM may not be explained by using Daymet forcing alone.

We compare the KGE between GWN-O and the other two models in Fig. 5. The performances of GWN-O and GWN-DropEdge are similar. Only slight KGE differences are noted in the headwater tributaries of the Taylor River, indicating that the GWN models are relatively robust to perturbations induced through dropping edges. Again, this is because the same NWM data are behind GWN-O and GWN-DropEdge and strong auto-correlation exists at the node level. Between GWN-O and GWN-Impute, GWN-O outperforms in the headwater subbasins. Nevertheless, the accuracy of GWN-Impute is high on the mainstems.

To elucidate why GWN-Impute gives a good performance on the ETW, we plot the node correlation heatmap in Fig. 6. Subbasins 1–13 are on the Taylor River side, while subbasins 14–23 are on the East River side of the ETW (Fig. 6a). An immediate observation is the block structure along the diagonal of the heatmap in Fig. 6b, suggesting that strong cross-node correlations exist inside each subbasin. Specifically, subbasin 13, the most downstream basin on the Taylor

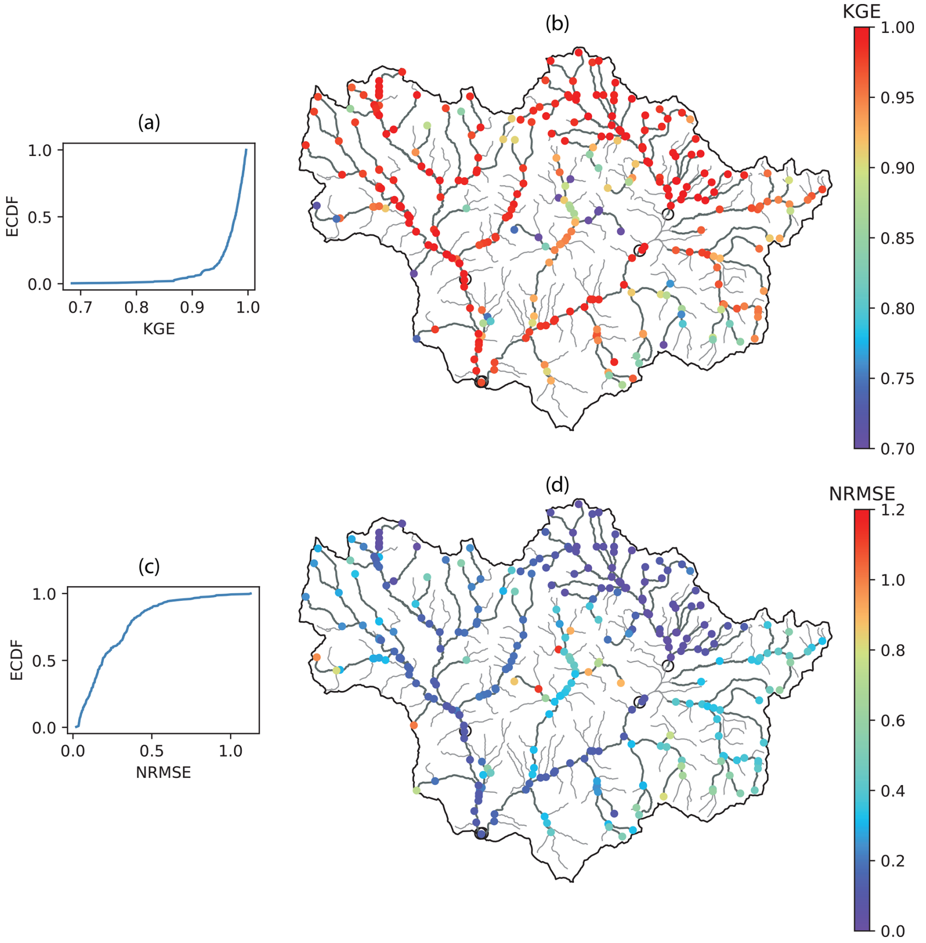

Figure 4. Test performance metrics obtained using the pretrained 295-node GraphWaveNet (GWN-O) model. Panels **(a, b)** represent the empirical cumulative distribution function (ECDF) and node map of KGE; **(c, d)** show the ECDF and node map of NRMSE. Results are obtained based on the ensemble mean of a 10-member ensemble. Gage locations are shown as open circles on the node maps. .

River side, exhibits the highest inner-basin correlation, while the upstream headwater basins (subbasins 1 and 2) show more inner-basin variations. Subbasin 13 also shows relatively strong cross-basin correlations with other subbasins on both the East River and Taylor River sides. This is because subbasin 13 is at the confluence of the two rivers, thus reflecting information passed from the upstream of both rivers. Similarly, the downstream basins on the East River side, namely, subbasins 22 and 23, also show relatively strong inter-basin correlations with other basins on the East River side. In contrast, isolated remote subbasins (e.g., subbasin 14 on the East River side, and subbasins 1 and 11 on the Taylor River side as labeled below Fig. 6b do not exhibit strong inter-basin correlations. These observations are further corroborated using the graph betweenness centrality, defined as the number of the shortest paths that pass through a node. Thus, nodes with a high betweenness centrality tend to have more influence on information propagation in the network (Donges et al., 2009b). Figure 6c shows that reaches in the downstream nodes of the East and Taylor rivers have higher betweenness centrality values than the upstream nodes. The heatmap analysis reveals the model parameter structure and the degree of freedom of the underlying the NWM which, in turn, determine how well the graph-coarsening process may work. Essentially, in this case, the ML imputation based on pour points is a physics-informed interpolation process. We expect that the information content is richest at the pour point of each subbasin, thus a surrogate model trained using only pour-point information may sufficiently capture the dominant flow dynamics in the watershed and satisfactorily interpolate to all unmodeled points.

We conducted an ablation study using the 348-node network corresponding to NWM v2.1 (see also Sect. 4.2). Results are shown in Fig. S3. The newly added nodes are low-flow nodes appearing at the headwaters and small tributaries in subbasins. In this case, the median of the KGE is 0.974 and the mean is 0.937. Compared to the model trained using NWM v2.0, the performance in subbasins 8, 10 and 11 are significantly improved, however, the performance in headwater subbasins 1 and 2 on the Taylor River side also show

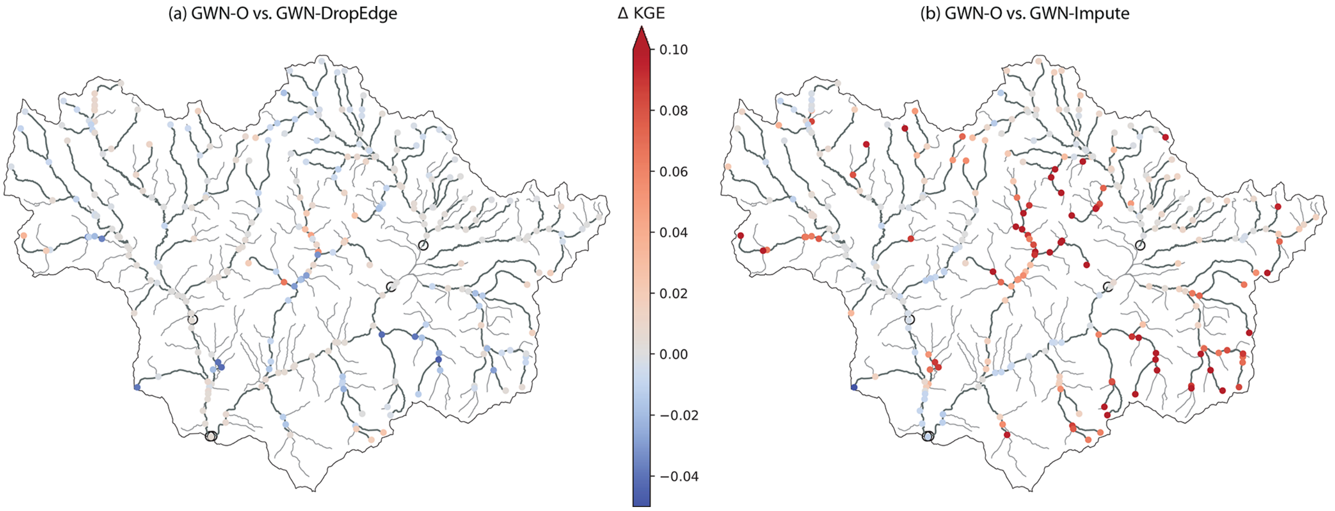

Figure 5. Node-level KGE comparison between the pretrained GWN-O and that of **(a)** GWN-DropEdge and **(b)** GWN-Impute.

Table 1. Performance metrics of pretrained GraphWaveNet (GWN) surrogate models. GWN-O is the original model of Wu et al. (2019), GWN-DropEdge implements random edge sampling, GWN-Impute is trained on a subset of 23 nodes corresponding to the HUC-12 pour points. All results reported are based on ensemble averages from 10 models trained with different random seeds.

Model	Median	Mean	Max	Min
KGE, range $(-\infty, 1]$				
GWN-O	0.978	0.947	0.998	0.574
GWN-DropEdge	0.978	0.947	0.998	0.584
GWN-Impute	0.975	0.915	0.997	0.445
NRMSE, range $[0, \infty]$				
GWN-O	0.188	0.256	1.126	0.021
GWN-DropEdge	0.186	0.262	1.132	0.023
GWN-Impute	0.214	0.300	1.455	0.039
R, range $[-1, 1]$				
GWN-O	0.991	0.985	1.000	0.877
GWN-DropEdge	0.990	0.984	1.000	0.912
GWN-Impute	0.990	0.982	0.999	0.906

some deterioration. This suggests that the effect of model calibration between NWM versions may not be uniform across the model domain.

Experiments presented thus far provide useful insights into how much physical realism is needed when implementing GNNs for the purpose of river network surrogate modeling. It depends on the watershed hydroclimatic and physiographic attributes, the parameter structure of the underlying process-based model, and the objective of the study. In the current case, we show that GWN models built on a coarsened graph give comparable performances to those built on more fine-grained representations of the river network, requiring only a fraction of training time. If the main objective of surrogate modeling is to capture the simulated streamflow patterns in the mainstem of a river (e.g., for estimating flood peaks), then all variants of the GWN models presented here should suffice. On the other hand, if the main objective is to simulate snowmelt and runoff in low-flow headwater basins, then more physical realism is required in the river network. These findings also shed light on scaling up the GNNs for modeling large river basins.

5.2 Fine-tuning

After pretraining, we fine-tune the GWN models by following the procedure described in Sect. 4.2. Table 2 reports the metrics of the NWM as well as pretrained and fine-tuned GWN-O models against the five USGS gages (see Fig. 3 for their locations) over the testing period. The corresponding hydrographs are presented in Fig. 7. The KGE values of NWM simulations are relatively low, only Gage 09107000 (upstream of Taylor Park Dam) and Gage 09110000 (at pour point Taylor River) have KGE values greater than 0.5. The pretrained GWN-O model reports similar KGE values to the NWM, which is by design. The fine-tuned GWN-O model, which is trained using the historical observations falling in the training period, improves the KGE values slightly over most gages, except for Gage 09112500 located in the midstream of the East River. We also calculate the KGE values for the winter season (NDJF) and summer season (MJJA) separately. The fine-tuned model captures the low flow on the East River relatively well (Gage 09112200 and 09112500). However, all peak flows are underestimated, as can be seen from Fig. 7. In comparison, the fine-tuned model generally makes more improvement on R, except at Gage 09107000.

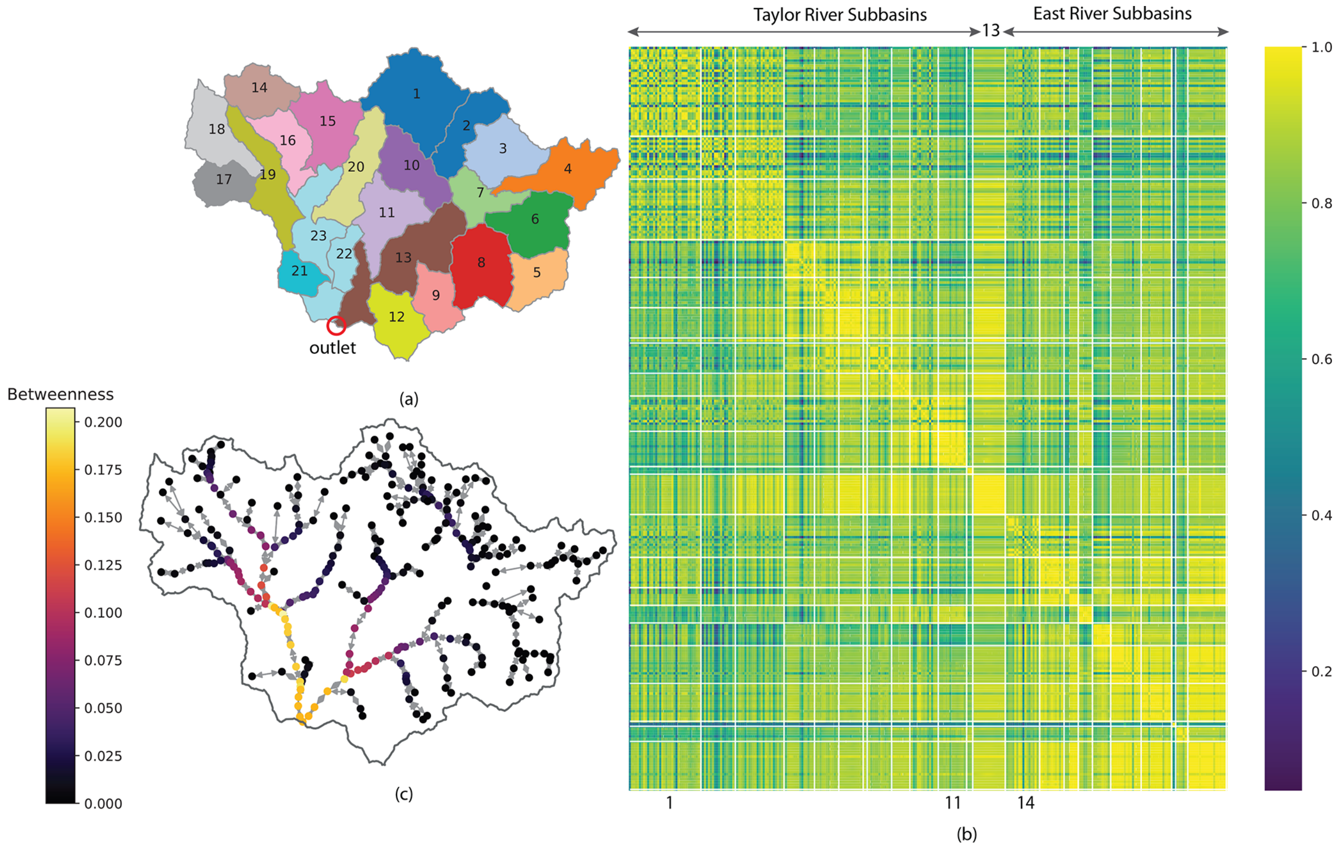

Figure 6. Node correlation heatmap generally exhibits a block structure within subbasins: **(a)** subbasin boundary map, where subbasins 1–13 are on the Taylor River side and subbasins 14–23 are on the East River side; **(b)** node correlation heatmap, where solid white lines separate nodes in different subbasins; **(c)** node betweenness centrality map .

The summer correlation values are higher than the winter values.

Overall, fine-tuning of the GWN-O only leads to mild performance improvements in this case. There can be multiple reasons. First, in this case, fine-tuning is restricted to a few nodes and the total effect is not as significant as when observations are available over many nodes/cells, such as the gridded data used in many climate applications (Ham et al., 2019). Second, the forcing data we used in driving the GWN are not accurate enough to allow the models to capture high flows. Third, the NWM may have underestimated snowmelt quantity and timing. Nevertheless, the bias corrections resulting from the fine-tuning stage, especially in the phase of the time series, are important for the subsequent data-fusion step. In this work, we mainly utilized streamflow data, but other types of Earth observation data may also be integrated in the fine-tuning step to further improve model performance.

5.3 Data fusion

We investigate the efficacy of data fusion on the five USGS gages. The value of hyperparameter ω is determined according to the LOOCV procedure described in Sect. 4.2. For the fine-tuned 295-node GWN-O model, the optimal ω value is found to be 1500. Figure 8 compares NWM v2.0, corrected GWN-O, and observed streamflow time series for the testing period, in which the KGE and correlation (R) between the corrected GWN-O and observations are shown in the subplot titles for each gage.

After data fusion, R becomes greater than 0.85 at all gage locations. In terms of KGE, the LOOCV data fusion has the greatest impact on Gage 09109000, achieving a value of 0.944. It also improves the two East River gages significantly (i.e., Gages 09112500 and 09112200). It has little effect on Gage 09107000, which is upstream of the Taylor Park Reservoir. It has a slightly negative impact on Gage 09110000, which is at the outlet of the Taylor River. In this case, Gages 091125000 and 09110000 are very close to each other (indirectly connected via the confluence node between the East and Taylor rivers), but reflecting different flow patterns. Specifically, Gage 09110000 is affected by the reservoir releases, as can be seen from the zigzag pattern in Fig. 8, while Gage 091125000 is not subject to such human-intervention impacts. The current data-fusion scheme does not differentiate these different dynamics. However, when all

Table 2. Comparison of the KGE and R of the NWM, pretrained GWN-O, fine-tuned GWO-O against the USGS gage data. The last four columns show the values for winter and summer seasons.

USGS gage	KGE						
	NWM	GWN-O pretrain	GWN-O fine-tune	NWM (winter)	GWN-O fine-tune (winter)	NWM (summer)	GWN-O fine-tune (summer)
09107000	0.563	0.559	0.596	0.169	0.205	0.467	0.485
09112200	0.332	0.328	0.348	0.192	0.625	0.266	0.258
09112500	0.355	0.348	0.341	−0.076	0.741	0.289	0.248
09109000	0.425	0.425	0.624	−1.418	−0.146	0.323	0.524
09110000	0.616	0.610	0.665	−0.632	0.148	0.534	0.614
USGS gage	R						
09107000	0.877	0.876	0.851	0.698	0.318	0.855	0.771
09112200	0.904	0.900	0.961	0.348	0.660	0.898	0.953
09112500	0.901	0.899	0.957	0.074	0.756	0.894	0.948
09109000	0.554	0.554	0.782	−0.471	−0.112	0.442	0.650
09110000	0.706	0.715	0.868	−0.337	0.195	0.618	0.788

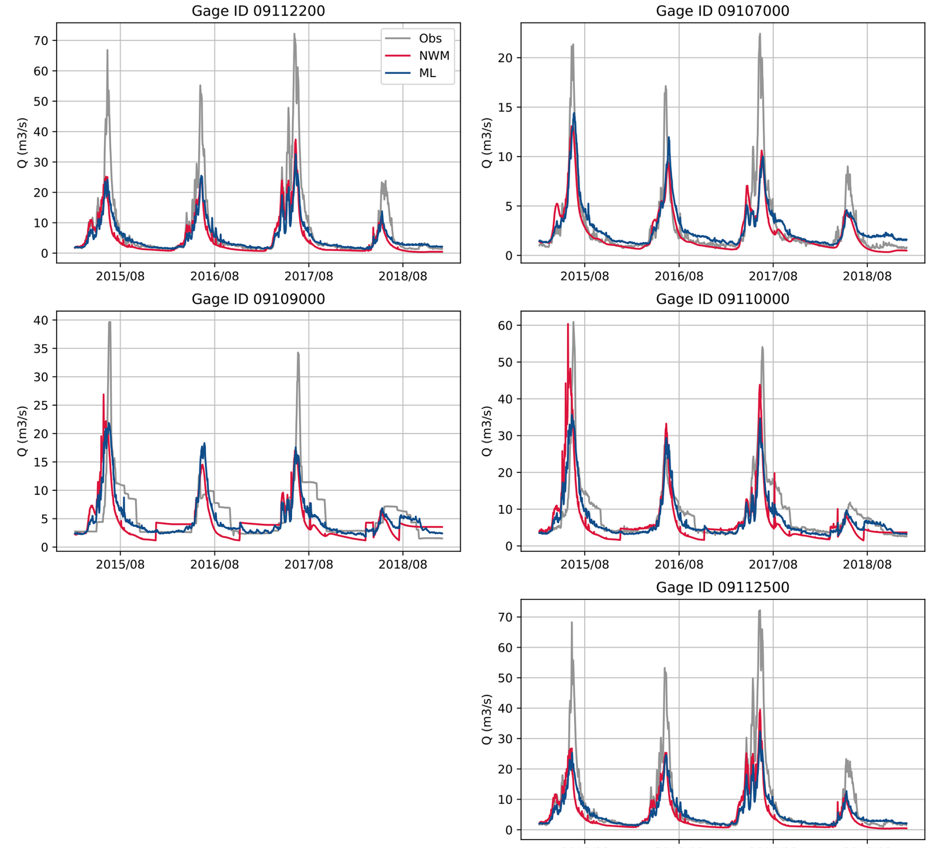

Figure 7. Hydrographs simulated by the NWM (dark blue) and GWN-O (ML, dark red) vs. the USGS data (Obs, gray) over the testing period.

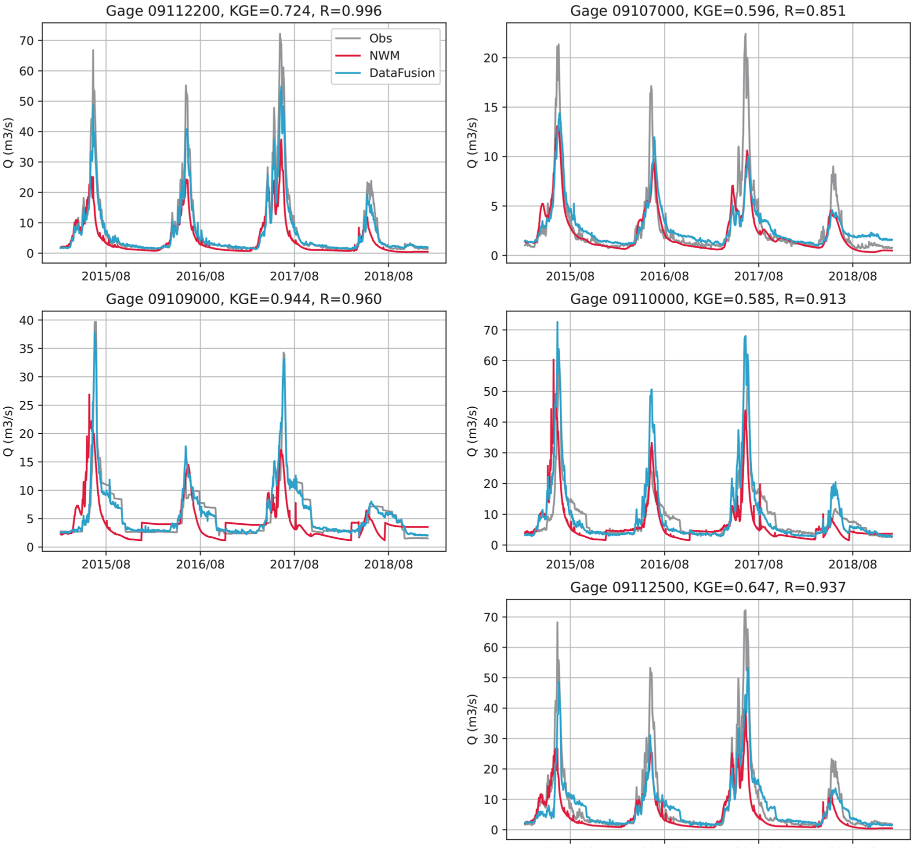

Figure 8. Hydrographs simulated by the NWM (dark red) and corrected GWN-O (data fusion, light blue) vs. the USGS data (observations (Obs), gray) over the testing period. KGE and *R* shown in the subplot titles are calculated between data fusion and observations.

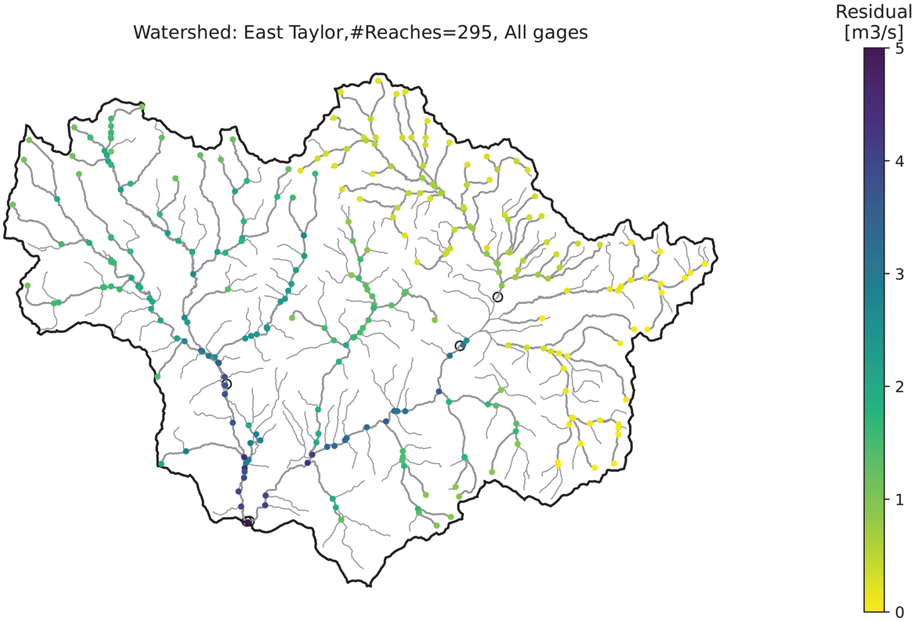

Figure 9. Residuals resulting from the data fusion using all five USGS gages. The greatest correction effect is seen at the downstream points.

gages are used simultaneously in data fusion, we expect such interference to be reduced.

Figure 9 shows the data-fusion residual map, defined as the flow difference between data fusion and the NWM (i.e., $\boldsymbol{r}$ in Eq. 9). The data-fusion effect is greatest at the most downstream locations of both East and Taylor rivers and then gradually fades out toward the upstream headwater basins, reflecting the reduction in flow magnitudes when traversing upstream, as well as the diminished influence of downstream gages.

The effect of data fusion is further validated using an extra USGS gage on the East River, Gage 385106106571000, that is not part of the model training and residual propagation (see Fig. 3 for its location). This is the only extra gage in the ETW that has a meaningful length of records (4475 d) for the study period. Results (Fig. 10) show that data fusion significantly improved streamflow, increasing the KGE from 0.056 to 0.892.

As an ablation study, we applied the same data-fusion procedure to two other models, the GWN-Impute model and the GWN-O model trained without using the NWM as predictors. The LOOCV results, shown in Tables S1 and S2, suggest that data fusion in general improves the results at most gage locations. The GWN-Impute model shows comparable performance to the GWN-O model, although it is challenging for the GWN-O trained without using the NWM to yield good results. Thus, these results further demonstrate the robustness of a coarsened graph network for modeling the study area.

6 Perspectives on GNN scaling up

So far, the performance of our GNN-based framework has been demonstrated over a single HUC-8 watershed (i.e., ETW). A remaining question is how well the proposed GNN framework can be adapted to larger river basins, which is an important aspect in practice. For this purpose, we consider the HUC-4 Gunnison River basin (GRB, drainage area 20 790 km^2) that encompasses the ETW as a subbasin (see Fig. 11). We focus on the surrogate modeling part, which allows us to investigate not only the scalability, but also transferability of the GNN to other basins.

After applying the same network extraction procedure that is used to obtain the ETW network, we get a 1708-node network, which is about 6 times larger than the ETW network used in previous examples. The GRB consists of 225 HUC-12 subbasins, the corresponding pour points are also extracted and used as inputs to the GWN-Impute model. We adopt the same training configurations as used for the ETW to test how the framework can be applied to other basins with little modifications. For the 1708-node GWN model, each training epoch takes about 6.2 min in wall-clock time, while it takes only 1 min for the GWN-Impute model on the same compute node.

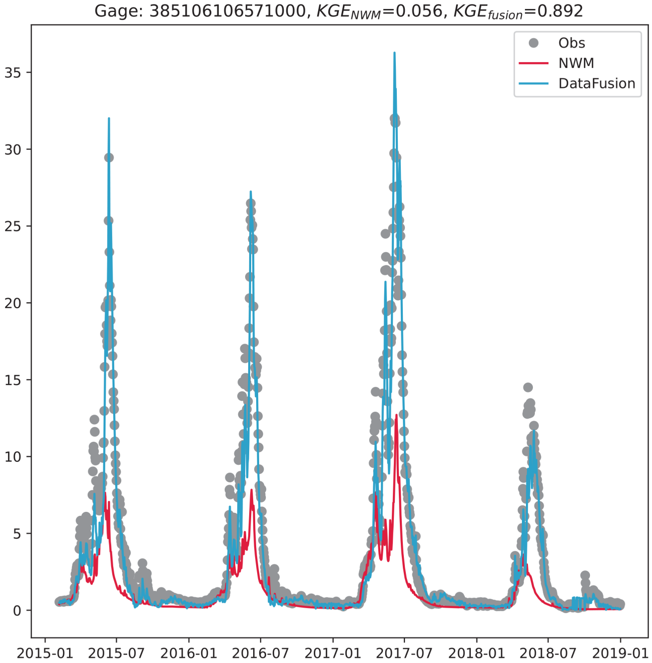

Figure 10. The effect of data fusion is validated on Gage 385106106571000 which is not used in any ML training and residual propagation. The KGE is improved from 0.056 to 0.892.

Again, the trained GWN and GWN-Impute models give good and comparable performance in terms of approximating the original NWM outputs over the GRB, with a median KGE value of 0.90 for both models (see Table 3 for all statistics). Areas of lower performance are mainly found in the high-elevation parts of the basin in the south and northwest. Thus, this additional use case provides additional evidence on the transferability of the GNN framework, at least in the same snow-dominated, mountainous region.

7 Conclusions

The GNNs provide a conceptually simple yet powerful ML framework for learning vector hydrography. This study presents a multi-stage, physics-guided ML framework that combines physics-based river network models with GNNs for streamflow forecasting. In particular, our workflow, consisting of pretraining, fine-tuning, and data-fusion stages, leverages existing investment in high-performance river network models (e.g., the US National Water Model) and Earth observation data.

We demonstrated the merits of the GNN-based framework over the HUC-8 East–Taylor watershed and HUC-4 Gunnison River basin. Both are located in the Upper Colorado River basin and representative of the snow-dominated river basins in the western United States. These snow-dominated basins present meaningful challenges from both the process understanding and the ML perspectives. On the science side,

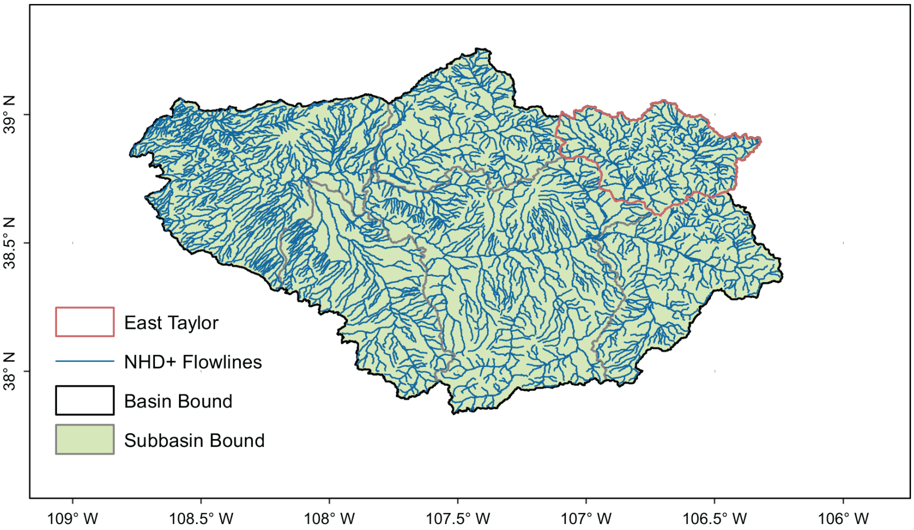

Figure 11. The HUC-4 Gunnison River basin (GRB) is located within the Upper Colorado River basin and encompasses the ETW as a subbasin (indicated by orange outline).

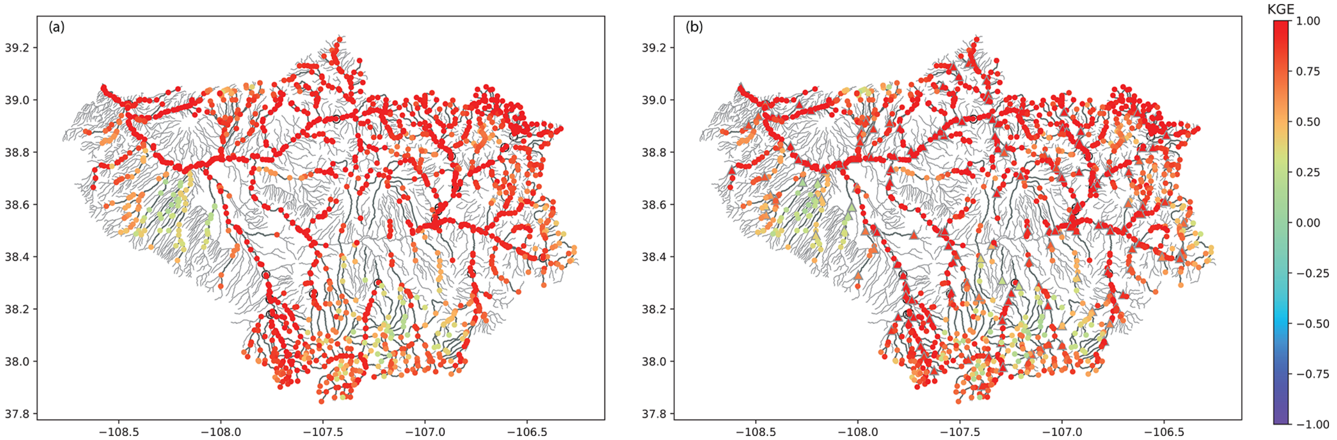

Figure 12. The node KGE maps obtained using **(a)** 1708-node GWN and **(b)** 225-basin GWN-Impute models for the testing period. Panel **(b)** shows the locations of the subbasins that are labeled as triangles.

significant research interests exist in understanding and predicting snowpack depth and shift in snowmelt timing under climate change. On the ML side, challenges remain on scaling graphs for solving large-scale graph-based regression problems. We showed that graph coarsening offers a feasible solution by exploiting the parameter structure of the underlying physics-based model. Thus, the designs of GNN and physics-based models can be asymmetric. When the physics-based model is already fine grained, which is the case in our work, the number of nodes in the GNN models can be significantly reduced. This same idea can be further expanded to embrace multiscale and/or multi-fidelity modeling to address different study objectives, such as coupling a fine-mesh network for biogeochemical transport with a coarse-mesh network for streamflow modeling. Finally, we show that graph-based data fusion provides a powerful post-processing tool for "nudging" streamflow observations, allowing bias corrections to traverse over the entire network. This study therefore provides important insights into adapting GNNs for large-scale river basin forecasting.

Appendix A: Definition of performance metrics

The performance metrics used in this work are defined as follows:

$$\mathrm{NRMSE} = \frac{1}{\overline{Q}_{\mathrm{obs}}}\sqrt{\frac{\sum_{t=1}^{n}\left(Q^t_{\mathrm{obs}} - Q^t_{\mathrm{sim}}\right)^2}{n}}, \tag{A1}$$

$$R = \frac{\sum\left(\left(Q^t_{\mathrm{sim}} - \overline{Q}_{\mathrm{sim}}\right)\left(Q^t_{\mathrm{obs}} - \overline{Q}_{\mathrm{obs}}\right)\right)}{\sqrt{\sum_{t=1}^{n}\left(Q^t_{\mathrm{sim}} - \overline{Q}_{\mathrm{sim}}\right)^2 \sum_{t=1}^{n}\left(Q^t_{\mathrm{obs}} - \overline{Q}_{\mathrm{obs}}\right)^2}}, \tag{A2}$$

$$\mathrm{KGE} = 1 - \sqrt{(1-R)^2 + (1 - \mu_{\mathrm{sim}}/\mu_{\mathrm{obs}})^2 + (1 - \sigma_{\mathrm{sim}}/\sigma_{\mathrm{obs}})^2}, \tag{A3}$$

$$\mathrm{MAE} = \frac{1}{n}\sum_{t=1}^{n} \mid Q^t_{\mathrm{obs}} - Q^t_{\mathrm{sim}} \mid, \tag{A4}$$

where Q_{obs} and Q_{sim} are observed and predicted values, respectively, $\overline{Q}$ denotes the mean values, and n is the total number of test data. The KGE score combines the linear Pearson correlation (R), the bias ratio $\mu_{\mathrm{sim}}/\mu_{\mathrm{obs}}$, and the variability ratio $\sigma_{\mathrm{sim}}/\sigma_{\mathrm{obs}}$ (Gupta et al., 2009). The range of KGE is $(-\infty, 1]$. A KGE value greater than -0.41 indicates that the model improves upon the mean flow (Knoben et al., 2019). The range of NRMSE is $[0, \infty)$. The mean absolute error (MAE), or L_1 norm, quantifies the absolute difference between simulated and measured values.

Code and data availability. We adapted the open-source Graph-Wavenet code from https://github.com/nnzhan/Graph-WaveNet (Wu et al., 2019), data-fusion code from https://github.com/000Justin000/gnn-residual-correlation (Jia and Benson, 2020), and DropEdge algorithm from https://github.com/DropEdge/DropEdge (Rong et al., 2019). The graph betweenness is generated using NetworkX https://networkx.org (last access: 22 March 2022, Hagberg et al., 2008). The NWM retrospective simulation (v2.0 and v2.1) data can be downloaded from AWS (https://registry.opendata.aws/nwm-archive, last access: 15 February 2022, NOAA, 2022). Alternatively, NWM data (v2.0) can be downloaded from Hydroshare (https://www.hydroshare.org, last access: 22 March 2022, Johnson and Blodgett, 2020). Daymet data can be downloaded from https://daymet.ornl.gov/ (last access: 10 March 2022, ORNL, 2022). NHDPlus database can be downloaded from EPA's NHDPlus site, https://www.epa.gov/waterdata/get-nhdplus-national-hydrography-dataset-plus-data (last access: 22 March 2022, EPA, 2022).

Supplement. The supplement related to this article is available online at: https://doi.org/10.5194/hess-26-5163-2022-supplement.

Author contributions. AYS was responsible for the conceptualization, methodology, software, data curation, formal analysis and writing of the original draft. PJ, ZLY, YX and XC assisted with the conceptualization, review and editing of the paper.

Competing interests. The contact author has declared that neither they nor their co-authors have any competing interests.

Disclaimer. Publisher's note: Copernicus Publications remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Acknowledgements. This work was funded by the ExaSheds project, which was supported by the US Department of Energy (DOE), Office of Science, Office of Biological and Environmental Research, Earth and Environmental Systems Sciences Division, Data Management Program. Pacific Northwest National Laboratory is operated for the DOE by Battelle Memorial Institute under contract DE-AC05-76RL01830. This paper describes objective technical results and analyses. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the US Department of Energy or the United States Government. Alexander Y. Sun wants to acknowledge the support of the Texas Advanced Computing Center. Alexander Y. Sun and Zong-Liang Yang are partly supported by DOE Advanced Scientific Computing Research under grant no. DE-SC0022211. The authors are grateful for the constructive comments from Dr. Uwe Ehret and an anonymous reviewer.

Financial support. This research has been supported by the Biological and Environmental Research (grant no. DE-AC05-76RL01830) and the Advanced Scientific Computing Research (grant no. DE-SC0022211).

Review statement. This paper was edited by Erwin Zehe and reviewed by Uwe Ehret and one anonymous referee.

References

Alfieri, L., Burek, P., Dutra, E., Krzeminski, B., Muraro, D., Thielen, J., and Pappenberger, F.: GloFAS – global ensemble streamflow forecasting and flood early warning, Hydrol. Earth Syst. Sci., 17, 1161–1175, https://doi.org/10.5194/hess-17-1161-2013, 2013.

Andersson, T. R., Hosking, J. S., Pérez-Ortiz, M., Paige, B., Elliott, A., Russell, C., Law, S., Jones, D. C., Wilkinson, J., Phillips, T., et al.: Seasonal Arctic sea ice forecasting with probabilistic deep learning, Nat. Commun., 12, 1–12, 2021.

Barnett, T. P., Adam, J. C., and Lettenmaier, D. P.: Potential impacts of a warming climate on water availability in snow-dominated regions, Nature, 438, 303–309, 2005.

Battaglin, W. A., Hay, L. E., and Markstrom, S. L.: Watershed Scale Response to Climate Change-East River Basin, Colorado, US Geological Survey Fact Sheet, 3126, 6 pp., 2011.

Bauer, P., Stevens, B., and Hazeleger, W.: A digital twin of Earth for the green transition, Nat. Clim. Change, 11, 80–83, 2021.

Beck, H. E., De Roo, A., and van Dijk, A. I.: Global maps of streamflow characteristics based on observations from several thousand catchments, J. Hydrometeorol., 16, 1478–1501, 2015.

Best, J.: Anthropogenic stresses on the world's big rivers, Nat. Geosci., 12, 7–21, 2019.

Biancamaria, S., Lettenmaier, D. P., and Pavelsky, T. M.: The SWOT mission and its capabilities for land hydrology, in: Remote sensing and water resources, Springer, 117–147, 2016.

Bierkens, M. F., Bell, V. A., Burek, P., Chaney, N., Condon, L. E., David, C. H., de Roo, A., Döll, P., Drost, N., Famiglietti, J. S., et al.: Hyper-resolution global hydrological modelling: what is next?, "Everywhere and locally relevant", Hydrol. Proc., 29, 310–320, 2015.

Bishop, C. M.: Pattern recognition and machine learning, Springer, ISBN: 978-1-4939-3843-8, 2006.

Blöschl, G., Hall, J., Parajka, J., Perdigão, R. A., Merz, B., Arheimer, B., Aronica, G. T., Bilibashi, A., Bonacci, O., Borga, M., et al.: Changing climate shifts timing of European floods, Science, 357, 588–590, 2017.

Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A., and Vandergheynst, P.: Geometric deep learning: going beyond Euclidean data, IEEE Signal Proc. Mag., 34, 18–42, 2017.

Bronstein, M. M., Bruna, J., Cohen, T., and Veličković, P.: Geometric deep learning: Grids, groups, graphs, geodesics, and gauges, arXiv [preprint], https://doi.org/10.48550/arXiv.2104.13478, 2021.

Bruna, J., Zaremba, W., Szlam, A., and LeCun, Y.: Spectral networks and locally connected networks on graphs, arXiv [preprint], https://doi.org/10.48550/arXiv.1312.6203, 2013.

Bryant, S. R., Sawyer, A. H., Briggs, M. A., Saup, C. M., Nelson, A. R., Wilkins, M. J., Christensen, J. N., and Williams, K. H.: Seasonal manganese transport in the hyporheic zone of a snowmelt-dominated river (East River, Colorado, USA), Hydrogeol. J., 28, 1323–1341, 2020.

Bureau of Reclamation: Taylor Park Reservoir, https://www.usbr.gov/projects/index.php?id=236, last access: 22 March 2022.

Camps-Valls, G., Tuia, D., Zhu, X. X., and Reichstein, M.: Deep learning for the Earth Sciences: A comprehensive approach to remote sensing, climate science and geosciences, John Wiley & Sons, ISBN: 9781119646143, 2021.

Chen, S., Appling, A., Oliver, S., Corson-Dosch, H., Read, J., Sadler, J., Zwart, J., and Jia, X.: Heterogeneous stream-reservoir graph networks with data assimilation, in: 2021 IEEE International Conference on Data Mining (ICDM), IEEE, 1024–1029, 2021.

Cosgrove, B., Gochis, D., Clark, E. P., Cui, Z., Dugger, A. L., Feng, X., Karsten, L. R., Khan, S., Kitzmiller, D., Lee, H. S., et al.: An overview of the National Weather Service national water model, in: AGU Fall Meeting Abstracts, 2016, H42B–05, 2016.

Dai, A. and Trenberth, K. E.: Estimates of freshwater discharge from continents: Latitudinal and seasonal variations, J. Hydrometeorol., 3, 660–687, 2002.

Dauphin, Y. N., Fan, A., Auli, M., and Grangier, D.: Language modeling with gated convolutional networks, Proceedings of the 34th International Conference on Machine Learning, PMLR, 70, 933–941, Sydney, Australia, 2017.

Davenport, F. V., Herrera-Estrada, J. E., Burke, M., and Diffenbaugh, N. S.: Flood size increases nonlinearly across the western United States in response to lower snow-precipitation ratios, Water Resour. Res., 56, e2019WR025571, https://doi.org/10.1029/2019WR025571, 2020.

David, C. H., Maidment, D. R., Niu, G.-Y., Yang, Z.-L., Habets, F., and Eijkhout, V.: River network routing on the NHDPlus dataset, J. Hydrometeorol., 12, 913–934, 2011.

De Groot, R. S., Wilson, M. A., and Boumans, R. M.: A typology for the classification, description and valuation of ecosystem functions, goods and services, Ecol. Econom., 41, 393–408, 2002.

Defferrard, M., Bresson, X., and Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering, Adv. Neur. Info. Proc. Syst., 29, 3844–3852, 2016.

Döll, P., Kaspar, F., and Lehner, B.: A global hydrological model for deriving water availability indicators: model tuning and validation, J. Hydrol., 270, 105–134, 2003.

Donges, J. F., Zou, Y., Marwan, N., and Kurths, J.: The backbone of the climate network, Europhys. Lett., 87, 48007, https://doi.org/10.1209/0295-5075/87/48007, 2009a.

Donges, J. F., Zou, Y., Marwan, N., and Kurths, J.: Complex networks in climate dynamics, Eur. Phys. J.-Spec. Top., 174, 157–179, 2009b.

Dottori, F., Szewczyk, W., Ciscar, J.-C., Zhao, F., Alfieri, L., Hirabayashi, Y., Bianchi, A., Mongelli, I., Frieler, K., Betts, R. A., and Feyen L.: Increased human and economic losses from river flooding with anthropogenic warming, Nat. Clim. Change, 8, 781–786, 2018.

EPA: National Hydrography Dataset Plus, https://www.epa.gov/waterdata/get-nhdplus-national-hydrography-dataset-plus-data, last access: 22 March 2022.

Feng, D., Fang, K., and Shen, C.: Enhancing streamflow forecast and extracting insights using long-short term memory networks with data integration at continental scales, Water Resour. Res., 56, e2019WR026793, https://doi.org/10.1029/2019WR026793, 2020.

Feng, D., Lawson, K., and Shen, C.: Mitigating prediction error of deep learning streamflow models in large data-sparse regions with ensemble modeling and soft data, Geophys. Res. Lett., 48, e2021GL092999, https://doi.org/10.1029/2021GL092999, 2021.

Gochis, D., Barlage, M., Dugger, A., FitzGerald, K., Karsten, L., McAllister, M., McCreight, J., Mills, J., RafieeiNasab, A., Read, L., Sampson, K., Yates, D., and Yu, W.: The WRF-Hydro modeling system technical description (Version 5.0), NCAR Technical Note, 107 pp., https://ral.ucar.edu/sites/default/files/public/WRFHydroV5TechnicalDescription.pdf (last access: 22 March 2022), 2018.

Grill, G., Lehner, B., Thieme, M., Geenen, B., Tickner, D., Antonelli, F., Babu, S., Borrelli, P., Cheng, L., Crochetiere, H., et al.: Mapping the world's free-flowing rivers, Nature, 569, 215–221, 2019.

Gupta, H. V., Kling, H., Yilmaz, K. K., and Martinez, G. F.: Decomposition of the mean squared error and NSE performance criteria: Implications for improving hydrological modelling, J. Hydrol., 377, 80–91, 2009.

Hagberg, A., Swart, P., and Schult, D.: Exploring network structure, dynamics, and function using NetworkX (No. LA-UR-08-05495; LA-UR-08-5495), Los Alamos National Lab.(LANL), Los Alamos, NM (United States), https://networkx.org (last access: 22 March 2022), 2008.

Ham, Y.-G., Kim, J.-H., and Luo, J.-J.: Deep learning for multi-year ENSO forecasts, Nature, 573, 568–572, 2019.

Hamilton, W. L.: Graph representation learning, Synthesis Lectures on Artifical Intelligence and Machine Learning, 46, Morgan & Claypool Publishers, Williston, VT, ISBN: 1681739631, 2020.

Hamilton, W. L., Ying, Z., and Leskovec, J.: Inductive representation learning on large graphs, in: Advances in Neural Information Processing Systems, 1024–1034, 2017.

Hochreiter, S. and Schmidhuber, J.: Long short-term memory, Neural Comput., 9, 1735–1780, 1997.

Hrachowitz, M., Savenije, H., Blöschl, G., McDonnell, J., Sivapalan, M., Pomeroy, J., Arheimer, B., Blume, T., Clark, M., Ehret, U., et al.: A decade of Predictions in Ungauged Basins (PUB) – a review, Hydrol. Sci. J., 58, 1198–1255, 2013.

Hu, Z., Dong, Y., Wang, K., Chang, K.-W., and Sun, Y.: Gpt-GNN: Generative pre-training of graph neural networks, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 1857–1867, 2020.

Hubbard, S. S., Williams, K. H., Agarwal, D., Banfield, J., Beller, H., Bouskill, N., Brodie, E., Carroll, R., Dafflon, B., Dwivedi, D., Falco, N., Faybishenko, B., Maxwell, R., Nico, P., Steefel, C., Steltzer, H., Tokunaga, T., Tran, P. A., Wainwright, H., and Varadharajan, C.: The East River, Colorado, Watershed: A mountainous community testbed for improving predictive understanding of multiscale hydrological–biogeochemical dynamics, Vadose Zone J., 17, 1–25, 2018.

Jia, J. and Benson, A. R.: Residual correlation in graph neural network regression, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 588–598, 2020.

Jia, J. and Benson, A. R.: A unifying generative model for graph learning algorithms: Label propagation, graph convolutions, and combinations, arXiv [preprint], https://doi.org/10.48550/arXiv.2101.07730, 2021.

Jia, X., Zwart, J., Sadler, J., Appling, A., Oliver, S., Markstrom, S., Willard, J., Xu, S., Steinbach, M., Read, J., and Kumar, V.: Physics-guided recurrent graph model for predicting flow and temperature in river networks, in: Proceedings of the 2021 SIAM International Conference on Data Mining (SDM), Society for Industrial and Applied Mathematics, 612–620, 2021.

Johnson, M. and Blodgett, D.: NOAA National Water Model Reanalysis Data at RENCI, HydroShare, https://doi.org/10.4211/hs.89b0952512dd4b378dc5be8d2093310f (last access: 22 March 2022), 2020.

Kashinath, K., Mustafa, M., Albert, A., Wu, J., Jiang, C., Esmaeilzadeh, S., Azizzadenesheli, K., Wang, R., Chattopadhyay, A., Singh, A., Manepalli, A., Chirila, D., Yu, R., Walters, R., White, B., Xiao, H., Tchelepi, H. A., Marcus, P., Anandkumar, A., Hassanzadeh, P., and Prabhat: Physics-informed machine learning: case studies for weather and climate modelling, Philos. T. Roy. Soc. A, 379, 20200093, https://doi.org/10.1098/rsta.2020.0093, 2021.

Kipf, T. N. and Welling, M.: Semi-supervised classification with graph convolutional networks, 5th International Conference on Learning Representations, Toulon, France, April 24–26, 2016.

Kitzmiller, D. H., Wu, W., Zhang, Z., Patrick, N., and Tan, X.: The analysis of record for calibration: a high-resolution precipitation and surface weather dataset for the united states, in: AGU Fall Meeting Abstracts, Vol. 2018, H41H–06, 2018.

Knoben, W. J. M., Freer, J. E., and Woods, R. A.: Technical note: Inherent benchmark or not? Comparing Nash–Sutcliffe and Kling–Gupta efficiency scores, Hydrol. Earth Syst. Sci., 23, 4323–4331, https://doi.org/10.5194/hess-23-4323-2019, 2019.

Kratzert, F., Klotz, D., Herrnegger, M., Sampson, A. K., Hochreiter, S., and Nearing, G. S.: Toward improved predictions in ungauged basins: Exploiting the power of machine learning, Water Resour. Res., 55, 11344–11354, 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, Hydrol. Earth Syst. Sci., 23, 5089–5110, https://doi.org/10.5194/hess-23-5089-2019, 2019b.

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.

Le, X.-H., Nguyen, D.-H., Jung, S., Yeon, M., and Lee, G.: Comparison of deep learning techniques for river streamflow forecasting, IEEE Access, 9, 71805–71820, 2021.

Lehner, B. and Grill, G.: Global river hydrography and network routing: baseline data and new approaches to study the world's large river systems, Hydrol. Proc., 27, 2171–2186, 2013.

Li, Q., Han, Z., and Wu, X.-M.: Deeper insights into graph convolutional networks for semi-supervised learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 32, ISBN 978-1-57735-800-8, 2018.

Li, W., Duan, Q., Miao, C., Ye, A., Gong, W., and Di, Z.: A review on statistical postprocessing methods for hydrometeorological ensemble forecasting, Wiley Interdisciplinary Reviews: Water, 4, e1246, https://doi.org/10.1002/wat2.1246, 2017.

Lin, P., Yang, Z.-L., Gochis, D. J., Yu, W., Maidment, D. R., Somos-Valenzuela, M. A., and David, C. H.: Implementation of a vector-based river network routing scheme in the community WRF-Hydro modeling framework for flood discharge simulation, Environ. Modell. Softw., 107, 1–11, 2018.

Lin, P., Pan, M., Beck, H. E., Yang, Y., Yamazaki, D., Frasson, R., David, C. H., Durand, M., Pavelsky, T. M., Allen, G. H., Gleason, C. J., and Wood, E. F.: Global reconstruction of naturalized river flows at 2.94 million reaches, Water Resour. Res., 55, 6499–6516, 2019.

Loshchilov, I. and Hutter, F.: Fixing weight decay regularization in adam, arXiv [preprint], https://doi.org/10.48550/arXiv.1711.05101, 2018.

Lu, D., Konapala, G., Painter, S. L., Kao, S.-C., and Gangrade, S.: Streamflow simulation in data-scarce basins using Bayesian and physics-informed machine learning models, J. Hydrometeorol., 22, 1421–1438, 2021.

Ma, N., Niu, G.-Y., Xia, Y., Cai, X., Zhang, Y., Ma, Y., and Fang, Y.: A systematic evaluation of Noah-MP in simulating land-atmosphere energy, water, and carbon exchanges over the continental United States, J. Geophys. Res.-Atmos., 122, 12–245, 2017.

Malik, N., Bookhagen, B., Marwan, N., and Kurths, J.: Analysis of spatial and temporal extreme monsoonal rainfall over South Asia using complex networks, Clim. Dynam., 39, 971–987, 2012.

McKay, L., Bondelid, T., Dewald, T., Johnston, J., Moore, R., and Rea, A.: NHDPlus version 2: User guide, US Environmental Protection Agency, 2012.

Mizukami, N., Clark, M. P., Gharari, S., Kluzek, E., Pan, M., Lin, P., Beck, H. E., and Yamazaki, D.: A Vector-Based River Routing Model for Earth System Models: Parallelization and Global Applications, J. Adv. Model. Earth Syst., 13, e2020MS002434, https://doi.org/10.1029/2020MS002434, 2021.

Moore, R. B. and Dewald, T. G.: The Road to NHDPlus–Advancements in Digital Stream Networks and Associated Catchments, J. Am. Water Resour. Assoc., 52, 890–900, 2016.

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.

NOAA: National Water Model CONUS Retrospective Dataset, https://registry.opendata.aws/nwm-archive, last access: 15 February 2022.

Nonnenmacher, M. and Greenberg, D. S.: Deep emulators for differentiation, forecasting, and parametrization in Earth science simulators, J. Adv. Model. Earth Syst., 13, e2021MS002554, https://doi.org/10.1029/2021MS002554, 2021.

ORNL: Daymet web service, https://daymet.ornl.gov, last access: 10 March 2022.

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

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S.: Pytorch: An imperative style, high-performance deep learning library, Adv. Neural Info. Proc. Syst., 32, , ISBN: 9781713807933, 2019.

Pathak, J., Subramanian, S., Harrington, P., Raja, S., Chattopadhyay, A., Mardani, M., Kurth, T., Hall, D., Li, Z., Azizzadenesheli, K., Hassanzadeh, P., Kashinath, K., and Anandkumar A.: FourCastNet: A Global Data-driven High-resolution Weather Model using Adaptive Fourier Neural Operators, arXiv [preprint], https://doi.org/10.48550/arXiv.2202.11214, 2022.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E.: Scikit-learn: Machine Learning in Python, J. Mach. Learn. Res., 12, 2825–2830, 2011.

Phillips, J. D., Schwanghart, W., and Heckmann, T.: Graph theory in the geosciences, Earth-Sci. Rev., 143, 147–160, 2015.

Pokhrel, Y., Hanasaki, N., Koirala, S., Cho, J., Yeh, P. J.-F., Kim, H., Kanae, S., and Oki, T.: Incorporating anthropogenic water regulation modules into a land surface model, J. Hydrometeorol., 13, 255–269, 2012.

Price, C.: WBD HU12 Pour Points derived from NHDPlus, https://www.sciencebase.gov/catalog/item/5762b664e4b07657d19a71ea, last access: 6 September 2022.

PRISM Climate Group: 30-year normals, https://prism.oregonstate.edu/normals/, last access: 6 September 2022.

Qin, Y., Abatzoglou, J. T., Siebert, S., Huning, L. S., AghaKouchak, A., Mankin, J. S., Hong, C., Tong, D., Davis, S. J., and Mueller, N. D.: Agricultural risks from changing snowmelt, Nat. Clim. Change, 10, 459–465, 2020.

Rasmussen, C. E. and Williams, C. K. I.: Gaussian processes for machine learning, The MIT Press, ISBN: 9780262182539, Cambridge, MA, 2006.

Rasp, S., Pritchard, M. S., and Gentine, P.: Deep learning to represent subgrid processes in climate models, P. Natl. Acad. Sci. USA, 115, 9684–9689, 2018.

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

Rong, Y., Huang, W., Xu, T., and Huang, J.: Dropedge: Towards deep graph convolutional networks on node classification, arXiv [preprint], https://doi.org/10.48550/arXiv.1907.10903, 2019.

Salas, F. R., Somos-Valenzuela, M. A., Dugger, A., Maidment, D. R., Gochis, D. J., David, C. H., Yu, W., Ding, D., Clark, E. P., and Noman, N.: Towards real-time continental scale streamflow simulation in continuous and discrete space, J. Am. Water Resour. Assoc., 54, 7–27, 2018.

Saunders, W.: Preparation of DEMs for use in environmental modeling analysis, Hydrologic and Hydraulic Modeling Support, Redlands, CA: ESRI, 29–51, 2000.

Schulthess, T. C., Bauer, P., Wedi, N., Fuhrer, O., Hoefler, T., and Schär, C.: Reflecting on the goal and baseline for exascale computing: a roadmap based on weather and climate simulations, Comput. Sci. Eng., 21, 30–41, 2018.

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

Simley, J. D. and Carswell Jr., W. J.: The national map–hydrography, US Geological Survey Fact Sheet, 2009–3054, 4 p., 2009.

Sun, A. Y. and Scanlon, B. R.: How can Big Data and machine learning benefit environment and water management: a survey of methods, applications, and future directions, Environ. Res. Lett., 14, 073001, https://doi.org/10.1088/1748-9326/ab1b7d, 2019.

Sun, A. Y., Wang, D., and Xu, X.: Monthly streamflow forecasting using Gaussian process regression, J. Hydrol., 511, 72–81, 2014.

Sun, A. Y., Scanlon, B. R., Zhang, Z., Walling, D., Bhanja, S. N., Mukherjee, A., and Zhong, Z.: Combining physically based modeling and deep learning for fusing GRACE satellite data: can we learn from mismatch?, Water Resour. Res., 55, 1179–1195, 2019.

Sun, A. Y., Jiang, P., Mudunuru, M. K., and Chen, X.: Explore Spatio-Temporal Learning of Large Sample Hydrology Using Graph Neural Networks, Water Resour. Res., 57, e2021WR030394, https://doi.org/10.1029/2021WR030394, 2021.

Tejedor, A., Longjas, A., Passalacqua, P., Moreno, Y., and Foufoula-Georgiou, E.: Multiplex networks: A framework for studying multiprocess multiscale connectivity via coupled-network theory with an application to river deltas, Geophys. Res. Lett., 45, 9681–9689, 2018.

Thornton, P. E., Thornton, M. M., Mayer, B. W., Wilhelmi, N., Wei, Y., Devarakonda, R., and Cook, R.: Daymet: Daily surface weather on a 1 km grid for North America, 1980–2008, Oak Ridge National Laboratory (ORNL) Distributed Active Archive Center for Biogeochemical Dynamics (DAAC), https://doi.org/10.3334/ORNLDAAC/1219, 2012.

Todini, E.: A model conditional processor to assess predictive uncertainty in flood forecasting, Int. J. River Basin Manage., 6, 123–137, 2008.

USGS: USGS water data retrieval library, https://github.com/USGS-python/dataretrieval, last access: 6 September 2022a.

USGS: Watershed Boundary Dataset, https://www.usgs.gov/national-hydrography/watershed-boundary-dataset, last access: 6 September 2022b.

Van Beek, L., Wada, Y., and Bierkens, M. F.: Global monthly water stress: 1. Water balance and water availability, Water Resour. Res., 47, 7, https://doi.org/10.1029/2010WR009791, 2011.

Wang, Y., Jin, J., Zhang, W., Yu, Y., Zhang, Z., and Wipf, D.: Bag of tricks for node classification with graph neural networks, arXiv [preprint], https://doi.org/10.48550/arXiv.2103.13355, 2021.

Weerts, A. H., Winsemius, H. C., and Verkade, J. S.: Estimation of predictive hydrological uncertainty using quantile regression: examples from the National Flood Forecasting System (England and Wales), Hydrol. Earth Syst. Sci., 15, 255–265, https://doi.org/10.5194/hess-15-255-2011, 2011.

Weiler, M., McGlynn, B. L., McGuire, K. J., and McDonnell, J. J.: How does rainfall become runoff?, A combined tracer and runoff transfer function approach, Water Resour. Res., 39, 11, https://doi.org/10.1029/2003WR002331, 2003.

Willard, J., Jia, X., Xu, S., Steinbach, M., and Kumar, V.: Integrating physics-based modeling with machine learning: A survey, ACM Computing Surveys (CSUR), https://dl.acm.org/doi/abs/10.1145/3514228, 2020.

Winnick, M. J., Carroll, R. W., Williams, K. H., Maxwell, R. M., Dong, W., and Maher, K.: Snowmelt controls on concentration-discharge relationships and the balance of oxidative and acid-base weathering fluxes in an alpine catchment, East River, Colorado, Water Resour. Res., 53, 2507–2523, 2017.

Winsemius, H. C., Aerts, J. C., Van Beek, L. P., Bierkens, M. F., Bouwman, A., Jongman, B., Kwadijk, J. C., Ligtvoet, W., Lucas, P. L., Van Vuuren, D. P., and Ward, P. J.: Global drivers of future river flood risk, Nat. Clim. Change, 6, 381–385, 2016.

Wu, Z., Pan, S., Long, G., Jiang, J., and Zhang, C.: Graph wavenet for deep spatial-temporal graph modeling, Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19), August 10–16 2019, Macao, China, 2019.

Xia, Y., Mitchell, K., Ek, M., Sheffield, J., Cosgrove, B., Wood, E., Luo, L., Alonge, C., Wei, H., Meng, J., et al.: 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, J. Geophys. Res.-Atmos., 117, D3, https://doi.org/10.1029/2011JD016048, 2012.

Yamazaki, D., de Almeida, G. A., and Bates, P. D.: Improving computational efficiency in global river models by implementing the local inertial flow equation and a vector-based river network map, Water Resour. Res., 49, 7221–7235, 2013.

Yang, T., Sun, F., Gentine, P., Liu, W., Wang, H., Yin, J., Du, M., and Liu, C.: Evaluation and machine learning improvement of global hydrological model-based flood simulations, Environ. Res. Lett., 14, 114027, https://doi.org/10.1088/1748-9326/ab4d5e, 2019.

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

Yeo, I.-K. and Johnson, R. A.: A new family of power transformations to improve normality or symmetry, Biometrika, 87, 954–959, 2000.

Yuval, J. and O'Gorman, P. A.: Stable machine-learning parameterization of subgrid processes for climate modeling at a range of resolutions, Nat. Commun., 11, 1–10, 2020.

Zhang, L., Shi, Z., Han, J., Shi, A., and Ma, D.: Furcanext: End-to-end monaural speech separation with dynamic gated dilated temporal convolutional networks, in: International Conference on Multimedia Modeling, Springer, 653–665, 2020.

Zhao, F., Veldkamp, T. I., Frieler, K., Schewe, J., Ostberg, S., Willner, S., Schauberger, B., Gosling, S. N., Schmied, H. M., and Portmann, F. T.: The critical role of the routing scheme in simulating peak river discharge in global hydrological models, Environ. Res. Lett., 12, 075003, https://doi.org/10.1088/1748-9326/aa7250, 2017.

Zhou, J., Cui, G., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., and Sun, M.: Graph neural networks: A review of methods and applications, AI Open, 1, 57–81, https://doi.org/10.1016/j.aiopen.2021.01.001, 2018.

Zhu, X. and Ghahramani, Z.: Learning from labeled and unlabeled data with label propagation, https://doi.org/10.1.1.14.3864, 2002.