Physics-Guided Graph Meta Learning for Predicting Water Temperature and Streamflow in Stream Networks

Shengyu Chen
University of Pittsburgh
Pittsburgh, PA, USA
shc160@pitt.edu

Jacob A. Zwart
U.S. Geological Survey
Pittsburgh, PA, USA
jzwart@usgs.gov

Xiaowei Jia
University of Pittsburgh
Pittsburgh, PA, USA
xiaowei@pitt.edu

ABSTRACT

This paper proposes a graph-based meta learning approach to separately predict water quantity and quality variables for river segments in stream networks. Given the heterogeneous water dynamic patterns in large-scale basins, we introduce an additional meta-learning condition based on physical characteristics of stream segments, which allows learning different sets of initial parameters for different stream segments. Specifically, we develop a representation learning method that leverages physical simulations to embed the physical characteristics of each segment. The obtained embeddings are then used to cluster river segments and add the condition for the meta-learning process. We have tested the performance of the proposed method for predicting daily water temperature and streamflow for the Delaware River Basin (DRB) over a 14 year period. The results confirm the effectiveness of our method in predicting target variables even using sparse training samples. We also show that our method can achieve robust performance with different numbers of clusterings.

CCS CONCEPTS

• **Information systems** → **Data mining**; **Spatial-temporal systems**.

KEYWORDS

physics-guided machine learning, meta learning, graph neural networks, spatio-temporal data mining, stream networks

ACM Reference Format:
Shengyu Chen, Jacob A. Zwart, and Xiaowei Jia. 2022. Physics-Guided Graph Meta Learning for Predicting Water Temperature and Streamflow in Stream Networks . In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22), August 14–18, 2022, Washington, DC, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3534678.3539115

1 INTRODUCTION

Healthy freshwater ecosystems are key to the future sustainability of our planet as fresh water plays a critical role in the global economic, food, energy, and water networks [18]. According to the Food and Agriculture Organization of the United Nations [10], 88% of available fresh water globally is used for agriculture and industry, including energy production. Stream networks are one of the most common freshwater ecosystems, which has encountered many challenges, such as rapidly degrading water quality, severe droughts and floods, due to the increasing demands for water-based ecosystem services [32, 33] and a shifting climate [41].

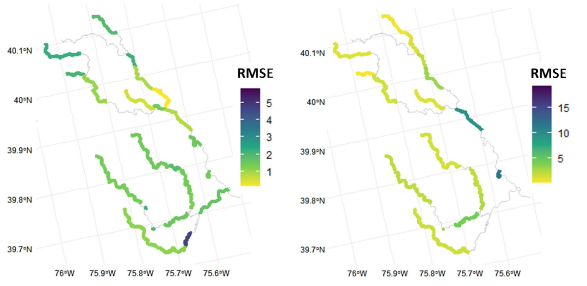
Our objective is to advance our ability for timely prediction of water quantity and quality (e.g., water temperature, streamflow) over large and diverse regions. Such prediction capacity will in turn provide useful information for sound policy and management decisions, establish relationships between ecological outcomes and water properties, and help understand other biogeochemical and ecological processes. For example, water resources managers in the Delaware River Basin need to supply safe drinking water to over 15 million people [40] while also maintaining sufficient streamflow and cool water temperatures in the river segments that are downstream from the reservoirs to maintain the desired habitat for aquatic life [27]. Accurate prediction of water properties in streams helps managers optimize when and how much water to release downstream to maintain the flow and temperature regimes.

The importance of monitoring steam networks for large river basins has been widely recognized as witnessed by the formation of large-scale high-quality data repositories [2, 28, 38]. Given the importance of this problem, scientists in multiple domains have developed physics-based models to simulate different components of water flow in stream networks. Even though these models are based on known physical laws that govern relationships between input and output variables (e.g., mass and energy conservation laws), most physics-based models are necessarily approximations of reality due to incomplete knowledge of certain processes or omission of processes to maintain computational efficiency. In particular, the model predictions often rely on qualitative parameterizations (approximations) based on soil and surficial geologic classification along with topography, land cover and climate input. Hence, such models tend to provide sub-optimal predictive performance. Furthermore, calibration of these physics-based models is often extremely time intensive due to interactions among parameters [4]. For example, the model proposed in this paper takes 8 hours for training while a process-based stream temperature model, the Precipitation-Runoff Modeling System (PRMS) [22] and the coupled Stream Network Temperature Model (SNTemp) [30], can take several days.

The advances in collecting water data have also provided unrealized potential for using data-driven methods to quickly predict water properties. Despite the success of advanced data-driven methods, e.g., deep learning models, in computer vision and natural language processing, these models face several major challenges in

KDD '22, August 14–18, 2022, Washington, DC, USA

ACM ISBN 978-1-4503-9385-0/22/08…$15.00
https://doi.org/10.1145/3534678.3539115



(a) (b)

Figure 1: The prediction root mean squared error (RMSE) of (a) water temperature (b) streamflow over different stream segments in Christina River watershed.

predicting water properties. First, stream networks contain a large number of river segments. These segments evolve over time while also interacting with each other through advected water flows. The water dynamics in river segments can also be affected by water flows from upstream human-built reservoirs. Second, the water dynamics often exhibit a strong variability across different segments due to the variation of their physical characteristics such as soil properties, elevation, and land cover. Many of these characteristics are approximated through various methods or are very costly to measure accurately, and thus physical characteristics may be inaccurate for many river segments. Third, the observations of water quantity and quality can be sparse over space and time due to the substantial cost needed for data collection.

To address the first challenge, prior work combined graph neural networks (GNNs) with additional recurrent network structures to capture both the spatial and temporal dependencies in predicting water properties [5, 17, 23]. For example, our prior work [17] has shown encouraging results in improving the predictive performance in small regions using a graph recurrent network model. However, these graph-based neural networks use a single set of parameters over all the locations, and thus have limits in capturing data variability over large regions. As a result, the model can prioritize certain regions while performing much worse over other regions, e.g., see the prediction of a global graph model trained using all the data samples from multiple river segments in Christina River watershed (a subset of Delaware River Basin) in Fig. 1. This not only limits the overall predictive performance, but may also lead to important problems such as unfair estimation of insurance and subsidy if such models are used as a reference for the risk of droughts or flooding.

To address this issue, we propose a **M**e**T**a-learning **H**eteregeneous **G**raph **N**etworks (MT-HGN) method. This method is based on a heteregeneous graph model, which represents the interactions amongst multiple river segments and reservoirs through advected water flows. The graph model is further enhanced through a meta-learning process to capture the data variability. Given a large number of heteregeneous river segments, traditional meta-learning approaches such as model-agnostic meta learning (MAML) [11] can often produce sub-optimal performance as they only use a single set of initial parameters for the internal algorithm of meta learning. The proposed MT-HGN method aims to further improve the meta-learning process by creating different initial models for different sets of river segments. Ideally, the parameters of these initial models have a better chance at capturing the water variability if they are conditioned on physical characteristics of river segments. However, we cannot directly apply such conditions given that available physical characteristics are often not complete and they are only measured or calibrated for certain segments. Hence, we propose a representation learning method to embed each stream segment using the simulated water property data generated by a physics-based model. The obtained embeddings are then used as additional conditions to build multiple initial models in the meta-learning process. The proposed meta-learning approach allows fine-tuning these initial models into different refined models for different segments, which helps mitigate the data variability over river segments and improve the performance for segments with limited data.

We evaluate our proposed method using the real streamflow and water temperature data from the Delaware River Basin, which covers large areas in New York, Pennsylvania, New Jersey, Delaware, and Maryland. The data is collected by U.S. Geological Survey over the past 40 years. The experiments are designed to evaluate the predictive performance of the proposed method in comparison with existing predictive and meta-learning methods using sparse or localized training data. The performance on different segment clusters and the sensitivity regarding different numbers of clusters are also tested. Our evaluations demonstrate that the proposed MT-HGN method can produce superior predictions over multiple baselines when applied to large and diverse regions. More importantly, we show that the predictions can be improved for regions with a small amount of observations. This confirms that the proposed MT-HGN method is generalizable to other basins with a lower data density.

2 RELATED WORK

Graph neural networks (GNN) have found immense success in commercial applications [15, 43]. Recently, they have also been applied to multiple scientific problems given the ability to model interacting processes in complex physical systems, which commonly requires substantial efforts in calibration for traditional physics-based modeling approaches. In particular, graph neural networks have shown a great promise for modeling water temperature and streamflow in river networks [5, 17, 23]. Despite the accuracy improvement brought by these methods, they are mostly evaluated in small regions or stream regions without reservoirs. The performance of these methods can be severely affected when reservoirs are present in the stream networks but unaccounted for in the graph network.

The graph model used in this paper is inspired by the heterogeneous graph, which is commonly used to represent multiple types of connections amongst multiple types of nodes [31]. Neural network models have been developed to represent such a graph structure and discover knowledge from heterogeneous data [37, 44, 46]. Our previous paper [5] also used heterogeneous graphs to represent the complex stream networks with both river segments and reservoirs. Compared to convolutional neural networks (CNNs), the graph-based model is more flexible in representing spatial dependencies amongst irregularly distributed locations, which are common in environmental applications. Graph-based models are often combined with other models, e.g., Long-Short Term Memory (LSTM), in neural networks to capture other types of data dependencies [7, 17].

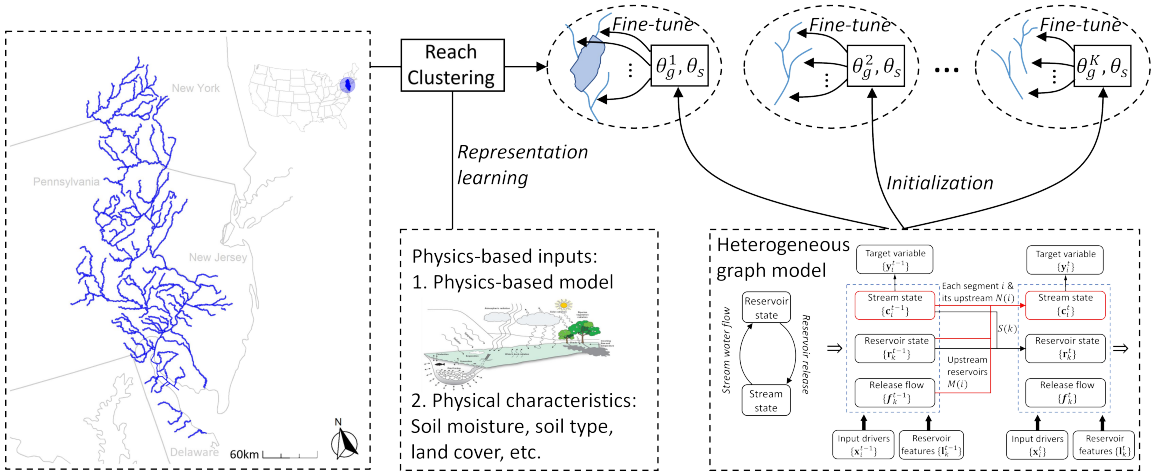


Figure 2: The overall flow of the proposed method, which contains two key components: (i) The heterogeneous graph model (HGN) is used to model the spatial interactions and temporal evolution of water properties in different stream segments (Section 4.1). The HGN model is then refined for different segments in each cluster using a meta-learning approach (Section 4.2). (ii) A representation learning method is created to embed different segments by leveraging their simulated water properties and physical characterstics (Section 4.3). The obtained embeddings are then used to create the segment clustering.

Despite its promise, the graph-based model has not been widely used to represent multiple complex interactions amongst different types of processes in scientific problems. The nature of scientific studies also requires adaptation of these neural network models based on scientific knowledge to better represent the influence amongst processes.

Also, traditional machine learning models or GNNs often cannot reach the same level of success when applied to a large and diverse set of locations in stream networks due to the water variability and the sparse training samples across different locations. Meta-learning has been widely used to adapt machine learning models to new tasks (e.g., a new location or a new environment) with a small number of data samples. The goal of meta-learning is to learn a learning algorithm that can generalize to multiple tasks. For example, MAML [11], which is one of the most popular meta-learning algorithms, aims to learn an initial model that can be quickly fine-tuned given few data samples from a new task. Some other meta-learning approaches build common representation across multiple tasks and also learn task-specific representation from data [34]. These traditional meta-learning methods seek to find a single set of parameters for the internal learning algorithm for all the tasks. For example, MAML only finds a single initial model for all the tasks. Such shared initialization can be limited given the heterogeneous data from large and diverse regions. To address this issue, conditional meta-learning has been proposed to create different initial models that are conditioned on target tasks [8, 36].

The GNN models often face the challenge of limited samples, and thus meta-learning methods have also been used to enhance GNN models. In particular, graph meta-learning has been used to adapt node classification models to new classes with few training samples [9, 42, 45]. These works cannot be directly used for our problem as they are not designed for handling the data variability issue across river segments. Huang et al. [12] also proposed another approach to transfer across multiple tasks in one or more graphs. The transfer process of this method is based on the prototype representation of each class [9, 12, 42]. However, the prototype representation requires the class notion and thus cannot be used for regression problems. Also, all these methods are not designed to leverage underlying physical relationships and characteristics, which can provide useful information in modeling the variability of water dynamics.

3 PROBLEM DEFINITION

We consider N river segments and M reservoirs in a stream network. For each river segment i, we are provided with input features over T daily time steps $\mathbf{X}_i = \{\mathbf{x}_i^1, \mathbf{x}_i^2, ..., \mathbf{x}_i^T\}$. Here input features $\mathbf{x}_i^t$ form a D_x-dimensional vector, which includes weather drivers (e.g., air temperature, precipitation, wind speed) and geometric parameters of the segment (more details can be found in Section 5). For each reservoir k, we are provided with its static D_m-dimensional characteristics $\mathbf{z}_k$ such as the height and width of the dam. We also have the D_s-dimensional physical characteristics $\mathbf{s}_i$ for a subset of stream segments, which include the information of soil property, elevation, land cover, etc. Additionally, we have observed water property (e.g., water temperature or streamflow) $\mathbf{Y} = \{y_i^t\}$ for certain segments and on certain dates. Our objective is to predict water temperature over all the N river segments in the stream network at a daily scale by leveraging the spatial and temporal contextual information.

We use a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{A}\}$ to represent dependencies amongst river segments and reservoirs. Here the node set $\mathcal{V} = \{\mathcal{V}_s, \mathcal{V}_r\}$ contains the set of river segments $\mathcal{V}_s$ and reservoirs $\mathcal{V}_r$. The edge set $\mathcal{E} = \{\mathcal{E}_{ss}, \mathcal{E}_{sr}, \mathcal{E}_{rs}\}$ contains three types of edges among river segments and reservoirs. Specifically, $\mathcal{E}_{ss}$ represents the edges between pairs of segments (i, j) where the segment i is anywhere upstream from the segment j, $\mathcal{E}_{sr}$ represents the edges between river segments and their downstream reservoirs, and $\mathcal{E}_{rs}$ represents the edges between reservoirs and their downstream river segments. The matrix $\mathbf{A} \in \mathbb{R}^{(N+M)\times(N+M)}$ represents the adjacency level

between each pair of river segments or between river segments and reservoirs in the graph. Specifically, $\mathbf{A}_{ij} = 0$ means there is no connection from node i to node j, and a higher value of $\mathbf{A}_{ij}$ indicates a smaller distance between node i and node j in terms of the stream distance.

4 METHOD

In this section, we formally describe our proposed MT-HGN method, as outlined in Fig. 2. We first introduce the heterogeneous graph model (HGN) to represent the dynamics and interactions amongst stream segments and reservoirs. Then we discuss the conditional meta-learning strategy to refine the graph model to different sets of stream segments based on their physical characteristics. Finally, we develop a representation learning method to create the segment clustering and use it as the condition in the meta-learning process.

4.1 Heterogeneous Graph Networks (HGN)

Streams and reservoirs have different patterns of water dynamics while also being affected by each other, i.e., stream water flowing into a reservoir affects the reservoir's temperature and volume, and water release from reservoirs also affects the temperature and streamflow of downstream river segments. Hence, the machine learning (ML) model needs to memorize the state of reservoirs and streams over time and capture their interactions. The intuition of HGN is to use two sets of state variables (stream states $\{\mathbf{c}_i\}$ and reservoir states $\{\mathbf{r}_k\}$, both of dimension D_h) to capture how streams and reservoirs evolve and interact with each other (Fig. 2). The state variable for each river segment or reservoir is a multi-dimensional vector that encodes the influence of weather and the spatio-temporal context. In the following, we describe how to update state variables over time.

State of river segments: For each river segment i, its water property at time t is affected by (1) the stream state at the previous time, (2) the weather input at the current time, (3) the water advected from upstream reservoirs, and (4) the water advected from upstream river segments. Similar to LSTM, we use multiple gating variables to filter the information from different sources and then combine the filtered information to update the stream state $\mathbf{c}_i^t$. This is analogous to the evolution of a dynamical system, in which the state of streams change over time in response to influences from different sources (e.g., solar radiation, advected water) filtered by specific physical conditions. This process is shown as:

$$\mathbf{c}_i^t = \tanh(\mathbf{gf}_i^t \odot \mathbf{c}_i^{t-1} + \mathbf{gi}_i^t \odot \bar{\mathbf{c}}_i^t + \mathbf{gr}_i^t \odot \mathbf{p}_i^{t-1} + \mathbf{gs}_i^t \odot \mathbf{q}_i^{t-1}), \tag{1}$$

where $\odot$ represents the element-wise product, and $\mathbf{gf}_i^t$, $\mathbf{gi}_i^t$, $\mathbf{gr}_i^t$, $\mathbf{gs}_i^t \in \mathbb{R}^{D_h}$ represent the gating variables used to filter the information from historical stream states, the current weather input, upstream reservoirs, and upstream river segments, respectively. The candidate state $\bar{\mathbf{c}}_i^t \in \mathbb{R}^{D_h}$ encodes the information of river segment i at the current time t, $\mathbf{p}_i^{t-1} \in \mathbb{R}^{D_h}$ and $\mathbf{q}_i^{t-1} \in \mathbb{R}^{D_h}$ are the latent variables (referred to as transferred variables) that embed the effect from upstream reservoirs and river segments, respectively. We use the transferred variables from the previous time step to account for the water travel time. We now describe how to compute these variables.

We first follow the same process in LSTM to compute the candidate state $\bar{\mathbf{c}}_i^t$ by combining weather drivers at the current time step $\mathbf{x}_i^t$ and the hidden representation at previous time step $\mathbf{h}_i^{t-1}$ (computed from $\mathbf{c}_i^{t-1}$ by Eq. 6), as follows:

$$\bar{\mathbf{c}}_i^t = \tanh(\mathbf{W}_c^h \mathbf{h}_i^{t-1} + \mathbf{U}_c^x \mathbf{x}_i^t + \mathbf{b}_c), \tag{2}$$

where $\mathbf{W}_c^h \in \mathbb{R}^{D_h \times D_h}$, $\mathbf{U}_c^x \in \mathbb{R}^{D_h \times D_x}$, and $\mathbf{b}_c \in \mathbb{R}^{D_h}$ are model parameters.

For a river segment i, we use the transferred variables $\mathbf{p}_i^{t-1}$ to embed the effect from from its upstream reservoirs. In particular, the effect one river segment i receives from a reservoir depends on the reservoir state, the reservoir's characteristics (e.g., reservoir depth) and its distance to the segment i, as well as the volume of water released from the reservoir (represented as $\mathbf{f}_k$). The transferred variables $\mathbf{p}_i^{t-1}$ combines such information from all the upstream reservoirs of the segment i (represented as $M(i)$) as:

$$\mathbf{p}_i^{t-1} = \tanh(\mathbf{W}_p \sum_{k \in M(i)} \mathbf{A}_{ki} f_1(\mathbf{z}_k) \odot (\mathbf{W}_p^r \mathbf{r}_k^{t-1} + \mathbf{f}_k^{t-1}) + \mathbf{b}_p), \tag{3}$$

where $\mathbf{W}_p \in \mathbb{R}^{D_h \times D_h}$, $\mathbf{W}_p^r \in \mathbb{R}^{D_h \times D_h}$, and $\mathbf{b}_p \in \mathbb{R}^{D_h}$ are trainable model parameters, the function $f_1(\cdot)$ transforms the static reservoir characteristics to the filtering variables in $\mathbb{R}^{D_h}$ with each output variable in the range of [0,1]. We implement $f_1(\cdot)$ using fully connected layers and the sigmoid activation function.

For each river segment i, we also use transferred variables $\mathbf{q}_i^{t-1}$ to capture the effect from all of its upstream river segments (represented as $N(i)$) as follows:

$$\mathbf{q}_i^{t-1} = \tanh(\mathbf{W}_q \sum_{j \in N(i)} \mathbf{A}_{ji} \mathbf{h}_j^{t-1} + \mathbf{b}_q). \tag{4}$$

Then we generate four sets of gating variables using the sigmoid function $\sigma(\cdot)$ as follows:

$$\begin{aligned} \mathbf{gf}_i^t &= \sigma(\mathbf{W}_f^h \mathbf{h}_i^{t-1} + \mathbf{U}_f^x \mathbf{x}_i^t + \mathbf{b}_f), \\ \mathbf{gi}_i^t &= \sigma(\mathbf{W}_g^h \mathbf{h}_i^{t-1} + \mathbf{U}_g^x \mathbf{x}_i^t + \mathbf{b}_g), \\ \mathbf{gr}_i^t &= \sigma(\mathbf{W}_r^p \mathbf{p}_i^{t-1} + \mathbf{U}_r^x \mathbf{x}_i^t + \mathbf{b}_r), \\ \mathbf{gs}_i^t &= \sigma(\mathbf{W}_s^q \mathbf{q}_i^{t-1} + \mathbf{U}_s^x \mathbf{x}_i^t + \mathbf{b}_s), \end{aligned} \tag{5}$$

where $\theta_g = \{\mathbf{W}_f^h, \mathbf{W}_g^h, \mathbf{W}_r^p, \mathbf{W}_s^q\} \in \mathbb{R}^{D_h \times D_h}$, $\{\mathbf{U}_f^x, \mathbf{U}_g^x, \mathbf{U}_r^x, \mathbf{U}_s^x\} \in \mathbb{R}^{D_h \times D_x}$, and $\{\mathbf{b}_f, \mathbf{b}_g, \mathbf{b}_r, \mathbf{b}_s\} \in \mathbb{R}^{D_h}$ are model parameters.

After obtaining the stream state $\mathbf{c}_i^t$ (Eq. 1), we generate the output gating variables $\mathbf{o}_i^t$ and use them to filter the model state to generate the hidden representation $\mathbf{h}^t$, as follows:

$$\begin{aligned} \mathbf{o}_i^t &= \sigma(\mathbf{W}_o^h \mathbf{h}_i^{t-1} + \mathbf{U}_o^x \mathbf{x}_i^t + \mathbf{b}_o), \\ \mathbf{h}_i^t &= \mathbf{o}_i^t \odot \tanh(\mathbf{c}_i^t), \end{aligned} \tag{6}$$

where $\mathbf{W}_o^h \in \mathbb{R}^{D_h \times D_h}$, $\mathbf{U}_o^x \in \mathbb{R}^{D_h \times D_x}$, and $\mathbf{b}_o \in \mathbb{R}^{D_h}$ are model parameters. Finally, we generate predicted target variables $\hat{\mathbf{y}}_i^t$ from the hidden representation, as follows:

$$\hat{y}_i^t = \mathbf{w}_y \mathbf{h}_i^t + b_y, \tag{7}$$

where $\mathbf{W}_y \in \mathbb{R}^{1 \times D_h}$ and $b_y \in \mathbb{R}^1$ are model parameters.

The HGN model is trained to minimize the mean squared error (MSE) loss between observed temperature or streamflow $\mathbf{Y} = \{\mathbf{y}_i^t\}$ and predicted values. The loss is only computed at time steps and locations for which observations are available.

State of reservoirs: Because water flows from upstream river segments can change the temperature and volume of reservoirs, we

update the reservoir state $\mathbf{r}_k^t$ for each reservoir k at time t by incorporating the influence from its upstream river segments at the previous time step $t-1$. The change of reservoir temperature given such influence also depends on the characteristics of the reservoir (e.g., the geometry of reservoirs). Hence, for each reservoir k, we combine the state variables $\mathbf{c}_i$ of its upstream river segments (represented as $S(k)$) and use the static features $\mathbf{z}_k$ to filter the influence from these river segments before updating the reservoir state, as:

$$\mathbf{r}_k^t = \tanh(\mathbf{W}_r \mathbf{r}_k^{t-1} + f_2(\mathbf{z}_k) \odot \sum_{i \in S(k)} \mathbf{A}_{ik} \mathbf{c}_i^{t-1} + \mathbf{b}_r), \tag{8}$$

where $\mathbf{W}_r \in \mathbb{R}^{D_h \times D_h}$ and $\mathbf{b}_r \in \mathbb{R}^{D_h}$ are model parameters, the function $f_2(\cdot)$ is used to convert reservoir characteristics to the filtering variables and is implemented using fully connected layers. Here the influence of each upstream river segment is also weighted by its adjacency level to the reservoir.

4.2 Conditional Meta-learning on HGN

Despite the expectation of the proposed GNNs in capturing the spatial and temporal context for each node, it is limited in handling heterogeneous data over a large number of river segments. In stream networks, water dynamics can be affected by soil properties, surrounding land covers, and other catchment characteristics. Such characteristics can be hard to measure for many river segments and thus are often not included in input features and are instead modeled as parameters in hydrologic process models. Omission of these characteristics makes it difficult for a global GNN model to produce good performance for all the segments. This issue can be exacerbated given the highly sparse or localized training data.

To address this issue, we propose a conditional meta-learning approach that aims to better adapt the graph model to different river segments, i.e., different tasks. Here each task i contains a set of training data $\{\mathbf{X}_i^{tr}, \mathbf{Y}_i^{tr}\}$ and validation data $\{\mathbf{X}_i^{val}, \mathbf{Y}_i^{val}\}$. Using the transformation HGN$(\cdot)$ defined by the HGN model, standard meta-learning approaches (e.g., MAML [11]) aim to learn a learning function $\mathcal{F}(\cdot;\theta)$ such that it can quickly produce task-specific parameters θ_i to help the HGN model adapt to a specific task i (e.g., a specific set of river segments) given some training samples for this task $\{\mathbf{X}_i^{tr}, \mathbf{Y}_i^{tr}\}$. Here we do not show the time dimension t explicitly as the meta-learning approach applies to all the time steps. More formally, the meta-learning aims to find optimal parameters θ for the learning function $\mathcal{F}(\cdot;\theta)$ such that the task-specific parameters it produces on all the tasks help the HGN model achieve the lowest error on their validation data. This can be expressed as:

$$\begin{aligned} &\theta_* = \arg\min_{\theta} \frac{1}{N} \sum_i \mathcal{L}(\text{HGN}(\mathbf{X}_i^{val}; \theta_i), \mathbf{Y}_i^{val}), \\ &\text{where } \theta_i = \mathcal{F}(\mathbf{X}_i^{tr}, \mathbf{Y}_i^{tr}; \theta), \text{for } i = 1 \text{ to } N, \end{aligned} \tag{9}$$

where $\mathcal{L}$ represents the MSE loss used in our work.

Similar to MAML [11], we consider the learning algorithm $\mathcal{F}$ to be a fine-tuning function, i.e., θ_* serves as initial model parameters that will be fine-tuned to each task using their training data. In the meta-learning process, the inner meta update (i.e., the fine-tuning function) can be solved by a few gradient steps, the first-order Taylor expansion, or other closed-form approximation [3]. We adopt the gradient step method, as $\theta_i = \theta - \alpha \frac{\partial \mathcal{L}(\text{HGN}(\mathbf{X}_i^{tr};\theta), \mathbf{Y}_i^{tr})}{\partial \theta}$, where α is the learning rate for the meta update process.

This standard method brings several challenges to our problem. First, given that different streams can have different physical characteristics, a global initial model may not be sufficient to adapt to a large number of river segments with strong variability of water dynamics. Second, direct application of the standard meta-learning approach can be unstable given the large parameter set of the HGN model and the interactions amongst nodes. As the embedding of each node (i.e., each segment) is affected by neighboring nodes, the validation loss on one task (i.e., one segment) could be affected by the task-specific parameters of other tasks.

To address these challenges, we first create different initial models to capture the variability of different streams by introducing an additional condition of physical characteristics (e.g., soil property, elevation, and surrounding land cover). The intuition is to enforce similar initial parameters for segments with similar physical characteristics. Specifically, assuming the physical characteristics $\mathbf{s}_i$ are available, we define a function $g(\mathbf{s}_i)$, which converts the physical characteristics $\mathbf{s}_i$ into the initial parameters of the segment i. Then we represent the meta-learning process as follows:

$$\begin{aligned} &\min_{g} \frac{1}{N} \sum_i \mathcal{L}(\text{HGN}(\mathbf{X}_i^{val}; \theta_i), \mathbf{Y}_i^{val}), \\ &\text{where } \theta_i = \mathcal{F}(\mathbf{X}_i^{tr}, \mathbf{Y}_i^{tr}; g(\mathbf{s}_i)), \text{for } i = 1 \text{ to } N. \end{aligned} \tag{10}$$

According to prior work [25, 36], this problem can be solved by the structured prediction. Inspired by [36], the structured prediction in meta-learning can be expressed as follows based on the structured encoding loss function principle [6]:

$$\begin{aligned} &g(\mathbf{s}_i) = \arg\min_{\theta} \sum_{j=1}^{N} \phi(\mathbf{s}_i, \mathbf{s}_j) \mathcal{L}(\text{HGN}(\mathbf{X}_j^{val}; \theta_j), \mathbf{Y}_j^{val}), \\ &\text{where } \theta_j = \mathcal{F}(\mathbf{X}_i^{tr}, \mathbf{Y}_i^{tr}; \theta), \text{for } i = 1 \text{ to } N, \end{aligned} \tag{11}$$

where $\phi(\mathbf{s}_i, \mathbf{s}_j)$ is the similarity between task i and task j based on their physical characteristics. According to this equation, the optimal parameters θ for each task i is estimated based on the weighted aggregated performance when the model is adapted to all the tasks, and the weights of aggregation depend on the similarity to the task i in terms of physical characteristics.

Given the computational complexity of this method, we propose a simplified method MT-HGN by considering the clustering structure of river segments. We will discuss a representation learning-based clustering method in Section 4.3. This clustering method avoids directly using physical characteristics $\mathbf{s}_i$ for computing the similarity ϕ in Eq. 11. This is useful as physical characteristics may not be measured for all the river segments, and available physical characteristics are often not complete, i.e., some important characteristics are not known and thus not included in the current data. Once we obtain the clustering structure, we rewrite the meta-learning process as follows:

$$\begin{aligned} &g(\mathbf{s}_i) = \arg\min_{\theta} \sum_{j \in C_i} \mathcal{L}(\text{HGN}(\mathbf{X}_j^{val}; \theta_j), \mathbf{Y}_j^{val}), \\ &\text{where } \theta_j = \mathcal{F}(\mathbf{X}_j^{tr}, \mathbf{Y}_j^{tr}; \theta), \text{for } j \in C_i. \end{aligned} \tag{12}$$

Here C_i represents the cluster of segments that contains the segment i. Hence, the proposed method considers only the aggregated effect from segments with similar physical characteristics, which is

also ensured by our clustering method (discussed in Section 4.3). Compared with Eq. 11, this new formulation considers the similarity ϕ between a pair of segments (i, j) in the same cluster to be 1, and 0 otherwise. It is also worth mentioning that this computation can be done even without using the physical characteristics $\mathbf{s}_i$ (as long as the clustering is available). Also, all the segments within the same cluster have the same initial parameters $g(\mathbf{s}_i)$.

To further reduce the complexity of the model, we only update model parameters for gating variables, i.e., $\theta_g = \{\mathbf{W}_f^h, \mathbf{W}_g^h, \mathbf{W}_r^p, \mathbf{W}_s^q, \mathbf{U}_f^x, \mathbf{U}_g^x, \mathbf{U}_r^x, \mathbf{U}_s^x, \mathbf{b}_f, \mathbf{b}_g, \mathbf{b}_r, \mathbf{b}_s\}$, in the internal meta update step. The other parameters (represented as θ_s) are shared across tasks. The selection of these parameters is justified by prior study [16, 19] on adjusting gating variables given different physical characteristics. It was shown in these works that the gating variables are more related to physical characteristics of catchment, and thus refining the parameters θ_g can reflect the variability of water properties.

In our work, we initialize the parameters $\{\theta_s, \theta_g\}$ using a global HGN model that is trained using all the training samples from all the segments. This turns out to contribute to a faster convergence of the conditional meta-learning process. In our implementation, we also freeze the gradient back-propagation for states of neighboring segments and upstream reservoirs when conducting the meta update for each segment. In this way, segments become independent with each other in the internal meta update step. In the following, we will describe how to create the clustering structure.

4.3 Stream Clustering via Physics-Guided Representation Learning

In real stream networks, physical characteristics are not always available (or are highly uncertain). In the absence of measured physical characteristics, there is a potential to identify the data variability from the joint distribution of input $\mathbf{X}_i$ and observed target variable $\mathbf{Y}_i$, i.e., $P(\mathbf{X}_i, \mathbf{Y}_i)$, across different river segments. Because real observations $\mathbf{Y}$ are very sparse over space and time, we use the simulated target variables $\tilde{\mathbf{Y}}$ produced by a physics-based PRMS-SNTemp model [30]. Because the physics-based model is built based on general physical relationships, it is often more generalizable over different scenarios.

To capture the water variability across segments, we concatenate the input features $\mathbf{x}_i$ and simulated label $\tilde{\mathbf{y}}_i$ as an augmented time series for each segment. Then we use a separate HGN model to produce hidden representation $\tilde{\mathbf{h}}_i^{t=1:T} = \text{HGN}([\mathbf{x}_i, \tilde{\mathbf{y}}_i]^{t=1:T})$ at T time steps. Our goal is to aggregate the hidden representation at multiple time steps to a fixed-length embedding. This aggregation process requires the recognition of important time steps, which can be achieved using the attention mechanism [14, 21]. Specifically, we create attention weights as follows:

$$\beta_{1:T} = \text{softmax}(\mathbf{W}_a \tilde{\mathbf{h}}_i^{1:T} + \mathbf{b}_a), \tag{13}$$

where $\mathbf{W}_a \in \mathbb{R}^{1\times D_h}$ and $\mathbf{b}_a \in \mathbb{R}^1$ are attention model parameters. The embedding for each segment can be obtained by the weighted mean over all the time steps using the attention weights, as

$$\mathbf{e}_i = \sum_t \beta_t \tilde{\mathbf{h}}_i^t. \tag{14}$$

To train the parameters involved in this representation learning process, we create a contrastive loss. The goal is to ensure that embeddings for two segments are close to each other if they share similar physical characteristics, and the embeddings are far way from each other if the they have different characteristics. To identify segments with similar characteristics, we conduct a K-means clustering over segments with available physical characteristics $\mathbf{s}_i$, and we use C_i^s to represent the obtained cluster that contains the segment i. The contrastive loss can be formally defined based on the obtained clustering, as follows:

$$\mathcal{L}_{ctr} = -\Big(\sum_{j\in C_i^s} \log\sigma(\mathbf{e}_i \mathbf{W}_{ctr} \mathbf{s}_j)/|C_j^s| - \sum_{j\notin C_i^s} \log\sigma(\mathbf{e}_i \mathbf{W}_{ctr} \mathbf{s}_j)/N_n\Big), \tag{15}$$

where $\mathbf{W}_{ctr} \in \mathbb{R}^{D_h\times D_s}$ is a trainable parameter matrix, and N_n denotes the number of negative pairs (i,j) used in the second term on the right. This contrastive loss is defined only on segments that have available physical characteristics. Here the training process is conducted to optimize the parameters $\mathbf{W}_a$, $\mathbf{b}_a$, $\mathbf{W}_{ctr}$, and the parameters in the HGN model. After training the representation learning model, we have estimates of the embedding $\mathbf{e}$ for all the segments. Then we conduct K-means over the obtained embeddings to obtain the clustering over all the segments. This clustering result differs from the previous clustering $\{C_i^s\}$ as $\{C_i^s\}$ is the K-means result for only segments with available characteristics, and it also does not consider the dynamic input features $\mathbf{x}$. Here we do not enforce the segments in each cluster to be connected in stream networks because distant segments may also share similar physical characteristics. However, the ignorance of upstream segments may degrade the performance of HGN when it is adapted to only the segments within each cluster. Hence, we augment each cluster with the direct upstream segments for every segment in the cluster. The reservoirs associated with the segments in each cluster are also included for this cluster.

5 DATASET

We evaluate the proposed method for predicting stream temperature data collected from the Delaware River Basin (DRB), which is an ecologically diverse region and a watershed along the east coast of the United States that provides drinking water to over 15 million people [40]. The dataset used in our evaluation is from the U.S. Geological Survey's National Water Information System [38] and the Water Quality Portal [28]. Observations at a specific latitude and longitude were matched to river segments that vary in length from 48 to 23,120 meters. The river segments were defined by the geospatial fabric used for the National Hydrologic Model [29], and the river segments are split up to have roughly a 1-day water travel time. We match observations to river segments by snapping observations to the nearest stream segment within a tolerance of 250 m. Observations farther than 5,000 m along the river channel to the outlet of a segment were omitted from our dataset. See [26] for the full observational dataset.

In particular, DRB contains 456 stream segments and 16 reservoirs. We use input features at the daily scale from Jan 01, 1980, to June 22, 2020 (14,784 dates). The input features have 10 dimensions, which include daily mean precipitation, daily mean air temperature, date of the year, solar radiation, shade fraction, potential

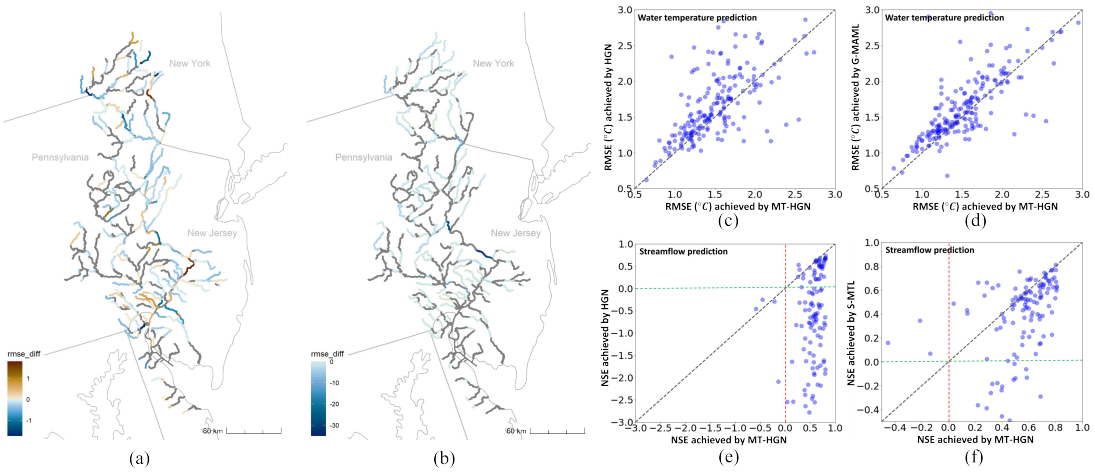


Figure 3: (a)-(b) The RMSE difference between MT-HGN and HGN for each segment in the DRB in (a) water temperature and (b) streamflow prediction. The gray colored segments have no observations in the test period for evaluation. (c)-(d) The per-segment water RMSE comparison for predicting water temperature between (c) MT-HGN and HGN, and (d) MT-HGN and G-MAML. (e)-(f) The per-segment NSE comparison for predicting streamflow between (e) MT-HGN and HGN, and (f) MT-HGN and S-MTL. The green line and the red line in (e) and (f) represent the NSE baseline.

evapotranspiration and the geometric features of each segment (elevation, length, slope, and width). Air temperature, precipitation, and solar radiation values were derived from the gridMET gridded meteorological dataset [1]. Other input features (e.g., shade fraction, potential evapotranspiration) are difficult to measure frequently, and we use values internally calculated by the physics-based PRMS-SNTemp model [30]. Amongst 456 segments, water temperature and streamflow observations were available for 290 and 183 segments, respectively. These observations are only available on certain dates. The number of temperature observations available for the 290 observed segments ranges from 1 to 13,000 with a total of 326,558 observations across all dates and segments. The number of streamflow observations available for the 183 segments range from 16 to 14,774 with a total of 1,928,445 streamflow observations across all dates and segments. The DRB has a higher data density than many other basins, which enables testing the model performance with different levels of data sparsity. For all the reservoirs, we also have meta-features of these reservoirs, including dam height, dam length, depth, elevation, and area of catchment [26, 40]. In addition, we have 39-dimension physical characteristics for a subset of segments, and these physical characteristics include the information about soil property, land cover, canopy, geographic location, etc. [35]. The characteristics for other segments are missing due to missing hydrologic response units (HRUs).

6 RESULTS

We compare to a set of baselines in our experiment:

Physics-based SNTemp: The Precipitation-Runoff Modeling System (PRMS) [22] and the coupled Stream Network Temperature Model (SNTemp) [30] is a physics-based model that simulates daily streamflow and water temperature for river networks. This model has been used to simulate catchment hydrologic variables at regional [20] to national scales [29] in support of resource management decisions. We treat it as a deterministic model and do not consider its parametric or structural uncertainty.

Table 1: Performance of predicting water temperature and streamflow in terms of RMSE. We report mean and standard deviation of RMSE out of five runs.

Method	Temperature ($^\circ C$)	Streamflow (m^3/s)
SNTemp	4.01(±NA)	16.22(±NA)
LSTM	1.91(±0.06)	20.74(±0.07)
HGN	1.72(±0.07)	17.18(±0.08)
G-MAML	1.70(±0.09)	18.24(±0.13)
S-MTL	1.71(±0.08)	16.18(±0.10)
MT-HGN	**1.54**(±0.07)	**14.17**(±0.08)

LSTM: Here we train a global recurrent neural network model (with the LSTM cell) for all the river segments in the DRB.

HGN: We train the HGN model (described in Section 4.1) for all the river segments in the DRB.

Graph-based Model Agnostic Meta Learning (G-MAML): To adapt the prior work [12] to our problem, we use the clustering obtained through our representation learning to create the local embedding for each node within its corresponding cluster, and then train the predictive network using MAML [11] over all the segments.

Static characteristics-based Meta Transfer Learning (S-MTL) [39]: As a meta transfer learning approach developed for studying freshwater systems, this approach learns a similarity function based on characteristics of stream segments (using our obtained segment embeddings $\mathbf{e}$), and then uses it to aggregate individual model predictions in an ensemble way. Here we train the individual models for each obtained cluster because we may have few training observations for some segments.

We train each model using data from January 01, 1980, to October 31, 2006 (using the latter half for validation), and evaluate the model performance using data from November 01, 2006, to March 31, 2020.

6.1 Predictive Performance

In Table 1, we summarize the performance of our method (using 10 clusters) against all the baselines in terms of the prediction RMSE. Because of the variation of streamflow magnitude and the number

Table 2: Performance of predicting water temperature and streamflow using different amounts of training samples.

Method	Water temperature ($^\circ C$)				Streamflow (m^3/s)			
	0.1%	1%	10%	100%	0.1%	1%	10%	100%
LSTM	2.88(±0.08)	2.37(±0.09)	2.04(±0.07)	1.91(±0.06)	22.27(±0.08)	21.97(±0.07)	21.54 (±0.07)	20.74(±0.07)
HGN	2.59(±0.09)	1.98(±0.07)	1.85(±0.06)	1.72(±0.07)	19.94(±0.10)	18.38(±0.09)	18.28(±0.08)	17.18(±0.08)
G-MAML	4.41(±0.14)	1.88(±0.11)	1.81(±0.09)	1.70(±0.09)	23.83(±0.17)	23.63(±0.15)	22.19(±0.15)	18.24(±0.13)
S-MTL	4.13(±0.13)	2.18(±0.13)	1.80(±0.11)	1.71(±0.08)	20.78(±0.14)	18.86(±0.13)	18.03(±0.14)	16.18(±0.10)
MT-HGN	**2.46**(±0.09)	**1.77**(±0.08)	**1.62**(±0.08)	**1.54**(±0.07)	**18.12**(±0.13)	**16.87**(±0.11)	**15.06**(±0.09)	**14.17**(±0.08)

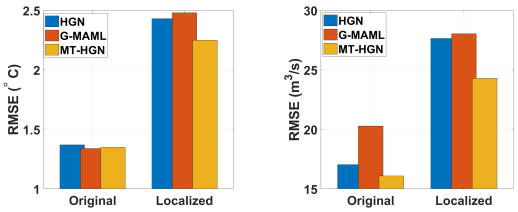
of streamflow observations across segments, we report segment-wise mean RMSE for the streamflow prediction. The segment-wise RMSE is measured by first computing the RMSE separately for each segment and then taking the mean over all the segments. We can observe that MT-HGN outperforms all the other methods. The meta-learning-based methods, i.e., G-MAML, S-MTL, and MT-HGN, generally achieve better performance than the other methods because of they capture the variability of water dynamic patterns over different segments. However, G-MAML gets worse performance than HGN for streamflow prediction because streamflow patterns are very different amongst segments, so a single set of initial parameters performs worse. S-MTL does not perform as well as MT-HGN because it only considers the data variability at the cluster level but does not model the variations of water dynamics for segments within a cluster.

The improvement from LSTM to HGN confirms the importance of incorporating spatial context in predicting water properties. It can be also seen that the physics-based SNTemp performs much better for streamflow as streamflow exhibits a higher degree of variability across segments, and the physical mechanism used in building SNTemp can better capture such variability than global data-driven models.

We compute the difference of the prediction RMSE between MT-HGN and HGN and visualize the spatial distribution of the RMSE difference in Fig. 3 (a)-(b). We also show the segment-wise performance comparison between MT-HGN and HGN (Fig. 3 (c) and (e)), and between MT-HGN and the second best performing method (G-MAML for water temperature and S-MTL for streamflow, Fig. 3 (d) and (f)) over each segment. For the ease of visualization, we compute the Nash–Sutcliffe model efficiency coefficient (NSE) [24] for measuring streamflow performance due to the variation of streamflow magnitude across segments. The NSE value ranges from $[-\inf, 1]$ and the higher value indicates the better performance. It can be seen that MT-HGN not only improves the overall performance, but also performs better for most individual segments. In particular, MT-HGN outperforms HGN for 72% segments and 95% segments in predicting water temperature and streamflow, respectively. We also study the effect of meta-learning to small streams in the Appendix.

6.2 Sparse and localized labels

We also aim to verify that the proposed MT-HGN method can produce good prediction even with sparse observations of target variables. This is especially important if the MT-HGN method is applied to other large basins with a small amount of observations. In particular, we conduct two sparse tests. First, we randomly remove a certain proportion of training samples and measure the performance of each method in the testing period. Second, we consider the collected data are highly localized and verify that the



(a) (b)

Figure 4: The performance of HGN, G-MAML, and MT-HGN for predicting (a) water temperature and (b) streamflow on target segments before and after we hide training samples from these segments.

method can produce good predictions for unobserved segments. Specifically, we remove data from 22 segments and 20 segments for predicting water temperature and streamflow, respectively, and then measure the testing performance only on these segments. For water temperature prediction, we select these 22 segments that have more than 400 training and more than 200 testing observations. For streamflow prediction, we randomly select 20 segments from a set of 84 segments that have observations for all the training and testing period. The goal is to examine the performance drop after removing the training data from the selected segments using sufficient testing data for evaluation.

We show the performance of each method using different proportion of training data in Table 2. The performance of each method becomes worse using less training data. The performance of MT-HGN is better compared to other methods when using sparse training data. Specifically, its performance when using 1% and 10% data is close to its performance using complete data. When we use 0.1% training data, all the methods perform poorly. In particular, G-MAML and S-MTL perform poorly using 0.1% data because G-MAML is more likely to overfit the small data when fine-tuning to each segment and S-MTL can learn inaccurate similarity estimation across different clusters.

For the localized test, we show the performance on the hidden segments before and after we hide the data (Fig. 4). Although MT-HGN produces better performance than HGN and G-MAML after the removal of training data, all the methods have much worse performance in this scenario. It would be beneficial to pursue improving the prediction in unobserved or poorly observed segments through active learning in future work.

6.3 Performance over the clustering

We measure the RMSE of HGN, G-MAML, and MT-HGN for streamflow prediction in each cluster, as shown in in Fig. 5. The results for water temperature prediction is also included in the Appendix. We

Figure 5: The RMSE in each cluster for streamflow prediction.

can see that segments in different clusters can have different RMSE values due to the variation of segment characteristics across space, and MT-HGN can improve the performance for most clusters using either full data or sparse data. We also report the model sensitivity to the number of clusters in the Appendix.

7 CONCLUSION

In this paper, we introduce a novel conditional meta learning approach for predicting water temperature and streamflow in stream networks. We create the representation of physical characteristics as the condition for the meta-learning, which allows learning different initial parameters for different sets of stream segments. Our results show the superiority of our method over other baselines with a considerable margin. Also, our method can perform well even with sparse observation samples. Finally, our method can improve the performance for most segment clusters, and the performance is robust against different numbers of clusters. Future investigation of active learning for data collection could benefit the prediction over unmonitored stream segments with no training data. The incorporation of physical knowledge [13, 17] could also help improve the model generalizability over space and time.

8 ACKNOWLEDGEMENTS

S. C. and X. J. were supported by the NSF award 2147195 and USGS Grants G21AC10207 and G21AC10564. This research was supported in part by the University of Pittsburgh CRC through the resources provided. We thank Jeremy Diaz and Jared Smith for their very helpful review on an earlier version of the manuscript. Any use of trade, firm, or product names is for descriptive purposes only and does not imply endorsement by the U.S. Government.

REFERENCES

[1] 2021. gridMET - Climatology Lab. http://www.climatologylab.org/gridmet.html.
[2] Nans Addor et al. 2017. The CAMELS data set: catchment attributes and meteorology for large-sample studies. *Hydrology and Earth System Sciences* (2017).
[3] Luca Bertinetto et al. 2018. Meta-learning with differentiable closed-form solvers. *arXiv preprint arXiv:1805.08136* (2018).
[4] Keith Beven. 2006. A manifesto for the equifinality thesis. *Journal of Hydrology* 320, 1-2 (2006), 18–36.
[5] Shengyu Chen et al. 2021. Heterogeneous stream-reservoir graph networks with data assimilation. In *ICDM*.
[6] Carlo Ciliberto, Francis Bach, and Alessandro Rudi. 2019. Localized structured prediction. *Advances in Neural Information Processing Systems* 32 (2019).
[7] Zhiyong Cui et al. 2019. Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting. *TITS* (2019).
[8] Giulia Denevi et al. 2020. The advantage of conditional meta-learning for biased regularization and fine tuning. *NeurIPS* (2020).
[9] Kaize Ding et al. 2020. Graph prototypical networks for few-shot learning on attributed networks. In *CIKM*.
[10] FAO. 2014. The state of the world's land and water resources for food and agriculture. http://www.fao.org/docrep/017/i1688e/i1688e00.htm
[11] Chelsea Finn et al. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*.
[12] Kexin Huang et al. 2020. Graph meta learning via local subgraphs. *NeurIPS* (2020).
[13] Xiaowei Jia et al. 2019. Physics guided RNNs for modeling dynamical systems: A case study in simulating lake temperature profiles. In *SDM*.
[14] Xiaowei Jia et al. 2019. Spatial context-aware networks for mining temporal discriminative period in land cover detection. In *SDM*. SIAM.
[15] Xiaowei Jia et al. 2020. Personalized Image Retrieval with Sparse Graph Representation Learning. In *SIGKDD*.
[16] Xiaowei Jia et al. 2021. Physics-guided machine learning from simulation data: An application in modeling lake and river systems. In *ICDM*.
[17] Xiaowei Jia et al. 2021. Physics-guided recurrent graph model for predicting flow and temperature in river networks. In *SDM*.
[18] Megan Konar et al. 2011. Water for food: The global virtual water trade network. *Water Resources Research* (2011).
[19] Frederik Kratzert et al. 2019. Towards learning universal, regional, and local hydrological behaviors via machine learning applied to large-sample datasets. *Hydrology and Earth System Sciences* (2019).
[20] Jacob H LaFontaine et al. 2013. Application of the Precipitation-Runoff Modeling System (PRMS) in the Apalachicola-Chattahoochee-Flint River Basin in the southeastern United States. *USGS* (2013).
[21] Fenglong Ma et al. 2017. Dipole: Diagnosis prediction in healthcare via attention-based bidirectional recurrent neural networks. In *SIGKDD*.
[22] Steven L Markstrom et al. 2015. PRMS-IV, the precipitation-runoff modeling system, version 4. *USGS* (2015).
[23] Zach Moshe et al. 2020. HydroNets: Leveraging River Structure for Hydrologic Modeling. In *ICLR*.
[24] J Eamonn Nash et al. 1970. River flow forecasting through conceptual models part I—A discussion of principles. *Journal of Hydrology* (1970).
[25] Sebastian Nowozin and Christoph H Lampert. 2011. *Structured learning and prediction in computer vision*. Vol. 6. Now publishers Inc.
[26] Samantha K Oliver et al. 2021. Predicting water temperature in the Delaware River Basin. U.S. Geological Survey Data Release. https://doi.org/10.5066/P9GD8I7A
[27] Arun Ravindranath et al. 2016. An environmental perspective on the water management policies of the Upper Delaware River Basin. *Water Policy* (2016).
[28] Emily K Read et al. 2017. Water quality data for national-scale aquatic research: The Water Quality Portal. *Water Resources Research* (2017).
[29] R Steven Regan et al. 2018. *Description of the national hydrologic model for use with the precipitation-runoff modeling system (PRMS).* Technical Report. US Geological Survey.
[30] Michael J Sanders et al. 2017. *Documentation of a daily mean stream temperature module—An enhancement to the Precipitation-Runoff Modeling System.* Technical Report. US Geological Survey.
[31] Chuan Shi et al. 2016. A survey of heterogeneous information network analysis. *TKDE* (2016).
[32] Samir Suweis et al. 2013. Water-controlled wealth of nations. *PNAS* (2013).
[33] Stefania Tamea et al. 2016. Global effects of local food-production crises: a virtual water perspective. *Scientific Reports* (2016).
[34] Nilesh Tripuraneni et al. 2021. Provable meta-learning of linear representations. In *ICML*. PMLR.
[35] RJ Viger. 2014. Preliminary spatial parameters for PRMS based on the Geospatial Fabric, NLCD2001 and SSURGO. *US Geological Survey, doi* 10 (2014), F7WM1BF7.
[36] Ruohan Wang et al. 2020. Structured prediction for conditional meta-learning. *NeurIPS* (2020).
[37] Xiao Wang et al. 2019. Heterogeneous graph attention network. In *WWW*.
[38] US Geological Survey. USGS water data for the Nation: U.S. Geological Survey National Water Information System database. 2016. http://doi.org/10.5066/F7P55KJN. Accessed: 2021-10-01.
[39] Jared D Willard et al. 2021. Predicting water temperature dynamics of unmonitored lakes with meta-transfer learning. *WRR* (2021).
[40] Tanja N Williamson et al. 2015. *Summary of hydrologic modeling for the Delaware River Basin using the Water Availability Tool for Environmental Resources (WATER).* Technical Report. U.S. Geological Survey Scientific Investigations Report.
[41] EM Wolkovich et al. 2014. Temporal ecology in the Anthropocene. *Ecology Letters* (2014).
[42] Huaxiu Yao et al. 2020. Graph few-shot learning via knowledge transfer. In *AAAI*.
[43] Rex Ying et al. 2018. Graph convolutional neural networks for web-scale recommender systems. In *SIGKDD*.
[44] Chuxu Zhang et al. 2019. Heterogeneous graph neural network. In *SIGKDD*.
[45] Fan Zhou et al. 2019. Meta-gnn: On few-shot node classification in graph meta-learning. In *CIKM*.
[46] Zhihua Zhu et al. 2020. HGCN: A heterogeneous graph convolutional network-based deep learning model toward collective classification. In *SIGKDD*.

A APPENDIX

A.1 Streamflow predictions on small streams

According to the prediction results, MT-HGN performs better than HGN for many small streams for predicting streamflow as these streams have less contribution to the overall RMSE loss used by the HGN model. In Fig. 6, we show the predictions made by HGN and MT-HGN in a stream segment. It can be seen that the HGN over-predict streamflow in this segment and the MT-HGN method matches observations much better, which confirms the effectiveness of fine-tuning the model to different sets of segments. It is noteworthy that this does not necessarily imply a larger RMSE improvement on small streams because the large stream often has a higher variance of streamflow and thus it is easier to achieve RMSE reduction.

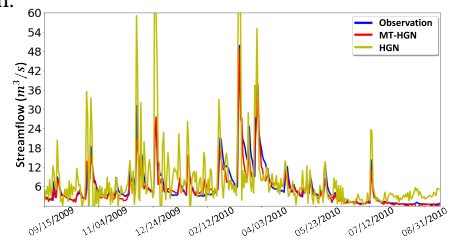


Figure 6: The streamflow predictions made by HGN and MT-HGN on a segment with low streamflow magnitude.

A.2 Performance of predicting water temperature over different clusters

We report the RMSE of HGN, G-MAML, and MT-HGN for water temperature prediction in each cluster, as shown in in Fig. 7. The RMSE values differs across clusters because the variance of water temperature changes over space due to the variation of segment characteristics and the amount of observations also vary across different clusters. MT-HGN can improve the performance for most clusters using either full data or sparse data. This confirms the effectiveness of the proposed method.

Figure 7: The prediction RMSE in each cluster for water temperature.

A.3 Performance variation using a different number of clusters

We also test the sensitivity of the performance using a different number of clusters (Fig. 8). The result shows that the MT-HGN model consistently outperforms HGN by a decent margin using different numbers of clusters. This also confirms that MT-HGN is relatively insensitive to the choice of number of clusters to use.

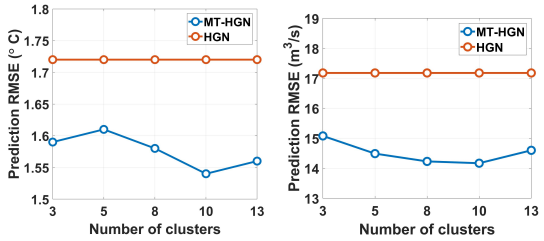


Figure 8: Variation of prediction RMSE for (a) water temperature prediction and (b) streamflow prediction using a different number of clusters.