NORTHWESTERN UNIVERSITY

Hydro-GRNNI: Hydrological Graph Recurrent Neural Network for Imputation

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

MASTER OF SCIENCE IN STATISTICS AND DATA SCIENCE

Field of Machine Learning

By

Shuo Han

EVANSTON, ILLINOIS

August 2024

ABSTRACT

Surface water quality observations across U.S. watersheds are often spatio-temporally sparse, complicating efforts to assess water quality goals and calibrate high-resolution models. Traditional statistical tools like the USGS's Exploration and Graphics for RivEr Trends (EGRET) [1] and Hydrologic and Climatic Analysis Toolkit (HydroClimATe) [2] primarily rely on regression methods to analyze hydrologic and climatic data and estimate pollutant loads based on flow and concentration relationships, but they overlook geographical relationships and struggle with extensive missing data. Recent advancements in spatial and temporal regression techniques have improved estimation accuracy, yet they lack real-time synchronization with upstream-downstream dynamics and spatio-temporal variations in hydrological parameters.

This study proposes a novel approach, *Hydro-GRNNI*, which enhances the resolution of spatial and temporal hydrological data by integrating river flow direction information with spatio-temporal inputs, leveraging Graph Recurrent Neural Network (GRNN). By constructing a physical flow direction graph through the novel Distance Matrix Converter (DMC), this approach establishes directional relationships among river monitoring stations and employs spatio-temporal encoders for data imputation.

Our methodology, applied to the Maumee River Basin, leverages extensive historical records of water quality on flow sediment concentrations. This paper further explores the application of flow direction information into the GRNN framework in predicting sediment and nutrient loading within the basin, presenting comparative analyses with traditional imputation methods to demonstrate the advantages of our approach. Empirical evaluations of *Hydro-GRNNI* have demonstrated its superior performance compared to state-of-the-art methods. Compared to existing benchmarks, *Hydro-GRNNI* consistently achieves superior performance in terms of mean absolute error for

hydrological data imputation. And the method does not rely on assumptions about the distribution of missing values or the presence and duration of transient dynamics, making it robust to various types of hydrological data incompleteness.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION AND BACKGROUND

A common issue in interconnected systems of surface water data observations, such as monitoring stations, is incomplete data caused by faults or network failures that frequently result in missing values. The spatio-temporal sparsity of hydrological data across many US watersheds further complicates the assessment of hydrological data goals, impedes effective management, and challenges the calibration of high-resolution hydrological models.

Popular statistical tools such as the United States Geological Survey (USGS)'s Load Estimator (LOADEST) [3] and EGRET [1] estimate daily pollutant loads using regression methods based on the relationship between flow and pollutant concentrations. However, these tools often neglect geographical relationships and struggle when faced with substantial amounts of missing data. Recent advancements aim to address these limitations through spatial regression, spatial-stepwise temporal regression, and lumped spatial characteristic approaches. Despite these efforts, existing methods lack real-time synchronization with upstream-downstream relationships and the spatio-temporal variations of hydrological parameters.

In parallel, imputation methods have advanced significantly, particularly through deep learning techniques for multivariate time series imputation (MTSI). Deep autoregressive models, such as GRU-D [4], handle missing data by managing the decay of hidden states, while BRITS improves spatial imputation by accounting for correlations across channels. Adversarial approaches, like GAIN [5], use GANs [6] for imputation, and other methods focus on generating synthetic sequences. Additionally, innovative models like graph-based spatio-temporal denoising autoencoders [7] and multiscale models [8] address specific data types and highly sparse time series.

Recently, the Graph Recurrent Imputation Network (GRIN) [9] has emerged, leveraging graph-based recurrent neural architectures to capture both spatial and temporal dependencies effectively. Despite these advancements in imputation methods, existing approaches often fall short for hydrological data, as they do not adequately consider the crucial upstream-downstream relationships necessary for accurate imputation. This gap highlights the need for a more robust and specialized framework to advance the state of the art in hydrological data.

In response, our work builds on GRIN [10] by incorporating Graph Neural Networks (GNNs) [11]–[13] to enhance the resolution of spatial and temporal hydrological data. Our method, Hydrological Graph Recurrent Neural Network for Imputation (*Hydro-GRNNI*), utilizes a physical flow direction graph established through the newly developed DMC to define numerical relationships among river monitoring stations. It employs spatio-temporal encoders to effectively manage time steps and decode spatial connections, resulting in continuous and accurate monitoring data for each station. We test our method in the Maumee River Basin, which provides extensive records of flow sediment concentrations, making it ideal for evaluating river water quality. Initial results are promising, showing accurate predictions of sediment concentrations even with missing data among monitoring stations. This paper demonstrates the effectiveness of our framework in predicting sediment concentrations within the basin. We compare our method against various baselines and alternative approaches, including those incorporating flow direction information and station locations, to highlight its performance. The paper is organized as follows: Chapter 2 reviews related research, setting the foundation for our study. Chapter 3 outlines the problem settings and the hydrological data imputation task. In Chapter 4, we introduce our novel imputation framework. Chapter 5 provides an overview of the dataset used for evaluation. Chapter 6 presents a detailed comparison of our method against state-of-the-art techniques. Finally, Chapter 7 concludes with a summary of our findings. The artifact is released at: `https://github.com/hshuoshuo/Flow-GRNNI`.

CHAPTER 2
RELATED WORK

2.1 Hydrological Data Analysis

The USGS provides several robust tools for hydrological data analysis and imputation of missing data. The EGRET [1] packages in R facilitate the analysis of long-term water quality and streamflow data, utilizing methods like Weighted Regressions on Time, Discharge, and Season (WRTDS) to identify trends. Hydrologic AnalySis Package (HASP) [14] and HYdrologic Surface Water Analysis Package (Hyswap) [15] packages focus on groundwater and surface water data analysis, respectively, providing statistical and visualization tools for understanding temporal changes. HydroClimATe offers a versatile suite of methods for analyzing climatic and hydrologic time-series data, including regression, correlation, and spectral analysis. LOADEST [3] is specifically designed for estimating constituent loads in streams and rivers, employing regression models to predict loads based on streamflow and concentration data. However, while these tools are highly effective, they primarily rely on traditional statistical methods, which may not fully leverage the complex dependencies in hydrological data, especially the upstream-downstream relationships and spatio-temporal variations.

2.2 Spatio-temporal Imputation

There exists a substantial body of literature on missing value imputation in time series data. Initial methods focused on simple interpolation techniques, such as imputing with the mean of observed values. More advanced methods leverage standard forecasting techniques and similarities among

time series, such as k-nearest neighbors, which takes geometrical information into consideration. Recently, deep learning approaches for multivariate time series imputation (MTSI) have emerged, including deep autoregressive methods based on recurrent neural networks (RNNs) and adversarial training frameworks. Notable models include BRITS [16], which uses bidirectional RNNs for spatial imputation, and GAIN, which employs GANs for imputation in i.i.d. settings.

2.3 GNNs for Spatio-temporal Analysis

GNNs have also been widely used in spatio-temporal forecasting by adapting standard neural network architectures for sequential data to the graph domain. Examples include GRU cells implemented with spectral GNNs and diffusion-convolutional networks. Other approaches involve spatio-temporal convolutional neural networks and attention-based models with Transformer-like architectures. Additionally, some research focuses on learning the graph structure underlying multivariate time series or predicting changes in graph topology, as seen in Temporal Graph Networks. Recently, GNNs have been proposed for imputing missing features in i.i.d. data, utilizing adversarial frameworks for data reconstruction and bipartite graph representations for feature imputation. They have also been applied to spatial interpolation [17]. A notable GNN-based method for generic multivariate time series imputation is the GRIN [10], which leverages relational information and nonlinear spatio-temporal dependencies, where specifically the location information of different stations are applied. However, these methods are all general imputation techniques, and none have been specifically designed for hydrological data, which has a unique spatial aspect in flow direction that significantly aids in imputation. This highlights a distinct advantage of our approach, as it directly addresses this specific need.

CHAPTER 3

PRELIMINARIES

3.1 Hydrological Data as Graph Structures

Hydrological data can be effectively modeled within this framework by representing each monitoring station as a node and constructing the adjacency matrix using directional information to define edge weights between nodes. Utilizing the GRIN framework [9], we analyze sequences of weighted directed graphs, where each graph G_t is characterized by a node-attribute matrix and an adjacency matrix derived from our direction matrix. Each graph comprises N_t nodes at each time step t. The node-attribute matrix at time t, denoted as X_t, has dimensions $N_t \times d$. Each row of X_t represents the attributes of a node, with the i-th node having an attribute vector of dimension d. The adjacency matrix W_t at time t, with size $N_t \times N_t$, contains elements $w_{i,j}^t$ that denote the weight of the edge connecting nodes i and j, converted from the directional information provided. This framework captures both direct and indirect upstream-downstream relationships. A direct relationship exists when stations are directly connected in an upstream-downstream manner, while an indirect relationship is established through intermediate stations mediating the connection between two nodes. We define W_t using standard similarity metrics and adjusted thresholded kernels. We assume the input hydrological data channels are homogeneous, meaning the monitoring stations are of the same type. In our process, nodes are uniquely identified to ensure consistent processing across time steps. We also assume a fixed topology over time, with the adjacency matrix W_t remaining constant and N_t unchanged.

3.2 Hydrological Data Imputation

In this section, we apply imputation techniques to hydrological data using a graph-based approach above. We explore two data partitioning strategies: in-sample and out-of-sample imputation. For in-sample imputation, the model is trained on the entire sequence $X_{[t,t+T]}$, excluding only the data that has been removed or masked to simulate additional failures for evaluation purposes. In contrast, out-of-sample imputation involves training and evaluating the model on disjoint sequences derived from $X_{[t,t+T]}$. This setup mimics the scenario where a model trained on available data is used to impute missing values in a completely new target sequence. In both cases, the model does not have access to the ground-truth data used for the final evaluation.

To handle missing values, we use a binary mask $M_t \in \{0,1\}^{N_t \times d}$ at each time step t. Each row m_t^i of the mask indicates the availability of node attributes in X_t: if $m_t^{i,j} = 0$, the attribute $x_t^{i,j}$ is missing during training; if $m_t^{i,j} = 1$, the attribute $x_t^{i,j}$ contains the actual monitored value.

We denote $\tilde{X}_t$ as the ground truth node-attribute matrix, which contains no missing data. For our experiments, we assume stationarity in the missing data distribution and adhere to the missing at random scenario [18]. We generate a random number of concurrent monitoring failures and specify the length of missing data blocks to simulate multiple failures over time. Consequently, the imputation performance is expected to vary with both the number of concurrent faults and the duration of missing data bursts.

The primary goal of hydrological data imputation is to accurately reconstruct missing values within a sequence of input data. To evaluate the imputation performance, we compute the reconstruction error for a given graph sequence $G_{[t,t+T]}$ of length T as follows:

$$L\left(\hat{X}_{[t,t+T]}, \tilde{X}_{[t,t+T]}, M_{[t,t+T]}\right) = \sum_{h=t}^{t+T} \sum_{i=1}^{N_t} m_i^h \left\langle \hat{x}_i^h - \tilde{x}_i^h, \hat{x}_i^h - \tilde{x}_i^h \right\rangle,$$

where $\hat{x}_i^h$ represents the imputed value at time h for node i, and $\tilde{x}_i^h$ denotes the ground truth value at time h for node i. The binary mask m_i^h indicates the availability of the attribute, ensuring that only observed values contribute to the reconstruction error. The term $\langle \cdot, \cdot \rangle$ represents the element-wise error function.

CHAPTER 4
METHODOLOGY

4.1 Distance Matrix Converter(DMC)

4.1.1 Converter Overview

An innovative feature of our paper is the incorporation of flow direction information among stations into *Hydro-GRNNI* through the distance matrix converter(DMC). The original flow direction matrix F captures only direct upstream-downstream connections, leaving indirect relationships, which involves passing-by stations, unrepresented. Relying solely on F for generating an adjacency matrix would overlook potential connections with nearby stations. To address this, we designed the converter to derive the distance matrix D from the flow direction matrix F.

As illustrated in Figure 4.1, this converter computes the shortest path step counts between all monitoring stations. By capturing detailed distance information, it greatly enhances our imputation process by leveraging the full network of inter-station relationships. This comprehensive spatial information allows for more accurate modeling and imputation of missing hydrological data, ensuring that the interdependencies between stations are fully considered.

4.1.2 Converter Details

Inspired by the Floyd-Warshall algorithm [19], our DMC transforms the flow direction matrix F into the distance matrix D through an iterative process. This converter incrementally calculates the shortest path step counts between stations by treating each station as an intermediate point along potential paths. At each iteration, the algorithm compares the current shortest step counts between

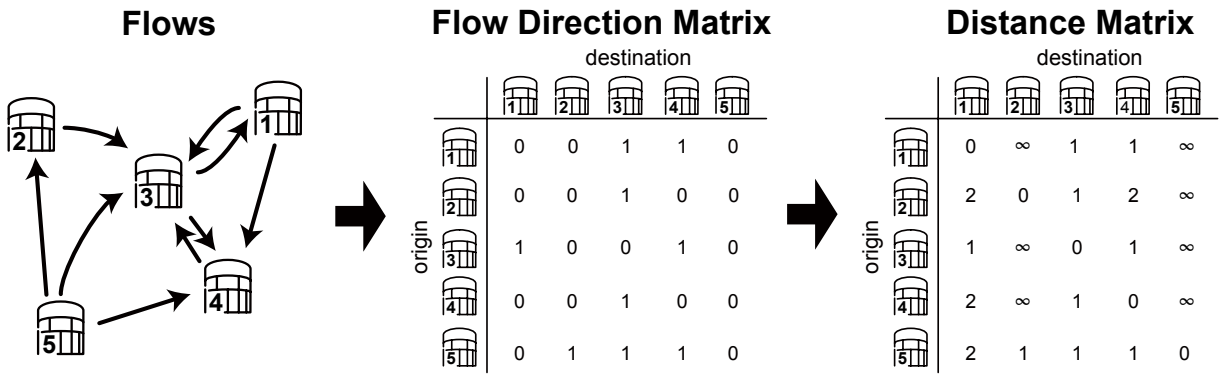

Figure 4.1: The process of converting a flow Graph to a Direction Matrix and subsequently to a Distance Matrix. This illustrates the steps involved in the transformation and the use of algorithms to achieve the final representation.

two vertices with those obtained by traversing through an intermediate vertex. If a shorter path is found, it updates the step count in the matrix accordingly. The process repeats until the matrix converges, meaning no further updates occur. The complete algorithm is detailed in Algorithm 1. The following sections offer a detailed explanation of the key steps involved.

Initialization: *Creating the distance matrix.* Begin by constructing a distance matrix D based on the initial flow direction matrix F. If there is a direct flow from station i to station j, set $D[i][j]$ to the corresponding value from F. For pairs of stations without direct flow, set $D[i][j]$ to infinity (∞) if $i \neq j$, and $D[i][j] = 0$ when $i = j$.

Iterative Update: *Applying updates until convergence.* Continuously update the matrix D through the following steps until the matrix reaches a stable state:

- **Intermediate updates:** For each station k, treat it as an intermediary and apply the update rule:

$$D[i][j] = \min(D[i][j], D[i][k] + D[k][j]) \tag{4.1}$$

- **Path evaluation:** This update checks if the route from station i to station j via station k offers a shorter path than the current known distance $D[i][j]$. If it does, the matrix D is adjusted to reflect this shorter path.

Convergence: *Determining when updates are complete.* Continue the iterative updates until no further changes are observed in the distance matrix D across all pairs of stations, indicating that the matrix has converged.

Completion: *Finalizing the distance matrix.* Once the matrix has stabilized and no more updates occur, the distance matrix D will represent the shortest paths between every pair of stations.

Algorithm 1 The distance matrix algorithm for DMC

Input: Flow graph G with n vertices, representing the number of monitoring stations and flow direction matrix F, where $F[i][j]$ represents the existence of flow from station i to station j
Output: Distance matrix D where $D[i][j]$ represents the step count between station i and station j. In this context, the step count is the number of segments or steps required to travel from station i to station j
Initialize matrix D such that $D[i][j] \leftarrow F[i][j]$ for all i, j, $D[i][j] \leftarrow \infty$ for pairs (i, j) where $D[i][i] = 0$ representing no direct edge between i and j, and set $D[i][i] \leftarrow 0$ for all diagonal entries.
while true **do**
    Create a copy of the matrix: previous_matrix $\leftarrow M$.
    for each vertex k from 1 to n **do**
        for each vertex i from 1 to n **do**
            for each vertex j from 1 to n **do**
                $D[i][j] \leftarrow \min(D[i][j], D[i][k] + D[k][j])$.
            end for
        end for
    end for
    if D is equal to previous_matrix **then**
        break
    end if
end while

4.1.3 Algorithm Complexity

The algorithm of the converter has a time complexity of $O(m \times n^3)$, where n is the number of monitoring stations in the graph, and m represents the maximum number of iterations required for convergence based on the longest step counts $m + 1$ in the output distance matrix. The space complexity remains $O(n^2)$ due to the storage requirements of the distance matrix. This complexity reflects the iterative updates necessary to compute the shortest paths between all pairs of stations. The algorithm is suitable for graphs where the number of stations are not excessively large.

4.2 Graph Recurrent Network for Imputation

In this section, we introduce the base architecture used for HDI that integrates both graph-based and recurrent neural network methodologies. Our goal is to reconstruct missing values in a hydrological data $X_{[t,t+T]}$ using the mask $M_{[t,t+T]}$ by leveraging information from both temporal and spatial dimensions. Referring to GRIN [9], we employs a bidirectional approach with two distinct phases of imputation in both forward and backward directions, culminating in a final refinement stage handled by a feed-forward network. Figure 4.2 provides a visual summary of the model architecture.

4.2.1 Bidirectional Processing Framework

The bidirectional processing framework consists of two primary modules that process the input sequence in opposite temporal directions. Each module mirrors the operations of the unidirectional model, executing two stages of imputation and generating intermediate representations. The final imputation result is achieved by aggregating these representations using a multi-layer perceptron

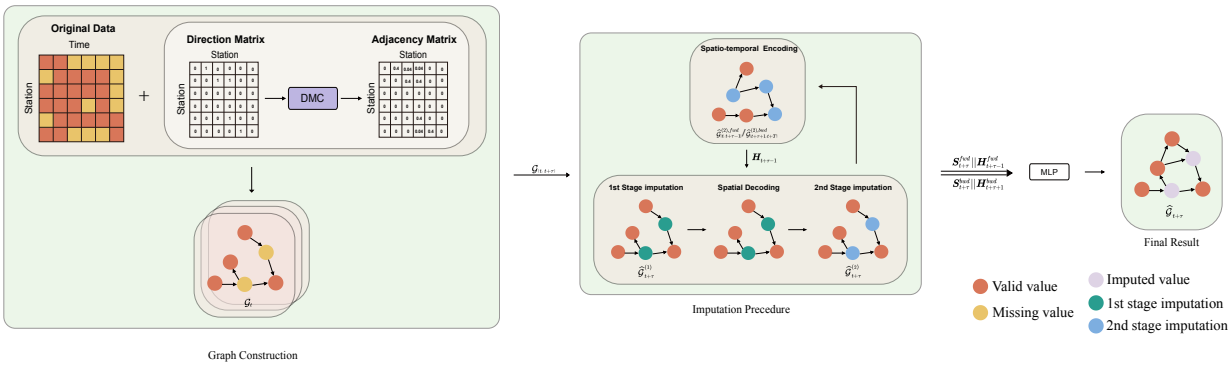

Figure 4.2: An overview of the *Hydro-GRNNI* architecture is presented. For illustration, consider 6 stations with τ steps in time. At each time step t, we construct a graph representing the entire input sequence, incorporating our direction information matrix. The series of graphs is then processed through a bidirectional structure, leading to the final imputation.

(MLP). The output imputation for node i at time t is given by:

$$\hat{y}_t^i = \text{MLP}\left(\begin{bmatrix} s_t^{\text{fwd},i} \\ h_{t-1}^{\text{fwd},i} \\ s_t^{\text{bwd},i} \\ h_{t+1}^{\text{bwd},i} \end{bmatrix}\right), \tag{1}$$

where MLP denotes the final MLP that combines features from both forward and backward modules. The terms $s_t^{\text{fwd/bwd},i}$ and $h_{t-1}^{\text{fwd/bwd},i}$ represent features and hidden states introduced below from the forward or backward module. This bidirectional structure ensures comprehensive utilization of both temporal directions, enhancing the model's ability to capture complex spatio-temporal dependencies.

4.2.2 Spatio-Temporal Feature Extraction

The spatio-temporal feature extractor transforms the input sequence $X_{[t,t+T]}$ into a spatio-temporal representation $H_{[t,t+T]} \in \mathbb{R}^{N_t \times l}$. This is achieved through a recurrent GNN architecture, where the message-passing neural network (MPNN) [20] is pivotal. The encoder's message-passing function at layer k is expressed as:

$$\text{MPNN}_k(v^i_{k-1,t}, W_t) = \phi_k \left(v^i_{k-1,t}, \sum_{j \in N(i)} \rho_k \left(v^i_{k-1,t}, v^j_{k-1,t} \right) \right), \tag{2}$$

Here, $v^i_{k-1,t}$ is the node features vector at layer $k-1$, and $N(i)$ denotes the neighborhood of node i, with ϕ_k and ρ_k being the update and message functions. For the temporal dynamics, gated recurrent units (GRUs) [21] are employed, with message-passing layers defining the GRU gates as follows, where $\circ$ represents Schur product:

$$\text{Reset gate: } r^i_t = \sigma \left(\text{MPNN} \left(\begin{bmatrix} \hat{x}^{(2)i}_t \\ m^i_t \\ h^i_{t-1} \end{bmatrix}, W_t \right) \right), \tag{3}$$

$$\text{Update gate: } u^i_t = \sigma \left(\text{MPNN} \left(\begin{bmatrix} \hat{x}^{(2)i}_t \\ m^i_t \\ h^i_{t-1} \end{bmatrix}, W_t \right) \right), \tag{4}$$

$$\text{Candidate hidden state: } c^i_t = \tanh \left(\text{MPNN} \left(\begin{bmatrix} \hat{x}^{(2)i}_t \\ m^i_t \\ r^i_t \circ h^i_{t-1} \end{bmatrix}, W_t \right) \right), \tag{5}$$

$$\text{Updated hidden state: } h_t^i = u_t^i \circ h_{t-1}^i + (1 - u_t^i) \circ c_t^i. \tag{6}$$

The GRU gates are implemented using message-passing operations to capture the node-level dynamics. The encoded sequence $H_{[t,t+T]}$ is derived by processing each time step and node individually.

4.2.3 Spatial Imputation Process

The spatial imputation process consists of two main stages: the first-stage imputation and the second-stage imputation. Each stage utilizes predictions generated from hidden states and updates the input sequence accordingly.

4.2.3 First-Stage Imputation

The initial step involves generating predictions from the hidden states using a linear transformation. Give that V_h and b_h are learnable parameters, the transformation is given by:

$$Y_t^{(1)} = H_{t-1} V_h + b_h, \tag{7}$$

Next, defining the filler operator here, with $\Psi(Y_t)$ denotes the updated sequence with missing values filled by Y_t, we updates the input sequence X_t by replacing the missing values with the first-stage predictions:

$$\Psi(Y_t) = M_t \circ X_t + (1 - M_t) \circ Y_t, \tag{8}$$

4.2.3 Second-Stage Imputation

Following the first-stage imputation, we first compute the final imputation representation. The imputation representation for each node i is obtained using the function below, where γ and ρ are functions applied to the hidden states and imputed values from neighboring nodes:

$$s_t^i = \gamma \left(h_{t-1}^i, \sum_{j \in N(i)/i} \rho \left(\begin{bmatrix} \Phi(\hat{x}_t^{(1)j}) \\ h_{t-1}^j \\ m_t^j \end{bmatrix} \right) \right). \tag{9}$$

Then, the second-stage imputation is performed using the formula, where $\hat{Y}_t^{(2)}$ and $\hat{X}_t^{(2)}$ represent the second-stage predictions and the final imputed sequence, respectively:

$$\hat{Y}_t^{(2)} = \begin{bmatrix} S_t \\ H_{t-1} \end{bmatrix} V_s + b_s, \tag{10}$$

$$\hat{X}_t^{(2)} = \Phi(\hat{Y}_t^{(2)}). \tag{11}$$

Finally, the imputed sequence $\hat{X}_t^{(2)}$ is used to update the hidden state in the GRU, preparing for the processing of the next input graph G_{t+1}.

CHAPTER 5

DATASET

5.1 Water Quality Data (Discharge)

In this paper, we include the sediment concentration data to analyze the water quality and flow dynamics across the stations within the Maumee River basin.

The daily river flow sediment concentrations dataset, named as *Discharge*, is detailed in Table 5.1. Measured in milligrams per liter (mg/l), this dataset was collected from 20 monitoring stations. The data were provided by three organizations: the United States Geological Survey (USGS), the Water Quality Portal (WQP), and the National Center for Water Quality Research (NCWQR). Covering the period from March 1, 2017, to September 30, 2022, the dataset offers comprehensive insights over these years.

Dataset	**Missing Rate(%)**	**# Stations**	**# Points in Time (Hourly)**
Discharge	0	20	2811

Table 5.1: Details of the adopted dataset.

5.2 Adjacency Information Source

In this paper, we use two types of information to construct the adjacency matrix for the 20 flow stations included in this paper: flow direction information and station location information. These sources provide a comprehensive representation of the spatial and directional relationships between the stations.

5.2.1 Flow Direction Information

Derived from Hydro Network-Linked Data Index (NLDI) dataset, we have also included the Flow Direction Map. The flow direction information is used to represent the directional flow relationships between the stations. This is captured in a binary manner where:

- A value of **1** indicates a direct flow from one station to another.
- A value of **0** indicates no direct flow between the stations.

We construct the distance matrix by calculating the shortest distances between stations using DMC and define edges based on a specified distance threshold. This ensures that stations with closer upstream-downstream relationships are connected in the adjacency matrix. The flow direction information is crucial in *Hydro-GRNNI*, as it helps to understand how water or other flow metrics move through the network of stations.

5.2.2 Station Location Information

In addition to flow direction information, we also incorporate the geographical location information of the stations from USGS in our work. This approach, used in earlier baseline methods, represents each station's location with its latitude and longitude coordinates. We construct the distance matrix by calculating the distances between stations and defining edges based on a specified distance threshold, ensuring that geographically close stations are connected in the adjacency matrix.
To summarize, the adjacency matrix is constructed by either of these two sources of information:

- The *flow direction information* provides the directional information.
- The *station location information* provides the spatial proximity information.

By incorporating directional information in our new method and geographical information in the old method, we ensure a more comprehensive and accurate representation of the relationships between the flow stations, which is essential for imputation tasks.

CHAPTER 6
EXPERIMENTS

6.1 Data Preparation

The water quality dataset is divided into training, validation, and test sets to ensure robust evaluation. To assess performance, we utilize several metrics: mean absolute error (MAE), root mean square error (RMSE), and mean relative error (MRE) [16], all calculated over the imputation window. Our analysis encompasses various baseline imputation methods, comparing their effectiveness in both in-sample and out-of-sample settings.

6.1.1 Data Masking Strategy

In our experiments, we employ both point and block missing strategies to simulate realistic sensor failures and data dropouts. Point missing involves the random dropout of individual data points, controlled by a dropout probability of $p = 5\%$. This approach simulates sporadic data loss due to transient sensor issues or communication errors. In contrast, block missing creates contiguous sequences of missing values, reflecting more prolonged sensor failures or systematic data transmission interruptions, and is managed with a dropout probability of $p = 0.15\%$. The block missing patterns are constrained to a minimum length of 12 time steps and a maximum length of 48 time steps. Additionally, the blocking masking function incorporates a noise probability of $p_{\text{noise}} = 5\%$ to introduce extra variability and randomness, further mimicking real-world conditions. By utilizing both point and block masking, we provide a comprehensive simulation of data loss and sensor failures, enabling us to assess the robustness of our method under these challenging scenarios.

6.2 Baseline Imputation Methods

Over the years, various methods have been developed to address the problem of missing data in multivariate time series. These methods range from simple statistical techniques to sophisticated machine learning algorithms and advanced neural network architectures. In our experiment, we incorporate multiple imputation methods to serve as baselines for our evaluations, enabling us to assess the performance of our method.

Our selection of baseline methods spans from traditional statistical approaches to cutting-edge neural architectures, each contributing unique strengths to the task of imputation. The MEAN imputation method provides a straightforward baseline by imputing missing values with the node-level average of observed values. This approach offers simplicity and ease of implementation. Complementing this, the K-Nearest Neighbors (KNN) Imputation method leverages the $k = 10$ neighboring nodes with the highest weight in the adjacency matrix W_t, utilizing similarity to average values and effectively fill in missing data.

Moving to more dynamic approaches, the Vector Autoregressive (VAR) Imputation method employs a VAR one-step-ahead predictor to capture linear inter-dependencies among multiple time series. This statistical model is crucial for scenarios where temporal relationships are pivotal. On the machine learning front, Recurrent Generative Adversarial Imputation Networks (rGAIN) [9] stands out by integrating generative adversarial networks with a bidirectional recurrent encoder and decoder, enhancing the modeling of data distribution for effective imputation.

The Bidirectional Recurrent Imputation for Time Series (BRITS) [16] further enriches our baseline suite by employing recurrent neural networks for imputation. It treats imputed values as RNN graph variables, allowing seamless updates during backpropagation without specific assumptions. Similarly, the Message-Passing GRU (MPGRU) method, inspired by DCRNN [22], utilizes Graph

Neural Networks with Gated Recurrent Units to adeptly capture both spatial and temporal dependencies.

Finally, the Graph Recurrent Imputation Network (GRIN) serves as a main baseline in our experiment, designed to reconstruct missing data across different channels of a multivariate time series. By learning spatio-temporal representations through message passing in a graph, GRIN [9] adeptly captures complex interactions. However, its reliance on geographical information highlights the need for more specialized hydrological data imputation methods, which our proposed approach aims to address.

6.3 Experiment Results

In this section, we present the results of our imputation experiments using the discharge dataset. The results are analyzed in two scenarios: the in-sample setting, detailed in Table 6.2, and the out-of-sample setting, shown in Table 6.1. These tables illustrate the performance of various methods. And the adjacency matrix derived from the flow direction information is visualized in 6.1 Notably, our proposed method, *Hydro-GRNNI*, exhibits substantial improvements over the baseline methods in both settings.

		Point Missing			Block Missing		
D	M	MAE	RMSE	MRE(%)	MAE	RMSE	MRE(%)
Discharge	Mean	998.34	2469.02	1.12	1264.21	3781.88	0.93
	KNN	1013.43	2732.56	1.14	1540.31	4369.41	1.13
	VAR	746.56	2596.45	0.57	597.58	1977.11	0.46
	rGAIN	607.34	2088.86	0.47	578.59	2126.00	0.44
	BRITS	505.68	1646.39	0.39	494.53	1627.71	0.38
	MPGRU	562.66	2339.57	0.43	625.12	2381.28	0.48
	GRIN	384.11	1639.11	0.29	504.77	1753.84	0.39
	Hydro-GRNNI	**246.01**	**910.06**	**0.19**	**327.84**	**1185.10**	**0.25**

Table 6.1: Performance results of imputation methods on Discharge dataset for out-of-sample evaluation

		Point Missing			Block Missing		
D	M	MAE	RMSE	MRE(%)	MAE	RMSE	MRE(%)
Discharge	Mean	967.04	2454.02	1.08	1235.57	3799.04	0.91
	KNN	1013.43	2732.56	1.14	1540.31	4369.41	1.13
	VAR	244.71	1040.65	0.18	257.19	1069.80	0.19
	rGAIN	171.43	576.50	0.13	199.94	687.44	0.15
	BRITS	85.86	236.36	0.07	113.02	322.21	0.08
	MPGRU	225.42	1160.06	0.17	273.73	1237.21	0.21
	GRIN	99.16	566.98	0.07	135.90	417.89	0.10
	Hydro-GRNNI	**79.46**	**273.01**	**0.06**	**110.13**	**335.47**	**0.08**

Table 6.2: Performance results of imputation methods on Discharge dataset for in-sample evaluation

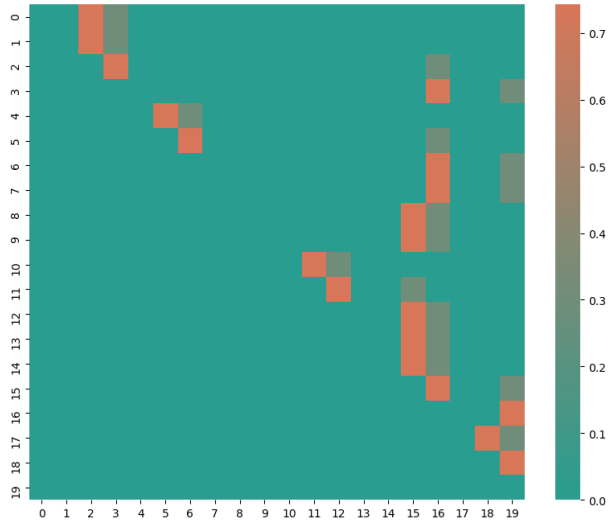

Figure 6.1: Visualization of adjacency matrix

6.4 Sensitivity Analysis

Additionally, we performed a sensitivity analysis as part of our experiments. In this analysis, we evaluated the performance of our method, *Hydro-GRNNI*, against GRIN—an established baseline method utilizing graph neural networks (GNNs)—in a challenging out-of-sample setting. We

evaluated performance across different expected masking ratios for both point missing and block missing setting. The results are presented in Figure 6.2 for the point missing setting and Figure 6.3 for the block missing setting.

In our sensitivity analysis, we control the masking ratio by calculating the expected masking ratio, which considers the effects of different masking strategies. For point masking, the expected masking ratio is simply:

$$\text{Expected Masking Ratio} = p_b$$

For block masking, the expected masking ratio accounts for both block and noise effects and is given by:

$$\text{Expected Masking Ratio} = p_b \cdot E[L_b] + p_n$$

where p_b is the dropout probability, $E[L_b]$ is the expected block length, and p_n is the noise probability. This formulation helps us assess how various masking approaches influence the imputation process.

Our analysis reveals that *Flow-GRNNI* consistently outperforms GRIN across different expected masking ratios. *Flow-GRNNI* demonstrates better stability and effectiveness, maintaining lower performance metric values compared to GRIN. While GRIN's performance deteriorates as the masking ratio increases, *Flow-GRNNI* shows improved and more consistent results. This highlights *Flow-GRNNI*'s robustness and suitability for scenarios involving varying levels of data masking, making it a more reliable choice.

Masking Ratio	10	20	30	40	50	60	70	80
Flow-GRNNI	**224.96**	**301.64**	**348.23**	**366.86**	**423.98**	**469.42**	**557.95**	**680.13**
GRIN	339.31	471.05	533.52	540.32	595.00	620.45	732.57	827.36

Table 6.3: Sensitivity analysis comparing Flow-GRNNI with the baseline GRIN under various masking ratios for point missing scenarios.

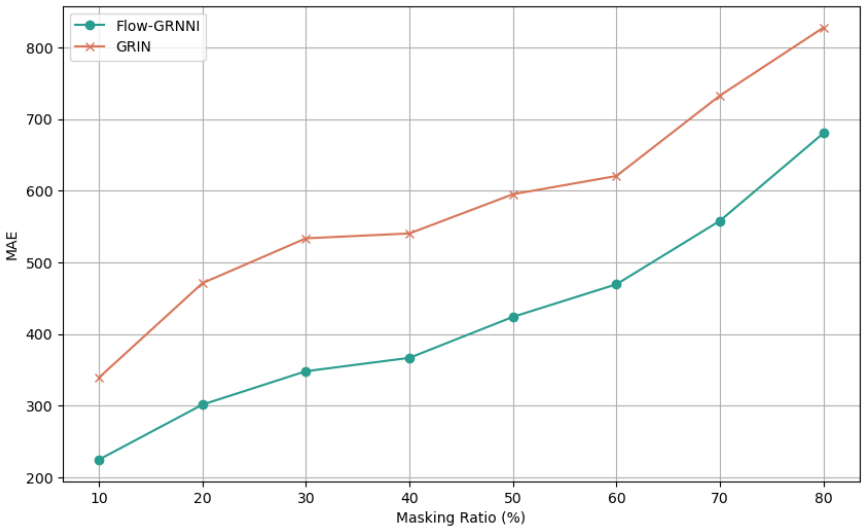

Figure 6.2: Graphical representation of the sensitivity analysis results shown in Table 6.3

Masking Ratio	10	20	30	40	50	60	70	80
Flow-GRNNI	**424.19**	**335.34**	**382.87**	**405.39**	**403.31**	**417.25**	**471.92**	**502.00**
GRIN	711.67	635.49	651.99	584.66	557.74	536.84	567.71	599.74

Table 6.4: Sensitivity analysis comparing Flow-GRNNI with the baseline GRIN under various masking ratios for block missing scenarios.

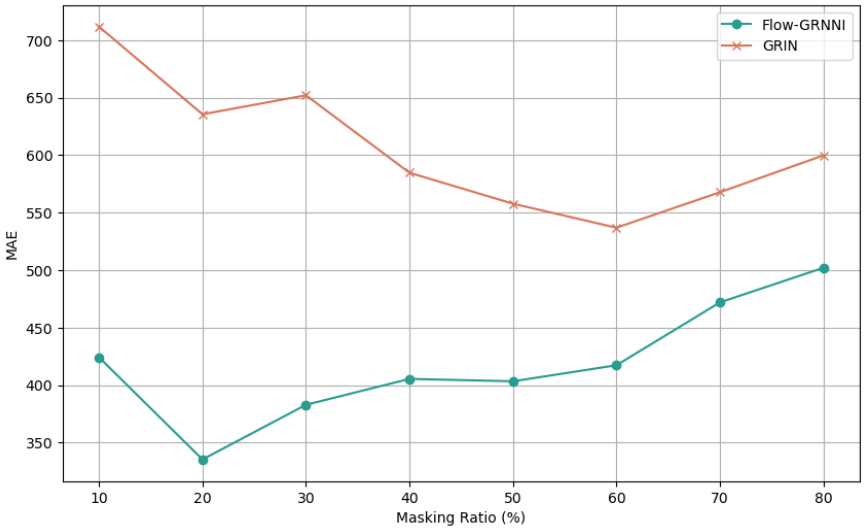

Figure 6.3: Graphical representation of the sensitivity analysis results shown in Table 6.4

CHAPTER 7
CONCLUSION AND FUTURE WORK

7.1 Summary of Key Findings and Significance

The results demonstrate that the proposed *Hydro-GRNNI* method significantly outperforms traditional techniques for predicting flow sediment concentrations in the Maumee River Basin. Specifically, *Hydro-GRNNI* achieved the lowest mean absolute error (MAE) in the out-of-sample setting, with values of 246.01 for point missing and 327.84 for block missing, and in the in-sample setting, with values of 79.46 for point missing and 110.13 for block missing. This superior performance underscores the method's enhanced predictive accuracy, stability, and reliability compared to other models such as KNN, rGAIN, and GRIN. The notable improvement highlights the effectiveness of incorporating flow direction information with graph neural networks (GNN), making *Hydro-GRNNI* a valuable tool for real-time hydrological data management.

7.2 Limitations

Despite the promising results, the study has certain limitations. The GNN model's performance may be sensitive to the quality and availability of input data, particularly with regard to spatial and temporal variations. Furthermore, the approach's scalability to larger river basins or regions with more complex hydrological interactions remains to be thoroughly evaluated. Additional validation in diverse hydrological contexts is necessary to ascertain the robustness of the model.

7.3 Opportunities for Future Research

Future research could explore the application of the GNN framework across various watersheds with differing hydrological characteristics, enabling a broader understanding of its applicability. Additionally, investigating the integration of more sophisticated machine learning techniques, such as ensemble methods, could further enhance prediction accuracy. Further studies could also examine the potential for real-time monitoring and adaptive modeling, utilizing sensor data to improve the responsiveness of water quality assessments in the face of changing environmental conditions.

REFERENCES

[1] R. Hirsch, L. DeCicco, and J. Murphy, *Exploration and graphics for river trends (egret)*, U.S. Geological Survey, 2023.

[2] J. Dickinson, R. Hanson, and S. Predmore, *Hydroclimate—hydrologic and climatic analysis toolkit*, 4–A9, Techniques and Methods, 2014, p. 49.

[3] R. L. Runkel, C. G. Crawford, and T. A. Cohn, *Load Estimator (LOADEST): A FORTRAN Program for Estimating Constituent Loads in Streams and Rivers*. U.S. Geological Survey Techniques and Methods Book 4, Chapter A5, 2004, 69 p.

[4] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, "Recurrent neural networks for multivariate time series with missing values," *Scientific Reports*, vol. 8, no. 1, pp. 1–12, 2018.

[5] J. Yoon, J. Jordon, and M. Schaar, "GAIN: Missing data imputation using generative adversarial nets," in *Proceedings of the 35th International Conference on Machine Learning*, ser. ICML, PMLR, 2018, pp. 5689–5698.

[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[7] S. R. Kuppannagari, Y. Fu, C. M. Chueng, and V. K. Prasanna, "Spatio-temporal missing data imputation for smart power grids," in *Proceedings of the Twelfth ACM International Conference on Future Energy Systems (e-Energy '21)*, New York, NY, USA: Association for Computing Machinery, 2021, pp. 458–465, ISBN: 9781450383332.

[8] Y. Liu, R. Yu, S. Zheng, E. Zhan, and Y. Yue, "Naomi: Non-autoregressive multiresolution sequence imputation," in *Advances in Neural Information Processing Systems*, vol. 32, 2019, pp. 11 238–11 248.

[9] A. Cini, I. Marisca, and C. Alippi, "Filling the g_ap_s: Multivariate time series imputation by graph neural networks," in *International Conference on Learning Representations*, 2022.

[10] X. Miao, Y. Wu, J. Wang, Y. Gao, X. Mao, and J. Yini, "Generative semi-supervised learning for multivariate time series imputation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, 2021, pp. 8983–8991.

[11] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE transactions on neural networks*, vol. 20, no. 1, pp. 61–80, 2008.

[12] P. W. Battaglia, J. B. Hamrick, V. Bapst, *et al.*, "Relational inductive biases, deep learning, and graph networks," *arXiv preprint arXiv:1806.01261*, 2018.

[13] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: Going beyond euclidean data," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, 2017.

[14] L. DeCicco, S. Prinos, P. Eslick-Huff, C. Hopkins, and T. Root, *Hasp: Hydrologic analysis package*, version 1.0.0, Reston, VA: U.S. Geological Survey, 2022.

[15] M. J. Sleckman, E. D. Hinman, S. D. Hamshaw, and L. Stanish, *Surface-water-geospatial-data-assembly*, Water Resources Mission Area - Headquarters, 2024.

[16] W. Cao, D. Wang, J. Li, H. Zhou, Y. Li, and L. Li, "Brits: Bidirectional recurrent imputation for time series," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2018, pp. 6776–6786.

[17] M. L. Stein, *Interpolation of Spatial Data: Some Theory for Kriging*. Springer Science & Business Media, 1999.

[18] D. B. Rubin, "Inference and missing data," *Biometrika*, vol. 63, no. 3, pp. 581–592, 1976.

[19] R. W. Floyd, "Algorithm 97: Shortest path," *Communications of the ACM*, vol. 5, no. 6, p. 345, Jun. 1962.

[20] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proceedings of the International Conference on Machine Learning (ICML)*, PMLR, 2017, pp. 1263–1272.

[21] K. Cho, B. Van Merrienboer, C. Gulcehre, *et al.*, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint*, 2014. arXiv: 1406.1078.

[22] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *International Conference on Learning Representations*, 2018.