Lifting Scheme for Streamflow Data in River Networks

Seoncheol Park
Pacific Climate Impacts Consortium
University of Victoria
Victoria, BC V8W 2Y2, Canada
seoncheolpark@uvic.ca

Hee-Seok Oh
Department of Statistics
Seoul National University
Seoul 08826, Korea
heeseok@stats.snu.ac.kr

Draft: version of July 6, 2021

Abstract

This paper presents a new multiscale method for analyzing water pollutant data located in river networks. The main idea of the proposed method is to adapt the conventional lifting scheme, reflecting the characteristics of streamflow data in the river network domain. Due to the complexity of the data domain structure, it is difficult to apply the lifting scheme to the streamflow data directly. To solve this problem, we propose a new lifting scheme algorithm for streamflow data that incorporates flow-adaptive neighborhood selection, flow proportional weight generation, and flow-length adaptive removal point selection. A nondecimated version of the proposed lifting scheme is also provided. The simulation study demonstrates that the proposed method successfully performs a multiscale analysis of streamflow data. Furthermore, we provide a real data analysis of water pollutant data observed on the Geum-River basin compared to the existing smoothing method.

Keywords: Lifting scheme; River network; Smoothing; Spatial adaptation; Spatial modeling; Streamflow data.

1 Introduction

Environmental monitoring is a collection of observations and studies for the evaluation of environmental data (Artiola *et al.*, 2004). Humans now know that the environment is crucial to our health and survival. So we cannot overemphasize environmental monitoring for humans. One of the main areas of environmental monitoring is water quality management. As human activities increase, more environmental costs are needed to rehabilitate water. Therefore, it is important to analyze the characteristics of water pollutants.

This paper focuses on the environmental pollutant called Total Organic Carbon (TOC, mg/L). Recently, the Korean Ministry of Environment announced that they changed the water pollution index for monitoring wastewater treatment performance of facilities from chemical oxygen demand (COD) to TOC. Both COD and TOC are the indirect representations of organic matter. COD is widely used in wastewater monitoring but produces hazardous wastes, including mercury and hexavalent chromium (Dubber and Gray, 2010). Therefore, analyzing TOC data is meaningful to society. The National Institute of Environmental Research (NIER) under the Ministry of Environment operates a Water Environment Information System to monitor water quality. This system provides an "Environment standard", which is a guideline for the amount of TOC, listed in Table 1.

Table 1: Environment standard for TOC provided by Water Environment Information System.

Status	Very good	Good	Slightly better	Normal	Poor	Bad	Very bad
TOC (mg/L)	≤ 2	≤ 3	≤ 4	≤ 5	≤ 6	≤ 8	> 8

Figure 1 shows the Geum-River basin in the heart of South Korea, which is divided into 14 sub-regions called catchments, and TOC data observed in the basin. The catchments are marked by solid lines. In the right panel of Figure 1, the gray lines represent streamflow segments with weights of different widths, and colored points at 127 observational locations over the 14 catchments denote logarithm values of TOC means from December 2011 to November 2017. Detailed information on the TOC data in the Geum-River basin is described in Section 3.

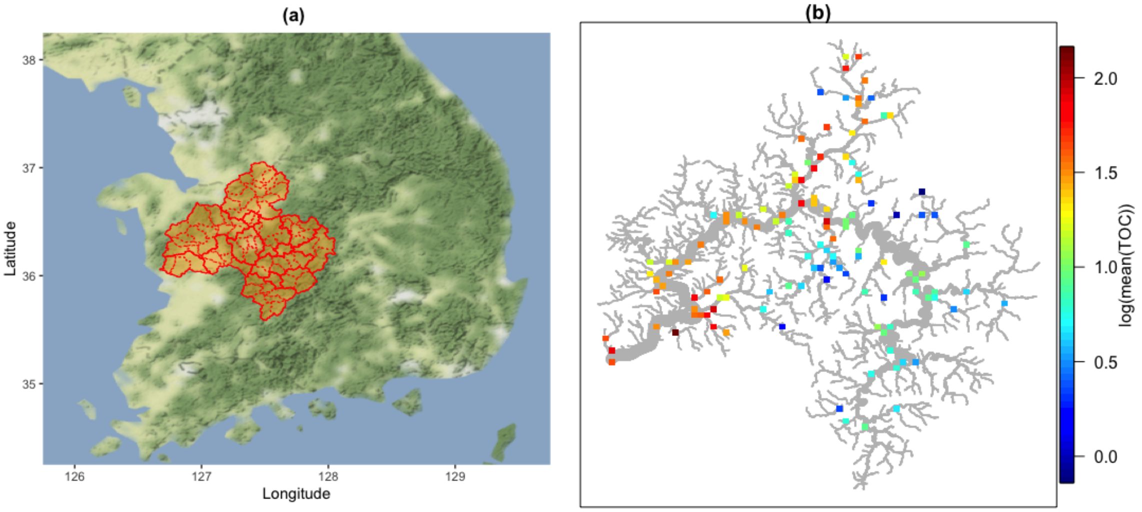

Figure 1: (a) The Geum-River basin and (b) TOC data observed in the basin.

From Figure 1, we observe some characteristics of the water quality index: (i) The TOC index is located in the river network. It means that the TOC data are observed in the river network, not the usual $\mathbb{R}^2$ domain. Most statistical models are interested in analyzing a spatial region, a subset of $\mathbb{R}^2$, where Euclidean distance works well. On the other hand, for the streamflow data in Figure 1, Euclidean distance does not work well as a natural metric. (ii) As shown in Figure 1, the data have spatially inhomogeneous features in various dependent structures along with the river network. (iii) The data are irregularly observed in the river network. Thus, classical methods, including smoothing splines and wavelet-based methods, cannot efficiently represent such water quality index data.

Therefore, an ideal method of analyzing the above data should have the following features: (i) It is capable of effectively representing streamflow data in the river network. (ii) It provides a spatially adaptive framework to estimate the inhomogeneous underlying function by reflecting the inherent multiscale characteristics of data. (iii) It applies to scattered data in the river network. In this paper, we would like to propose a multiscale approach that satisfies all of the features mentioned above.

Suppose that a set of scattered data (x_i, y_i), $i = 1, \ldots, n$ is observed from the following model

$$y_i = g(x_i) + \varepsilon_i, \tag{1}$$

where x_i denote the locations of observations in the data domain, ε_i represent error terms that are assumed to be independent and identically distributed (i.i.d) random variables with finite variance, and g denotes an unknown underlying function of interest. Our goal is to estimate the underlying field $g(x)$ for every location x on the river network. That is, we want to represent the underlying field of the water quality index in the river network domain, as shown in Figure 1.

In the literature, there exist some studies of streamflow data analysis. Ver Hoef *et al.* (2006) proposed the use of stream distance defined by the shortest distance between two locations on river networks as a reasonable distance measure for data analysis on the river network. They showed that it could construct a large class of valid spatial autocovariance models using the stream distance. They also suggested a kernel convolution-based method to generate a class of covariance models for streamflow data. O'Donnell *et al.* (2014) used non-parametric flexible regression approaches, such as kernel methods and penalized splines, to build spatio-temporal models in river networks. They proposed a piecewise simple regression analysis by dividing the network into a large number of small pieces called *stream segments.* They provided regression-based estimates assuming that the function values g's are constant within the same stream segments.

Meanwhile, due to the complexity of the streamflow data, it is not easy to fully understand the underlying structure of the data. A multiscale analysis is a possible way to solve such a problem by analyzing the data on multiple scales. As a conventional multiscale method, wavelets are the most popular choice. However, wavelets do not properly work when the data are not observed on regular grids, or the number of observations is not dyadic, i.e., $n = 2^J$, for some $J \in \mathbb{Z}$. To overcome these problems, Sweldens (1996, 1998) proposed a kind of second-generation wavelet called *lifting scheme.* The lifting scheme has been extensively studied in signal processing and image analysis (Jansen and Oonincx, 2005).

However, there is a limit that all of the previous works cannot provide a multiscale structure for streamflow data. As far as we know, there is no direct literature describing multiscale methods for streamflow data. In this paper, a new lifting method for streamflow data is proposed by combining the conventional lifting method and novel modifications of neighborhood selection, filter prediction, and removal point selection, taking into account the characteristics of the data. The proposed method has two advantages. First, by following the argument of the lifting scheme, it gives a multiscale structure of streamflow data. Second, the proposed method is advantageous compared to the conventional smoothing methods for river networks from the signal denoising point of view.

The rest of the paper is organized as follows. Section 2 reviews the existing lifting schemes and smoothing method in the river network. Section 3 describes the streamflow data used in this study. Section 4 presents a new method termed the *streamflow lifting scheme*. Simulation studies and real data analysis are conducted in Sections 5 and 6 to evaluate the proposed method. Finally, concluding remarks are provided in Section 7.

2 Background

2.1 Lifting scheme

We briefly summarize the lifting scheme of Sweldens (1996, 1998) for the self-contained material. Suppose that we observe a set of n irregular locations $\boldsymbol{x} = (x_1, \ldots, x_n)^T$ and have function values $y_1, \ldots, y_n$ at every location, where n may not be dyadic. Given the jth level data $\boldsymbol{y}_j$, the lifting scheme at the $j-1$th level consists of the following four steps: (i) Split $\boldsymbol{y}_j$ into two subsets, $\mathcal{I}_{j-1}$ and $\mathcal{I}_{j-1}^c$ at level $j-1$. (ii) Predict $y_{j,i} \in \mathcal{I}_{j-1}^c$ from $y_{j,k} \in \mathcal{I}_{j-1}$ with a prediction filter $\mathbf{p}_{j-1,i}$, and store the error $d_{j-1,i} := y_{j,i} - \hat{y}_{j,i} = y_{j,i} - \sum_{k \in \mathcal{N}_{j-1,i} \cap \mathcal{I}_{j-1}} p_{j-1,i,k} y_{j,k}$, where $\mathcal{N}_{j-1,i}$ is the set of neighbors of node i, and $\hat{y}_{j,i}$ represents the predicted value constructed from $\mathcal{I}_{j-1}$ neighbors of node i. Note that i and k denote the location in $\mathcal{I}_{j-1}^c$ and $\mathcal{I}_{j-1}$, respectively. (iii) Update the $j-1$th level data $y_{j-1,k}$ in $\mathcal{I}_{j-1}$ with a filter $\mathbf{u}_{j-1,k}$, i.e., $y_{j-1,k} := y_{j,k} + \sum_{i \in \mathcal{N}_{j-1,k} \cap \mathcal{I}_{j-1}^c} u_{j-1,k,i} d_{j-1,i}$, to preserve important statistics of the original data such as mean or median (Nunes *et al.*, 2006). (iv) Repeat the above steps until the desired resolution level is achieved.

By performing these steps, we construct coarse signals of data from updated subsamples. Meanwhile, the reverse version of the lifting scheme can be easily obtained by undoing the forward scheme operations at the level $j-1$: (i) Undo update: $y_{j,k} = y_{j-1,k} - \sum_{i \in \mathcal{N}_{j-1,k} \cap \mathcal{I}_{j-1}^c} u_{j-1,k,i} d_{j-1,i}$. (ii) Undo predict: $y_{j,i} = d_{j-1,i} + \sum_{k \in \mathcal{N}_{j-1,i} \cap \mathcal{I}_{j-1}} p_{j-1,i,k} y_{j,k}$. (iii) Undo split. (iv) Repeat the above steps at the next level.

There are several crucial components to choose from in the construction of the lifting scheme, such as the number of points remaining at the next (coarser) level, prediction filter, removal order of points, and neighborhood. For more information, refer to Jansen and Ooninckx (2005).

The lifting one coefficient at a time (LOCAAT) algorithm of Jansen *et al.* (2009) constructs a removal order of data points and sequentially decomposes the data with the order. Suppose that we have values $y_1, \ldots, y_n$ at n irregularly spaced points $x_1, \ldots, x_n$ on the real line. The lifting scheme approximates the function g in (1) as $\tilde{g}(y) = \sum_{k=1}^{n} c_{n,k}\phi_{n,k}(x)$, where $c_{n,i} := g(x_i)$, $\phi_{n,k}(x_i) = \delta_{i,k}$ for $k, i \in \{1, \ldots, n\}$, and $\delta_{i,k}$ denotes the Kronecker delta.

The LOCAAT algorithm first defines the index set of the scaling coefficients as $\mathcal{I}_n = \{1, \ldots, n\}$ and the index set of wavelet coefficients as $\mathcal{I}_n^c = \emptyset$. At the next step $n-1$, a point to be lifted is selected and denoted as j_n, which is the point to be removed from the current set of scaling coefficients and to be converted into a detailed coefficient. The new set of indices corresponding to the scaling coefficients is $\mathcal{I}_{n-1} = \mathcal{I}_n \setminus \mathcal{I}_{n-1}^c$, while $\mathcal{I}_{n-1}^c = \{j_n\}$ is the index set of the wavelet coefficients constructed at this stage. To select the point to be lifted, Jansen *et al.* (2009) used the minimum of the integral of scaling function $\phi_{n,k}$, $I_{n,k}$, as a measure. For the configuration of update filters, Jansen *et al.* (2009) proposed a minimum norm solution-based update weights at level r due to the numerical stability,

$$u_{r,j,i_r} = I_{r,i_r} I_{r-1,j} / \sum_{k \in \mathcal{N}_r} I_{r-1,k}^2, \tag{2}$$

where i_r is an index of the candidate points for removal.

Nunes *et al.* (2006) proposed a lifting scheme called *adaptive lifting.* The key ingredients of the adaptive lifting are the data-adaptive selection of the removal order and the neighborhood size in the prediction step. They flexibly constructed prediction filters in the one-dimensional signal denoising setting. To enhance the performance of the lifting scheme in nonparametric regression settings, Knight and Nason (2009) proposed a “nondecimated” concept in the lifting scheme. It borrows the idea from a nondecimated wavelet transform that uses over-complete basis functions to improve the performance of the wavelet transform. Knight and Nason (2009) generated several removal order sequences called paths or trajectories by permutation.

Before closing this section, we remark that the conventional lifting scheme is limited to analyzing the TOC data in Figure 1 observed in a river network, which is not a subset of $\mathbb{R}^2$ domain. Thus, it is necessary to develop a new lifting scheme that takes into account all the important features of streamflow data.

2.2 Shrinkage by lifting scheme

Lifting schemes have also been applied to nonparametric regression problems by incorporating a shrinkage approach. The main idea of shrinkage is based on the assumption that the true signal information is contained only in large values of the elements. Thus, by setting the coefficient less than a specific threshold to zero, the reconstruction results may be more similar to the true signal. As previous studies that are closely related to our analysis, Nunes *et al.* (2006) applied existing shrinkage rules to their adaptive lifting scheme for denoising signals, and Knight and Nason (2009) proposed a shrinkage estimator using the following steps: (i) generate P estimates $\hat{g}^{(p)}(x)$ $(p = 1, \ldots, P)$ by combining their nondecimated lifting transform and classical shrinkage techniques, and (ii) compute an averages estimator $\hat{\bar{g}}(x) = \sum_{p=1}^{P} \hat{g}(x)/P$. For details, refer to Nunes *et al.* (2006) and Knight and Nason (2009).

In the proposed streamflow lifting scheme to be discussed in Section 4, we use the same shrinkage strategies used in Nunes *et al.* (2006) and Knight and Nason (2009). In this paper, we use `EbayesThresh` with median and hard threshold rules, which are implemented by `median` and `hard` in R packages `adlift` and `nlt`. To use the lifting scheme, one must decide the number of scaling coefficients to be kept in the final representation of the initial signal. The user also specifies `nkeep` in `adlift` and `nlt`. In this paper, we use the fully decomposed result (`nkeep=2`) in Knight and Nason (2009), which produces $(n-2)$ detail coefficients in the length-n dataset.

2.3 Smoothing method on river networks

In this section, we briefly summarize the approach of O'Donnell *et al.* (2014). O'Donnell *et al.* (2014) simplified the information in a given network using the concept of stream segments and suggested a penalized spline-based method with spatial, seasonal, temporal, and interaction bases. The current study focuses on the analysis of the spatial behavior of pollutants, taking into account the structure of river networks. For this purpose, we consider a straightforward spatial additive model as

$$y_i = \mu + m_x(x_i) + \varepsilon_i = g(x_i) + \varepsilon_i, \tag{3}$$

where m_x describes spatial trends. The spline method uses a set of basis functions to estimate g in (1). So, with p basis functions, the estimator is expressed as $\hat{g}(x) = \sum_{j=1}^{p} \beta_j \phi_j(x)$. O'Donnell *et al.* (2014) used a B-spline model which is formulated as $\mathbf{y} = B\boldsymbol{\beta} + \boldsymbol{\varepsilon}$, where $B = (1, B_s)$, B_s is a design matrix of spatial components, and $\boldsymbol{\beta}$ is an $n \times p$ response vector. The model is fitted by minimizing the following penalized sum of squares

$$(\mathbf{y} - B\boldsymbol{\beta})^T(\mathbf{y} - B\boldsymbol{\beta}) + \lambda\boldsymbol{\beta}^T D^T D\boldsymbol{\beta}, \tag{4}$$

where D denotes the penalty matrix. The solution of (4) is $\hat{\boldsymbol{\beta}} = (B^T B + \lambda D^T D)^{-1} B^T \mathbf{y}$, where λ is a smoothing parameter. For the optimal value of λ, O'Donnell *et al.* (2014) selected λ to minimize $\log(\hat{\sigma}^2) + 1 + \frac{2+2\mathrm{df}}{n-\mathrm{df}-2}$, where df denotes the degree of freedom. For detailed information of smoothing of the river network, refer to O'Donnell *et al.* (2014).

3 Geum-River TOC data

According to the Water Environment Information System operated by the Ministry of Environment, the Geum-River basin is divided into 14 sub-regions, called catchments, which are plotted with solid lines in Figure 2(a) that is an enlarged map of Figure 1. All 14 catchments are also divided into several sub-catchments, which are plotted with dotted lines. Among them, the Miho-Cheon catchment marked by green in Figure 2(a) is one of the sub-regions. It contains many observational stations compared to other catchments, and there are several cities and factories around it. We believe it is meaningful to take a closer look at the area. Note that this river network is used to build a network model for simulation studies in Section 5.

The colored lines in the Geum-River catchment of Figure 2(a) represent stream segments defined by lines between junctions of the river network (Ver Hoef *et al.*, 2006, 2010). We note that there are 113 stream segments and 28 observation stations in the Miho-Cheon catchment. The Geum-River network has a total of 942 stream segments and 127 observation points.

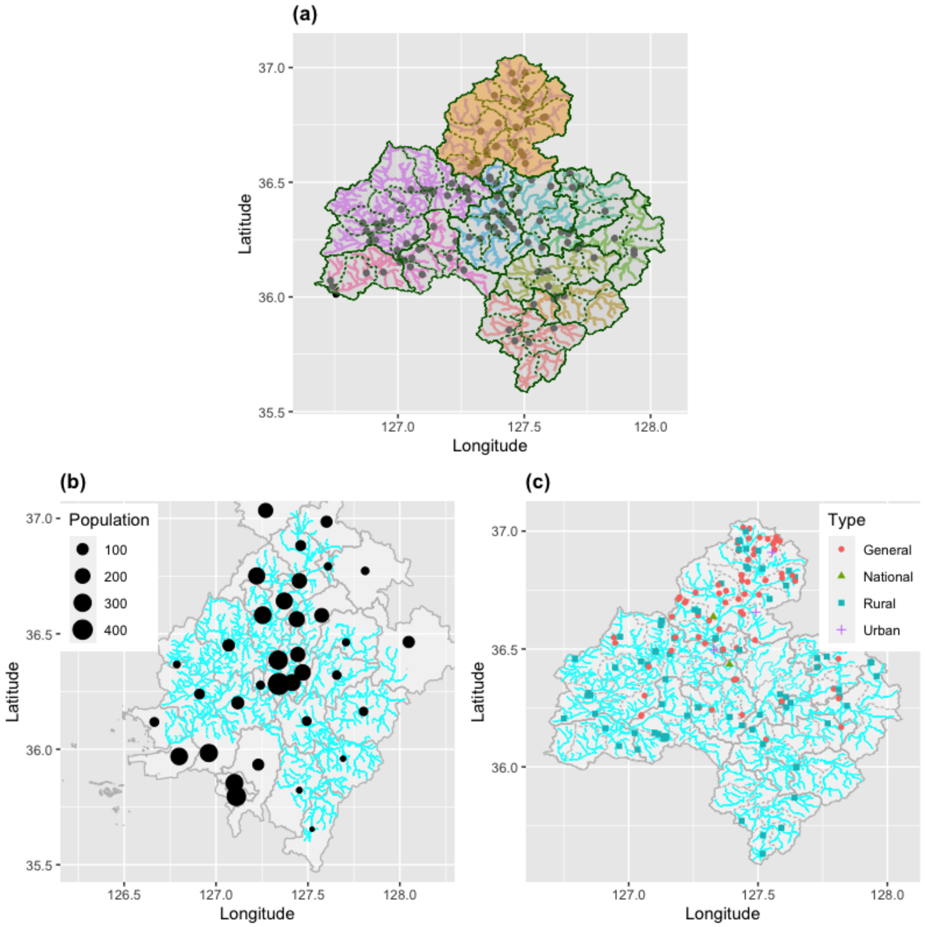

Figure 2: (a) An enlarged map of Figure 1(a). Black dots are 127 observation points. (b) Populations (31/12/2017, thousands) in the Geum-River basin. (c) Locations of industrial areas in the Geum-River basin.

Figure 2(b) shows the cities, counties, and districts populations located in the Geum-River basin. Note that these administrative areas do not fully match the Geum-River catchments. From Figure 2(b), we observe that most of the populations are concentrated in the Northern and Central parts of the Geum-River basin. Figure 2(c) shows the locations of industrial areas in Geum-River basins. Note that general, national, and urban industrial sites are clustered in the Miho-Cheon and its nearby areas. Therefore, it is possible to assume that many water pollutants occur in the Miho-Cheon and its adjacent river basin.

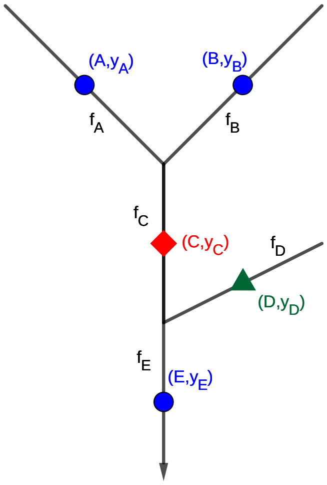

Figure 3: A simple example of streamflow data. Five solid black lines represent stream segments, indexed by A, B, C, D, and E. Each line segment has its flow volume, called $f_A, f_B, \ldots, f_E$. $y_A, y_B, \ldots, y_E$ denote water quality values of each segment. Suppose that red diamond point C is the removal point at a specific level. Then blue circle points A, B, and E are its neighbors, and green triangle point D is a neighbor of point C since points C and D are not flow-connected.

4 Streamflow lifting scheme

This section presents a new lifting scheme for streamflow data by modifying the LOCAAT algorithm of Jansen *et al.* (2009) to adapt some characteristics of streamflow data. Our main idea is to develop a multiscale method for streamflow data analysis by incorporating the idea of Nunes *et al.* (2006) into O'Donnell *et al.* (2014). The necessary modifications for developing the streamflow lifting scheme are as follows: (i) performing a network-adaptive neighborhood selection, (ii) constructing a prediction filter with flow-adaptive weighted averages, and (iii) determining a removal order by defining a proper contribution measure of each observation point to the river network.

We consider a toy network shown in Figure 3. Suppose that there are five observation points (A, B, C, D, E) in the different stream segments of a river network. Assume that each segment has a flow volume of f. Let $f_A, f_B, \ldots, f_E$ denote the flow volume of the station

$A, B, \dots, E$, respectively. We further denote $y_A, y_B, \dots, y_E$ as water quality observations at station $A, B, \dots, E$, respectively.

4.1 Neighborhood selection

The concept of "flow-connected" introduced in Ver Hoef *et al.* (2006) is useful to build a neighborhood set of a point in a river network. Ver Hoef *et al.* (2006) defined that two locations are connected when the intersection of upstreams of two stations is a non-empty set. In our example, segments A, C and B, C are "flow-connected" because the water in A and B can go to location C. On the other hand, C and D are not flow-connected since the water in C cannot go to station D or vice versa.

We use the concept of "flow-connected" to determine whether the two segments are neighbors or not. In this study, when two points are flow-connected, we consider each other neighbors. In Figure 3, suppose that we are interested in removing point C at a specific resolution level. By following the concept of flow-connected, A, B, and E (blue circles) are defined as its neighbors, and D (green triangle) is excluded from the neighborhood of C.

One of the distinct characteristics of the proposed neighborhood selection is that it considers both upstream and downstream neighborhoods. By doing so, it can reduce the number of boundary points. At first glance, including downstream points into the neighborhood seems awkward. However, by combining an appropriate prediction filter construction in Section 4.2, it can generate reasonable prediction filters.

4.2 Construction of the prediction filter

In this section, we consider the problem of the prediction filter construction. The simplest prediction filter is constructed using an equally weighted value vector. However, every river network has its mainstream and substreams. It is plausible that observations on the mainstream usually have a stronger effect on nearby observations. Therefore, the effect of each stream on a given segment should be different. To take into account the influence of each stream segment, we use flow volumes. For the construction of the prediction filter, we consider the size of flow volumes compared to others, called "relative flow volumes" (O'Donnell *et al.*, 2014).

Suppose that we have neighbors of a specific point in a river network. An easy way to weigh is to give the same weight to all neighbors, which may not be desirable. For example, if f_A is much larger than f_B, y_A has a more significant effect on y_C than y_B. Also, if f_D is larger than f_C, y_E is much different from y_C. Therefore, we intend to construct flow-adaptive weights that reflect the above considerations. We now consider predicting the response value of point C with the neighbors in the toy example in Figure 3. Since $f_C = f_A + f_B$, flow-adaptive weights for point C can be defined as ratios of flows,

$$w_A = \frac{f_A}{f_C}, \quad w_B = \frac{f_B}{f_C}, \quad \text{and} \quad w_E = \frac{f_C}{f_E}. \tag{5}$$

Then we obtain a predicted value of y_C as $\hat{y}_C = \tilde{w}_A y_A + \tilde{w}_B y_B + \tilde{w}_E y_E$, where

$$\begin{aligned} \tilde{w}_A &= \frac{f_A/f_C}{f_A/f_C + f_B/f_C + f_C/f_E}, \\ \tilde{w}_B &= \frac{f_B/f_C}{f_A/f_C + f_B/f_C + f_C/f_E}, \text{ and} \\ \tilde{w}_E &= \frac{f_C/f_E}{f_A/f_C + f_B/f_C + f_C/f_E}, \end{aligned} \tag{6}$$

which are normalized flow-adaptive weights to make the sum of weights to be 1, i.e., $\tilde{w}_A + \tilde{w}_B + \tilde{w}_E = 1$. Therefore, the predicted value of the segment C, $\hat{y}_C$ is

$$\hat{y}_C = \tilde{w}_A y_A + \tilde{w}_B y_B + \tilde{w}_E y_E.$$

Hence, we provide a lifting scheme for streamflow data by combining flow-adaptive weights of (5) and (6) with the conventional lifting scheme. In practice, it is rare to know all f values on the entire streamlines. Therefore, it is necessary to estimate flow values. Ver Hoef *et al.* (2006) used equal weights for each split. In this study, it is assumed that the flow volume f in most upstream segments is proportional to their stream order and segment length. Note that the stream order is a positive whole number that is often used in hydrology to define stream-based distance in river networks. There are several stream orders in the literature. Among them, the Shreve stream order of Shreve (1966) is one of the most straightforward stream orders (Ver Hoef *et al.*, 2006, 2010). Cressie *et al.* (2006) defined the stream order as the number of sources in the upstream portion of the river network. The Shreve stream order starts from setting all most upstream segments to 1. Magnitudes increase at all junctions in the river network. For example, if a stream has a magnitude one and combines with a new stream having magnitude 2, it becomes magnitude 3. By doing so, it is able to configure all magnitudes of the given network.

To approximate f values, we use the Shreve stream order and assume that the flow of the most upstream segments is proportional to their lengths to prevent multiple tie values of flow volumes. After defining flow volumes of most upstream segments, one can define flow volumes of the next upstream segments as a sum of their upstream segments. By repeating this approach, we obtain all f values in the river network. It is also assumed that the weights associated with the flow volumes are known to generate $\log(\sqrt{f})$ values. Following O'Donnell *et al.* (2014), we normalize the $\log(\sqrt{f})$ values, which are between 0.2 to 1.5.

4.3 Removal point selection

The removal order should be determined for the streamflow lifting scheme. If the data lie in the real line, it is easy to apply the conventional approach, such as Nunes *et al.* (2006). They used the length of points on the real line for integral calculations. Moreover, it can be extended to the two-dimensional data proposed by Jansen *et al.* (2009). To determine the removal point, Jansen *et al.* (2009) found the highest density observation in the Euclidean domain by considering the integral of the scaling function. In addition, they proposed measuring the Voronoi polygon-based area as a candidate for proper integrals and chose to have the smallest integration point as a removal point in the LOCAAT algorithm.

However, these methods cannot be applied directly to streamflow data because the river network is not easily projected into one- or two-dimensional data. In the streamflow lifting scheme, a simple approach is proposed to measure the contribution of each segment in the data to distinguish the points located in the densest areas of the river network. We define an integral as the contribution of each observation point to the network. More specifically, to define the contribution of each point in streamflow data, we use flow-adaptive weights defined in (6). A simple example is illustrated in Figure 3. Suppose that at the jth level, we want to remove point C with neighborhood points A, B, and E. Let I_A^j denote the integral of point A at the jth level, which is defined by the volume of the segment where A is located, say V_A defined by the product of flow f_A and length of the segment ℓ_A,

$$\begin{aligned} I_A^j &= V_A = f_A \times \ell_A, \\ I_B^j &= V_B = f_B \times \ell_B, \text{ and} \\ I_E^j &= V_E = f_E \times \ell_E. \end{aligned}$$

At the next level $j-1$ after point C is removed, we need to update the integral of neighborhood points. For this purpose, we use a weighted volume of point C according to the weights of neighbors in (6). Thus, I_A^j, I_B^j, and I_E^j are updated to

$$\begin{aligned} I_A^{j-1} &= I_A^j + \tilde{w}_A \times V_C, \\ I_B^{j-1} &= I_B^j + \tilde{w}_B \times V_C, \text{ and} \\ I_E^{j-1} &= I_E^j + \tilde{w}_E \times V_C. \end{aligned}$$

Note that since $\tilde{w}_A \times V_C + \tilde{w}_B \times V_C + \tilde{w}_E \times V_C = I_C^j$, the sum of integrals does not change. Jansen *et al.* (2009) and Nunes *et al.* (2006) used a similar approach for their update step. We select a point that has the minimum value of I^{j-1} as the removal point at the $j-1$th level. For the update filter, we use the minimum norm solution-based filter in (2).

4.4 Nondecimated lifting scheme for streamflow data

In this section, the proposed lifting scheme is generalized to a nondecimated version of the streamflow lifting scheme that can reduce the mean squared error of the lifting scheme in nonparametric regression settings, as mentioned in Section 2.1. According to Knight and Nason (2009), any removal order of lifting algorithm can be considered as a trajectory (or path), $T = (x_{o_1}, \ldots, x_{o_n})$, where $(o_1, o_2, \ldots, o_n)$ is a permutation of the index set, $\{1, \ldots, n\}$. For this purpose, we assume that the current stream distance-based removal order is one of the well-behaved trajectories in terms of the root mean squared error. Then we generate multiple trajectories by permutations. Before generating such trajectories, we assume that suitable clusters of regions of interest are known. To generate these well-behaved trajectories, we first make clusters of observations and do permutation to those within the same cluster. For implementation, we need to choose two tuning parameters, the number of trajectories, Q, and the number of permutations within a single trajectory, v. We note that we use $Q = 10$ and $v = 5$ for simulation study in Section 5.

Before closing this section, we discuss some aspects of the proposed streamflow lifting scheme, such as scaling function integral and prediction filter and the dependent structure of coefficients, following Nunes *et al.* (2006) and Jansen *et al.* (2009) that presented the main ideas of theoretical aspects of the lifting scheme.

Scaling function integral and prediction filter : To represent the initial function $g(x_i)$ at points x_i on the river network, we consider a linear combination of scaling functions $\varphi_{n,k}(x_i) = \delta_{i,k},\ k,\ i \in \{1, \dots, n\}$ as $g(x) = \sum_{k=1}^{n} c_{n,k}\varphi_{n,k}(x)$, where $g(x_i) = \sum_{k=1}^{n} c_{n,k}\delta_{i,k} = c_{n,i}$ and $c_{n,i}$ denote the observation at points x_i on the river network. Moreover, we select the point j_n to be lifted at the nth stage such that

$$\int \varphi_{n,j_n}(x)dx = \min_{k\in\{1,\dots,n\}} \int \varphi_{n,k}(x)dx.$$

In our streamflow lifting scheme, the integral is defined by stream segments and their connectedness structure. For example, in Figure 3, $\int \varphi_{n,C}(x)dx = V_C = f_C \times \ell_C = I_C^n$.

Dependent structure of coefficients : Suppose that we remove the observation point j_n. From the derivations of Nunes *et al.* (2006), the detail coefficient of the lifting transform is

$$d_{j_n} = c_{n,j_n} - \sum_{i\in\mathcal{N}_{n-1,j_n}\cap\mathcal{I}_{n-1}} p_{n-1,j_n,i}c_{n,i}, \tag{7}$$

where $p_{n-1,j_n,i} = \frac{\tilde{w}_i}{\sum_{j\in\mathcal{N}_{n-1,j_n}\cap\mathcal{I}_{n-1}} \tilde{w}_j}$ is a prediction filter. For notational simplicity, we omit index j_n for the prediction filter $p_{n-1,j_n,i}$ and the update filter u_{n-1,k,j_n}. Let $\mathcal{N}_{n-1,j_n} \cap \mathcal{I}_{n-1} = \mathcal{J}_{n-1}$. According to independence assumption of initial observations, we have $\text{var}(d_{j_n}) = \sigma^2\big\{1 + \sum_{i\in\mathcal{J}_{n-1}} p_{n-1,i}^2\big\}$, where σ^2 is the error variance. The update step gives, $\text{var}(c_{n-1,i}) = \text{var}(c_{n,i}) + u_{n-1,i}^2\text{var}(d_{j_n}) + 2u_{n-1,i}\text{cov}(c_{n,i}, d_{j_n})$, for all $i \in \mathcal{J}_{n-1}$, $i \neq j_n$, where $\text{cov}(c_{n,i}, d_{j_n}) = -p_{n-1,i}\sigma^2$. We then have the following covariance terms between the coarser coefficients $\text{cov}(c_{n-1,i}, c_{n-1,j}) = (-p_{n-1,i}u_{n-1,j} - p_{n-1,j}u_{n-1,i})\sigma^2 + u_{n-1,i}u_{n-1,j}\text{var}(d_{j_n})$ for $i, j \in \mathcal{J}_{n-1}$, $i \neq j$, $i, j \neq j_n$, $\text{cov}(c_{n-1,i}, c_{n-1,j}) = 0$ for $i \in \mathcal{J}_{n-1}$, $j \notin \mathcal{J}_{n-1}$, $i, j \neq j_n$, and $\text{cov}(c_{n-1,i}, c_{n-1,j}) = 0$ for any $i, j \notin \mathcal{J}_{n-1}$. This derivation coincides the result of Nunes *et al.* (2006).

5 Simulation study

This section conducts numerical experiments for the evaluation of our approach. Assume that the data are observed from the regression model of (1). We focus on the situation in which the underlying mean-field of the data is piecewise constant. Thus, there are several discontinuous function values in a river network, which may not be properly estimated using conventional smoothing-based methods. For comparison, we consider the flexible smoothing approach of O'Donnell *et al.* (2014) and three variants of the proposed method: streamflow lifting scheme with median thresholding (`S-Lifting (M)`), streamflow lifting scheme with hard thresholding (`S-Lifting (H)`), and nondecimated streamflow lifting scheme with median thresholding (`S-Lifting (N)`).

For simulation setup, two types of river networks are considered: one is the Miho-Cheon streamflow segments in Figures 4(a) and (b), and the other is the simulated river network in Figures 4(c) and (d), which was used in Gallacher *et al.* (2017). The two networks consist of 113 stream segments and 80 stream segments, respectively. For each river network, the entire stream segments are divided into two groups: most upstream segments and non-most upstream segments, as shown in Figure 4(a). Assume that there are no intrinsic sources to change the simulated signal values. The signal values in the non-most upstream segments are then generated from a weighted average of nearby upstream signal values. It implies that the simulation is sufficient to generate only the signal values in the most upstream segments.

In addition, we divide the most upstream segments into several clusters, marked by red circles in Figures 4(a) and (c), to generate inhomogeneous stream network data. The construction of clusters are affected by the sub-catchments provided by NIER, as shown in Figure 4(a), and clusters for the simulated stream network used in Gallacher *et al.* (2017) are roughly built using junctions located in downstream areas marked with green circles in 4(c). We assume that the signal values for all most upstream segments within the cluster are the same. For each simulated data set, $g(x_i)$ values of the most upstream segments are generated as follows: (i) All $g(x_i)$ values in the most upstream segments are set 9. (ii) A cluster is randomly selected from the clusters in Figure 4, and (iii) $g(x_i)$ values in the selected cluster are replaced with a value that is randomly chosen from $\{12, 15, 18\}$. This procedure is repeated until at least 30 most upstream segments have values greater than 9.

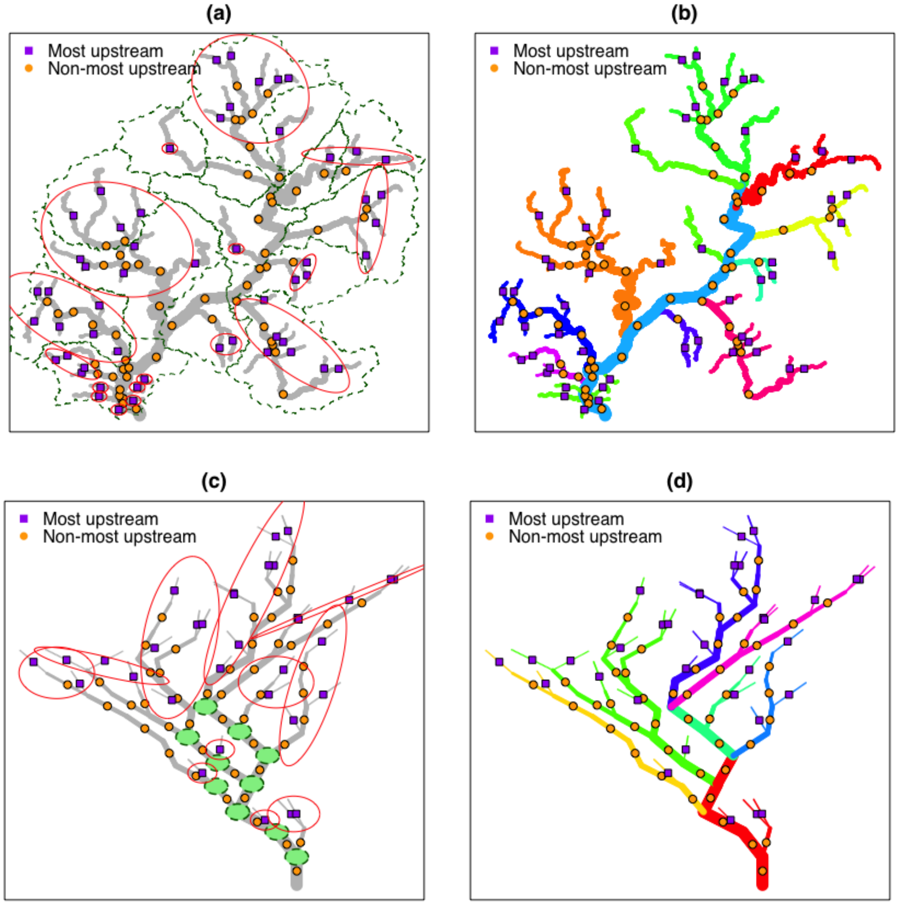

Figure 4: (a) Clusters (red circles) in the Miho-Cheon stream network. Note that the purple squares represent the most upstream segments and the orange circles denote the non-most upstream segments. The green dotted line represents the sub-catchments provided by the NIER. (b) Colors represent sub-streams for the sampling procedure. The sampling probability is proportional to the number of streams of each sub-stream. (c) and (d) show the same information as (a) and (b) for the simulated river network used in Gallacher *et al.* (2017).

Realizations of the simulated data generation are shown in Figures 6(a) and 7(a).

Three spatial sampling designs are also considered for simulation data in river networks: (i) For a sparse design, among a total of 113 Miho-Cheon stream segments, 40 stations located on 40 different segments are considered. A realization is shown in Figure 4(b). (ii) 80 stations are used, which is nearly two-thirds of the number of the Miho-Cheon streams. (iii) 113 stations are considered as a dense case. Along the same line, we analyze the simulated network of Gallacher *et al.* (2017) in two designs: (i) observations are generated at 40 stations, and (ii) one observation is simulated in each segment. To select stations in river networks, we use a spatial stratification sampling method to distribute the resulting stations evenly in the network. See Figures 4(b) and (d).

The noise terms are generated in two ways: (i) For uncorrelated noises, $\varepsilon_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2)$ with $\sigma = 1, 1.5$ and 2. (ii) For correlated noises, we first generate an error value of the river mouth from the normal distribution with mean 0 and standard deviation 0.1. We then recursively generate error values of upper neighborhood stream segments using the conditional normal distribution. For example, an error of location A, ε_A is generated with downstream error values ε_B observed at location B as follows,

$$\varepsilon_A | \varepsilon_B = \mu_B \sim \mathcal{N}\Big(\mu_B, \frac{1}{2}\big(1 - \rho^2\big)\Big), \tag{8}$$

where $\rho = w_A = \frac{f_A}{f_B}$ and w_A is defined in (5). We generate all error values repeatedly for all stream segments. Then, we scale all generated error values to follow $\sigma = \sigma^*$, where $\sigma^* = 1, 1.5$, and 2. Figure 5(a) shows a realization of correlated data generated in the river network used in the above procedure. The y-axis represents the magnitude of the data value, and the x-axis denotes the distance between the mouth of the river network and the upstream of the stream segment. Figure 5(b) shows the plot of the same realization in the river network of Gallacher *et al.* (2017).

As for the evaluation measure, we consider the root mean square error (RMSE) as

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N_{tot}} (g(x_i) - \hat{g}(x_i))^2}{N_{tot}}},$$

where $\hat{g}(x_i)$ is an estimate of segment i, and $N_{tot} = 113$ denotes the total number of stream segments in the river network. For each combination of three spatial designs and three σ's, we compute RMSE values according to our methods and O'Donnell *et al.* (2014) over 100 simulated data sets.

Tables 2, 3, 4, and 5 show the averages of RMSE values under the two river networks. From the results in the tables, we have some observations: (i) The proposed methods outperform O'Donnell *et al.* (2014) for most of the combinations. (ii) The proposed methods work well under the given simulation settings, especially when σ is small. (iii) The method

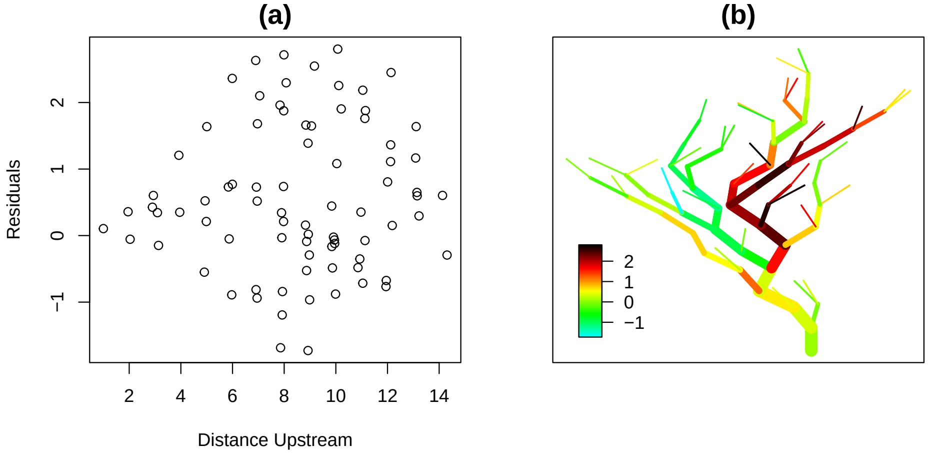

Figure 5: (a) a realization of the correlated error according to the distance upstream, which is the distance between the mouth of the river network and the upstream of the stream segment; and (b) the same realization of the correlated error plotted in the river network used in Gallacher *et al.* (2017).

by O'Donnell *et al.* (2014) provides stable results across the number of observations and σ's, while the performance of the proposed methods is affected by both scenarios. (iv) The nondecimated version of the proposed lifting scheme has lower RMSE compared to other methods in most cases. For visual inspection, we look at one realization example of the fitting results of the spatial design with 40 stations and $\sigma = 1$ shown in Figures 6 and 7. The proposed methods appear to reflect the inhomogeneous features of the underlying fields in the two river networks efficiently.

For additional justification, we consider the streamflow data observed at 60 stations in Gallacher *et al.* (2017), which are shown in Figure 8. The same data are provided in the R package `stpca` (http://researchdata.gla.ac.uk/277/). To evaluate the performance, we generate a simulated data set by adding (i) i.i.d. Gaussian error terms or (ii) correlated errors generated by (8) into the observations. We use three noise levels as $\sigma = 1, 1.5, 2$. Then we compute RMSE values over 100 simulated data sets. As listed in Tables 6 and 7, the proposed nondecimated lifting scheme provides the lowest RMSE values for all cases.

Before closing this section, we compare the computation time for each method. Under the Miho-Cheon stream network, Obs=113 and $\sigma = 1$ setting, the method of O'Donnell *et al.*

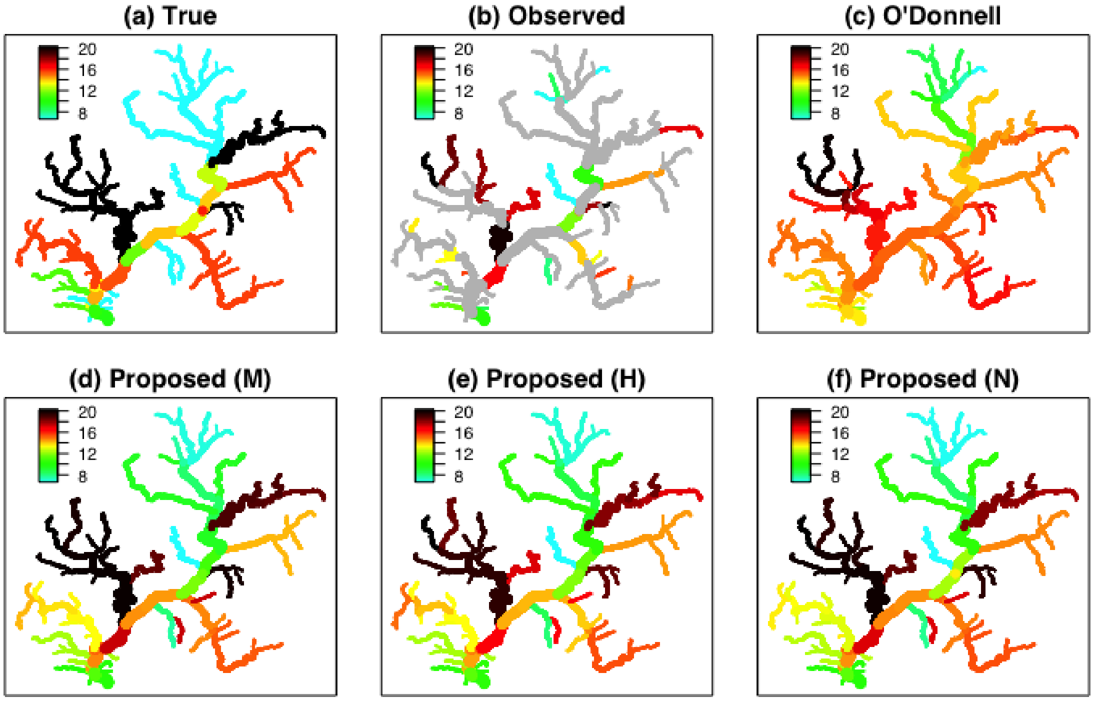

Figure 6: (a) True signal, (b) noisy observations when Obs=40 (unobserved segments are marked by gray lines), (c) fit by O'Donnell *et al.* (2014), (d)-(e) fits by the proposed method with median thresholding and hard thresholding, and (f) fit by the proposed nondecimated method with median thresholding.

Table 2: Averages of RMSE values and their standard errors of 100 simulated datasets with i.i.d Gaussian errors on the Miho-Cheon river network.

RMSE	Obs=40			Obs=80			Obs=113		
(Std. error)	$\sigma = 1$	$\sigma = 1.5$	$\sigma = 2$	$\sigma = 1$	$\sigma = 1.5$	$\sigma = 2$	$\sigma = 1$	$\sigma = 1.5$	$\sigma = 2$
O'Donnell	2.0988	2.2101	**2.2882**	1.5270	1.6336	1.7542	1.3252	1.4458	1.5798
	(0.1957)	(0.2052)	**(0.2317)**	(0.1561)	(0.1513)	(0.1654)	(0.1364)	(0.1320)	(0.1609)
Proposed	1.7089	2.1410	2.3588	1.0662	1.3488	1.6377	0.8362	1.1656	1.4141
(Median)	(0.4181)	(0.5202)	(0.5134)	(0.1877)	(0.2343)	(0.2686)	(0.1187)	(0.1929)	(0.2119)
Proposed	**1.6287**	2.1281	2.3647	**1.0382**	1.3440	1.6400	0.8386	1.2142	1.5172
(Hard)	**(0.3882)**	(0.4913)	(0.4704)	**(0.1528)**	(0.1903)	(0.2524)	(0.1074)	(0.1569)	(0.1985)
Proposed	1.7210	**2.1143**	2.3292	1.0428	**1.3212**	**1.6076**	**0.8000**	**1.1225**	**1.3705**
(Median, nlt)	(0.3907)	**(0.4517)**	(0.4460)	(0.1849)	**(0.2241)**	**(0.2588)**	**(0.1158)**	**(0.1760)**	**(0.2081)**

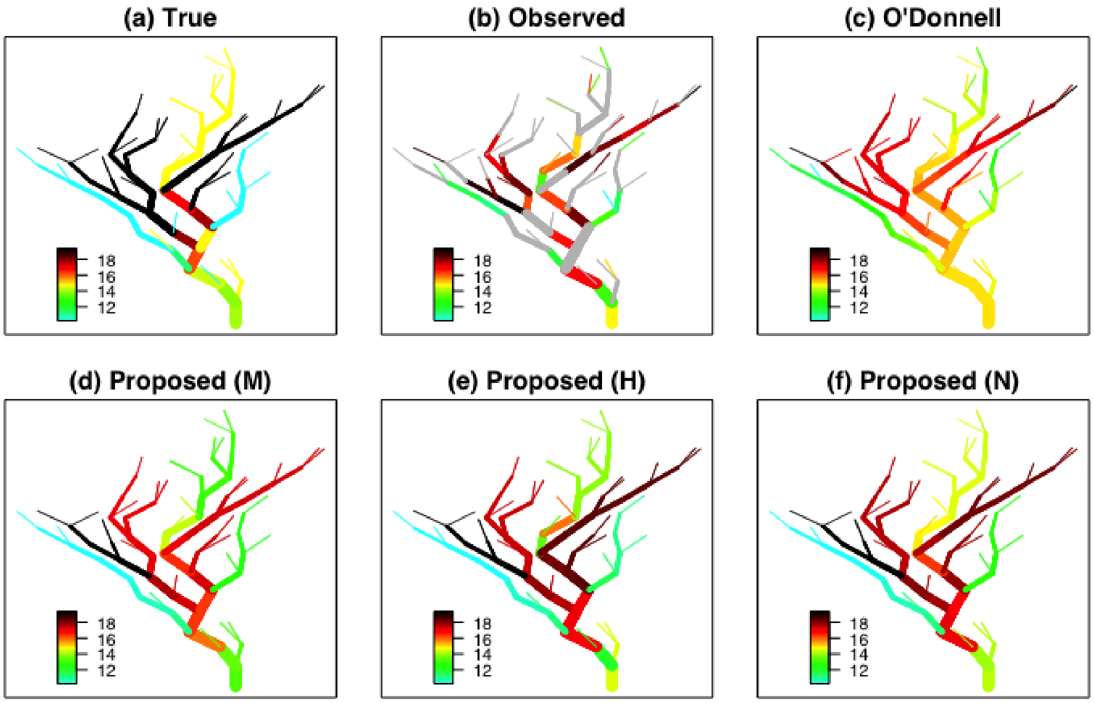

Figure 7: (a) True signal, (b) noisy observations when Obs=40 (unobserved segments are marked by gray lines), (c) fit by O'Donnell *et al.* (2014), (d)-(e) fits by the proposed method with median thresholding and hard thresholding, and (f) fit by the proposed nondecimated method with median thresholding.

Table 3: Averages of RMSE values and their standard errors of 100 simulated datasets with correlated errors on the Miho-Cheon river network.

RMSE	Obs=40			Obs=80			Obs=113		
(Std. error)	$\sigma = 1$	$\sigma = 1.5$	$\sigma = 2$	$\sigma = 1$	$\sigma = 1.5$	$\sigma = 2$	$\sigma = 1$	$\sigma = 1.5$	$\sigma = 2$
O'Donnell	2.1240 (0.1944)	2.1384 (0.1935)	2.2039 (0.2474)	1.5378 (0.1502)	1.6094 (0.1756)	1.7199 (0.1687)	1.3214 (0.1324)	1.4481 (0.1157)	1.5337 (0.1147)
Proposed (Median)	1.4786 (0.2775)	1.6576 (0.2201)	1.7782 (0.2852)	1.0446 (0.0924)	1.2429 (0.1065)	1.3853 (0.1119)	0.9249 (0.0416)	1.1368 (0.0542)	1.3185 (0.0575)
Proposed (Hard)	**1.4477** **(0.2522)**	**1.6108** **(0.2042)**	**1.7573** **(0.2675)**	1.0638 (0.0833)	1.2715 (0.0937)	1.4227 (0.0928)	0.9615 (0.0283)	1.1784 (0.0379)	1.3650 (0.0390)
Proposed (Median, nlt)	1.4879 (0.2648)	1.6738 (0.2267)	1.7928 (0.3097)	**1.0355** **(0.0936)**	**1.2337** **(0.1080)**	**1.3775** **(0.1112)**	**0.9180** **(0.0434)**	**1.1341** **(0.0508)**	**1.1253** **(0.0547)**

Table 4: Averages of RMSE values and their standard errors of 100 simulated datasets with i.i.d Gaussian errors on the river network in Gallacher *et al.* (2017).

RMSE	Obs=40			Obs=80		
(Std. error)	$\sigma = 1$	$\sigma = 1.5$	$\sigma = 2$	$\sigma = 1$	$\sigma = 1.5$	$\sigma = 2$
O'Donnell	3.1470	1.8695	1.9698	1.2698	1.3815	1.5421
	(1.1271)	(0.2835)	(0.2839)	(0.1251)	(0.1727)	(0.2017)
Proposed	2.9422	1.5464	1.8121	0.7265	1.0249	1.2818
(Median)	(1.2386)	(0.3611)	(0.3539)	(0.1212)	(0.1510)	(0.2599)
Proposed	**3.0418**	1.5769	1.9040	0.7396	1.0666	1.3705
(Hard)	**(1.2299)**	(0.3050)	(0.3286)	(0.1317)	(0.1678)	(0.2495)
Proposed	2.9221	**1.4988**	**1.7329**	**0.7162**	**1.0106**	**1.2816**
(Median, nlt)	(1.2432)	**(0.3267)**	**(0.3385)**	**(0.1219)**	**(0.1678)**	**(0.2712)**

Table 5: Averages of RMSE values and their standard errors of 100 simulated datasets with correlated errors on the river network in Gallacher *et al.* (2017).

RMSE	Obs=40			Obs=80		
(Std. error)	$\sigma = 1$	$\sigma = 1.5$	$\sigma = 2$	$\sigma = 1$	$\sigma = 1.5$	$\sigma = 2$
O'Donnell	1.6830	1.6969	1.7092	1.2975	1.3348	1.4520
	(0.2739)	(0.2796)	(0.3021)	(0.1556)	(0.1828)	(0.1563)
Proposed	**1.0913**	1.2489	1.3685	0.8987	1.0963	1.2512
(Median)	**(0.1591)**	(0.1452)	(0.1830)	(0.0581)	(0.0845)	(0.0962)
Proposed	1.1116	1.2748	1.4181	0.9342	1.1403	1.3021
(Hard)	(0.1488)	(0.1248)	(0.1329)	(0.0438)	(0.0597)	(0.0835)
Proposed	1.0933	**1.2405**	**1.3565**	**0.8921**	**1.0847**	**1.2365**
(Median, nlt)	(0.1560)	**(0.1437)**	**(0.1787)**	**(0.0594)**	**(0.0821)**	**(0.1016)**

(2014), the proposed decimated method (median and hard) and the proposed nondecimated method took 57.39 seconds, 3.67 seconds, 3.69 seconds, and 22.58 seconds, respectively, to run a single simulation on the R with CPU 2.80GHz Quad-core Intel Core i7 processor and 16GB memory. Under the river network in Gallacher *et al.* (2017), Obs=80 and $\sigma = 1$ setting, the four methods took 63.89 seconds, 1.61 seconds, 1.60 seconds, and 16.98 seconds, respectively. The proposed streamflow lifting scheme methods are relatively fast when the number of stream segments is close to 100, as listed in Tables 2 and 4, compared to O'Donnell's method that is computationally intensive for selecting the penalty parameter. On the other hand,

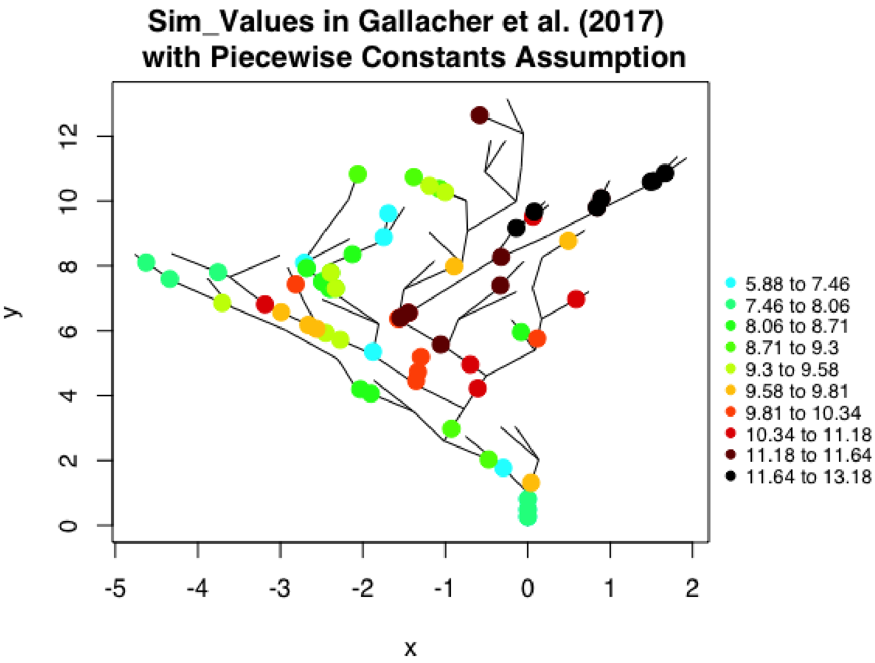

Figure 8: Streamflow data used in Gallacher *et al.* (2017).

Table 6: Averages of RMSE values and their standard errors of the streamflow data in Gallacher *et al.* (2017) with i.i.d. Gaussian errors.

RMSE (Std. error)	Streamflow data in Figure 8		
	$\sigma = 1$	$\sigma = 1.5$	$\sigma = 2$
O'Donnell	1.0853 (0.0648)	1.1624 (0.1027)	1.2572 (0.1403)
Proposed (Median)	1.0197 (0.1229)	1.2159 (0.1497)	1.3015 (0.1467)
Proposed (Hard)	1.0710 (0.1175)	1.3186 (0.1750)	1.4349 (0.2252)
Proposed (Median, nlt)	**0.9816** **(0.1170)**	**1.1440** **(0.1368)**	**1.2368** **(0.1259)**

Table 7: Averages of RMSE values and their standard errors of the streamflow data in Gallacher *et al.* (2017) with correlated errors.

RMSE (Std. error)	Streamflow data in Figure 8		
	$\sigma = 1$	$\sigma = 1.5$	$\sigma = 2$
O'Donnell	1.0511 (0.0491)	1.1143 (0.0889)	1.1694 (0.1102)
Proposed (Median)	1.0510 (0.0975)	1.2107 (0.1081)	1.3015 (0.1467)
Proposed (Hard)	1.0764 (0.1039)	1.1916 (0.1200)	1.3291 (0.1833)
Proposed (Median, nlt)	**1.0231** **(0.0925)**	**1.0792** **(0.0970)**	**1.1460** **(0.0940)**

the proposed methods require finding neighbors at each level, which takes more computation time when the size of the river network is large.

We finally note that R codes used to implement the methods and to carry out some experiments are available at `https://github.com/SeoncheolPark/paper_StreamflowLifting/tree/master/code` in order that one can reproduce the same results.

6 Real data analysis

In this section, we apply the proposed lifting scheme to the real data set in Section 3. We consider the TOC water pollutant observed from 2012 to 2017. Water pollutants typically have some extreme values, which results in skewed empirical distributions. Therefore, we consider the average values of the log transformation of TOC data from 2012 to 2017 at each station shown in Figure 1.

For the configuration of the results, we use an interpolation method based on equations in (6). We consider the river network in Figure 3. Suppose that there are no observations in segment C. That is, we assume that the value of y_c is unknown. Then we interpolate the value of y_c with observations y_A, y_B, and y_E, which results in $\hat{y}_C = w_A y_A + w_B y_B + w_E y_E$, where $w_A + w_B + w_E = 1$. For the nondecimated version of the streamflow lifting scheme, clusters should be set up to achieve stable smoothing results. In this analysis, we only consider

Table 8: $\widetilde{\text{RMSE}}$ results of the interpolation of the Geum-River data set.

	O'Donnell	S-Lifting (M)	S-Lifting (N)
$\widetilde{\text{RMSE}}$	0.1240	0.0818	0.1350

a permutation of observations in the same stream segments, assuming that the original removal path defined through Section 4 is such a well-behaved removal order. Twelve of the 127 stations are located in segments with two or more stations.

Assume that the underlying model of TOC data follows the model of (3). Although the function g is unknown, the similarity of interpolation can be evaluated by the approximated root mean square error ($\widetilde{\text{RMSE}}$),

$$\widetilde{\text{RMSE}} = \sqrt{\frac{\sum_{i=1}^{N_{tot}} (\tilde{g}(x_i) - \hat{g}(x_i))^2}{N_{tot}}},$$

where $N_{tot} = 942$ is the total number of stream segments in the Geum-River network, $\tilde{g}(x_i)$ denote the interpolation of raw data, and $\hat{g}(x_i)$ represents the interpolation of estimates. This $\widetilde{\text{RMSE}}$ can be considered as a measure of global goodness-of-fit in the physical domain.

Figure 9 shows the results of the proposed streamflow lifting scheme for the Miho-Cheon TOC data set. The interpolation for the raw data is shown in panel (a). For comparison, we consider the method of O'Donnell *et al.* (2014), which gives the result in panel (b). The interpolation results of the proposed methods are in panels (c) and (d), respectively. The nondecimated streamflow lifting scheme uses random trajectories so that the results can vary over executions. To obtain a stable $\widetilde{\text{RMSE}}$, we use $Q = 100$ trajectories for the nondecimated streamflow lifting scheme. We used the sub-catchment information, shown as dotted lines in Figure 2 (a), for the cluster construction of the nondecimated lifting scheme. From Figure 9, we observe that the high TOC values of the Geum-River downstream are affected by the water quality of the Miho-Cheon. In other words, TOC from the Miho-Cheon dominates the water pollution in the Geum-River downstream. As for Figure 2, we find that many industrial factories are near the Miho-Cheon catchment area. It is a plausible conclusion that industrial factories may affect the amount of TOC in the river network.

In addition, as shown in Figure 9, the representations by all methods are smoother than the raw data. From panels (c) and (d), the proposed streamflow lifting schemes provide more

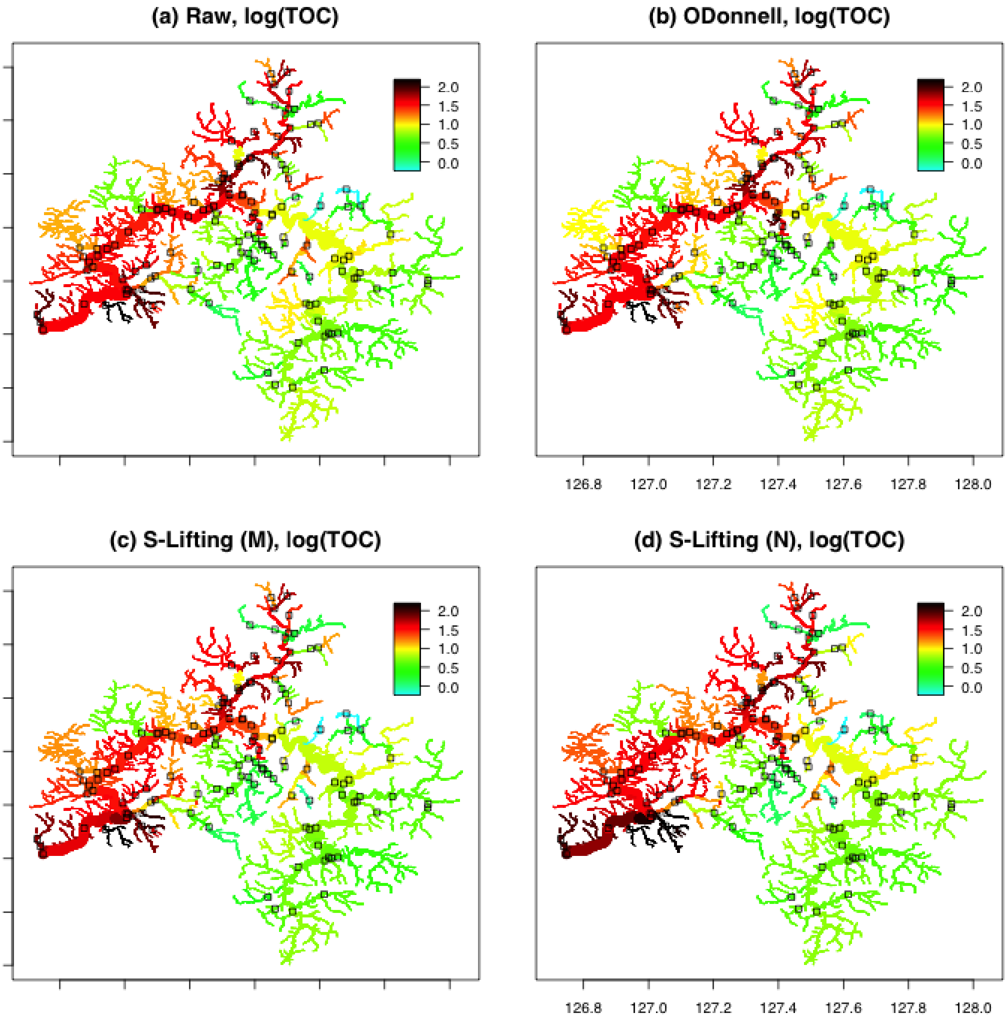

Figure 9: Real data analysis for TOC data. (a) Interpolation of the raw dataset, (b) interpolation of estimates by O'Donnell's method, (c) interpolation of estimates by the proposed streamflow lifting scheme with median thresholding, and (d) interpolation of estimates by the proposed nondecimated streamflow lifting scheme with median thresholding.

smoothed representations in all regions, especially the southeast region around longitude 127.4-128.0 and latitude 35.6-36.2. From the nondecimated version result of the panel (d), we find that some values in the region of longitude 126.8-127.1 and latitude 36.0-36.2 are high. The data seem to be a mixed pattern instead of a piecewise function. So, it is challenging to establish appropriate clusters to carry out the nondecimated streamflow lifting scheme. Compared to this result, the original streamflow lifting scheme and O'Donnell *et al.* (2014)

provide more stable results. This observation is supported by the $\widetilde{\text{RMSE}}$ values listed in Table 8.

To check the normality assumption of this study, we obtain a residual Q-Q plot of the Geum-River data analysis in the proposed method. Figure 10 shows the Q-Q plot of the residuals obtained by the streamflow nondecimated lifting scheme. There are several differences between the distribution of the residuals and the standard normal distribution, especially the distribution of low quantities, but the error distribution is generally considered to follow the standard normal distribution.

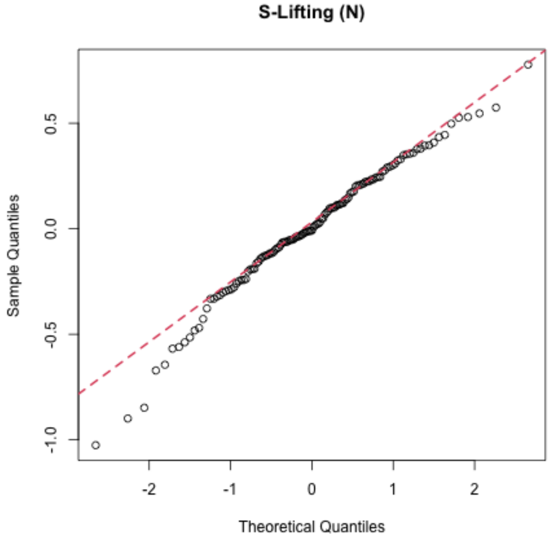

Figure 10: Residual Q-Q plot of the Geum-River data analysis in the nondecimated lifting scheme.

For further evaluation, we compute the leave-one-out cross-validation (LOCV) score as a prediction error measure,

$$\text{LOCV} = \sqrt{\frac{\sum_{i=1}^{N_{obs}} (\tilde{g}(x_i) - \hat{g}^{-i}(x_i))^2}{N_{obs}}},$$

where $\hat{g}^{-i}(x_i)$ is the prediction value of each method estimated using the data without the point x_i. We compute the LOCV score by applying the proposed methods and the method of O'Donnell *et al.* (2014) to $N_{obs} = 115$ observations in the Geum-River network, after eliminating duplicated points at the same stream segments. As listed in Table 9, the proposed lifting scheme is slightly better than O'Donnell *et al.* (2014). Table 9 shows that

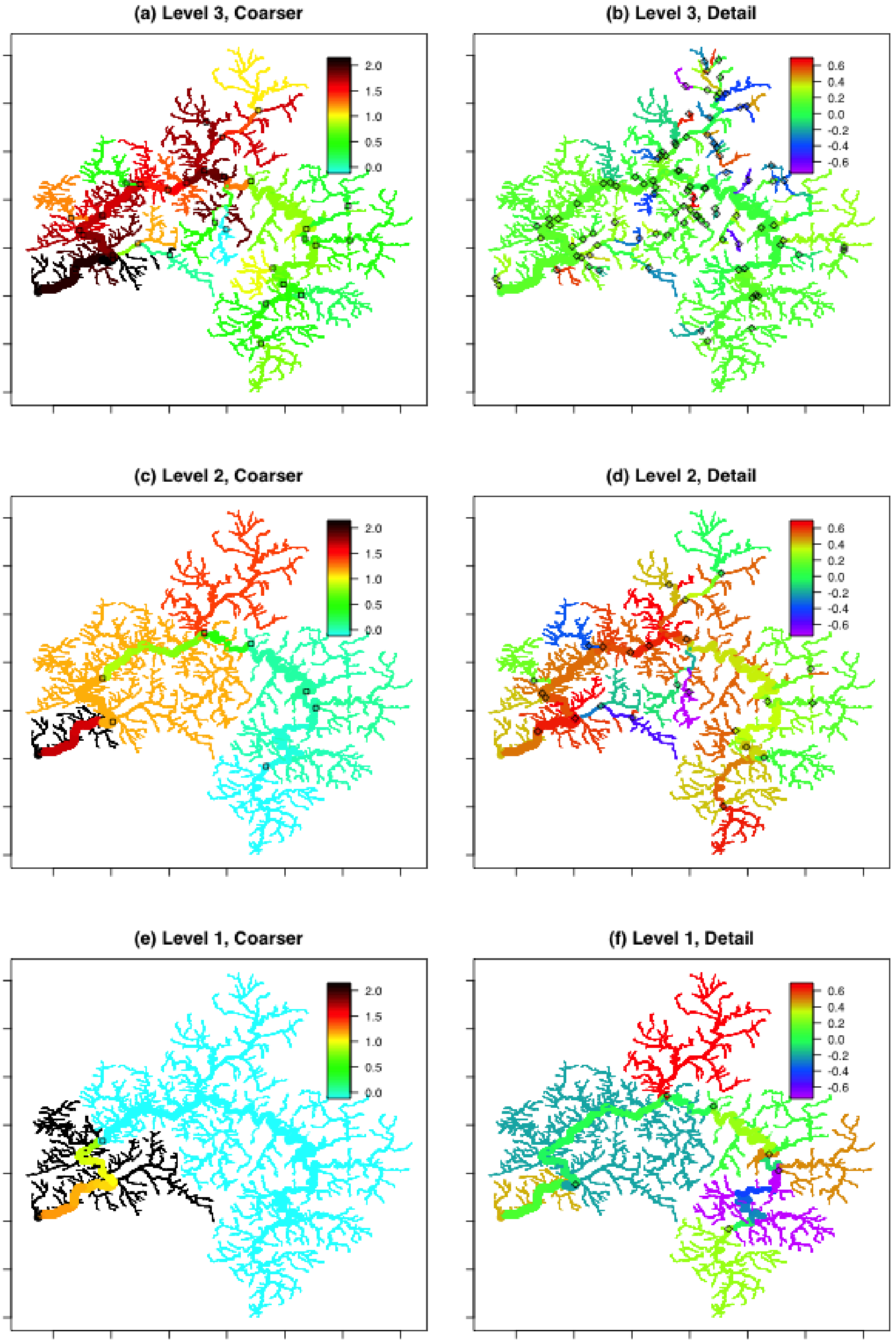

Figure 11: A multiscale analysis of TOC data. (a), (c) and (e) show global components of streamflow data at three different levels. (b), (d) and (f) are corresponding detail components. The number of points at each level (`nkeep`) are `32, 8, 2,` respectively.

Table 9: Leave-one-out cross-validation (LOCV) scores by applying each method to the Geum-River data set.

	O'Donnell	S-Lifting (M)	S-Lifting (N)
LOCV	0.3946	0.4027	0.3916

the proposed nondecimated version of the streamflow lifting scheme has the lowest LOCV score.

To conclude this section, we perform a multiscale analysis of streamflow data, which is one of the advantages of the proposed streamflow lifting scheme method. Let $g_L(x)$ be a representation at the finest level. We then decompose the function $g_L(x)$ into global component $g_{L-1}(x)$ and detailed component $d_{L-1}(x)$. It further breaks down the function $g_{L-1}(x)$ into global component $g_{L-2}(x)$ and detailed component $d_{L-2}(x)$. By repeating the above steps until the coarsest level 1, we finally decompose the $g_L(x)$ as

$$g_L(x) = g_1(x) + \sum_{\ell=1}^{L-1} d_\ell(x),$$

where ℓ denotes the resolution index and the formula for the detailed coefficient d_ℓ is given in (7). As ℓ decreases, the corresponding representation becomes coarser. To perform the above multiscale analysis of the TOC data in the Geum-River network, we consider the representation in Figure 9(c) as the finest level representation $g_{L:=4}(x)$. Figure 11 shows the multiscale representations by the proposed streamflow lifting scheme. In Figure 11(a), we reconstruct the river network field only using 32 stations out of 127 stations, which still holds global features of the representation in Figure 9(c). The difference between the two representations is shown in Figure 11(b) as a detailed field $d_3(x)$. Figures 11(c) and (e) show the global components $g_2(x)$ and $g_1(x)$ using 8 stations and 2 stations, respectively, and Figures 11(d) and (f) show the corresponding differences $d_2(x)$ and $d_1(x)$. As the number of data points for reconstruction decreases, the corresponding representations are becoming rougher with focusing on global patterns. Instead, detail fields at each level provide some important information about networks that global components cannot represent.

7 Concluding remarks

In this paper, we have proposed a new lifting scheme for streamflow data. The proposed methods enable lifting scheme to streamflow data by (i) adopting a stream network adaptive neighborhood selection, (ii) constructing a prediction filter with flow-adaptive weighted averages, and (iii) setting a removal order by defining neighborhood flows of each observation point. By using the proposed neighborhood selection method, we reduce the number of boundary points and predict the values of upstream streamflow points. Besides, we have developed a nondecimated version of the proposed streamflow lifting scheme as a generalization. Simulation studies show that the proposed method works better than the conventional smoothing approach for streamflow data in particular situations, especially if there are some discontinuities in the data.

However, the proposed approach has some limitations. First, it is assumed that the volume of the water flow is proportional to the length of the segments and the Shreve order. In practice, however, the volume of the water flow may differ from the segment length and the Shreve order. The volume of the water varies over seasons. For example, precipitation in Korea is mostly is concentrated in the summer season. Second, in the simulation study, we have gathered segments in the given river network into several artificial groups to enhance the performance of the proposed lifting scheme. However, it is not easy to define optimal clusters in real data analysis. Therefore, one of the future studies will be to suggest an appropriate way to find optimal groups. Third, the removal order of the proposed method is not determined by the value of the streamflow data set, but based on location only. If possible, a data-adaptive removal order selection algorithm is useful to enhance the performance of the proposed method.

Finally, the approach proposed in this study does not provide spatio-temporal data analysis. Since the TOC data are observed irregularly in both space and time domains, it is necessary to have a novel method to carry out spatio-temporal streamflow data analysis. Lindström *et al.* (2014) and O'Donnell *et al.* (2014) solved this problem by calculating biweekly or monthly average data for each station. The method of O'Donnell *et al.* (2014) can then be used to build space-time basis functions with tensor products. However, if we find a way to construct multiscale spatio-temporal bases without merging the data, it will be more useful to capture the multiscale spatio-temporal behavior of the data. It is reserved for future research.

Acknowledgement

Seoncheol Park would like to acknowledge that this research was undertaken thanks in part to funding from the Canada First Research Excellence Fund (Global Water Futures: Solutions to Water Threats in an Era of Global Change, Climate-Related Precipitation Extremes project), the Pacific Climate Impacts Consortium and the Canadian Statistical Sciences Institute. The work by Hee-Seok Oh was supported by the National Research Foundation of Korea (NRF) funded by the Korea government (2018R1D1A1B07042933; 2020R1A4A1018207).

References

Artiola, J, Pepper, I. L., and Brusseau, M. L. (2004). *Environmental Monitoring and Characterization.* Elsevier Academic Press, Burlington.

Cressie, N., Frey, J., Harch, B., and Smith, M. (2006). Spatial prediction on a river network. *Journal of Agricultural, Biological, and Environmental Statistics*, **11**, 127–150.

Dubber, D. and Gray, N. F. (2010) Replacement of chemical oxygen demand (COD) with total organic carbon (TOC) for monitoring wastewater treatment performance to minimize disposal of toxic analytical waste. *Journal of Environmental Science and Health* Part A, **45**, 1595–1600.

Gallacher, K., Miller, C., Scott, E. M., Willows, R., Pope, L., and Douglass, J. (2017). Flow-directed PCA for monitoring networks. *Environmetrics*, **28**, e2434.

Jansen, M. H., and Ooninc x, P. (2005). *Second Generation Wavelets and Applications.* Springer Science and Business Media, London.

Jansen, M., Nason, G. P., and Silverman, B. W. (2009). Multiscale methods for data on graphs and irregular multidimensional situations. *Journal of the Royal Statistical Society* Series B, **71**, 97–125.

Knight, M. I., and Nason, G. P. (2009). A 'nondecimated' lifting transform. *Statistics and Computing*, **19**, 1–16.

Lindström, J., Szpiro, A. A., Sampson, P. D., Oron, A. P., Richards, M., Larson, T. V., and Sheppard, L. (2014). A flexible spatio-temporal model for air pollution with spatial and spatio-temporal covariates. *Environmental and Ecological Statistics*, **21**, 411–433.

Nunes, M. A., Knight, M. I., and Nason, G. P. (2006). Adaptive lifting for nonparametric regression. *Statistics and Computing*, **16**, 143–159.

O'Donnell, D., Rushworth, A., Bowman, A. W., Scott, E. M., and Hallard, M. (2014). Flexible regression models over river networks. *Journal of the Royal Statistical Society* Series C, **63**, 47–63.

Shreve, R. L. (1966). Statistical law of stream numbers. *The Journal of Geology*, **74**, 17–37.

Sweldens, W. (1996). The lifting scheme: A custom-design construction of biorthogonal wavelets. *Applied and Computational Harmonic Analysis*, **3**, 186–200.

Sweldens, W. (1998). The lifting scheme: A construction of second generation wavelets. *SIAM Journal on Mathematical Analysis*, **29**, 511–546.

Ver Hoef, J. M. and Peterson, E. E., and Theobald, D. (2006). Spatial statistical models that use flow and stream distance. *Environmental and Ecological Statistics*, **13**, 449–464.

Ver Hoef, J. M., and Peterson, E. E. (2010). A moving average approach for spatial statistical models of stream networks. *Journal of the American Statistical Association*, **105**, 6–18.