Climate Downscaling Using YNet: A Deep Convolutional Network with Skip Connections and Fusion

Yumin Liu
Northeastern University
Boston, MA, USA
yuminliu@ece.neu.edu

Auroop R. Ganguly
Northeastern University
Boston, MA, USA
a.ganguly@neu.edu

Jennifer Dy
Northeastern University
Boston, MA, USA
jdy@ece.neu.edu

ABSTRACT

Climate change is one of the major challenges to human beings in our time. It brings many unexpected disasters which cause drastic losses including lives and properties. To better understand climate change, scientists developed various Global Climate Models (GCMs) to simulate the global climate and make projections for future climate values. These global climate models have coarse grids (i.e., low resolutions both in space and time) due to limitations of computing power and simulation time. Although they are helpful in predicting large scale long term trend in climate, they are too coarse for impact analysis in smaller scales such as in regional or local scale. However, climate conditions in regional or local scale are very important in making decisions related to climate conditions such as infrastructure, transportation and evacuation, as they highly depend on small scale climate conditions. In this paper, we proposed YNet, a novel deep convolutional neural network (CNN) with skip connections and fusion capabilities to perform downscaling for climate variables, on multiple GCMs directly rather than on reanalysis data. We analyzed and compared our proposed method with four other methods on datasets of three climate variables: mean precipitation, and extreme values (maximum temperature and minimum temperature). The results show the effectiveness of the proposed method.

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**; • **Applied computing** → **Environmental sciences**.

KEYWORDS

Deep Neural Networks; Climate Downscaling; Super Resolution

ACM Reference Format:
Yumin Liu, Auroop R. Ganguly, and Jennifer Dy. 2020. Climate Downscaling Using YNet: A Deep Convolutional Network with Skip Connections and Fusion. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining USB Stick (KDD '20), August 23–27, 2020, Virtual Event, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3394486.3403366

KDD '20, August 23–27, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-7998-4/20/08…$15.00
https://doi.org/10.1145/3394486.3403366

1 INTRODUCTION

Climate change has growing impact on human society as global warming brings more unexpected extreme disasters such as widely-spread wild fires, severe droughts, frequent floods and extreme heat waves, and cause huge loss of lives and properties. To better understand the future climate and be better prepared, scientists developed various Global Climate Models (GCMs) to simulate the global climate and make projections for future climate values. GCMs use mathematical equations based on physical processes to describe the interactions of energy and materials among ocean, atmosphere and land. They are sophisticated systems with different boundary conditions and initial conditions and are very computationally expensive to run. GCMs usually have a coarse spatial resolution ($\geq 1^\circ$ or $100km$) due to limited computational costs.

While GCMs are able to capture the long term climate trends in large scales such as global scale or continental scale, they are not accurate enough in smaller scales such as regional scale or local scale [10]. However, it is crucial to access the climate impacts in a finer scale as many human activities and climate impacts are in smaller scales [27, 30]. To this end, people come up with approaches called "downscaling" to derive finer scale climate information from larger scale climate information such as GCMs. There are mainly two types of downscaling methods: dynamical downscaling and statistical downscaling. Dynamical downscaling utilize regional climate models (RCMs) which are similar to GCMs but have higher resolution and smaller covered area than that of GCMs. Dynamical downscaling methods are useful in simulating extreme values but also require extensive computation and need expertise to explain. Statistical downscaling methods try to find out the empirical relationships between the lower resolution GCM output with the higher resolution local climate variables (usually produced from historical observations) and applied them to GCMs to get higher resolution predictions. Statistical downscaling methods have the advantage of requiring less computing resources.

Statistical downscaling is very similar to image super-resolution (SR) in the computer vision field, which enjoys a rapid progress and great success as the fast development of machine learning especially convolutional neural network based deep learning methods. They both aim at getting higher resolution images from lower resolution images if we regard climate variable data as images. However, compared with the rapid advancing image super-resolution approaches using machine learning / deep learning methods, there are relatively fewer climate downscaling approaches utilizing these new techniques. For these reasons, we want to leverage the advanced super-resolution techniques in climate downscaling to bridge the gap and inspire new insights for climate science.

However, we should also highlight some differences between climate downscaling and image super-resolution. Firstly, an image usually has one or three channels and the pixel values are bounded (e.g., $0-255$) with the same range, but climate variables are not real images, and there may be multiple channels consisting of different climate variables with different value ranges and more skewed distributions. The values for climate variables are unbounded real values which may have no theoretical maximum / minimum values, which may affect the estimation of extreme values.

Secondly, in image super-resolution, the low-resolution input images and high-resolution target images come from the same sources (usually the high-resolution target images are downsampled / interpolated to low-resolution images as the inputs), while in climate downscaling, the inputs and outputs come from different sources. For example, in statistical downscaling, the inputs are the simulated data from GCMs (and may contain other auxiliary variables, as explained later) and the target outputs are the historical observational data. This makes it harder for downscaling as it requires an additional "regression" from input variables to output variables in addition to the regular function of increasing resolution.

Thirdly, climate downscaling can use additional auxiliary variables to help improve the performance. Different climate variables may be related to each other subject to constraints such as physical processes. One example is that elevation would affect temperature: higher elevation, lower temperature. Topography will affect precipitation [23] (e.g., mountains may facilitate or prevent rainfall depending on the locations and heights) and therefore topographical information can be helpful in climate downscaling.

In addition, climate variables often exhibit periodic property in temporal aspect and thus the climatological information (i.e., long term average) can also help to predict climate variables. Since climatological information and topographical information can be acquired in high resolution and remain roughly unchanged in a relatively long time (e.g., years), they are very reliable and useful information. Therefore we would like to take advantage of both the image super-resolution techniques and these useful characteristics for climate downscaling.

In this paper, we adopt an image super-resolution method to climate downscaling. The main contributions are as follows:

- We propose a neural network architecture that utilizes effective image super-resolution techniques and at the same time incorporates advantageous climate characteristics for climate downscaling. The proposed architecture, we call *YNet*, consists of an encoder-decoder-like architecture with residual learning through skip connections and fusion layers to enable the incorporation of topological and climatological data as auxiliary data. This architecture is effective in producing high-resolution outputs.
- To our knowledge, this work is the first time a systematic deep learning approach is being tried to actually use the full blown GCM simulations. The previous climate downscaling methods using machine learning do not use GCMs simulated data as predictors, instead they first upscaled observational / reanalysis data and then use them as predictors, which makes it a less challenging and less interesting problem.
- We examine the performance of the proposed method on downscaling three climate variables at three different downscaling factors (the downscaling factor means how many times the resolution increases) and compare the results with four state-of-the-art methods in climate downscaling and image super-resolution. Our proposed method significantly outperforms these methods.

2 RELATED WORK

2.1 Image Super-Resolution

Convolutional neural networks have shown great success in dealing with images, including image restoration, image classification, image segmentation, image style transformation, image super-resolution and so on. Deep learning methods based on neural networks quickly dominate the image super-resolution field since the seminal work by Dong et al. [5]. Different deep learning methods have been proposed for image super-resolution [1, 7, 8, 11, 11–15, 21, 24, 31, 32, 36]. Many of these methods take advantage of one or more techniques proven to be effective in improving the performance of image super-resolution. The first technique is that almost all methods are based on convolutional neural networks, and most of them use small size (3 or 5) kernels for the networks. The effective receptive field of multiple small kernels is equivalent to that of a larger kernel but the former architecture requires less parameters and is more flexible. The second technique is the skip connections between earlier and later network layers. Most of the skip connections are additive to corresponding channels but a few are attached as extra channels. Skip connections help prevent gradients from vanishing in backpropagation during training and in addition transform the problem from directly mapping input to output to residual mapping, which is more effective. Some methods also utilize encoder-decoder network or generative adversarial network (GAN) architecture.

2.2 Downscaling

Downscaling methods have long been applied in climate science [27, 33] and researchers in climate science proposed different dynamical or statistical downscaling methods. Wood et al. [33] proposed a statistical downscaling method called bias-corrected spatial disaggregation (BCSD) to downscale precipitation and temperature from climate model outputs. Zobel et al. [37] used ensemble of dynamical downscaled model simulated data to study the extreme temperatures over CONUS in the mid-late 21st century. Tanaka et al. [25] proposed a method to downscale spatial data using auxiliary data. Xu et al. [35] compared various dynamical downscaling methods and found out that in general bias correction technique help improve results. Recently Xu et al. [34] proposed dictionary learning-based method to downscale short-duration precipitation.

Although there are many downscaling methods proposed in climate science [2, 10], there are very few downscaling methods utilizing advanced computer vision / machine learning methods especially deep learning methods [17, 28, 29]. One of the first works was by Vandal et al. [29], who proposed a neural network based method called DeepSD which stacked multiple SRCNN [5] together to do the downscaling. But as the first attempt to match climate downscaling to image super-resolution, they oversimplified the

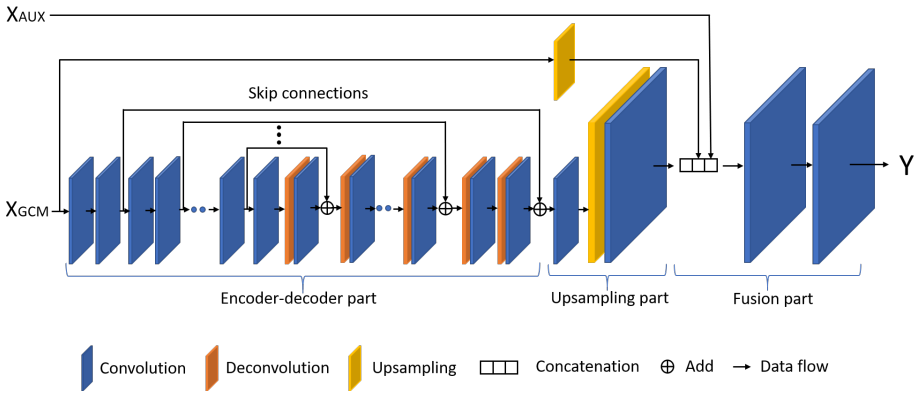


Figure 1: Proposed model architecture, YNet. Every convolutional layer (except the last one) and deconvolutional layer is followed by a ReLU activation which is not shown in the diagram. See context for more details.

problem by taking the upscaled observation rather than GCMs data as the input. In addition, their model lacked flexibility in adjusting downscaling factor due to the stacking architecture and failed to utilize more effective architecture such as skip connections and encoder-decoder scheme. Our proposed method improves over DeepSD in those aspects and outperforms it significantly. In another paper [28] they also compared several methods including ordinary least square, elastic net, support vector machine (SVM), BCSD and autoencoder neural networks for downscaling daily extreme precipitation in the Northeastern United States and found BCSD still have merits compared to other methods. Recently Miao et al. [16] proposed a CNN combined LSTM neural network architecture to downscale monsoon precipitation, but they also used reanalysis data and not GCM data as the predictors. Tran Anh et al. [26] claimed to use LSTM and feedforward network to downscale GCMs to observation, but they focused on only two specific locations instead of general grid downscaling. Rodrigues et al. [19] proposed a neural network architecture to downscale weather forecasts.

3 METHOD

In this section we will first introduce two image super-resolution methods closely related to our proposed YNet method and then describe the YNet.

3.1 Related Methods

Our proposed model is inspired mainly by two methods for image super-resolution: Residual encoder-decoder network (REDNet) [15] and efficient sub-pixel convolutional neural network (ESPCN) [21]. REDNet consists mainly of 15 symmetric convolutional layers and 15 deconvolutional layers with skip connections between corresponding convolutional and deconvolutional layers. The convolutional layers serve as encoders while the deconvolutional layers serve as decoders. The encoder-decoder scheme has been used to extract essential information, eliminate noise and corruption and reconstruct the original image. The scheme has been proven effective [20]. The skip connections have several benefits. They can convey detailed feature information in the convolutional layers to the corresponding deconvolutional layers in order to better recover features. With skip connections, the encoder-decoder learns to fit the residual of decoder output and encoder input instead of directly learning the mapping from encoder input to decoder output, which is more difficult [6]. During training, skip connections help backpropagate gradients to the previous layers and make training the network easier and more effective. Due to the symmetric architecture, REDNet requires input and output images to have the same size and the same number of channels. In order to perform image super-resolution, the input low-resolution image must be upsampled to the same size with the desired high-resolution output before being fed into the network.

Instead of using deconvolutional layers to increase the image resolution, the authors of ESPCN [21] use a so called sub-pixel convolutional layer to increase resolution and they show that sub-pixel convolution is essentially equivalent to deconvolution [22]. The original ESPCN model consists of 3 convolutional layers and a final sub-pixel convolutional layer. The convolutional layers operate on the lower resolution images and serve as a feature extractor. If the output image has C channels and the upsampling factor of the model is r, then the input feature map for the sub-pixel convolutional layer will have r^2C channels. The sub-pixel convolutional layer increases the resolution by interlacing the r^2C-channel feature map into a C-channel high-resolution output image. One advantage of this model is that it is computational and memory complexity efficient as most of its layers operate in the low-resolution space. In addition, the input image can be arbitrary in size and it is very flexible to adjust to different scaling factors. However, as Odena et al. [18] pointed out, the deconvolution (as well as sub-pixel convolution) operation may introduce an undesired pattern called "checkerboard artifacts" which will degrade the quality of the output image or even ruin it with serious artifact corruption. In fact, we did find severe checkerboard artifacts when applying the sub-pixel convolution to the climate datasets we used (this will be explained in the experiment section).

3.2 Proposed Method, YNet

We name our model "YNet" since the architecture looks like a lying down "Y" shape. Our model consists of several two-dimensional

convolutional/deconvolutional layers, an upsampling layer and a concatenation layer. The model architecture is shown in Figure 1. The low-resolution input image X_{GCM} can be arbitrary size while high-resolution $\boldsymbol{X}_{AUX}$ and $\boldsymbol{Y}$ have the same size. Each convolutional/deconvolutional layer has a kernel of size 3×3 with stride 1, padding 1 and ReLU activation except the last convolutional layer whose kernel size is 1×1 without padding nor ReLU. The model can be divided into three parts: the first part is a quasisymmetric structure of encoder-decoder with skip connections that operates on low-resolution space. It consists of $2M_1$ convolutional layers and M_1 deconvolutional layers, each with D_1 feature depth. The skip connections connect every L convolutional layers to the corresponding deconvolutional layers. In the experiment we set $M_1 = 15$, $D_1 = 64$ and $L = 2$ following [15]. This is similar to REDNet taking the advantage of an encoder-decoder network and the skip connections. It can be regarded as a feature extractor taking as input "noisy" low-resolution images and get "cleaner" low-resolution features. To alleviate the possible checkerboard artifacts introduced by deconvolution, we add a convolutional layer after every deconvolutional layer. The second part is an upsampling part consists of two convolutional layers with feature depth D_2 the same as the input image channel and an upsampling layer with upsampling scale S. In our experiment $D_2 = 35, 33$ and $S = 2, 4, 8$, respectively. To address the checkerboard problem, we use a bilinear interpolation upsampling followed by a convolutional layer to replace the sub-pixel convolutional layer, as suggested in Odena et al. [18]. The following convolutional layer is used to compensate the upsampling which is a fixed operation that does not have learnable parameters. This works well and eliminates the checkerboard artifacts. The third part is a fusion part which consists of a concatenation and two convolutional layers with feature depth $D_3 = D_1$ and 1, respectively. The input $\boldsymbol{X}_{GCM}$ is also unsampled using bilinear interpolation and concatenated to the upsampling part output. The fusion part combines the upsampled feature maps with the auxiliary inputs to reconstruct the final high-resolution outputs. We use squared loss as the loss function and minimize this objective:

$$L(\theta) = \frac{1}{N} \sum_{n=1}^{N} ||f(\boldsymbol{X}_n, \theta) - \boldsymbol{Y}_n||^2 \tag{1}$$

where θ is the parameters of the neural network to be optimized, N is number of training data, $f(\cdot, \cdot)$ is the learnt function of the network, $\boldsymbol{X}_n = \{\boldsymbol{X}_{GCM}, \boldsymbol{X}_{AUX}\}_n$ is the input, and $\boldsymbol{Y}_n$ is the high-resolution target.

Inspired by Vandal et al. [29] and BCSD, we also use auxiliary inputs to help improve the results. We term them "auxiliary" inputs instead of normal inputs because of two reasons. The first is that they remain the same for different months through out the training and testing period. The other is that they are high-resolution (the same as the target resolution) rather than low-resolution and they are fed into the neural network at the last layers rather than at the first layer along with GCMs. The auxiliary inputs are concatenated to the output of the upsampling layer as extra channels.

4 EXPERIMENT

We applied our model to three climate data sets: monthly mean precipitation (ppt) in mm/day, monthly maximum temperature (tmax) and monthly minimum temperature (tmin) in degree Celsius ($^\circ C$), and compared with four other methods: two climate downscaling methods BCSD [3, 33] and DeepSD [29], and two image super-resolution methods ESPCN [21] and REDNet [15].

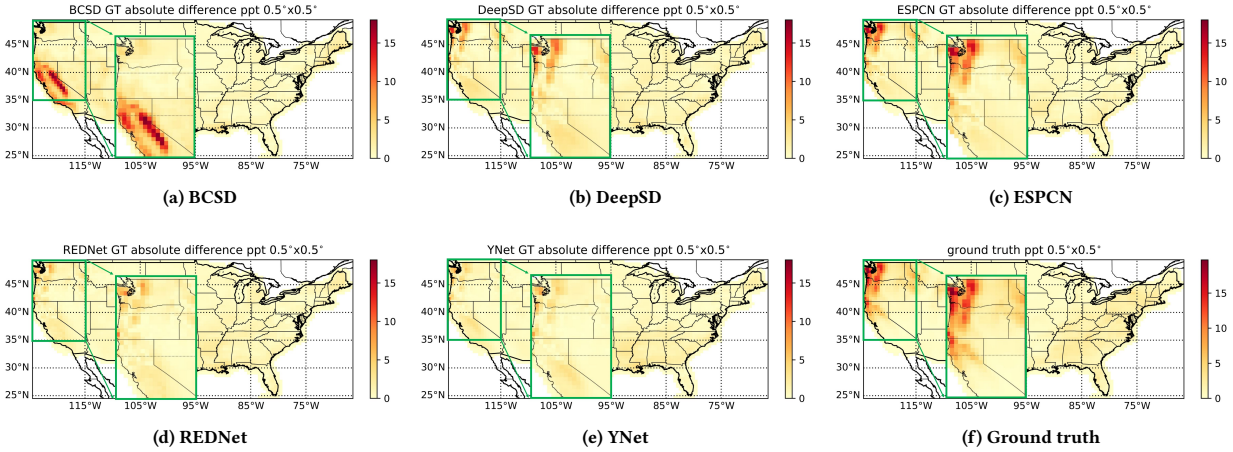
4.1 Datasets

We use the Global Climate Models (GCMs) simulated data provided by NASA [1] as our low-resolution input $\boldsymbol{X}_{GCM}$. We downloaded 35 and 33 different GCMs simulated data from different institutions around the world [2] and thus $\boldsymbol{X}_{GCM}$ has 35 and 33 channels for precipitation and temperature data, respectively. The original GCMs data cover the whole earth, with latitude grid from 89.5S to 89.5N by 1° and longitude grid from 0.5E to 359.5E by 1°, and with every month from January 1950 to December 2005, a total of 672 months. We focus on the Continental United States (CONUS) and extract the data with latitude from 24.5N to 49.5N and longitude from 235.5E to 293.5E. There are 26 points for latitude and 59 for longitude and therefore the input $\boldsymbol{X}_{GCM}$ has a size of 26×59 with 35 and 33 channels for precipitation and temperature, respectively.

We use reanalysis data as our high-resolution target $\boldsymbol{Y}$. The reanalysis data are combinations of sparse on-site observation with other sources such as remote sensing and satellite images to produce high-resolution data. It is common to use reanalysis data as the proxy of true observational data because the site-based observational data are very sparse. The reanalysis data we use is the PRISM [4] data which were downloaded from Oregon State University [3]. In addition, we use high-resolution elevation and climatological data calculated from PRISM data as the auxiliary input $\boldsymbol{X}_{AUX}$. For the elevation data, we use the same data from Global 30 Arc-Second Elevation Data Set (GTOPO30) as with [29]. The original PRISM and GTOPO30 data cover the CONUS with the same spatial resolution of 1/24 degree ($\sim 4km$), with latitude grid from 24.104N to 49.9375N by 1/24 degree, and longitude grid from 234.979E to 293.479E by 1/24 degree. The original latitude and longitude grids are not aligned with GCMs' therefore we have to aligned them by averaging 4 nearest neighbouring locations to produce the central location value. We test our method using three downscaling factors: 2, 4 and 8 in our experiments, but other factors can also be used. After alignment, we reduce the spatial resolution to our desired resolutions (0.5° by 0.5°, 0.25° by 0.25° and 0.125° by 0.125°) accordingly by downsampling interpolation. Although GCMs have values for both land and oceans, the PRISM data only have values for CONUS, therefore we zero out those values outside CONUS. Figure 3 shows an example month (January 2003) of the mean precipitation of 35 GCMs and the corresponding PRISM precipitation at different resolutions. Figure 3a is the average of the 35 different GCMs. The GCM data are very different with PRISM data in addition to the resolution gap. We use the first 600 months (Jan. 1950 to Dec. 1990) data as the training data, the following 36 months (Jan. 2000 to Dec. 2002) data as validation data and the last 36 months (Jan. 2003 to Dec. 2005) data as the test data.

The values for the precipitation are all non-negative and have a long tail distribution with many small values and a few large values.

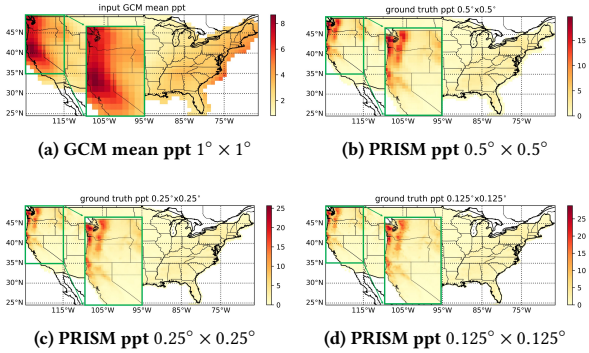
[1] https://registry.opendata.aws/nasanex/, last access: Feb. 2020.
[2] Two GCMs for temperature have missing months so we discard them.
[3] http://www.prism.oregonstate.edu,last access: Feb. 2020.



(a) BCSD (b) DeepSD (c) ESPCN

(d) REDNet (e) YNet (f) Ground truth

Figure 2: Mean precipitation: MAE of different methods with downscaling factor 2 in a single month. The green boxes are zoomed in areas.



(a) **GCM mean ppt** $1^\circ \times 1^\circ$ (b) **PRISM ppt** $0.5^\circ \times 0.5^\circ$

(c) **PRISM ppt** $0.25^\circ \times 0.25^\circ$ (d) **PRISM ppt** $0.125^\circ \times 0.125^\circ$

Figure 3: GCM and PRISM ppt at different resolutions.

The range of precipitation for GCMs is 0 to 33.0 mm/day while the range of that for PRISM is 0 to 67.9 mm/day and the range of elevation is -74.1 to 3677.8 meters. In order to relieve the long-tail effect and align the value ranges, we apply log1p transformation (first add 1.0 and then take logarithm) to the precipitation and elevation and then normalize the values by dividing the maximum value (5 for precipitation and 10 for elevation in the experiments). We set negative values to be zero for elevation before applying the transformation, which should not affect too much for the experiment results since the negative values only appear in a very small part of the image. After that, the range of the inputs are between 0.0 and 1.0, which is common for normalized image pixel value range. We did not perform the log1p transformation for temperature data since they contain both positive and negative values. We just normalize the values to between -1.0 and 1.0 by dividing the maximum absolute value (50 in the experiments). After that, we shift and scale the input such that the range of the inputs are between 0 and 1.0.

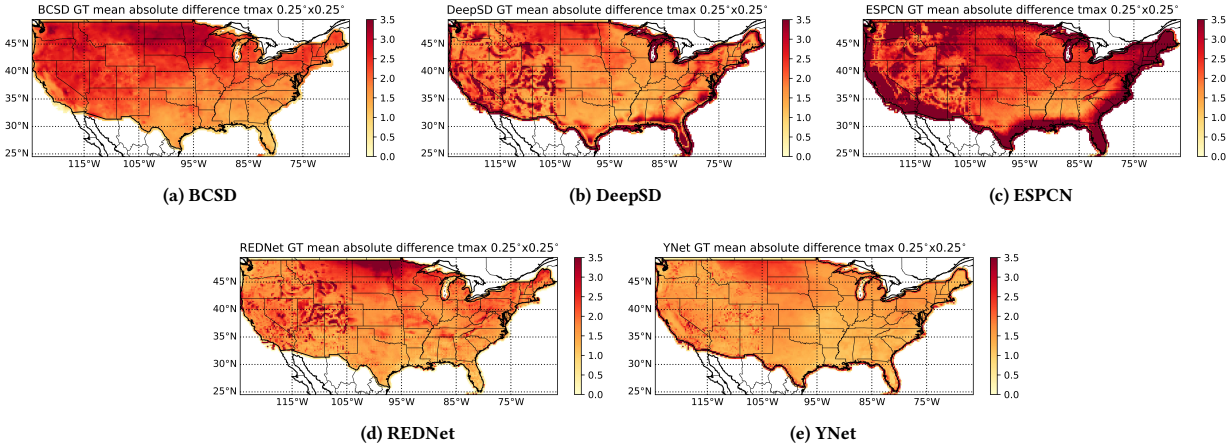
We use two auxiliary variables: high-resolution elevation and climatological variables (precipitation / temperature). The elevation is a topographical variable and is supposed to remain the same for a long time. This variable may have a great effect on precipitation and temperature. For example, high mountains may obstruct the cloud and thus create rainfall on one side while preventing rainfall on the other side of the mountain; higher elevation, lower temperature, etc. We average over the training data for each natural month (January to December) for precipitation, maximum and minimum temperature separately to get the climatological data. These same climatological data are used during both training and test as auxiliary variables. The climatological data go through the same transformation and normalization procedure as the GCM variables.

4.2 Experiment Setup

In computer vision the Peak-Signal-to-Noise Ratio (PSNR) is often used as a metric to compare the qualities of high-resolution images produced by different methods. It is defined as $PSNR = 10log_{10}(\frac{MAX^2}{MSE})$ where MAX is the maximum possible value (e.g., 255 for 8-bit image) and MSE is the mean square error. However, in our case there is no bound for the pixel value and PSNR can not be computed. Therefore we use MSE as the metric. The MSE is calculated as $MSE = \frac{1}{TK}\sum_{t=1}^{T}||\hat{Y}_t - Y_t||^2$ where T is the number of months (i.e., images) in test data, K is the number of locations [4] for each month, and $\hat{Y}_t$ and Y_t is the prediction and the ground truth values for month t, respectively.

The implementations of both REDNet and ESPCN are from Github [5][6]. We implemented BCSD, DeepSD and our proposed YNet using python and pytorch. We use Adam [9] optimizer with ReduceLROnPlateau scheduler. The initial learning rate is 1.0×10^{-4} and the batch size is 32. The input X_{GCM} has a size of 26×59 with 35 or 33 channels and the output has one channel with different sizes corresponding to different downscaling factors. We choose

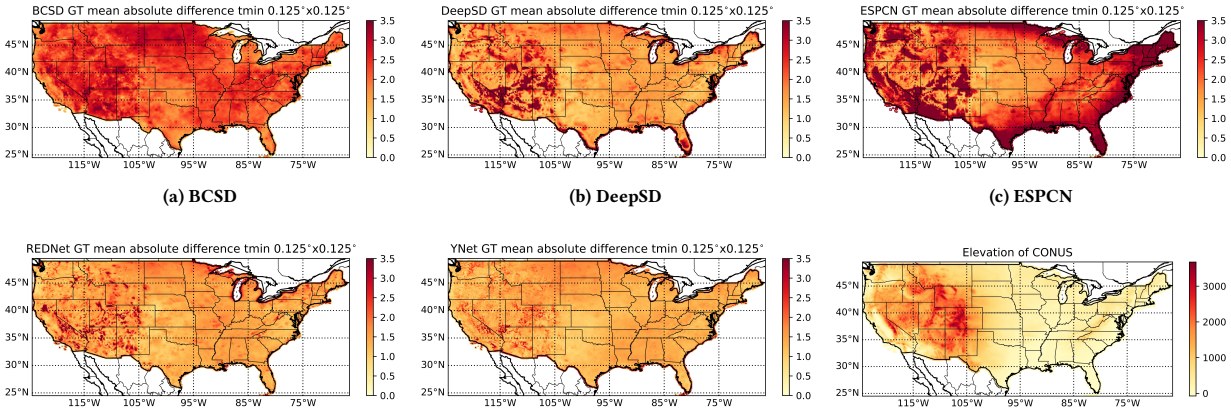
[4] We use the number of pixels in each image as the number of locations for simplicity. This simplification should not affect the comparison results since the number of invalid locations (pixels outside CONUS) are roughly the same for different methods.
[5] https://github.com/yjn870/REDNet-pytorch; last access: Feb. 2020.
[6] https://github.com/leftthomas/ESPCN; last access: Feb. 2020.



(a) BCSD (b) DeepSD (c) ESPCN

(d) REDNet (e) YNet

Figure 4: Maximum temperature: MAE of different methods with downscaling factor 4.



(a) BCSD (b) DeepSD (c) ESPCN

(d) REDNet (e) YNet (f) Elevation map*

Figure 5: Minimum temperature: MAE of different methods with downscaling factor 8. *We put Elevation map here to show the overlap between the erroneous prediction areas and the mountainous / plateau areas. See context for details.

the best number of training epochs using the validation data for all neural network methods and report the results using the test data.

We briefly describe BCSD here. BCSD consists of two steps: a bias correction step and a spatial disaggregation step. Bias correction is applied to match simulated data with observations so that they have the same cumulative distribution function (CDF). Specifically, quantile mapping is used to map every simulated value to a corresponding observation value of the same quantile. The bias-corrected value v_{bc} of a climate variable is $v_{bc} = F^{-1}_{obs}[F_s(v_s)]$ where v_s is the simulated value, $F^{-1}_{obs}(\cdot)$ is the inverse CDF of observation data, and $F_s(\cdot)$ is the CDF of simulated data. The second step is the spatial disaggregation, which is slightly different for precipitation and temperature since precipitation is non-negative but temperature can be negative. The step for precipitation is

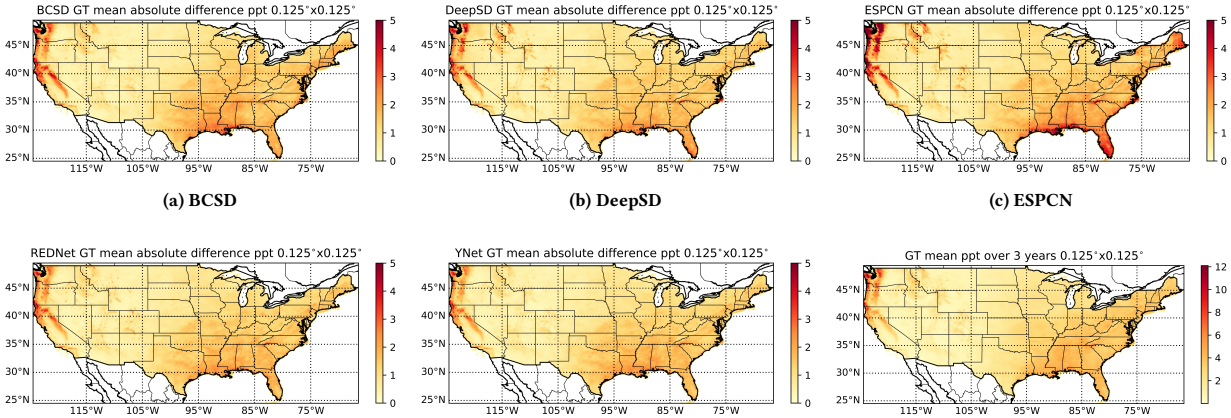
$$v_{bcsd} = v_{ch} + h_{int}(v_{bc} - v_{cl})\left(\frac{v_{ch}}{1 + v_{cl}}\right) \tag{2}$$

where v_{ch} and v_{cl} is the climatological data (i.e., long term average) in low/high-resolution, respectively. $h_{int}(\cdot)$ is an interpolation operation (e.g., bilinear interpolation). For temperature, this step is

$$v_{bcsd} = v_{ch} + h_{int}(v_{bc} - v_{cl}) \tag{3}$$

4.3 Results

We compare the MSE of different methods for the three climate variables: monthly mean precipitation (ppt), monthly maximum temperature (tmax) and monthly minimum temperature (tmin) using three downscaling factors: 2, 4 and 8. The results are shown in Table 1, 2 and 3. The bold font represents the best performance. Our proposed method YNet has the best performance with regards to MSE for all but one case. In Table 1, all methods except ESPCN perform reasonably when the downscaling factor is small and degrade as the factor goes larger. In Table 2 and 3 we can see that the methods REDNet and YNet still have a very good performance but

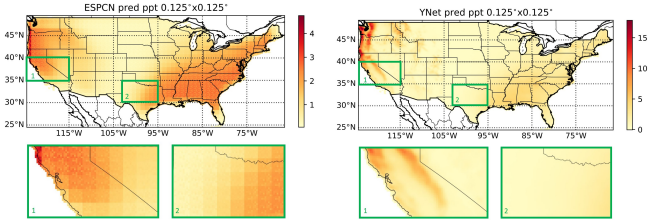


(a) BCSD (b) DeepSD (c) ESPCN

(d) REDNet (e) YNet (f) GT mean*

Figure 6: MAE of predicted mean precipitation of different methods with downscaling factor 8. *We put GT mean here to show the overlap between the erroneous prediction areas and the high precipitation areas. See context for details.

ESPCN deteriorates and cannot get a reasonable result. One possible reason why ESPCN perform worse may be that it has a severe checkerboard artifact, which is inherent in its architecture, and it does not handle the out of boundary pixels very well. In Figure 7 we show the prediction results of ESPCN and our proposed YNet for one month. We can clearly see that the result of ESPCN has serious checkerboard artifacts while YNet does not.



(a) ESPCN predicted ppt (b) YNet predicted ppt

Figure 7: ESPCN vs YNet predicted results for ppt in one month. The green boxes are the zoomed in areas. ESPCN predictions exhibit severe checkerboard artifacts, while YNet does not have this problem.

To examine the performances of methods for different seasons, in Table 4 we show the predictive MSE of these methods on mean precipitation for four seasons when the downscaling factor is 8. The proposed YNet performs best for all four seasons. In addition, comparing to the last column of Table 1, we observe that the predictive ability varies through different seasons. Winter ("DJF") and Spring ("MAM") have smaller MSEs while Summer ("JJA") and Autumn ("SON") have larger MSEs for most of the methods (except for BCSD). This suggests that precipitation in Summer and Autumn are more difficult to predict than that in Winter and Spring.

Table 1: MSE of mean precipitation.

method \ scale factor	2	4	8
BCSD	1.554	1.688	1.763
DeepSD	1.523	1.682	1.701
ESPCN	1.986	2.217	2.498
REDNet	1.345	1.467	1.543
YNet	***1.291***	***1.396***	***1.454***

Table 2: MSE of maximum temperature.

method \ scale factor	2	4	8
BCSD	7.339	7.256	7.289
DeepSD	6.118	6.891	6.482
ESPCN	21.353	24.280	29.212
REDNet	3.124	3.649	3.747
YNet	***2.365***	***2.611***	***2.853***

Table 3: MSE of minimum temperature.

method \ scale factor	2	4	8
BCSD	7.188	6.908	6.690
DeepSD	3.555	3.460	3.800
ESPCN	11.344	13.027	13.989
REDNet	***1.806***	2.500	2.694
YNet	2.070	***2.090***	***2.087***

Table 4: Seasonal MSE of mean precipitation for downscaling factor 8.

method \ season	DJF	MAM	JJA	SON
BCSD	1.620	1.783	1.747	1.901
DeepSD	1.509	1.451	1.873	1.971
ESPCN	2.394	2.109	2.756	2.735
REDNet	1.311	1.365	1.676	1.821
YNet	***1.220***	***1.285***	***1.54***	***1.772***

DJF:Dec.-Feb.; MAM:Mar.-May; JJA:June-Aug.; SON:Sep.-Nov.

In Table 5 we also show the bias and Pearson correlation coefficients for different methods on three data sets when using downscaling factor 8. The bold font is the best. We can see that our proposed method YNet gets the best values for most of the data. From the bias term we observe that BCSD and YNet tend to underestimate both precipitation and temperature, and the other three methods underestimate precipitation while overestimate temperature. One possible reason for BCSD to get low bias for precipitation but high bias for temperature is that the shift between the past and future CDFs for temperature is larger than that for mean precipitation.

Table 5: Bias and correlation of different methods with downscaling factor 8.

method \ data	ppt		tmax		tmin	
	Bias	Corr.	Bias	Corr.	Bias	Corr.
BCSD	***-0.021***	0.757	-1.135	0.977	-1.332	0.958
DeepSD	-0.220	0.765	0.338	0.978	0.024	0.970
ESPCN	-0.295	0.649	0.399	0.891	0.160	0.873
REDNet	-0.199	0.788	0.415	0.988	***0.023***	0.977
YNet	-0.197	***0.802***	***-0.074***	***0.991***	-0.335	***0.983***

To visually examine and compare the results of different methods, we take the prediction results of precipitation for one month as an example. The results are shown in Figure 2. Figure 2f is the corresponding ground truth (GT) precipitation with resolution 0.5° by 0.5°. Figures 2a - 2e are the absolute error maps for different methods. An absolute error map is defined as the absolute value of the difference between a prediction image and the corresponding ground truth image. The deeper the red color, the larger the absolute error for the prediction. We can see from Figure 2 that in this case, most of the precipitations are in the Northwestern coastal area of CONUS, and most errors are concentrated in this area. We zoom in the area in the green boxes and compare them. The proposed YNet has the smallest error area, followed by the REDNet. The other three methods have relatively larger error areas. The neural network based methods have larger errors mainly in areas of the Northwestern corner (Washington State especially around Seattle) where the precipitation is high and few errors in other areas where the precipitation is low. For non neural network based method BCSD, the errors mainly appear in the California state, just around the Central Valley.

For the two data sets monthly maximum temperature and monthly minimum temperature, we show the mean absolute error (MAE) map over all test months. The MAE is defined as $MAE = \frac{1}{T}\sum_{t=1}^{T}|\hat{Y}_t - Y_t|$ where $|\cdot|$ is element-wise absolution. Figure 4 shows the MAE maps of predicted maximum temperature of different methods with downscaling factor 4. The deeper the red color, the larger MAE. For better visualization and fair comparison, we clip the color bar to the same range (i.e., 0 - 3.5) for all methods. From the figures we can see that ESPCN is the worst with severe mismatch near the boundaries. DeepSD also suffers around the boundaries. The other three has no or very mild boundary issues. For the BCSD method, the high error areas are more evenly distributed and cover a large part of CONUS. The reason may be that BCSD heavily relies on climatological data (i.e., long term average in the past years) and tend to underestimate the temperature due to the global warming effect. REDNet has high error in the mid-north area. YNet also has red color (errors) in the mid-north area but they are much less severe.

In addition, we notice that all the methods tend to have more mismatched areas in the mid-west part of CONUS. This phenomenon is also found in the MAE maps of predicted monthly minimum temperature of different methods, as shown in Figure 5. Interestingly, these areas are mostly mountainous areas and plateaus with high elevation, which include the Rocky Mountains, Sierra Nevada, Cascade Range, Columbia Plateau and Colorado Plateau, as shown in red color in Figure 5f. The complex terrain and rapid change of elevations have high impact on temperature and make it more difficult to predict. One hypothesis is that GCMs fail to model the complicated effects of the mountains and plateaus on temperature and the simulated values are more far away from observations than that in other areas. This may need more attention from climate scientists when they want to use or improve GCMs.

In Figure 6 we show the MAE maps of predicted monthly mean precipitation of different methods and the mean precipitation over 3 years in Figure 6f. Unlike temperature, the errors mainly concentrate on the west coastal area and the southern area. These areas are known for having higher mean precipitation than the rest of CONUS. Again, YNet has the smallest area of high error among these methods.

5 CONCLUSION

In this paper we proposed a deep neural network architecture, YNet, to downscale climate variables from low-resolution GCMs simulated data. The proposed method utilizes skip connections in low-resolution image space and incorporates high-resolution topological and climatological variables as auxiliary data. The proposed method can have arbitrary size of input and is very flexible at different downscaling factors. We compare its performance with four other methods on three climate data sets representing different climate variables. We analyze the results using metrics including MSE and MAE maps. The results show that our method significantly outperforms the competing methods.

Although machine learning / deep learning methods have shown strength over traditional climate downscaling methods, there is still a long way to go for the climate science community to fully embrace these methods. One barrier is that deep learning methods are thought to be black boxes without physical explanations. One

future work direction would be extending this work to interpret the physical information from the results, such as explanation of the MAE pattern found in the results and strategies to improve accordingly. It maybe also interesting to develop / investigate uncertainty quantification techniques for deep learning methods applied in climate science since they are important for climate science community. Another direction is to design physics-guided machine learning methods to better incorporate scientific expertise and lead to easier understanding of why they work.

ACKNOWLEDGMENTS

This work was supported by National Science Foundation CyberSEES under grant number: NSF CCF-1442728. We would like to acknowledge computing support from AI for Earth Grant provided by Microsoft and Cheyenne (doi:10.5065/D6RX99HX) provided by NCAR's Computational and Information Systems Laboratory, sponsored by the National Science Foundation.

REFERENCES

[1] Jose Caballero, Christian Ledig, Andrew Aitken, Alejandro Acosta, Johannes Totz, Zehan Wang, and Wenzhe Shi. 2017. Real-time video super-resolution with spatio-temporal networks and motion compensation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4778–4787.

[2] P Caffrey and A Farmer. 2014. A review of Downscaling methods for climate change projections. *Tetra Tech ARD* (2014).

[3] Kai Chen, Arlene M Fiore, Renjie Chen, Leiwen Jiang, Bryan Jones, Alexandra Schneider, Annette Peters, Jun Bi, Haidong Kan, and Patrick L Kinney. 2018. Future ozone-related acute excess mortality under climate and population change scenarios in China: A modeling study. *PLoS medicine* 15, 7 (2018).

[4] Christopher Daly, Michael Halbleib, Joseph I Smith, Wayne P Gibson, Matthew K Doggett, George H Taylor, Jan Curtis, and Phillip P Pasteris. 2008. Physiographically sensitive mapping of climatological temperature and precipitation across the conterminous United States. *International Journal of Climatology: a Journal of the Royal Meteorological Society* 28, 15 (2008), 2031–2064.

[5] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. 2014. Learning a deep convolutional network for image super-resolution. In *European conference on computer vision*. Springer, 184–199.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[7] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. 2016. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1646–1654.

[8] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. 2016. Deeply-recursive convolutional network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1637–1645.

[9] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[10] Frank Kreienkamp, Andreas Paxian, Barbara Früh, Philip Lorenz, and Christoph Matulla. 2019. Evaluation of the empirical–statistical downscaling method EPISODES. *Climate dynamics* 52, 1-2 (2019), 991–1026.

[11] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. 2017. Deep laplacian pyramid networks for fast and accurate super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 624–632.

[12] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. 2017. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4681–4690.

[13] Zhen Li, Jinglei Yang, Zheng Liu, Xiaomin Yang, Gwanggil Jeon, and Wei Wu. 2019. Feedback network for image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3867–3876.

[14] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. 2017. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 136–144.

[15] Xiaojiao Mao, Chunhua Shen, and Yu-Bin Yang. 2016. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In *Advances in neural information processing systems*. 2802–2810.

[16] Q Miao, B Pan, H Wang, K Hsu, and S Sorooshian. 2019. Improving Monsoon Precipitation Prediction Using Combined Convolutional and Long Short Term Memory Neural Network, Water, 11, 977.

[17] Saptarshi Misra, Sudeshna Sarkar, and Pabitra Mitra. 2018. Statistical downscaling of precipitation using long short-term memory recurrent neural networks. *Theoretical and Applied Climatology* 134, 3-4 (2018), 1179–1196.

[18] Augustus Odena, Vincent Dumoulin, and Chris Olah. 2016. Deconvolution and Checkerboard Artifacts. *Distill* (2016). https://doi.org/10.23915/distill.00003

[19] Eduardo Rocha Rodrigues, Igor Oliveira, Renato Cunha, and Marco Netto. 2018. DeepDownscale: a deep learning strategy for high-resolution weather forecast. In *2018 IEEE 14th International Conference on e-Science (e-Science)*. IEEE, 415–422.

[20] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*. Springer, 234–241.

[21] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. 2016. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1874–1883.

[22] Wenzhe Shi, Jose Caballero, Lucas Theis, Ferenc Huszar, Andrew Aitken, Christian Ledig, and Zehan Wang. 2016. Is the deconvolution layer the same as a convolutional layer? *arXiv preprint arXiv:1609.07009* (2016).

[23] Craig D Smith. 2008. The relationship between monthly precipitation and elevation in the Alberta foothills during the foothills orographic precipitation experiment. In *Cold Region Atmospheric and Hydrologic Studies. The Mackenzie GEWEX Experience*. Springer, 167–185.

[24] Ying Tai, Jian Yang, and Xiaoming Liu. 2017. Image super-resolution via deep recursive residual network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3147–3155.

[25] Yusuke Tanaka, Tomoharu Iwata, Toshiyuki Tanaka, Takeshi Kurashima, Maya Okawa, and Hiroyuki Toda. 2019. Refining coarse-grained spatial data using auxiliary spatial data sets with various granularities. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 5091–5099.

[26] Duong Tran Anh, Song P Van, Thanh D Dang, and Long P Hoang. 2019. Downscaling rainfall using deep learning long short-term memory and feedforward neural network. *International Journal of Climatology* 39, 10 (2019), 4170–4188.

[27] Sylwia Trzaska and Emilie Schnarr. 2014. A review of downscaling methods for climate change projections. *United States Agency for International Development by Tetra Tech ARD* (2014), 1–42.

[28] Thomas Vandal, Evan Kodra, and Auroop R Ganguly. 2019. Intercomparison of machine learning methods for statistical downscaling: The case of daily and extreme precipitation. *Theoretical and Applied Climatology* 137, 1-2 (2019), 557–570.

[29] Thomas Vandal, Evan Kodra, Sangram Ganguly, Andrew Michaelis, Ramakrishna Nemani, and Auroop R Ganguly. 2017. 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, 1663–1672.

[30] John E Walsh, Uma S Bhatt, Jeremy S Littell, Matthew Leonawicz, Michael Lindgren, Thomas A Kurkowski, Peter A Bieniek, Richard Thoman, Stephen Gray, and T Scott Rupp. 2018. Downscaling of climate model output for Alaskan stakeholders. *Environmental modelling & software* 110 (2018), 38–51.

[31] Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. 2018. Recovering realistic texture in image super-resolution by deep spatial feature transform. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 606–615.

[32] Zhaowen Wang, Ding Liu, Jianchao Yang, Wei Han, and Thomas Huang. 2015. Deep Networks for Image Super-Resolution With Sparse Prior. In *The IEEE International Conference on Computer Vision (ICCV)*.

[33] Andrew W Wood, Lai R Leung, Venkataramana Sridhar, and DP Lettenmaier. 2004. Hydrologic implications of dynamical and statistical approaches to downscaling climate model outputs. *Climatic change* 62, 1-3 (2004), 189–216.

[34] Mengchao Xu, Qian Liu, Dexuan Sha, Manzhu Yu, Daniel Q Duffy, William M Putman, Mark Carroll, Tsengdar Lee, and Chaowei Yang. 2020. PreciPatch: A Dictionary-based Precipitation Downscaling Method. *Remote Sensing* 12, 6 (2020), 1030.

[35] Zhongfeng Xu, Ying Han, and Zongliang Yang. 2019. Dynamical downscaling of regional climate: A review of methods and limitations. *Science China Earth Sciences* 62, 2 (2019), 365–375.

[36] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. 2018. Residual dense network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2472–2481.

[37] Zachary Zobel, Jiali Wang, Donald J Wuebbles, and V Rao Kotamarthi. 2017. High-resolution dynamical downscaling ensemble projections of future extreme temperature distributions for the United States. *Earth's Future* 5, 12 (2017), 1234–1251.