

Article

# PreciPatch: A Dictionary-based Precipitation Downscaling Method

Mengchao Xu <sup>1,2</sup>, Qian Liu <sup>1,2</sup> , Dexuan Sha <sup>1,2</sup> , Manzhu Yu <sup>1,3</sup> , Daniel Q. Duffy <sup>4</sup>, William M Putman <sup>4</sup>, Mark Carroll <sup>4</sup>, Tsengdar Lee <sup>5</sup> and Chaowei Yang <sup>1,2,\*</sup> 

<sup>1</sup> NSF Spatiotemporal Innovation Center, George Mason Univ., Fairfax, VA 22030, USA;

mxu6@gmu.edu (M.X.); qliu6@gmu.edu (Q.L.); dsha@gmu.edu (D.S.); mqy5198@psu.edu (M.Y.)

<sup>2</sup> Department of Geography and GeoInformation Science, George Mason Univ., Fairfax, VA 22030, USA

<sup>3</sup> Department of Geography, Pennsylvania State University, State College, PA 16801, USA

<sup>4</sup> NASA Center for Climate Simulation, Greenbelt, MD 20771, USA; daniel.q.duffy@nasa.gov (D.Q.D.); William.M.Putman@nasa.gov (W.M.P.); mark.carroll@nasa.gov (M.C.)

<sup>5</sup> NASA Headquarters, Washington, DC 20546, USA; tsengdar.j.lee@nasa.gov

\* Correspondence: cyang3@gmu.edu

Received: 11 February 2020; Accepted: 17 March 2020; Published: 23 March 2020



**Abstract:** Climate and weather data such as precipitation derived from Global Climate Models (GCMs) and satellite observations are essential for the global and local hydrological assessment. However, most climatic popular precipitation products (with spatial resolutions coarser than 10km) are too coarse for local impact studies and require “downscaling” to obtain higher resolutions. Traditional precipitation downscaling methods such as statistical and dynamic downscaling require an input of additional meteorological variables, and very few are applicable for downscaling hourly precipitation for higher spatial resolution. Based on dynamic dictionary learning, we propose a new downscaling method, PreciPatch, to address this challenge by producing spatially distributed higher resolution precipitation fields with only precipitation input from GCMs at hourly temporal resolution and a large geographical extent. Using aggregated Integrated Multi-satellitE Retrievals for GPM (IMERG) data, an experiment was conducted to evaluate the performance of PreciPatch, in comparison with bicubic interpolation using RainFARM—a stochastic downscaling method, and DeepSD—a Super-Resolution Convolutional Neural Network (SRCNN) based downscaling method. PreciPatch demonstrates better performance than other methods for downscaling short-duration precipitation events (used historical data from 2014 to 2017 as the training set to estimate high-resolution hourly events in 2018).

**Keywords:** GeoAI; natural disaster; flooding; global water cycle; spatiotemporal data analytics

## 1. Introduction

The studies of local climate impact are critical for environmental management [1], including applications such as water resources, ecosystem services, and agricultural productivity [2,3]. Global Climate Models, also known as general circulation models or GCMs [4], are our primary tools for simulating future climatological variables such as temperature, wind, and precipitation [5]. GCMs help us obtain critical scientific insights into the climate system changes [5], and the outputs (gridded climate datasets) from GCMs played a crucial role in assessing the impact of large-scale climate variation and human activities. However, GCM’s resolutions are often too coarse to resolve important topographical features, and not sufficient to estimate and resolve mesoscale processes that govern local climate changes [6–8]. Downscaling the low-resolution climate variables is used as a primary method to obtain high-resolution information [9]. For example, Regional Climate Models (RCMs) are nested in GCMs to produce higher resolution outputs in a dynamic downscaling fashion with limitations of

computing-intensive and small study areas. Statistical downscaling methods are also developed to enhance GCM outputs' resolution directly to generate high-resolution results at low cost and with high efficiency.

For hydrological impact studies, precipitation and temperature are the most relevant meteorological variables [10–12]. Long-term precipitation prediction is significant for planning flood responses, as well as strategic planning for agriculture, water resources, and other water-related hazards. Meanwhile, shorter-term predictions have more immediate applications, such as hydrological extreme prediction and management [13], and crop yields prediction for a specific area [10]. Precipitation is one of the most challenging data in downscaling studies because precipitation events are not simulated well by models, and precipitation is not spatiotemporally continuous [10,14–16]. Previous studies on precipitation downscaling can only provide downscaled results as time-series predictions for a single station point or multi-points. Thus, traditional downscaling methods are insufficient for understanding the lifecycle of precipitation events in space, especially for short-duration events [17] than point scale or areal averages. However, downscaling precipitation as spatially distributed fields is a challenging task [10]. It usually cannot be achieved by using statistical downscaling methods when high-resolution observation data are not available for the study area. The benefit of the proposed method is that it can downscale the precipitation of short duration events with full coverage in any region.

Downscaling could be addressed as a disaggregation problem, in which theoretically, any number of small-scale weather sequences can be associated with a given set of large-scale values [18]. Statistical downscaling is popular in precipitation downscaling but the recent opportunities in Single Image Super-Resolution (SISR) for downscaling brought the advantage without requiring additional downscaling parameters such as temperature, pressure, wind speed and direction. Many studies have reviewed downscaling concepts and methods. For example, Hewitson and Crane [18] reviewed process-based and empirical methods. Wilby and Wigley [19] investigated regression methods, weather pattern (circulation)-based approaches, stochastic weather generators, and limited-area climate models. Xu [11] and Wilby et al. [20] illustrated the guideline for statistical downscaling methods for climate scenarios. Fowler et al. [21] discussed the hydrological community impact from downscaling, and Maraun et al. [9] tried to connect dynamic downscaling with end-users. We review downscaling according to the methodology as follows.

Statistical downscaling and dynamic downscaling are the two main methods to generate high-resolution regional precipitation based on global climate model output or reanalysis data [22]. Dynamical downscaling [6,23] relies on the availability of GCM and climate model outputs. Statistical downscaling adopts the relationship between local climate variables and large-scale variables [9] depending significantly on the accessibility to other climate variables and the target parameter such as precipitation. Most statistical downscaling methods rely on empirical mathematical relationships from coarse resolution predictors to finer resolution predictands. These relationships are often much faster to implement and compute than dynamical downscaling, which is the main reason for using statistical downscaling. However, such relationships are subject to the stationarity assumption that the relationship between predictors and predictands continues to hold, even in a changing climate [8,19]. Practically, statistical downscaling is coupled with dynamic downscaling methods to provide more reliable results [24]. However, climate variables like precipitation are not well simulated in either GCM or RCM, therefore it is difficult to downscale through statistical methods. Meanwhile, most statistical methods are area limited and require the availability of higher resolution observation data [25], while it is challenging for many regions to get observation data (e.g., mountain areas).

Additionally, statistical methods can hardly produce precipitation field output, but the precipitation field is essential for driving distributed hydrological models for climate change impact studies [26]. Most researches in statistical downscaling focus on specific areas and lead to case-by-case studies. The downscaled quality is highly dependent on the availability of local observation data. Specifically, traditional statistical downscaling methods have the following limitations: (1) their dependency on higher-resolution observations or other variables in addition to precipitation, (2) their results vary in

quality across different regions and often not in the form of precipitation fields, and (3) their ability to downscale future estimations of precipitation is limited to an hourly basis.

Stochastic downscaling can be realized without the consideration of other parameters, however, it yields high-resolution output only through mathematical transformations and ignores the natural features of the target. Machine learning methods are also proposed with the advancement of AI technology [27] but lack in considering the natural characteristics of precipitation. Stochastic downscaling can be considered as a weather pattern approach because of its similar assumption of probability distributions. A more recent recognition is to treat the stochastic downscaling methods as disaggregation techniques that could generate small-scale ensemble rainfall predictions [28–30]. In operations, the stochastic downscaling method is one of the few available techniques that could produce downscaled precipitation output. Rainfall Filtered Autoregressive Model (RainFARM) is one of the most successful stochastic downscaling models [29], which is based on the nonlinear transformation of a Gaussian random field and constrained by rainfall amounts at larger scales. Many have attempted to enhance the RainFARM. For example, Terzago et al. [30] added a map of extra weights to adjust RainFARM output. Recent developments in the image processing field have shown promising performance in increasing spatial resolution of a single image through prior knowledge learned using machine learning approaches [31–34]. By adopting similar techniques or ideas and using them in downscaling applications, Super-Resolution (SR) algorithms and frameworks have drawn significant attention because of their potentials. The dictionary learning-based method, or sparse regularization [35], and the Super-Resolution Convolutional Neural Networks (SRCNN) based method [36] are two main approaches in precipitation downscaling applications. Dictionary learning is an old method while SRCNN is newer and is one of the most popular methods for super-resolution reconstruction in image processing. Although applying SISR algorithms to gridded datasets and downscaling precipitation fields have been attempted by a few researchers [36], providing an appealing result is still a difficult task.

We plan to fill this gap by proposing a novel downscaling method, called PreciPatch, to produce hourly precipitation at a higher spatial resolution based on GCM outputs. The proposed method uses dictionary learning to construct a dictionary for relationships between low-resolution and high-resolution datasets based on historical data, and applies the dictionary to downscaling low-resolution data for a certain timestamp. The remaining sections report our research as follows: This section introduces and reviews related works in statistical downscaling, stochastic downscaling, Single Image Super-Resolution (SISR) based spatial downscaling, and types of downscaled precipitation products. Section 2 introduces the proposed methods in detail. Section 3 evaluates the proposed methods using several study areas across CONUS and compares results generated by other state-of-the-art downscaling methods. Section 4 concludes the study with discussions.

## 2. Methodology

The development of the PreciPatch is inspired by the idea of the Analog Method (AM) and sparse regularization or, more recently called, *dictionary learning*. The general assumption is made that future weather will behave similarly to the past. By constructing a dictionary for low-resolution and high-resolution patches and using the dictionary with mapping information from historical data, the downscaling process could be achieved through a dictionary learning approach.

### 2.1. Precipitation Downscaling Based on Dictionary Learning

#### 2.1.1. Dictionary Learning for Super-Resolution

A classical dictionary trained for sparse coding consists of a limited collection of static atoms, i.e. a set of sample-based fixed patches. It is not a problem in the image super-resolution field since the goal for image enhancement is different from precipitation downscaling (sharpen the image vs. create local variation), and the resolution enhancement rates are different (2 to 4 vs. 8 to 12) [16]. In a

standard dictionary learning approach for image super-resolution, the estimation of a high-resolution image,  $x = [x_1, x_2, x_3, \dots, x_m]^T \in \mathbb{R}^m$ , from its low-resolution counterpart,  $y \in \mathbb{R}^n$ , where  $n \leq m$ , can be recast as an inverse problem.  $x$  may be estimated from  $y$  through a linear structured degradation operator,  $H \in \mathbb{R}^{n \times m}$  and a noise  $e$  [13]:

Equation (1): Inverse estimation

$$y = Hx + e \quad (1)$$

The sparsity of  $x$  implies that it can be approximated by its orthogonal projection  $x_S$  by a few atoms,  $\{\phi_i\}_{i=1}^M$  of a dictionary,  $\Phi \in \mathbb{R}^{m \times M}$  with a vector of representation coefficient  $c$ . By approximating  $x$  with  $x_S$ , the equation could be rewritten as:

Equation (2): Inverse estimation with a dictionary

$$y = H\Phi c + e', \text{ where } e' = H(x - x_S) + e \quad (2)$$

$y$  could also be approximated by using a low-resolution dictionary. The idea of dictionary learning is to train two dictionaries: one low-resolution dictionary and one high-resolution dictionary. By solving the approximation problem for low-resolution, the high-resolution can be recovered using the gathered information. The explanation above is a highly simplified version of sparse coding and dictionary learning, but it could be accomplished in many ways. For example, the SRCNN super-resolution method could be treated as a revised version of dictionary learning with modifications under a deep learning framework [32].

### 2.1.2. Dynamic Dictionary Learning

A critical aspect of precipitation downscaling is to generate local variations from large-scale areal averages. Ideally, the atom number in a dictionary should be large enough to create variations under complex conditions. The generality of large-scale and local variables could be achieved by merging closer patches, regularizing patches, or using regularized searching functions [37]. The dictionary learning method is still inadequate in precipitation downscaling circumstances. One important reason is that gridded data sets are different from pure image files, and variations in precipitation are difficult to simulate and, hence hard to generalize.

The proposed approach is inspired by the idea of dictionary learning; however, it differs in many ways and does not follow any dictionary learning frameworks. For example, the atoms in the dictionary are not used directly as patches; instead, they were used as base patch values to simulate new patches. Therefore, the dictionary learning process becomes a dynamic approach.

### 2.1.3. Downscaling Workflow

The general workflow for the proposed downscaling method is to construct a dictionary consisting of linked low and high-resolution patches using historical datasets. A similarity search is then performed on the low-resolution patch with the low-resolution GCM product as input to find the linked high-resolution patch in the dictionary. The high-resolution output will be produced by linearly aligning the corresponding high-resolution patches. In detail, the dynamic dictionary learning approach includes the following steps:

1. Select a GCM precipitation product that needs to be downscaled.
2. Select a downscaling scale ratio or a desired spatial resolution. Let  $L_s$  denote the scaling factor, for  $L_s \in \mathbb{Z}^+$ .
3. Select an existing gridded precipitation dataset  $A_{HR}$  (observation and bias-corrected data are preferred) that has a native spatial resolution match or close to the desired output spatial resolution  $L_s$ . The temporal resolution should also be matched on the same level (e.g., hourly, 6-hourly, daily, etc.).

4. Upscale  $A_{HR}$  to the spatial resolution level that matches or is close to the GCM precipitation product (a widely used method is to aggregate grids as averages from the higher resolution [30]).
5. Use the upscaled output and  $A_{HR}$  as inputs to construct the patch dictionary.
6. Use the GCM precipitation product as input and perform a similarity search on the constructed dictionary to find an estimated high-resolution patch for each grid.
7. Align each high-resolution patch to form a weight mask for the high-resolution precipitation field.
8. Use the weight mask as a basis, and the original GCM product as the rain rate constraint to produce the downscaled output.

The workflow of the PreciPatch method (Figure 1) is detailed in the following subsections.

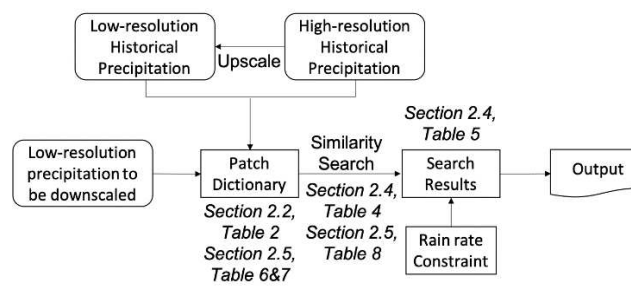


Figure 1. PreciPatch method workflow.

## 2.2. High-Resolution Dictionary Construction

The core of the proposed dynamic dictionary learning approach is to construct a dictionary consisting of both low-resolution and high-resolution patches. Patches are defined as subsets from the original gridded datasets, and the dictionary is enriched by ingesting more datasets. This section introduces algorithms to construct the dictionary.

### 2.2.1. Recording the Spatial Difference

Before constructing the dictionary, the Diff Array is proposed and defined in Table 1 as a record of space difference between grid cells, and is used in further analysis of other algorithms. Figure 2 shows the construction of the Diff Array; the difference between cells is defined as “top minus bottom” and “left minus right.” In the Diff Array, precipitation value differences between nearby grids are recorded and will be used to classify the spatial differences between grids. The Diff Array stores 12 values in total for a three by three array. The Diff Array will be used for dictionary construction (Section 2.2.2) and similarity search (Section 2.4).

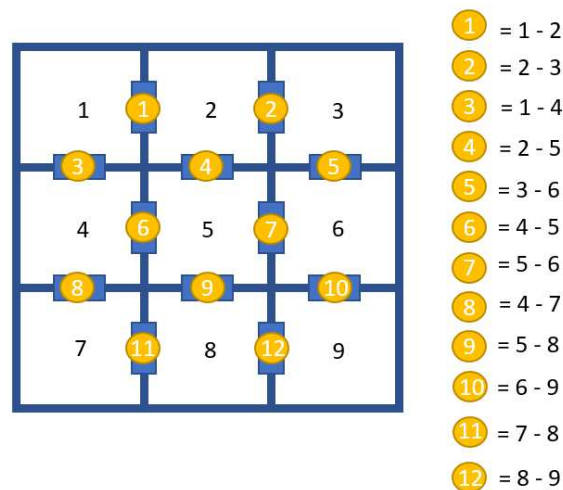


Figure 2. A graph representation of Diff Array.

**Table 1.** The Diff Array.

Name: Create the Diff Array	
<b>Input:</b>	Array $B_{LR}[3][3]$ , a non-empty array with size $3 \times 3$
<b>Output:</b>	$\text{Diff}(B_{LR}[3][3]) = B_{DIFF}[12]$ , the Diff array of $B_{LR}[3][3]$ with size 12
<b>Procedure:</b>	<p>(1) <math>\text{Diff}(B_{LR}[3][3]) = B_{DIFF}[12]</math> is defined as:</p> $B_{DIFF}[0] = B_{LR}[0][0] - B_{LR}[0][1],$ $B_{DIFF}[1] = B_{LR}[0][1] - B_{LR}[0][2],$ $B_{DIFF}[2] = B_{LR}[0][0] - B_{LR}[1][0],$ $B_{DIFF}[3] = B_{LR}[0][1] - B_{LR}[1][1],$ $B_{DIFF}[4] = B_{LR}[0][2] - B_{LR}[1][2],$ $B_{DIFF}[5] = B_{LR}[1][0] - B_{LR}[1][1],$ $B_{DIFF}[6] = B_{LR}[1][1] - B_{LR}[1][2],$ $B_{DIFF}[7] = B_{LR}[1][0] - B_{LR}[2][0],$ $B_{DIFF}[8] = B_{LR}[1][1] - B_{LR}[2][1],$ $B_{DIFF}[9] = B_{LR}[1][2] - B_{LR}[2][2],$ $B_{DIFF}[10] = B_{LR}[2][0] - B_{LR}[2][1],$ $B_{DIFF}[11] = B_{LR}[2][1] - B_{LR}[2][2]$
<b>(2) Return</b>	$B_{DIFF}[12]$

### 2.2.2. The Dictionary Construction Algorithm

The patch dictionary used in this downscaling method is constructed using the algorithm introduced in Table 2. The high-resolution and low-resolution dictionaries are integrated as one, and the spatial information (i.e., Diff Array) of the low-resolution patches is stored. High-resolution datasets and low-resolution datasets are decomposed into small patches and reorganized into a patch dictionary. In this case, the low-resolution datasets are divided into three by three low-resolution patches, and the high-resolution datasets are divided into high-resolution patches with sizes corresponding to the scaling factor. The Diff Array of each low-resolution patch is also calculated and stored, which provides additional information about the spatial relationships between nearby grids of each low-resolution patch.

### 2.3. A Dynamic Time Warping (DTW) Based Similarity Search

After constructing the dictionary, a similarity search is performed against the dictionary using low-resolution inputs. It is different from the widely used Mean Squared Error (MSE) or Euclidean Distance-based similarity search. Dynamic Time Warping (DTW) distance is utilized in this research to help to measure spatial similarities between two low-resolution patches. For example, the following algorithm is used to calculate the DTW distance of two data series with equal length. The original version of DTW distance is used for measuring the similarity between two-time series with different lengths. See related research done by Itakura [38] and Keogh and Ratanamahatana [39]. Operational speaking, the DTW distance could be applied to any data series in the form of arrays. Table 3 provides the algorithm for an application, revised from the original version of DTW [39].



**Table 2.** The dictionary construction algorithm.

Name: Construction of the High-Resolution Dictionary	
<b>Definition:</b>	
$L_s$ is the scaling factor, for $L_s \in \mathbb{Z}^+$	
$A_{HR}$ is one-time slice from a gridded precipitation dataset	
$A_{LR}$ is the aggregated (upscaled) result from $A_{HR}$ using area average based on the scaling factor $L_s$	
<b>Input:</b>	
Array $A_{LR}[m][n]$	$= \begin{bmatrix} l_{0,0} & \cdots & l_{0,n} \\ \vdots & \ddots & \vdots \\ l_{m,0} & \cdots & l_{m,n} \end{bmatrix}$ , for $m, n \in \mathbb{Z}^+$
Array $A_{HR}[q][p]$	$= \begin{bmatrix} h_{0,0} & \cdots & h_{0,p} \\ \vdots & \ddots & \vdots \\ h_{q,0} & \cdots & h_{q,p} \end{bmatrix}$ , for $q, p \in \mathbb{Z}^+, q = (m+1) * L_s - 1, p = (n+1) * L_s - 1$
<b>Output:</b>	
$D\{d_{LR}, d_{HR}, d_{DIFF}\}$ , the patch dictionary consists of the low-resolution patch set $d_{LR}$ , high-resolution patch set $d_{HR}$ , and spatial difference set $d_{DIFF}$ (defined in Table 1)	
<b>Procedure:</b>	
(1) For ( $j = 1$ to $j = m - 1$ ) do {	
For ( $i = 1$ to $i = n - 1$ ) do {	
New Array $B_{LR}[3][3]$ , Array $B_{HR}[L_s][L_s]$ , Array $B_{DIFF}[12]$	
For ( $k = -1$ to $k = 1$ ) do {	
For ( $r = -1$ to $r = 1$ ) do {	
$B_{LR}[k+1][r+1] = A_{LR}[j+k][i+r]$	
}	
}	
Appending $B_{LR}$ as the last element of $d_{LR}$	
$B_{DIFF}[12] = \text{Diff}(B_{LR}[3][3])$	
Appending $B_{DIFF}$ as the last element of $d_{DIFF}$	
$B_{HR} = A_{HR}[(j * L_s) \text{ to } ((j+1) * L_s - 1)][(i * L_s) \text{ to } ((i+1) * L_s - 1)]$	
Appending $B_{HR}$ as the last element of $d_{HR}$	
}	
(2) Return $D\{d_{LR}, d_{HR}, d_{DIFF}\}$ as the integration of sets $d_{LR}$ , $d_{HR}$ , and $d_{DIFF}$	

**Table 3.** DTW distance review.

Name: DTW Distance of Two Data Series with Equal Length ( $n$ )	
<b>Input:</b>	
Array $A$	$= [a_1, a_2, \dots, a_i, \dots, a_n]$
Array $B$	$= [b_1, b_2, \dots, b_j, \dots, b_n]$
<b>Output:</b>	
$distance_{DTW}(A, B)$ , the DTW distance between $A$ and $B$	
<b>Procedure:</b>	
(1) Construct an $n$ -by- $n$ matrix where the $(i^{th}, j^{th})$ element of the matrix contains the distance $dist(a_i, b_j)$ between the two points $a_i$ and $b_j$ , (i.e. $dist(a_i, b_j) = (a_i - b_j)^2$ ).	
(2) A warping path $W$ is a contiguous set of matrix elements that defines a mapping between $A$ and $B$ . The $k^{th}$ element of $W$ is defined as $w_k = (i, j)_k$ , therefore,	
$W = w_1, w_2, \dots, w_k, \dots, w_K$ , where $n \leq K \leq 2n - 1$	
$W$ is subject to several constrains:	
• Boundary conditions: $w_1 = (1, 1)$ and $w_K = (n, n)$	
• Continuity: given $w_k = (c, d)$ , then $w_{k-1} = (c', d')$ , where $c - c' \leq 1$ and $d - d' \leq 1$	
• Monotonicity: given $w_k = (c, d)$ , then $w_{k-1} = (c', d')$ , where $c - c' \geq 0$ and $d - d' \geq 0$	
(3) The DTW path is the path that minimizes the warping cost:	
$DTW(A, B) = \min \left\{ \sqrt{\sum_{k=1}^K w_k} \right\}$	
(4) Define the cumulative distance $\gamma(i, j)$ as the distance found in the current cell and the minimum of the cumulative distances of the adjacent elements:	
$\gamma(i, j) = dist(a_i, b_j) + \min\{\gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1)\}$	
(5) $distance_{DTW}(A, B) = \gamma(n, n)$ , where the path is $DTW(A, B)$	
(6) Return $distance_{DTW}(A, B)$	

#### 2.4. Generate the High-Resolution Weight Mask Through a Double-Layer DTW Similarity Fuzzy Search Algorithm

DTW distance offers a more comprehensive estimation for similarity than simple MSE. This approach could capture linear changes; however, precipitation field downscaling requires estimation of spatial changes. A double-layer DTW similarity search is then proposed in Table 4, which utilizes the Diff Array stored in the previous steps to compare rainfall peaks and valleys in the neighborhood cells spatially. Additionally, a fuzzy search layer is added into the algorithm to avoid the overfitting of patches. In Table 4, each grid from the low-resolution input is compared with the patches in the constructed dictionary along with its spatial relationship with nearby eight grids (the Diff Array). The final high-resolution patches will be estimated by comparing the similarity of the low-resolution input grids with low-resolution dictionary patches. Specifically, closest patches from the dictionary are selected for each input grid, and their corresponding high-resolution patches are averaged to produce a final high-resolution patch for the input grid. The final high-resolution output field is produced by aligning each high-resolution patch found for each input grid.

**Table 4.** A double-layer Dynamic Time Warping (DTW) similarity fuzzy search algorithm.

Name: Similarity Fuzzy Search Based on a Double-layer DTW Distance	
<b>Input:</b>	$L_s$ , for $L_s \in \mathbb{Z}^+$ , is the scaling factor $D\{d_{LR}, d_{HR}, d_{DIFF}\}$ , is the patch dictionary with size $T_D$ for each set Array $C_{LR}[y][x] = \begin{bmatrix} l_{0,0} & \cdots & l_{0,x} \\ \vdots & \ddots & \vdots \\ l_{y,0} & \cdots & l_{y,x} \end{bmatrix}$ , for $y, x \in \mathbb{Z}^+$ and $y, x \geq 2$ , is the low-resolution dataset
<b>Output:</b>	Array $C_{HR}[g][k] = \begin{bmatrix} h_{0,0} & \cdots & h_{0,k} \\ \vdots & \ddots & \vdots \\ h_{g,0} & \cdots & h_{g,k} \end{bmatrix}$ , for $g, k \in \mathbb{Z}^+$ , $g = (y-1) * L_s - 1$ , $k = (x-1) * L_s - 1$ , is the downscaled weight mask
<b>Procedure:</b>	<pre> (1) For (j = 1 to j = y - 1) do {     For (i = 1 to i = x - 1) do {         New Array E<sub>LR</sub>[3][3], Array E<sub>DIFF</sub>[12],         New Array E<sub>1st</sub>[L<sub>s</sub>][L<sub>s</sub>], Array E<sub>2nd</sub>[L<sub>s</sub>][L<sub>s</sub>], Array E<sub>3rd</sub>[L<sub>s</sub>][L<sub>s</sub>]         For (K = -1 to k = 1) do {             For (r = -1 to r = 1) do {                 E<sub>LR</sub>[k + 1][r + 1] = C<sub>LR</sub>[j + k][i + r]             }         }         E<sub>DIFF</sub>[12] = Diff(E<sub>LR</sub>[3][3])         Let 1st_best_so_far = infinity         Let 2nd_best_so_far = infinity         Let 3rd_best_so_far = infinity         For each t in T<sub>D</sub> do {             dist<sub>part1</sub> = distance<sub>DTW</sub>(flat(E<sub>LR</sub>), flat(d<sub>LR</sub>[t]))             dist<sub>part2</sub> = distance<sub>DTW</sub>(E<sub>DIFF</sub>, d<sub>DIFF</sub>[t])             dist<sub>all</sub> = dist<sub>part1</sub> + dist<sub>part2</sub>             If (dist<sub>all</sub> &lt; 1st_best_so_far) then {                 1st_best_so_far = dist<sub>all</sub>                 E<sub>1st</sub> = d<sub>HR</sub>[t]             }             Else if (dist<sub>all</sub> &lt; 2nd_best_so_far) then {                 2nd_best_so_far = dist<sub>all</sub>                 E<sub>2nd</sub> = d<sub>HR</sub>[t]             }             Else if (dist<sub>all</sub> &lt; 3rd_best_so_far) then {                 3rd_best_so_far = dist<sub>all</sub>                 E<sub>3rd</sub> = d<sub>HR</sub>[t]             }             Else {Continue}         }         C<sub>HR</sub>[((j - 1) * L<sub>s</sub>) to (j * L<sub>s</sub> - 1)][((i - 1) * L<sub>s</sub>) to (i * L<sub>s</sub> - 1)] =         Average(E<sub>1st</sub> + E<sub>2nd</sub> + E<sub>3rd</sub>)     } } (2) Return C<sub>HR</sub> </pre>



The result from the algorithm in Table 4 is a high-resolution weight mask, but it cannot be treated as the final downscaling result, yet it is very close to the aimed output. The downscaling algorithm described in Table 5 is a necessary step to make sure the final output is constrained by the original low-resolution input, i.e., consistent with the large-scale precipitation amount.

**Table 5.** Downscaling algorithm based on a weighted mask and constrains.

---

**Name: Produce Downscaled Results from a Weighted Mask and Constrains**

---

**Input:**  
 $L_s$ , for  $L_s \in \mathbb{Z}^+$ , is the scaling factor

Array  $C_{LR}[y][x] = \begin{bmatrix} l_{0,0} & \cdots & l_{0,x} \\ \vdots & \ddots & \vdots \\ l_{y,0} & \cdots & l_{y,x} \end{bmatrix}$ , for  $y, x \in \mathbb{Z}^+$  and  $y, x \geq 2$ , is the low-resolution dataset

Array  $C_{HR}[g][k] = \begin{bmatrix} h_{0,0} & \cdots & h_{0,k} \\ \vdots & \ddots & \vdots \\ h_{g,0} & \cdots & h_{g,k} \end{bmatrix}$ , for  $g, k \in \mathbb{Z}^+$ ,  $g = (y-1) * L_s - 1$ ,  $k = (x-1) * L_s - 1$ , is the downsampled weight mask

**Output:**

Array  $DS_{HR}[g][k] = \begin{bmatrix} h_{0,0} & \cdots & h_{0,k} \\ \vdots & \ddots & \vdots \\ h_{g,0} & \cdots & h_{g,k} \end{bmatrix}$ , for  $g, k \in \mathbb{Z}^+$ ,  $g = (y-1) * L_s - 1$ ,  $k = (x-1) * L_s - 1$ , is the downsampled precipitation field from  $C_{LR}$

**Procedure:**

(1) **For** ( $j = 1$  **to**  $j = y - 1$ ) **do** {  
     **For** ( $i = 1$  **to**  $i = x - 1$ ) **do** {  
          $DS_{HR}[(j-1) * L_s \text{ to } (j * L_s - 1)][[(i-1) * L_s \text{ to } (i * L_s - 1)]] =$   
          $Sum(C_{LR}[j][i]) * (\frac{C_{HR}[(j-1)*L_s \text{ to } (j*L_s-1)][[(i-1)*L_s \text{ to } (i*L_s-1)]]}{Sum(C_{HR}[(j-1)*L_s \text{ to } (j*L_s-1)][[(i-1)*L_s \text{ to } (i*L_s-1)]])})$   
     }  
 }

(2) **Return**  $DS_{HR}$

---

### 2.5. Patch Dictionary Classification and Loose Index

The spatiotemporal complexity of DTW is  $O(n^2)$  for the data series with the same length  $n$  [39]. A patch dictionary containing  $K$  data series would have a complexity of  $O(K \cdot n^2)$ . The databases or dictionary used in prior research is relatively small, typically less than 10,000 samples [40]. However, the dictionary constructed in previous sections will easily exceed 1,000,000 samples after absorbing years of data. Therefore, classification and indexing become necessary if we want to maintain searching performance at a practical level. The following sections will introduce one classification algorithm to divide the patch dictionary into sub-dictionaries, an effective loose index method to generate an individual index for each sub-dictionary, and an updated similarity search that utilizes the classification and indexing system.

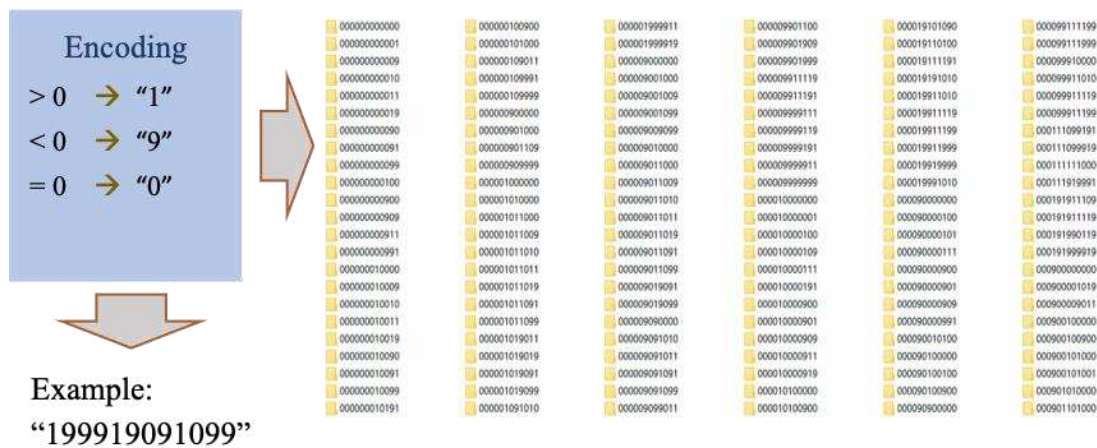
#### 2.5.1. Dictionary Classification Algorithm

The patch classification algorithm in Table 6 is a simple but effective coding method based on the spatial difference of low-resolution patches. Sub-dictionaries are classified according to the code. A similarity search will be done against individual sub-dictionary instead of the whole dictionary, which saves tremendous scan time.

**Table 6.** Patch dictionary classification algorithm.

Name: Build up Classification
<p><b>Input:</b>  <math>D\{d_{LR}, d_{HR}, d_{DIFF}\}</math>, is the patch dictionary with size <math>T_D</math> for each set</p> <p><b>Output:</b>  The classification code <math>Code_D</math> of an element <math>D\{d_{LR}[t], d_{HR}[t], d_{DIFF}[t]\}</math> in <math>D\{d_{LR}, d_{HR}, d_{DIFF}\}</math> for an element <math>t</math> in <math>T_D</math></p> <p><b>Procedure:</b></p> <p>(1) Let <math>Code_D = ""</math></p> <p>(2) For each <math>d</math> in <math>d_{DIFF}[t]</math> do {            If (<math>d &lt; 0</math>) then {<math>Code_D + = string(9)</math>}            Else if (<math>d &gt; 0</math>) then {<math>Code_D + = string(1)</math>}            Else if (<math>d = 0</math>) then {<math>Code_D + = string(0)</math>}            Else {print (NaN)}            }  }</p> <p>(3) Return <math>Code_D</math></p>

Figure 3 gives a preview of the classification system when implementing. Each file folder in the figure represents a sub-dictionary that has been classified according to the encoding system.

**Figure 3.** Dictionary classification.

### 2.5.2. Loose Index Algorithm for Sub-Dictionaries

Although the atom number in sub-dictionaries is much less than the original dictionary, the scanning process will still hold if large amounts of historical data are used in constructing the dictionary. Therefore, an indexing system is always needed to reduce lookup time. An index is introduced in Table 7 to store comparative DTW distance between the patches and the x-axis (sets with all zeros). The idea is to filter potential solutions for arbitrary similarity searches. In Table 7, the combined DTW distance is pre-calculated for each low-resolution patch between the zero-vector and its Diff Array between the zero-vector. The combined DTW distance could not be treated as the exact index but enough for a loose indexing system.

### 2.5.3. Updated Similarity Search Based on Dictionary Classification and Loose Index

After classifying the dictionary and building the loose index, the similarity search algorithm should be revised to leverage the index structure. Table 8 offers the updated version part of the double-layer similarity fuzzy search algorithm from Table 4. In Table 8, each low-resolution input grid is compared with the patches in the corresponding sub-dictionaries that follows the same spatial relationship with its 8 nearby grids. The DTW distance between the patches and the zero-vector have been pre-calculated and is compared with the DTW distance between the input grid with the

zero-vector to form a candidate set of patches. Then the exact DTW distance is calculated among the candidate patches to find the three closest patches and produce the final patch piece.

**Table 7.** The sub-dictionary loose indexing algorithm.

Name: Build up a Loose Index for the Sub-dictionary
<p><b>Input:</b>  <math>SD\{sd_{LR}, sd_{HR}, sd_{DIFF}\}</math>, is the sub-dictionary under a classification code <math>Code_{SD}</math> with size <math>T_{SD}</math> for each set</p> <p><b>Output:</b>            Index set <math>sd_{index}</math> with size <math>T_{SD}</math></p> <p><b>Procedure:</b></p> <p>(1) <b>For each</b> <math>t</math> <b>in</b> <math>T_{SD}</math> <b>do</b> {                <math>dist_{part1} = distance_{DTW}(flat(sd_{LR}[t]), flat(zeros(sd_{LR}[t])))</math>                <math>dist_{part2} = distance_{DTW}(sd_{DIFF}[t], zeros(sd_{DIFF}[t]))</math>                <math>dist_{zero} = dist_{part1} + dist_{part2}</math>                <b>Appending</b> <math>dist_{zero}</math> <b>as the last element of</b> <math>sd_{index}</math>            }            (2) <b>Return</b> <math>sd_{index}</math></p>

**Table 8.** The similarity search algorithm based on dictionary classification and the index.

Name: Using the Classification and Index to Search
<p><b>Input:</b>  <math>L_s</math>, for <math>L_s \in \mathbb{Z}^+</math>, is the scaling factor  <math>N_{patch}</math>, for <math>N_{patch} \in \mathbb{Z}^+</math>, is the minimum number of patches will be examined            Fully classified patch dictionary <math>D\{d_{LR}, d_{HR}, d_{DIFF}\}</math>, for each sub-class with code <math>Code_{SD}</math>, the indexed sub-dictionary is <math>SD\{sd_{LR}, sd_{HR}, sd_{DIFF}, sd_{index}\}</math> with size <math>T_{SD}</math>, for <math>SD \in \mathbb{Z}^+</math>            A non-empty array <math>E_{LR}[3][3]</math>, and its Diff array <math>E_{DIFF}[12]</math></p> <p><b>Output:</b>            The similarity fuzzy search results: <math>E_{1st}[L_s][L_s]</math>, <math>E_{2nd}[L_s][L_s]</math>, and <math>E_{3rd}[L_s][L_s]</math></p> <p><b>Procedure:</b></p> <p>(1) <b>Find</b> code <math>Code_{SD}</math> for <math>E_{DIFF}[12]</math> and <b>locate</b> the sub-dictionary <math>SD\{sd_{LR}, sd_{HR}, sd_{DIFF}, sd_{index}\}</math>            (2) <b>Calculate</b> <math>dist_{zero}</math> for <math>E_{LR}[3][3]</math> and <math>E_{DIFF}[12]</math>                <math>dist_{part1} = distance_{DTW}(flat(E_{LR}[3][3]), flat(zeros(E_{LR}[3][3])))</math>                <math>dist_{part2} = distance_{DTW}(E_{DIFF}[12], zeros(E_{DIFF}[12]))</math>                <math>dist_{zero} = dist_{part1} + dist_{part2}</math>            (3) <b>Calculate</b> absolute differences between <math>dist_{zero}</math> and each element in <math>sd_{index}</math>            (4) <b>Sort</b> <math>sd_{index}</math> based on the absolute differences in ascending order            (5) <b>Find</b> <math>dist_{thd}</math> threshold as <math>dist_{thd} = sd_{index}[N_{patch}]</math>, <b>if</b> (<math>T_{SD} \leq N_{patch}</math>), <b>then</b> { <math>dist_{thd} = sd_{index}[T_{SD}]</math> }            (6) <b>For each</b> <math>t</math> <b>in</b> <math>T_{SD}</math> <b>do</b> {                <b>If</b> (<math>sd_{index}[t] \leq dist_{thd}</math>) <b>then</b> {                    <b>Let</b> 1st_best_so_far = infinity                    <b>Let</b> 2nd_best_so_far = infinity                    <b>Let</b> 3rd_best_so_far = infinity                    <b>For each</b> <math>t</math> <b>in</b> <math>T_{SD}</math> <b>do</b> {                        <math>dist_{part1} = distance_{DTW}(flat(E_{LR}), flat(sd_{LR}[t]))</math>                        <math>dist_{part2} = distance_{DTW}(E_{DIFF}, sd_{DIFF}[t])</math>                        <math>dist_{all} = dist_{part1} + dist_{part2}</math>                        <b>If</b> (<math>dist_{all} &lt; 1st\_best\_so\_far</math>) <b>then</b> {                            1st_best_so_far = <math>dist_{all}</math>                            <math>E_{1st} = sd_{HR}[t]</math> }                        <b>Else if</b> (<math>dist_{all} &lt; 2nd\_best\_so\_far</math>) <b>then</b> {                            2nd_best_so_far = <math>dist_{all}</math>                            <math>E_{2nd} = sd_{HR}[t]</math> }                        <b>Else if</b> (<math>dist_{all} &lt; 3rd\_best\_so\_far</math>) <b>then</b> {                            3rd_best_so_far = <math>dist_{all}</math>                            <math>E_{3rd} = sd_{HR}[t]</math> }                        <b>Else</b> {Continue}                      }                    }                } <b>Else</b> {Continue}            }            (7) <b>Return</b> <math>E_{1st}[L_s][L_s]</math>, <math>E_{2nd}[L_s][L_s]</math>, and <math>E_{3rd}[L_s][L_s]</math></p>

### 3. Implementation and Results

To exam the performance of the proposed precipitation downscaling method on the gridded precipitation product, we use the Integrated Multi-satellitE Retrievals for GPM (IMERG) precipitation estimates as the testing data source. The original IMERG data is aggregated to a lower resolution, to which the downscaling method is applied. Then, the results are compared and assessed with the original high-resolution data, as well as the output from other approaches such as bicubic, DeepSD, and RainFARM.

The downscaling method was implemented using Python and open-source scientific libraries, including NumPy, NetCDF, numba, and itertools. The proposed downscaling method is implemented as a batch of computation processes from dictionary construction to downscaled output production. The system accepts NetCDF and HDF files as input and output downscaled results in NetCDF formats. The current implementation method is not under any existing frameworks (to maximize implementation flexibility), the implementation of dictionary learning is based on programming each algorithm in individual programs, and could be treated as a complete downscaling system.

#### 3.1. Synthetic Experiments

A common problem in evaluating the downscaling method is the lack of ground truth data. Climate variables like precipitation are simulated differently across models and show different patterns and rainfall amounts. So, instead of downscaling a GCM output directly, synthetic experiments are often used to evaluate downscaling performance. Synthetic experiments are conducted through upscaling higher resolution rainfall products and use them as inputs and compare downscaled results to original products to evaluate overall performance [31]. The upscaling process is achieved through aggregate grids as averages from the higher resolution.

##### 3.1.1. Study Areas

The main study area is the Chesapeake Bay, which is an estuary in the Eastern U.S. Additional areas are also included for parallel comparison of the downscaling results. The specific areas are separated regions within California, Kansas, and Florida. Downscaling methods are used for all four regions, and more sets are generated for the Chesapeake Bay for long-term comparison. Figure 4 maps the geographical coverage of each study region on the U.S. map.

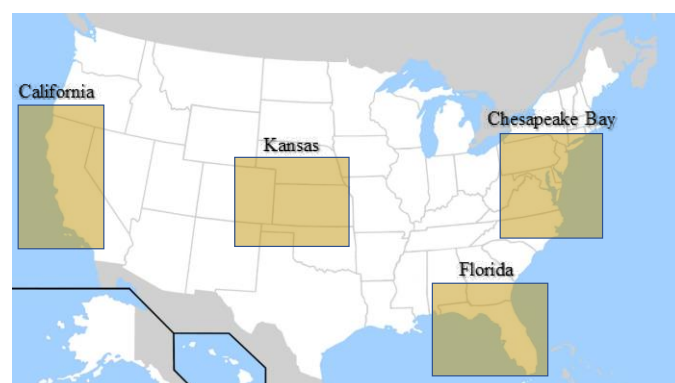


Figure 4. Study areas for downscaling studies.

##### 3.1.2. Dataset

IMERG dataset for the U.S. coverage is collected and selected as experimental data, which is generated by NASA's Precipitation Processing System every half hour with 4-hours (Early) to 3.5-months (Final) latency from acquisition time. IMERG V06 Final is chosen, which is a satellite-gauge product. The spatiotemporal resolution of the precipitation variable in IMERG is  $0.1^\circ$  with half-hour

reads for each day of the year. The year 2014 to 2018 are selected in the experiment, and 2014 to 2017 were used for any model or dictionary training process, while 2018 is used for method comparison in the production process.

### 3.1.3. Downscaling Methods

The methods used in downscaling case studies are chosen based on the ability to provide precipitation field as the final output and whether they require climate variables other than precipitation when producing downscaled results. Besides the statistical downscaling methods, bilinear interpolation (e.g., bicubic) is considered the baseline for spatial downscaling of precipitation fields [32]. Specifically, three methods are selected to evaluate besides the proposed method: (1) bicubic interpolation, (2) DeepSD, as an SRCNN based super-resolution downscaling method [36], and (3) RainFARM [28] as a stochastic downscaling method. Although there are other downscaling methods available in the research, most of them are not able to produce precipitation fields as output, require other datasets when downscaling, or they are limited to a specific area. Statistical methods like regression and PP methods are not chosen because they failed to produce continuous precipitation fields and require additional variables other than precipitation, which are not suitable for downscaling requirements in the previous section. Most of the statistical methods are involved using either observation data from the same time frame or other variables if precipitation field is desired as the downscaled result. Under this consideration, statistical methods are chosen.

## 3.2. Results

For each study area, two images from different time slices are selected to show a brief view of the general performance of the bicubic method, DeepSD, RainFARM, and PreciPatch. The original high-resolution IMERG data set is also visualized to compare with the downscaled results from different methods with a direct visual comparison. DeepSD is trained using the data from 2014 to 2017 at the same region, bicubic and RainFARM do not need a training process, and PreciPatch dictionary is trained using the whole U.S. coverage data from 2014 to 2017.

Figure 5 presents the downscaled results for different study areas. The input column in each figure contains the inputs for different downscaling methods and is used as the only input in downscale methods. The results from bicubic, DeepSD, RainFARM, and PreciPatch are in the next four columns, respectively. All the images from the same row share the same color scale. Original IMERG data are in the last column for visual comparison with different results. Generally, PreciPatch outperformed RainFARM, DeepSD, and bicubic, showing results closer to the ground truth in most cases. RainFARM method is a valid solution that meets downscaling requirements and can produce downscaling results with fast speed. However, due to the theoretical limitations of this method, which assumes rainfall fields as Gaussian random fields at all scale levels [27], the practical shapes and textures of rainfall are not sufficiently simulated by RainFARM. Bicubic is often treated as the baseline for downscaling, and the results from DeepSD are very close to those from bicubic. DeepSD is showing great potential for improvements.

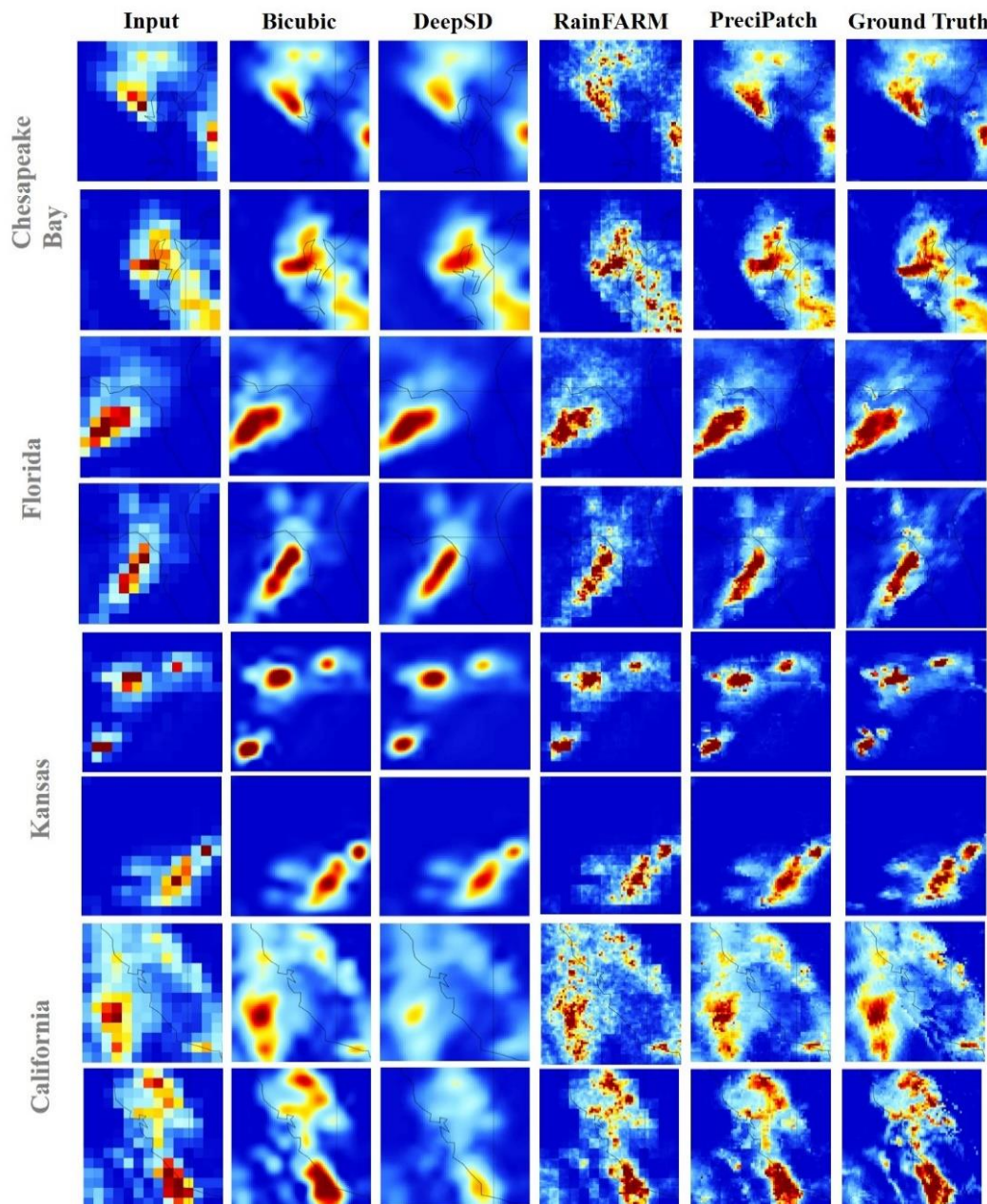
## 3.3. Validation and Evaluation

The downscaled results are compared with ground truth statistically.

Figures 6 and 7 show the direct comparison between the downscaled point values (predicted values) to the ground truth in the cases corresponding to the ones showing in Figure 5. Each plot represents a time slice. Statistically speaking, for  $R^2$ , bicubic and PreciPatch performs better than DeepSD and RainFARM in most cases. Six more criteria are used to evaluate the performance between different methods. A total of 2208 time slices (hourly precipitation from 2018 June, July, and August) are used from all four study regions to generate results. The statistical comparison is made through a grid-to-grid comparison between the predicted values from each method to the ground truth at each time slice. The detailed statistical comparison is shown in Table 9; the values are averaged from all

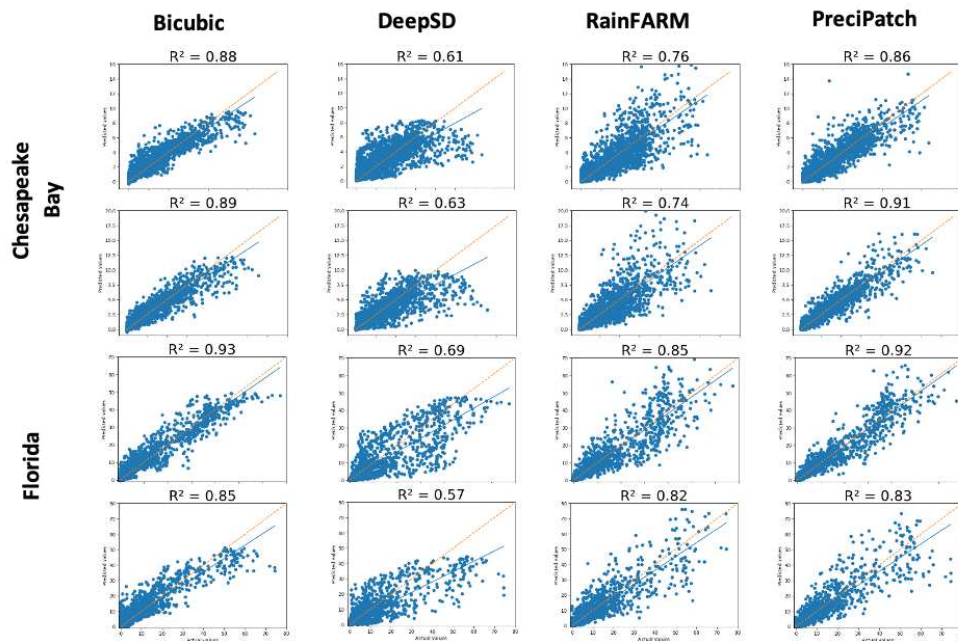


time slices, where PreciPatch shows the best statistical accuracies in almost all aspects, followed by RainFARM and bicubic. DeepSD has comparatively low performance in this case study, which agrees with the results from the pre-test and visual comparison. PreciPatch outperforms RainFARM, bicubic, and DeepSD in this synthetic experiment regarding statistical accuracy.

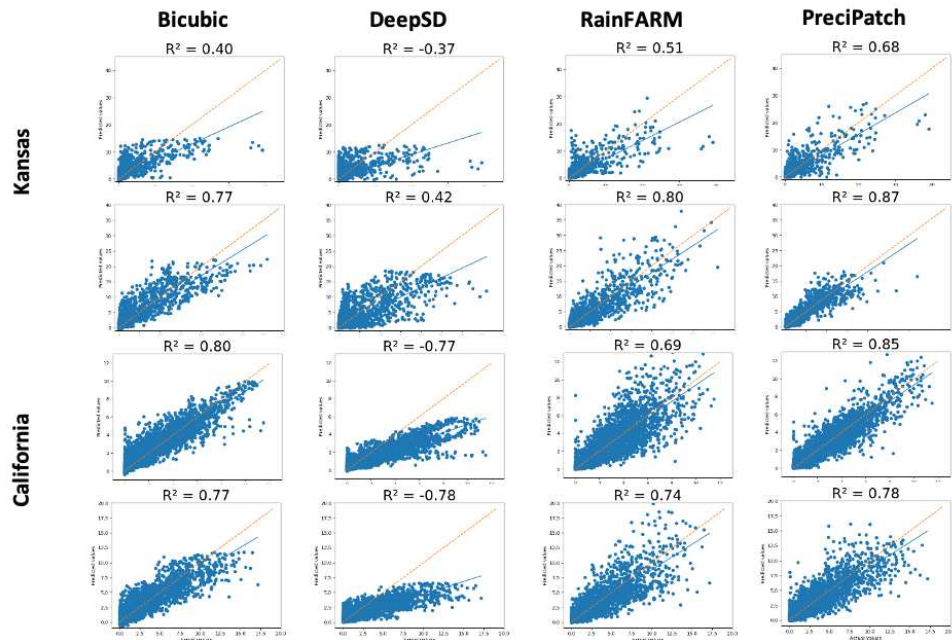


**Figure 5.** Downscaling results. The x-axis refers to the longitude, and the y-axis refers to the latitude. Each row is the input aggregated Integrated Multi-satellite Retrievals for GPM (IMERG) rain rate, Bicubic result, DeepSD result, RainFARM result, PreciPatch result and original high-resolution IMERG rain rate respectively for each location. 1st row: hourly precipitation at Chesapeake Bay area on 2018/7/21 (23:30:00 UTC), 2nd row: hourly precipitation at Chesapeake Bay area on 2018/7/21 (16:30:00 UTC), 3rd row: hourly precipitation at Florida area on 2018/12/14 (05:30:00 UTC), 4th row: hourly precipitation at Florida area on 2018/12/14 (18:30:00 UTC), 5th row: hourly precipitation at Kansas area on 2018/7/29 (04:30:00 UTC), 6th row: hourly precipitation at Kansas area on 2018/7/29 (08:30:00 UTC), 7th row: hourly precipitation at California area on 2018/1/9 (00:30:00 UTC), 8th row: hourly precipitation at California area on 2018/1/9 (08:30:00 UTC).





**Figure 6.** The Chesapeake Bay and Florida— $R^2$ . The x-axis refers to the ground truth, and the y-axis is the prediction. The 45 degrees line in each graph is the perfect match reference line between the prediction and the ground truth, showing as the dashed line, the solid line is the trending line. 1st row: hourly precipitation at Chesapeake Bay area on 2018/7/21 (23:30:00 UTC), 2nd row: hourly precipitation at Chesapeake Bay area on 2018/7/21 (16:30:00 UTC), 3rd row: hourly precipitation at Florida area on 2018/12/14 (05:30:00 UTC), 4th row: hourly precipitation at Florida area on 2018/12/14 (18:30:00 UTC).



**Figure 7.** Kansas and California— $R^2$ . The x-axis refers to the ground truth, and the y-axis is the prediction. The 45 degrees line in each graph is the perfect match reference line between the prediction and the ground truth, showing as the dashed line, the solid line is the trending line. 1st row: hourly precipitation at Kansas area on 2018/7/29 (04:30:00 UTC), 2nd row: hourly precipitation at Kansas area on 2018/7/29 (08:30:00 UTC), 3rd row: hourly precipitation at California area on 2018/1/9 (00:30:00 UTC), 4th row: hourly precipitation at California area on 2018/1/9 (08:30:00 UTC).

**Table 9.** Statistical comparison.

mm/hour	Bicubic	DeepSD	RainFARM	PreciPatch
<b>Average Bias per grid</b>	0.03	0.12	$4.3 \times 10^{-9}$	0.003
<b>Average R<sup>2</sup></b>	0.18	-0.16	0.29	0.42
<b>Average (+) R<sup>2</sup></b>	0.44	0.3	0.47	0.57
<b>RMSE</b>	0.45	0.61	0.49	0.42
<b>Mean Absolute Error</b>	0.13	0.28	0.13	0.1
<b>Maximum Absolute Error</b>	7.75	7.95	4.2	3.87
<b>STDV Absolute Error</b>	0.2	0.27	0.09	0.07

In addition to the visual and statistical comparison between methods, aggregation of the downscaled results was implemented to test the downscaling method's ability to hold precipitation consistency from large-scale to small-scale. As designed with constrains for both RainFARM and PreciPatch, the downscaled results from these two methods could be upscaled to the original coarse-resolution input with visually unnoticeable variations. However, the downscaled results from bicubic interpolation and DeepSD methods could not be aggregated back to low-resolution inputs, which means such downscaling approaches represent a biased transformation from large-scale data to small-scale outputs.

### 3.4. Discussion

Major differences between our method and previous precipitation downscaling research include:

- Full-field generation instead of area limited generation or fixed size.
- Precipitation fields as target production instead of station networks (e.g., single or multiple), providing continuous information in 2D space.
- Precipitation field distribution simulated closer to precipitation observation
- Data independence in production (available for downscaling future predictions).
- Hourly precipitation downscaling to precipitation fields (downscaling short-duration precipitation events) instead of a daily average or seasonal downscaling used in other research.
- Using the prior knowledge learned in one data set and successfully applied to another related data set.
- The rainfall amount is consistent with large-scale vs. rainfall inconsistency in others.

Major differences between PreciPatch and other downscaling approaches regarding methodology includes: (1) comparing to statistical downscaling method: there is no need for other climate variables in the downscaling process, no regression process or probability assumption process, (2) comparing to stochastic downscaling method: the proposed method is making use of prior knowledge, not purely hypothesis and mathematical transformation based, (3) comparing to SRCNN based methods like DeepSD: when interpolation process is involved, the final output is constrained by coarse-resolution inputs, which is equal to an unbiased disaggregation of precipitation amounting to space, (4) comparing to traditional dictionary learning: one master dictionary is constructed instead of two in traditional approaches, the proposed method uses dynamic patches to recover the high-resolution image, while the traditional approach uses stored fixed patches, DTW distance is used instead of MSE or Euclidean distance, fuzzy search is utilized in the new method instead of exact search for similarity patches, and rainfall amount is consistent in the downscaled result, 5) comparing to Analog Method (AM): similarity searching is executed spatially with a fuzzy search instead of linearly with a direct search.

In this research, there are also some limitations to PreciPatch. For example, the computing time is longer than other downscaling methods, which makes it hard to apply to large datasets. The detailed comparison is in Table 10.

**Table 10.** Comparison of tested downscaling method.

	Bicubic	DeepSD	RainFARM	PreciPatch
<b>Training time</b>	N/A	~1 hour for 3 months of data (national coverage)	N/A	~200 hours for 300 Million patches (national coverage)
<b>Applicability to different areas?</b>	Direct apply, no training is needed	Additional training is needed if the area is different or the input size is different	Direct apply, no training is needed	Direct apply, no need for additional training
<b>Computing time</b>	< 0.001 sec per grid point	~0.005 sec per grid point	~0.005 sec per grid point	~2 sec per grid point per process
<b>Overall downscaling performance</b>	Baseline	Not good—close to the baseline	Rainfall clusters are not well simulated	Best
<b>Holds rainfall consistency with large-scale input?</b>	No	No	Yes	Yes

However, this limitation could be resolved by including an additional index layer for the dictionary and reprogramming the system using more efficient languages such as C/C++. Another limitation for the proposed method is its assumption: future precipitation would behave similarly as its past, relationships developed for present climate also hold for possible future climates [39] (the same assumption is used in all statistical downscaling methods). One advantage compares with other models like DeepSD is that PreciPatch does not need additional training when applying to different areas, even for regions outside of the training area.

#### 4. Conclusions

The importance of precipitation downscaling is well recognized in hydrological impact studies and local climate change-related domains. However, few statistical downscaling methods can provide universally applicable downscaling solutions for arbitrary GCM outputs, large amounts of precipitation downscaling related studies are area limited and highly additional data-dependent, not applicable for random locations. Meanwhile, major research can only provide the downscaled result on a level of single or multiple stations. Providing precipitation fields as outputs are still beyond the reach for most traditional methods. Recent developments in the stochastic downscaling method and the SR based method brought new opportunities to enhance downscaling studies. A new precipitation downscaling method based on dynamic dictionary learning is proposed with detailed algorithms and procedures. Four precipitation downscaling study cases are conducted to evaluate the performance of the proposed method and compare it with other downscaling methods (RainFARM and DeepSD), along with the comparison baseline, the bicubic interpolation method. Results from case studies have shown positive feedback on the proposed downscaling method, demonstrating good quality for simulating spatially distributed precipitation fields and yielding the best performance among all other tested methods through visual inspection and quantitative analyses. It is hard for traditional and machine learning super-resolution based methods to exclude the interpolation process and include the non-linear characteristics of precipitation event into the method design [41]. Meanwhile, many downscaling methods like bicubic interpolation and DeepSD do not have extra constraints to enforce the final outputs to hold rainfall consistency with the initial inputs. The proposed method interprets the downscaling task as a constrained single image super-resolution problem, different from traditional approaches and other downscaling methods, it addresses the precipitation downscaling issue using a novel approach which excludes the interpolation process and is able to generate non-linear spatial patterns that are closer to rainfall observations. Additionally, rainfall consistency is ensured by adding extra constraints in the proposed method. As a result, PreciPatch achieved better results than other downscaling methods in precipitation downscaling study cases because it takes the spatial

characteristics of precipitation into the method design consideration and optimizes the method to fit precisely for the precipitation downscaling scenario.

Precipitation field downscaling is a challenging task, and related research topics have been explored by many studies. The proposed precipitation downscaling method, PreciPatch has shown its ability to spatially downscale short-duration precipitation events to precipitation fields and can be applied to downscale future estimations from GCMs or other gridded data sources. The bias correction is not addressed by PreciPatch, and the bias from GCMs are preserved at the same spatial scale as the input. This research contributes to hydrological impact studies and local climate change studies by providing a precipitation downscaling method and associated algorithms for simulating precipitation fields in local scale and being consistent with large-scale information, in detail:

- This method does not require additional data as predictors to produce downscaled results, and it could be used for downscaling future estimations from GCMs.
- This downscaling method has the potential to aid climate models like WRF by providing higher resolution inputs.
- The downscaled results from this method could be used to force mountain glacier models for local impact studies, where the complete absence of climate monitoring activities within the regions of interest presents a data challenge.

For future work, more (spatiotemporal) indexing techniques [42] will be integrated to accelerate the similarity search process in PreciPatch. PreciPatch will be converted and rebuilt on top of a machine learning framework to generalize the dictionary into a model, accelerate the downscaling process, and reduce storage requirement. More observation data will be used to evaluate the downscaling method regarding expandability and transferability.

**Author Contributions:** C.Y., D.Q.D., W.M.P., M.C., and T.L. came up with the original research idea; C.Y. advised M.X., Q.L., D.S., and M.Y. on the experiment design and paper structure; M.X., Q.L., and D.S. designed the experiments, developed the scripts for experiments, conducted the experiments, and analyzed the experiment results; M.X. developed the PreciPatch method and algorithms; Q.L. conducted the RainFARM experiment; D.S. conducted the DeepSD experiment; M.X., Q.L., and D.S. wrote the paper; C.Y. and M.Y. revised the paper. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by NSF (1841520) and NASA Goddard (NNG14HH38I).

**Acknowledgments:** NASA Goddard Space Flight center provides the IMERG data. Kyla Carte helped proofread the manuscript.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Sheehan, M.C.; Fox, M.A.; Kaye, C.; Resnick, B. Integrating health into local climate response: Lessons from the US CDC Climate-Ready States and cities initiative. *Environ. Health Perspect.* **2017**, *125*, 094501. [CrossRef]
2. Fatichi, S.; Ivanov, V.Y.; Caporali, E. Assessment of a stochastic downscaling methodology in generating an ensemble of hourly future climate time series. *Clim. Dyn.* **2013**, *40*, 1841–1861. [CrossRef]
3. Yang, C.; Yu, M.; Li, Y.; Hu, F.; Jiang, Y.; Liu, Q.; Sha, D.; Xu, M.; Gu, J. Big Earth data analytics: A survey. *Big Earth Data* **2019**, *3*, 83–107. [CrossRef]
4. Climate.gov, n.d. Climate Models|NOAA Climate.gov. Available online: <https://www.climate.gov/maps-data/primer/climate-models> (accessed on 24 August 2019).
5. Benestad, R. Downscaling Climate Information. Oxford Research Encyclopedia of Climate Science. 2016. Available online: <https://oxfordre.com/climatescience/view/10.1093/acrefore/9780190228620.001.0001/acrefore-9780190228620-e-27> (accessed on 24 August 2019).
6. Giorgi, F.; Mearns, L.O. Approaches to regional climate change simulation: A review. *Rev. Geophys* **1991**, *29*, 1–2. [CrossRef]
7. Arritt, R.W.; Rummukainen, M. Challenges in regional-scale climate modeling. *Bull. Am. Meteorol. Soc.* **2011**, *92*, 365–368. [CrossRef]

8. Walton, D.B.; Sun, F.; Hall, A.; Capps, S. A hybrid dynamical–statistical downscaling technique. Part I: Development and validation of the technique. *J. Clim.* **2015**, *28*, 4597–4617. [[CrossRef](#)]
9. Wilby, R.L. Statistical downscaling of daily precipitation using daily airflow and seasonal teleconnection indices. *Clim. Res.* **1998**, *10*, 163–178. [[CrossRef](#)]
10. Maraun, D.; Wetterhall, F.; Ireson, A.M.; Chandler, R.E.; Kendon, E.J.; Widmann, M.; Brienen, S.; Rust, H.W.; Sauter, T.; Themeßl, M.; et al. Precipitation downscaling under climate change: Recent developments to bridge the gap between dynamical models and the end user. *Rev. Geophys.* **2010**, *48*. [[CrossRef](#)]
11. Xu, C.Y. Climate change and hydrologic models: A review of existing gaps and recent research developments. *Water Resour. Manag.* **1999**, *13*, 369–382. [[CrossRef](#)]
12. Bronstert, A.; Kolokotronis, V.; Schwandt, D.; Straub, H. Comparison and evaluation of regional climate scenarios for hydrological impact analysis: General scheme and application example. *Int. J. Climatol. A J. R. Meteorol. Soc.* **2007**, *27*, 1579–1594. [[CrossRef](#)]
13. Rau, M.; He, Y.; Goodess, C.; Bárdossy, A. Statistical downscaling to project extreme hourly precipitation over the United Kingdom. *Int. J. Climatol.* **2019**, *40*, 1805–1823. [[CrossRef](#)]
14. Kundzewicz, Z.W.; Mata, L.J.; Arnell, N.W.; Doll, P.; Kabat, P.; Jimenez, B.; Miller, K.; Oki, T.; Zekai, S.; Shiklomanov, I. *Freshwater Resources and Their Management*; Parry, M.L., Canziani, O.F., Palutikof, J.P., van der Linden, P.J., Hanson, C.E., Eds.; Climate Change 2007: Impacts, Adaptation and Vulnerability. Contribution of Working Group II to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change; Cambridge University Press: Cambridge, UK, 2007; pp. 173–210. ISBN 9780521880091.
15. Vrac, M.; Naveau, P. Stochastic downscaling of precipitation: From dry events to heavy rainfalls. *Water Resour. Res.* **2007**, *43*, 7. [[CrossRef](#)]
16. Vandal, T.J. Statistical Downscaling of Global Climate Models with Image Super-resolution and Uncertainty Quantification. Ph.D Thesis, Northeastern University, Boston, MA, USA, 2018.
17. Lenderink, G.; Van Meijgaard, E. Increase in hourly precipitation extremes beyond expectations from temperature changes. *Nat. Geosci.* **2008**, *1*, 511. [[CrossRef](#)]
18. Wilks, D.S. Use of stochastic weather generators for precipitation downscaling. *Wiley Interdiscip. Rev. Clim. Chang.* **2010**, *1*, 898–907.
19. Wilby, R.L.; Wigley, T.M.L. Downscaling general circulation model output: A review of methods and limitations. *Prog. Phys. Geogr.* **1997**, *21*, 530–548. [[CrossRef](#)]
20. Wilby, R.L.; Charles, S.P.; Zorita, E.; Timbal, B.; Whetton, P.; Mearns, L.O. Guidelines for use of climate scenarios developed from statistical downscaling methods. In *Supporting Material of the Intergovernmental Panel on Climate Change*; DDC of IPCC TGCI: Geneva, Switzerland, 2004.
21. Fowler, H.J.; Blenkinsop, S.; Tebaldi, C. Linking climate change modelling to impacts studies: Recent advances in downscaling techniques for hydrological modelling. *Int. J. Climatol. A J. R. Meteorol. Soc.* **2007**, *27*, 1547–1578. [[CrossRef](#)]
22. Tang, J.; Niu, X.; Wang, S.; Gao, H.; Wang, X.; Wu, J. Statistical downscaling and dynamical downscaling of regional climate in China: Present climate evaluations and future climate projections. *J. Geophys. Res. Atmos.* **2016**, *121*, 2110–2129. [[CrossRef](#)]
23. Wang, Y.Q.; Leung, L.R.; McGregor, J.L.; Lee, D.K.; Wang, W.C.; Ding, Y.H.; Kimura, F. Regional climate modeling: Progress, challenges, and prospects. *J. Meteorol. Soc. Jpn.* **2004**, *82*, 1599–1628. [[CrossRef](#)]
24. Chen, J.; Brissette, F.P.; Leconte, R. Coupling statistical and dynamical methods for spatial downscaling of precipitation. *Clim. Chang.* **2012**, *114*, 509–526. [[CrossRef](#)]
25. Jarosch, A.H.; Anslow, F.S.; Clarke, G.K. High-resolution precipitation and temperature downscaling for glacier models. *Clim. Dyn.* **2012**, *38*, 391–409. [[CrossRef](#)]
26. Chen, J.; Chen, H.; Guo, S. Multi-site precipitation downscaling using a stochastic weather generator. *Clim. Dyn.* **2018**, *50*, 1975–1992. [[CrossRef](#)]
27. Liu, Q.; Li, Y.; Yu, M.; Chiu, L.S.; Hao, X.; Duffy, D.Q.; Yang, C. Daytime Rainy Cloud Detection and Convective Precipitation Delineation Based on a Deep Neural Network Method Using GOES-R ABI Images. *Remote Sens.* **2019**, *11*, 2555. [[CrossRef](#)]
28. Rebola, N.; Ferraris, L.; von Hardenberg, J.; Provenzale, A. RainFARM: Rainfall downscaling by a filtered autoregressive model. *J. Hydrometeorol.* **2006**, *7*, 724–738. [[CrossRef](#)]



29. Posadas, A.; Duffaut Espinosa, L.A.; Yarlequé, C.; Carbajal, M.; Heidinger, H.; Carvalho, L.; Jones, C.; Quiroz, R. Spatial random downscaling of rainfall signals in Andean heterogeneous terrain. *Nonlinear Process. Geophys.* **2015**, *22*, 383–402. [[CrossRef](#)]
30. Terzago, S.; Palazzi, E.; Hardenberg, J.V. Stochastic downscaling of precipitation in complex orography: A simple method to reproduce a realistic fine-scale climatology. *Nat. Hazards Earth Syst. Sci.* **2018**, *18*, 2825–2840. [[CrossRef](#)]
31. He, X.; Chaney, N.W.; Schleiss, M.; Sheffield, J. Spatial downscaling of precipitation using adaptable random forests. *Water Resour. Res.* **2016**, *52*, 8217–8237. [[CrossRef](#)]
32. Dong, C.; Loy, C.C.; He, K.; Tang, X. Learning a deep convolutional network for image super-resolution. In Proceedings of the European conference on computer vision, Zurich, Switzerland, 6–12 September 2014; pp. 184–199.
33. Kim, J.; Kwon Lee, J.; Mu Lee, K. Accurate image super-resolution using very deep convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1646–1654.
34. Zhang, Y.; Li, K.; Li, K.; Wang, L.; Zhong, B.; Fu, Y. Image super-resolution using very deep residual channel attention networks. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 286–301.
35. Ebtehaj, A.M.; Foufoula-Georgiou, E.; Lerman, G. Sparse regularization for precipitation downscaling. *J. Geophys. Res. Atmos.* **2012**, *117*. [[CrossRef](#)]
36. Vandal, T.; Kodra, E.; Ganguly, S.; Michaelis, A.; Nemani, R.; Ganguly, R.A. DeepSD: Generating high resolution climate change projections through single image super-resolution. In Proceedings of the 23rd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining, ACM, Halifax, NS, Canada, 13–17 August 2017; pp. 1663–1672.
37. Freeman, W.T.; Jones, T.R.; Pasztor, E.C. Example-based super-resolution. *IEEE Comput. Graph. Appl.* **2002**, *22*, 56–65. [[CrossRef](#)]
38. Itakura, F. Minimum prediction residual principle applied to speech recognition. *IEEE Trans. Acoust. Speech Signal Process.* **1975**, *23*, 67–72. [[CrossRef](#)]
39. Keogh, E.; Ratanamahatana, C.A. Exact indexing of dynamic time warping. *Knowl. Inf. Syst.* **2005**, *7*, 358–386. [[CrossRef](#)]
40. Tan, C.W.; Webb, G.I.; Petitjean, F. Indexing and classifying gigabytes of time series under time warping. In Proceedings of the 2017 SIAM international Conference on Data Mining. Society for Industrial and Applied Mathematics, Houston, TX, USA, 27–29 April 2017; pp. 282–290.
41. Yu, M.; Bambacus, M.; Cervone, G.; Clarke, K.; Duffy, D.; Huang, Q.; Li, J.; Li, W.; Li, Z.; Liu, Q.; et al. Spatiotemporal event detection: A review. *Int. J. Digit. Earth* **2020**. [[CrossRef](#)]
42. Yang, C.; Clarke, K.; Shekhar, S.; Tao, C.V. Big Spatiotemporal Data Analytics: A research and innovation frontier. *Int. J. Geogr. Inf. Sci.* **2019**, 1–14. [[CrossRef](#)]

