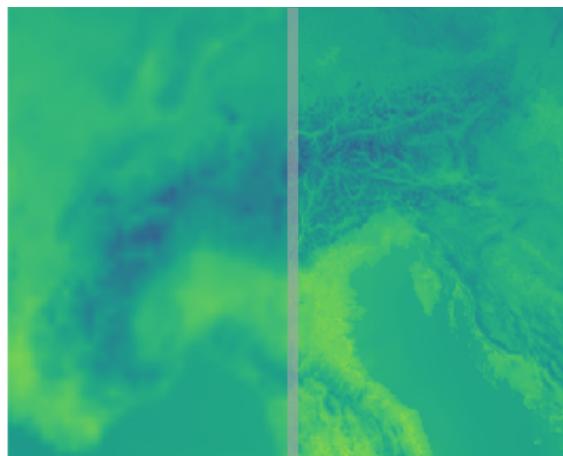


Research Internship (PRE)

Field of Study: Applied Mathematics

Scholar Year: 2021-2022

Downscaling of climate models using Multiple-Point Geostatistics approaches



Confidentiality Notice

Non-confidential report and publishable on Internet

Author:

Léo Micollet

ENSTA ParisTech Tutor:

David Lefèvre

Promotion:

2023

Host Organism Tutor:

Grégoire Mariéthoz

Internship from May 16, 2022 to August 26, 2022

Name of the host organism: Université de Lausanne

**Address: Université de Lausanne - Château de Dornigny 1015
Lausanne - Switzerland**

Confidentiality Notice

This present document is not confidential. It can be communicated outside in paper format or distributed in electronic format.

Abstract

Climate models are physics-based simulation of the Earth surface that, given historical data as input. They are used for analysis of the climate (Temperature, humidity, wind velocity, etc.) over short or long periods, small or large regions, with different resolutions. The computation is really long for good prediction of the small scale evolution of the climate. It is where downscaling methods can help increase the resolution. This work presents baselines for a bigger project that aims to compare different downscaling methods on a regional and global set. The downscaling methods presented in the project are based on multiple-point geostatistics and machine learning. A pre-processing step is performed on the regional data to prepare it for downscaling. Metrics are implemented using baseline downscaling methods to allow a clear comparison with the methods that will be implemented after this work.

Acknowledgments

I'm extremely grateful to Prof. Grégoire Mariéthoz, director of the IDYST institute, for this opportunity and his guidance throughout the course of my internship, and Tom Beucler, assistant professor, for the continuous accompaniment during those 3 months and the great help he provided with his project proposal.

I am also grateful to Dr. David Leutwyler for the climate models and all the answers he had for my question.

Lastly, many thanks to the GAIA lab, Naomie Kayitesi, Fatemeh Zakeri, Pau Wiersma, Hu Guanghui, and also Wenyue Zou, for the help and your presence during my internship.

Contents

| | |
|---|-----------|
| Confidentiality Notice | 3 |
| Abstract | 5 |
| Acknowledgments | 7 |
| Contents | 9 |
| Introduction | 11 |
| I Climate models downscaling | 13 |
| I.1 Climate modeling | 13 |
| I.1.1 Climate models | 13 |
| I.1.2 Storm-resolving model (SRM) | 13 |
| I.1.3 Historical data and pseudo global warming simulations | 14 |
| I.2 Downscaling and climate models | 15 |
| I.2.1 The problem of high resolution models and future predictions models | 15 |
| I.2.2 Multiple-point geostatistics | 15 |
| I.2.3 Deep Learning methods | 16 |
| II Getting ready for downscaling | 19 |
| II.1 Data pre-processing | 19 |
| II.1.1 Remote location | 19 |
| II.1.2 Data exploration | 20 |
| II.1.3 Creating the upscaled data | 21 |
| II.2 Metrics implementation | 21 |
| II.2.1 Pixel-wise metrics | 21 |
| II.2.2 Structural Similarity Index (SSIM) | 22 |
| II.2.3 Probability density function metrics | 23 |
| II.2.4 Temporal and spatial autocorrelation | 24 |
| III Downscaling implementation | 27 |
| III.1 Baselines methods | 27 |
| III.1.1 Temperature downscaling | 27 |
| III.1.2 Relative humidity downscaling | 31 |
| III.1.3 Total precipitation downscaling | 32 |
| III.1.4 Future climate models | 34 |
| III.2 Other downscaling algorithms | 34 |
| Conclusion | 37 |

| | |
|--|-----------|
| Bibliography | 39 |
| List of Tables | 41 |
| List of Figures | 43 |
| Appendices | 45 |
| A Autocorrelation plots for the baselines methods | 47 |
| A.1 Temperature autocorrelation plots | 47 |
| A.2 Relative humidity autocorrelation plots | 49 |
| A.3 Total precipitation autocorrelation plots | 51 |
| A.4 Future temperature autocorrelation plots | 53 |

Introduction

Environmental science aims to give a better understanding of the mechanisms at the surface of the earth. These mechanisms are being altered by climate change, resulting in modification of the climate and increases in the frequency of severe events, thus the need for forecasting models. Climate models can predict the evolution of the earth surface for multiple variables, but it takes a lot of resources to obtain a high resolution model on large areas. Moreover, our existing models are based on historical data, but the future predictions are likely to differ from the past. It is where downscaling methods can play a role. In addition to improving climate model resolution, these methods can be trained with historical data, but their performance on coarse resolution must be controlled.

I worked in the GAIA lab (Geostatistical Algorithms Image Analysis) with 3 PhD students, with the presence of Prof. Grégoire Mariéthoz, director of the Institute of Earth Surface Dynamics at UNIL. This laboratory is specialized in stochastic and geostatistical methods. My work was part of a bigger project, and I was supported by both Grégoire Mariéthoz and Tom Beucler, principal investigator of the DAWN lab (Data-Driven Atmospheric and Water Dynamics). This project is an in-depth comparison of multiples downscaling methods. Tom Beucler's project proposal is extremely complete and detailed, and keeps track of all the project related documentation and steps involved in the project [1].

My part was more focused on laying the foundation for the future PhD student's work. I had to get familiar with working on a cluster. Then, I had to understand and prepare the data, then implement the first metrics so I could gain a deeper understanding of how the methods performed. The downscaling methods will take as input 12km resolutions models and will have to upscale it to a 2 km resolution. The rest of the project will be done by two PHD students. One PHD student will test statistical downscaling methods for regional analysis, and the other will test stochastic downscaling methods for global analysis. At the end of this project, it is expected to have a clear understanding of how different downscaling methods respond to climate change.

Part I

Climate models downscaling

In this section, climate models will be briefly defined, with a focus on the models used in this project. The importance of downscaling is then explained, with several methods already used.

I.1 Climate modeling

COSMO SRM climate models are presented in this section.

I.1.1 Climate models

Climate models are derived from fundamental physics equation, that are computed on large zones using approximations and discretization of the equations [2]. These models take into account initial conditions (such as temperature, humidity, topography ...) and the different equations that describe the Earth surface in order to give realistic simulations. In addition to simulating future seasons, they can also be used to predict the effects of global warming. The models are quite reliable for global warming prediction, but small physics phenomenon and precise regional simulation are more complicated to obtain [2]. It is possible to use those models on a variety of scales, from the entire Earth to small areas.

The resolution of such models can be large ($\sim 100\text{km}$) for big scale models, which simplifies the simulations and reduces the computation time. However, high resolution ($\sim 10\text{km}$) climate models are really useful for meteorological simulations, but are mostly used at regional scales.

I.1.2 Storm-resolving model (SRM)

As mentioned previously, climates models can have different resolution. However, to predict small events, 10km models are too coarse. Indeed, for convective storm prediction, a finer resolution of at least 5km is necessary [3].

A good prediction of small and violent events, such as storms and heavy rainfall, are really interesting for countries and cities. Indeed, in July 2021, severe floods due to important rainfalls caused deaths and important damages in Central Europe [4]. Such events could become more frequent in a warming climate, with a likelihood between 1.2 and 9 for a similar 1-day or 2-days event to happen compared to a cooler climate [4]. An

effective SRM simulation could reduce risk and damage from these future disasters by developing a precise and widely available method of predicting the places where it could occur and the intensity. It can also be used for simulation of the physical aspects of earth surface with a warming climate [5].

I.1.3 Historical data and pseudo global warming simulations

In this project, two years of hourly data are available. All data are centered on Europe, with latitude and longitude rotated to prevent distortion due to the poles. Each dataset is a 3084×3084 km image, with a resolution of 2km for each dimension, which is perfect for convective storm simulation [3].

There are 26 variables in each hourly dataset, but the project will focus on only three: temperature at 2 meters elevation(T_2M), relative humidity at 2 meters elevation (RELHUM_2M) and total precipitation (TOT_PR). Indeed, those three variables are supposed to be quick to downscale once the models are trained, and can be used as inputs in climate models.

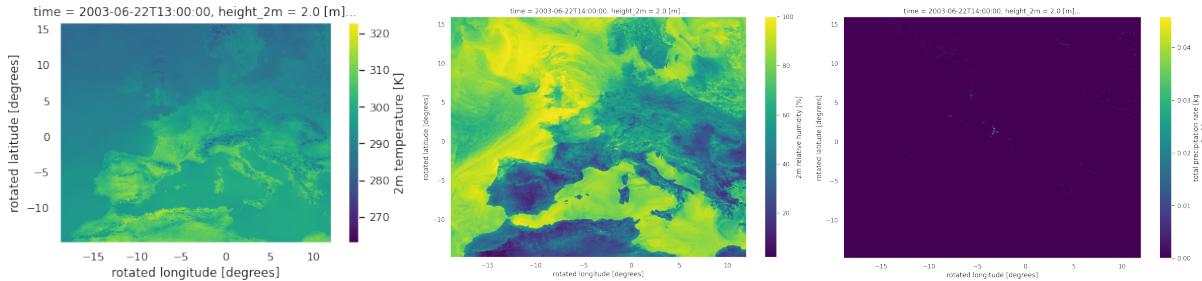


Figure I.1: Representation of the three variables (respectively T_2M, RELHUM_2M and TOT_PR) on May 22, 2003

The 2003 dataset is made of past measurements, whereas the 2083 values are based on a pseudo-global warming scenario. To create this future simulation, the historical data is imposed new initial and boundary conditions [5]. The temperature is increased on this simulation, the relative humidity is lower on the southern part of Europe, and rain is of course still present on the continent.

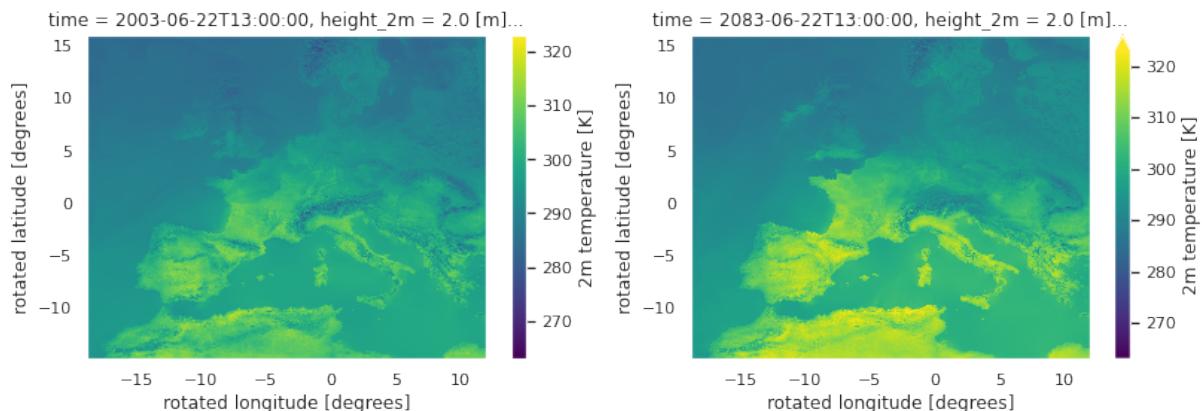


Figure I.2: Representation of T_2M on May 22, 2003 and May 22, 2083

The models available in the project are already in a high 2km resolution, it is however interesting to see if a coarse resolution model could be enhanced with methods trained on historical data.

I.2 Downscaling and climate models

It is explained here why downscaling is necessary and the different methods available.

I.2.1 The problem of high resolution models and future predictions models

With climate models, it is difficult to obtain a high resolution simulation over a big region because of a too high computation time. However, SRM are becoming increasingly necessary to have a better physical understanding and evolution forecast of the potential climate change impacts and can play a role in the preparation of countries and regions against more frequent extreme events. Downscaling methods could be a solution to this problem. Indeed, if a low resolution model is available, downscaling methods could increase the resolution of this model, using past high resolution data. Such methods have already been used to increase the resolution from 100×100 km to 2×2 km [6], 50km to 10km [7], 39 km to ~ 9 km [8], or 3m to 1.24m [9]. The methods tested in this internship include multiple-point geostatistics and deep learning.

There is however an uncertainty with these methods. The training data only reflect past and present climates, but future climates are likely to differ from the historical values. It is important to test how well the different downscaling methods trained on historical data will perform in a future warmer climate.

I.2.2 Multiple-point geostatistics

Geostatistics is a field of study that works with spatiotemporal variables, for instance in hydrology or climate science, and can be used to predict the evolution of a certain animal population, weather or future climate. Sometimes, big historical datasets or complete datasets are hard to come by, or the opposite can also occur if there is an abundance of data. A too coarse input or too long computation time can make the training of an algorithm difficult in any of these cases. The multiple point geostatistic methods are aiming to avoid such issues by using models that could only require one sufficient training image, gathering the most useful information about the targeted area [10].

QuickSampling (QS) [11] is one of the methods used in this project to downscale the coarse resolution data, and part of the multiple-point geostatistics family. The algorithm takes training images, that can be either univariate or multivariate, and a target grid that needs only to match with the training image number of variables. This target grid can be empty in a unconditional simulation, or can contain some information in a conditional simulation.

The algorithm takes an empty pixel and the N closest neighbours that are not empty, and based on a given distance metric, that can be the L^2 distance or the MAE for example, looks for the most similar pattern in one of the training images. Two important parameters

are also given: the number of neighbors N which is the number of closest pixels to take for the pattern recognition, and k , the number of similar patterns to consider.

The best pattern will be chosen depending on k : a frequent k is 1.2, it means that the best pattern will be chosen with a chance of $\frac{1}{1.2} * 100 \sim 83\%$, and the second best with a chance of $\frac{0.2}{1.2} * 100 \sim 17\%$.

N and k need to be chosen wisely : $k = 1$ or a too big N can lead to the reproduction of entire parts of the training image on the targeted one. This is called a verbatim copy [11].

The training image can be multivariate, therefore it is possible to use a training image with coarse and fine resolution as the two variables, and apply the QS algorithm on a target data with a known low resolution and an empty high resolution [9]. An other advantage is that compared to machine learning methods, few training images are needed to use the QS algorithm. This open-source method has already been used for downscaling problems [9].

QS tutorial is given with the g2s website¹.

I.2.3 Deep Learning methods

Deep Learning algorithm are really powerful, and are used in domain dealing with a lot of data, from image analysis to YouTube algorithm. Artificial Neural Networks (ANN) are the basis of Deep Learning methods. ANN are based on the neurons in our brain, and are connected together in the deep learning algorithms to process data. Two Deep Learning methods are presented for this project: Super Resolution Generative Adversarial Networks (SRGAN)[12] and Convolutional Neural Network (CNN)[8].

A GAN is composed of two neural networks, a generator G and a discriminator D . The generator's goal is to give as an output data that corresponds to the high resolution training set with only the coarse data as an input. The discriminator takes data as an input, and must tell if the data is coming from the training set or not. The objective of the GAN is to have the discriminator having a loss around 0.5. If that is the case, the discriminator cannot make a distinction between the actual training data and the output of the generator. When this loss is more than 0.6, the discriminator is trained multiple times until the loss goes towards 0.5. If it is less than 0.45, the generator is then trained multiple times [6].

The first trained neural network is the discriminator. Half of the inputs given are from the training set, the other is just random noise. The point is to train the discriminator to differentiate the real from the fake data. After this, the discriminator and the generator are trained one after the other.

The loss function for the discriminator is given by $\mathcal{L}_D(\mathbf{x}, \mathbf{y}) = -\log(D(\mathbf{y})) - \log(D(G(\mathbf{x})))$ and the loss function for the generator is given by $\mathcal{L}_G(\mathbf{x}, \mathbf{y}) = \|\mathbf{y} - G(\mathbf{x})\|_2^2 - \alpha \log(D(G(\mathbf{x})))$. The SRGAN model used for the project² has already be trained with climate models data from an other article. It was however for wind and solar data, from a 100×100 km to a

¹<https://gaia-unil.github.io/G2S/>

²<https://github.com/NREL/PhIRE>

2×2 km [6].

A CNN can also be used for downscaling, for example of wind velocity [8]. The CNN is made of multiple layers, which decrease the size of the image to only keep feature maps on the layers. In the case of the wind velocity downscaling article, the MSE was used for the loss function. The results from this method are not precise enough, delivering blurry images. The CNN GitHub is going to be the first CNN tried on this project data ³.

³<https://github.com/khoehlein/CNNs-for-Wind-Field-Downscaling/tree/master/data>

Part II

Getting ready for downscaling

The available data cannot be directly used for the different downscaling methods, and it is not possible to conclude whether the methods are effective based only on the visual aspects of the downscaled images. The downscaling results need to be easier to interpret after this preliminary work.

II.1 Data pre-processing

The first preliminary step was to be able to work with the data, and arrange it so that all methods could be used.

II.1.1 Remote location

The entire project is hosted on a remote cluster, called Curnagl. After managing to connect to the cluster through the login node, and viewing the repository from the command line, it was necessary to request for CPU allocations. Indeed, the two years of data are available in two different repositories, and represent a large amount of memory, 1.9Tb for each year. It is therefore not possible to work on a personal computer directly with the data. Once the login node is reached, an interactive partition is created on an other node to access directly the resources. It is then necessary to tunnel, which means accessing the allocated node, in order to start coding.

With a conda environment installed, the first data exploration was done using a Jupyter notebook. It is a really simple tool that allows to keep track of the past work, and quite easy to handle. The interactive session allows to write code or modify other files while an other one is running, and visualize the functions result during the process. The entirety of the functions and notebooks are saved in a repository linked on GitHub (called Downscaling_CM)¹, respectively in /utils and /dev repositories. This GitHub repository is a safety in case of issue with the cluster, but also gives a clear presentation of the work done for the next students that will continue this project. A documentation keeping track of all the useful command lines (to achieve the cluster connection, tunnelling...) was made and is also present in the GitHub repository.

¹https://github.com/LeoMicollet/Downscaling_CM

II.1.2 Data exploration

The hourly datasets are saved in .netCDF format, frequently used among geoscientist and really useful to save multidimensional arrays, for instance Earth surface observations. Xarray is the perfect tool to work with such files. In all the hourly datasets, all the 23 useless data for the project were dropped. All the hourly datasets were previously saved separately in the yearly repository, the second step was to keep all of them in a single .netCDF file following the time dimension. When all the files were grouped together, the hourly datasets were randomly shuffled in the process. Sorting was therefore necessary to keep track of the evolution of the variables. Daily, weekly and seasonal datasets were finally created. We clearly see with figure II.1 the transformation of the datasets, with the main variables and coordinates kept and the augmentation of the time dimension.

| | |
|-------------------|---|
| xarray.Dataset | |
| ► Dimensions: | (time: 1, bnds: 2, rlon: 1542, rlat: 1542, srlon: 1542, srlat: 1542, level1: 61, soil1: 10) |
| ▼ Coordinates: | |
| time | (time) datetime64[ns] 2003-01-01T03:00:00 |
| rlon | (rlon) float32 -18.86 -18.84 ... 11.94 11.96 |
| rlat | (rlat) float32 -14.86 -14.84 ... 15.94 15.96 |
| srlon | (srlon) float32 -18.85 -18.83 ... 11.95 11.97 |
| srlat | (srlat) float32 -14.85 -14.83 ... 15.95 15.97 |
| lon | (rlat, rlon) float32 ... |
| lat | (rlat, rlon) float32 ... |
| height_2m | () float32 ... |
| height_10m | () float32 ... |
| soil1 | (soil1) float32 0.005 0.025 0.07 ... 2.86 5.74 11.5 |
| ► Data variables: | (28) |
| ▼ Attributes: | |
| title : | CLM driven by EralINT |
| source : | COSMO_5X_pompa |
| experiment_id : | evaluation |
| realization : | 1 |
| Conventions : | CF-1.4 |
| conventionsURL : | http://www.cfconventions.org/ |
| creation_date : | 2019-03-12 09:10:11 |
| xarray.Dataset | |
| ► Dimensions: | (time: 24, rlon: 1542, rlat: 1542) |
| ▼ Coordinates: | |
| time | (time) datetime64[ns] 2003-06-22 ... 2003-06-22T23:00:00 |
| rlon | (rlon) float32 -18.86 -18.84 ... 11.94 11.96 |
| rlat | (rlat) float32 -14.86 -14.84 ... 15.94 15.96 |
| height_2m | () float32 2.0 |
| height_10m | () float32 10.0 |
| ▼ Data variables: | |
| T_2M | (time, rlat, rlon) float32 293.9 293.9 293.9 ... 283.5 283.5 |
| RELHUM_2M | (time, rlat, rlon) float32 79.83 79.32 79.08 ... 79.15 79.35 |
| TOT_PR | (time, rlat, rlon) float32 0.0 0.0 0.0 0.0 ... 0.0 0.0 0.0 0.0 |
| ▼ Attributes: | |
| standard_name : | air_temperature |
| long_name : | 2m temperature |
| units : | K |
| grid_mapping : | rotated_pole |

Figure II.1: Structure of a random hourly dataset and structure of a modified dataset (values of an entire day)

The plotting with xarray is simple, but it is important to watch out for the scale. Indeed, depending on the time and the season, the value of the different variables changes, and so is the scale.

However, netCDF doesn't works with all the methods. QS needs numpy arrays inputs for the training images. The two deep learning methods are using tensorflow format for the training images. In deep learning, large datasets are used for training, so this format

keeps them in a memory-efficient file.

Functions are therefore made to change the datasets format, from .netCDF to numpy arrays for the algorithms, and from the different outputs to .netCDF in order to use the different metrics functions, presented later in the report.

II.1.3 Creating the upscaled data

To run the different algorithms is important that the training image has both coarse and fine resolution. Thus, the high resolution data must be upscaled to the coarse resolution wanted. It was advised to use function remapcon from the Climate Data Operator software, which uses first order conservative remapping for regridding from the 2km resolution to a 12 km resolution, using a spherical coordinates system.[13].

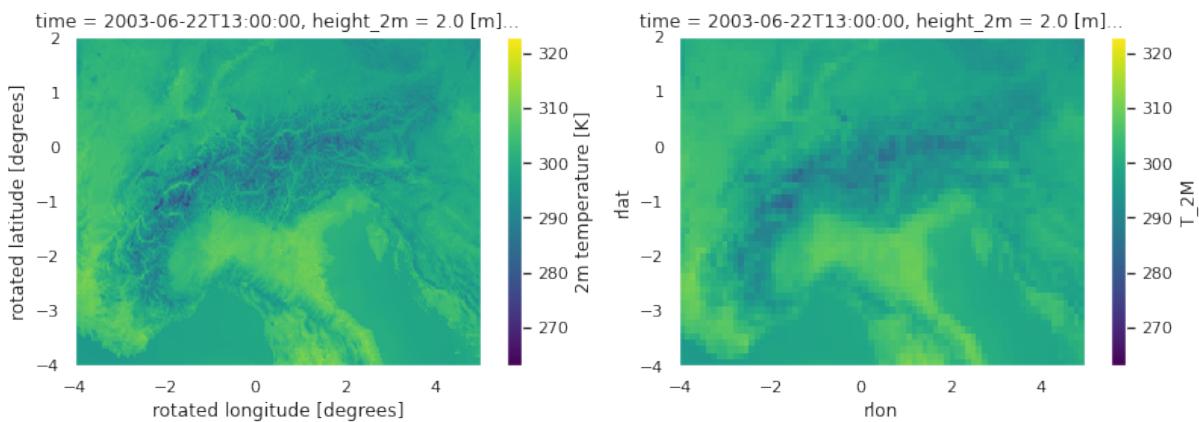


Figure II.2: Representation of T_{2M} in the original model and after the upscaling in a mountainous area

The loss of information on discontinuous terrain is clearly visible in the figure II.2 after the upscaling.

II.2 Metrics implementation

It is great to visualize the data, but it is better to know how to interpret the results before. Metrics are used to gather more information about the methods. The metrics were tested with results from the baselines methods in .netCDF format.

II.2.1 Pixel-wise metrics

The Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) are giving a simple result that compares directly the pixels with one another. These two metrics are quite similar, with the following expressions :

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (x_{i \text{ original}} - x_{i \text{ downscaled}})^2}{n}}$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |x_{i \text{ original}} - x_{i \text{ downscaled}}|$$

n is the total number of pixels, x_i *original* is a pixel from the original data and x_i *downscaled* is the same pixel from the downscaled data. Calculations were performed using the function rmse and mae from the xskscore package, which works directly with xarrays. For both methods, a result of 0 means that the images are similar.

II.2.2 Structural Similarity Index (SSIM)

This metric is quite different, because it aims to compare images with human criteria. Three aspects of the images are taken in account: the luminance, the contrast and the structure [14].

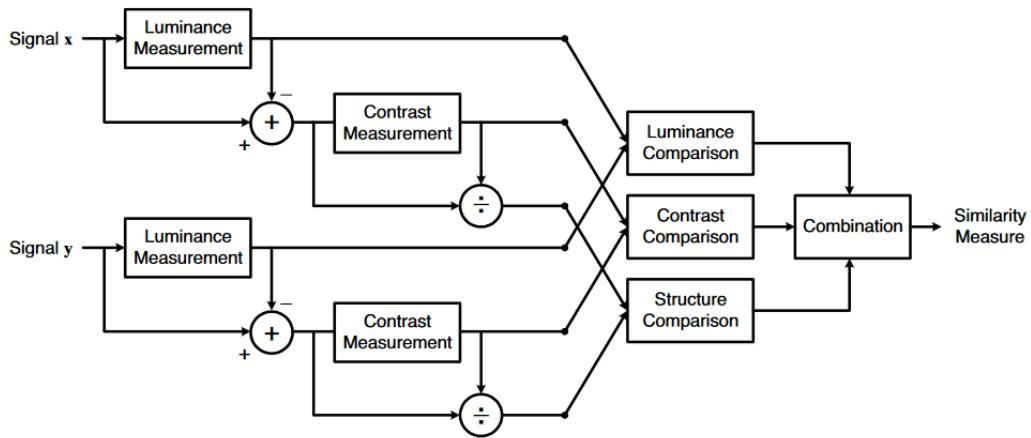


Figure II.3: Diagram of how the SIM works, from the original article [14]

The luminance takes the average of all the pixels for each image $\mu_x = \frac{1}{N} \sum_{i=1}^N x_i$. An unbiased estimator of the standard deviation is then taken to get the contrast $\sigma_x = \left(\frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x)^2 \right)^{1/2}$. The contrast comparison function $c(x, y)$ is similar to the luminance one. Finally, the structure comparison takes the correlation coefficient between the two images $\sigma_{xy} = \frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x)(y_i - \mu_y)$ and is given with the comparison function $s(x, y)$.

$$l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}; \quad c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}; \quad s(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}$$

C_1 , C_2 and C_3 are constants that exclude the possibility of a denominator close to zero for each comparison. The combination of those different comparison is then summarised to return the SSIM score :

$$SSIM(x, y) = f(l(x, y), c(x, y), s(x, y)) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

The SSIM function for this project is giving a result between 0 and 1, with 1 for the same images and 0 for two very different images. The SSIM is symmetric and $SSIM(x, y) =$

$1 \Leftrightarrow x = y$ in the implementation chosen for the project. However, the triangle inequality is not verified, hence the SSIM is not a distance function.

It is not sure if this metric is appropriate for earth surface measures, but implementation was done using the SSIM function from the scikit image package. An appropriate function is made to give the SSIM result from a xarray.

II.2.3 Probability density function metrics

Two different types of metrics comparing the probability density functions are made.

The first one is 1-dimensional probability density function comparison, using Hellinger distance $H^2(P, Q) = \frac{1}{\sqrt{2}} \cdot \|\sqrt{P} - \sqrt{Q}\|_2$, with P and Q two distributions, and Perkins Skill Score $P(f, g) = \int \min(f(x), g(x))dx$ [15].

The Hellinger distance verifies the three conditions of a distance. $H(f, g) \geq 0$ and $H(f, g) = 0 \Leftrightarrow f = g$, H is symmetrical and H verifies the triangle inequality. A value of 0 indicates that the functions have the same density, and a value of 1 means that if one density is zero, the other must be strictly positive, and neither can be strictly positive at the same time.

The Perkins skill score is really simple to interpret as it is only taking the minimum between two densities over the definition domain. The downscaled distribution perfectly fits the original distribution if it obtains a score of 1. It is a symmetric measure, positive, however $P(x, y) = 1$ if $x = y$, it can not be called a distance.

These metrics are going to quantify how well the original density is reproduced by the different downscaling methods. For both methods, the probability density function is given with the function np.histogram, with the parameter density set to *True* to obtain a normalized density, and the integration using the corresponding function with the np.trapz function. This function is based on the trapezoidal rule for the integral.

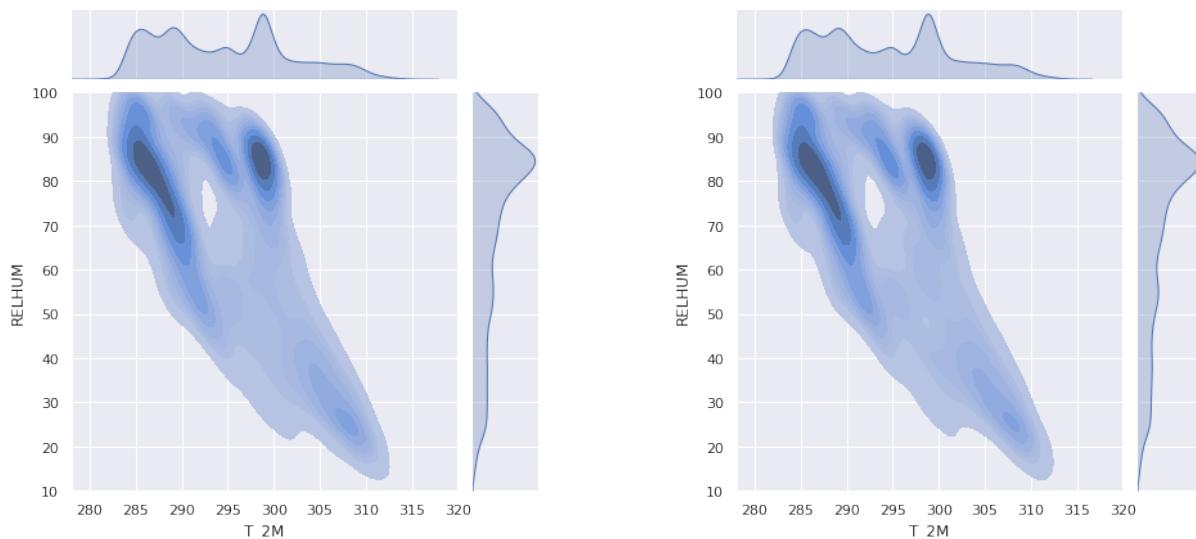


Figure II.4: Bidimensional probability density function between temperature and relative humidity, for the original data the quintic interpolation method with 5% of the values

The second type is 2-dimensional probability density function comparison. Perkins is

used again, because the function is simple to implement in two dimensions. Hellinger was too complicated to implement and test, because of the high computing time due to the huge number of value for a daily dataset ($57 * 10^6$ values).

`np.histogram2d` gives the 2-dimensional probability density function, and `Perkins` is then computed with the `scipy integrate` function `nquad`.

II.2.4 Temporal and spatial autocorrelation

The autocorrelation shows the relation between the present dataset and its evolution following one of its dimension.

Temporal autocorrelation works with two dataset: the original one and the same temporally-lagged dataset. Autocorrelation is then calculated between all those lagged dataset, and averaged over the number of day. The step is of 1h, therefore there is 1 comparison less each time. The plotted temporal autocorrelation should follow a 24h cycle for the temperature, as we can see with the figure II.5, with the lowest autocorrelation for a $n+12$ h lag. For the precipitation rate, no cycles should appear with this metric

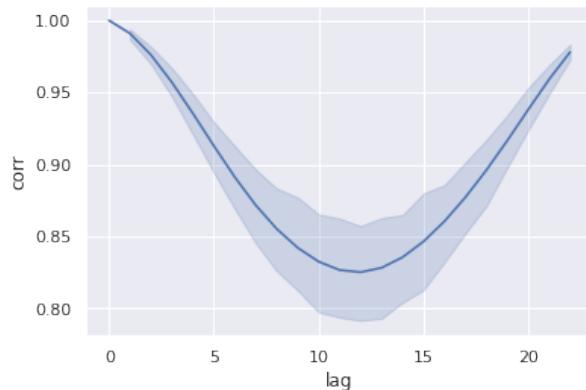


Figure II.5: Temporal autocorrelation for a random day, with the 95% confidence interval

Spatial autocorrelation is calculated with the same dataset. The lags are in km, and one score is taken on latitude and longitude. The dataset is this time lagged following one of its dimensions. The different datasets are going to decrease in size following the chosen spatial dimension, it wont be interesting to put a too high value of autocorrelation.

The autocorrelation for lags lower than 12km is completely missing from the upscaled dataset, it will be really interesting to see how the downscaling methods are performing without this information. We should also expect to see autocorrelation constantly decreasing following the latitude, and reaching a minimum following the longitude for the temperature. This phenomenon is due to the gradient of temperature following the latitude.

All the autocorrelations from the downscaling methods are plotted with the original data to have a visual comparison. To keep track of the best methods, a score is calculated using the RMSE between the different autocorrelations. The autocorrelation is computed with the numpy function `corrcoef`. The output are the Pearson product-moment correlation coefficients : $r = \frac{1}{n-1} \sum \left(\frac{x-\mu_x}{\sigma_x} \right) \left(\frac{y-\mu_y}{\sigma_y} \right)$. It will represent a graph drawn from multiple

autocorrelation coefficients

Part III

Downscaling implementation

All the preliminary work is done for the use of the downscaling methods, it is now technically possible to train the downscaling methods.

All the plotted result for each methods can be found in the Appendix

III.1 Baselines methods

Once the baseline are done, it is not enough to only try the downscaling models directly. Those models need to be compared to something simpler to conclude to check if they are efficient. Three baselines are taken : bilinear interpolation, bicubic interpolation and quintic interpolation. Bilinear interpolation $f_{bilinear}(x, y) = axy + b_1x + b_2y + c$ is the 2-dimensional version of linear interpolation, bicubic is taking a cubic function in 2 dimensions $f_{bicubic}(x, y) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij}x^i y^j$, and for quintic $f_{quintic}(x, y) = \sum_{i=0}^5 \sum_{j=0}^5 a_{ij}x^i y^j$. The result is expected to be smoother than the original data. The following results are calculated for an entire week of data.

III.1.1 Temperature downscaling

The temperature tends to be irregular in mountainous areas, but some of this information was lost during the upscaling. The smoothness of the values from the upscaled data is clearly seen with the bilinear method. The bicubic method seems to extrapolate slightly more than the bilinear method, but still less than the quintic method. This overall smoothness result was expected with the different expressions for each method. The slightly better performance from the quintic is not a surprise.

| method | RMSE | MAE | SSIM | Hellinger | Perkins |
|----------|-------|-------|--------|-----------|---------|
| bilinear | 0.756 | 0.378 | 0.5444 | 0.028 | 0.976 |
| bicubic | 0.76 | 0.378 | 0.5541 | 0.024 | 0.979 |
| quintic | 0.764 | 0.38 | 0.550 | 0.023 | 0.98 |

Table III.1: Baselines results for the temperature downscaling

The table III.3 gives some interesting insights. We first see that all the results are quite similar. Overall, the quintic method seems to be better in almost each metrics. The quintic method performs better at reproducing the original density, it is the best version

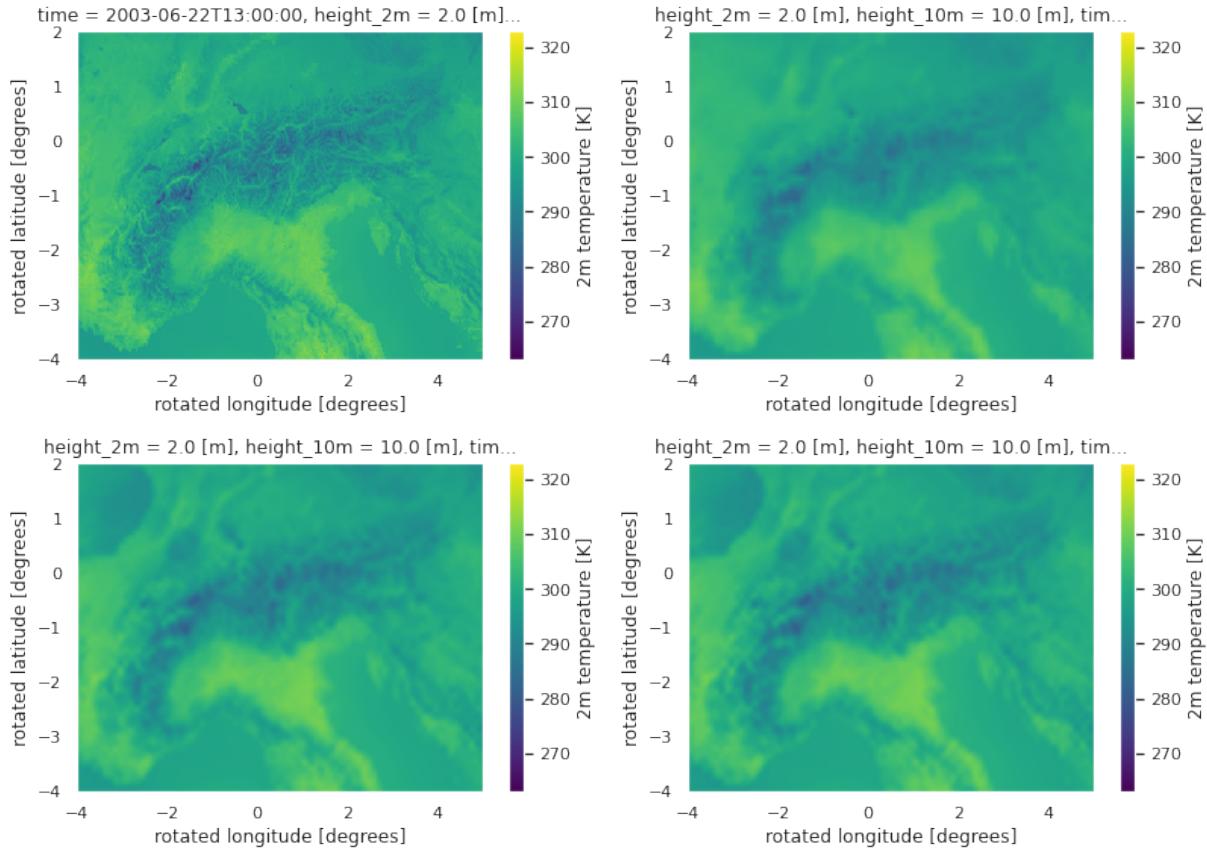


Figure III.1: Temperature in a mountainous region, with the original data and the three methods

for the RMSE and MAE, that could indicate that the highest or lowest values from the original data are less further away from the downscaling ones.

It was unexpected to see that the bilinear is seen as performing better with the SSIM metric, that could mean that on the SSIM aspects (luminance, contrast and structure) the bilinear method tends to resemble more the original data.

Only the spatial autocorrelation for the longitude in figure III.2 is represented here because the latitude results are really similar. Only the end of the autocorrelation is changing due to the temperature climate evolution following the latitude as mentioned earlier in this report. We see that the global variability is kept on the large scale, however for the tiny steps, the loss of information is present.

The upscaled dataset spatial autocorrelation is above the original dataset spatial autocorrelation, and the 3 downscaling methods are following the upscaled path. All the high and low values have been smoothed, it explains why the autocorrelation curves are above the original one, with the highest being the bilinear. The autocorrelation remains quite high even after a 700km lag.

The RMSE between the different autocorrelation and the original data autocorrelation is really low, approximately $1 * 10^{-4}$ for latitude and longitude. The autocorrelation between the variables seems to be well kept on average.

Finally, with the figure III.3, the temporal autocorrelation is present. The 24h cycles are easily seen because of the day and night temperature cycle. The MSE plot also shows

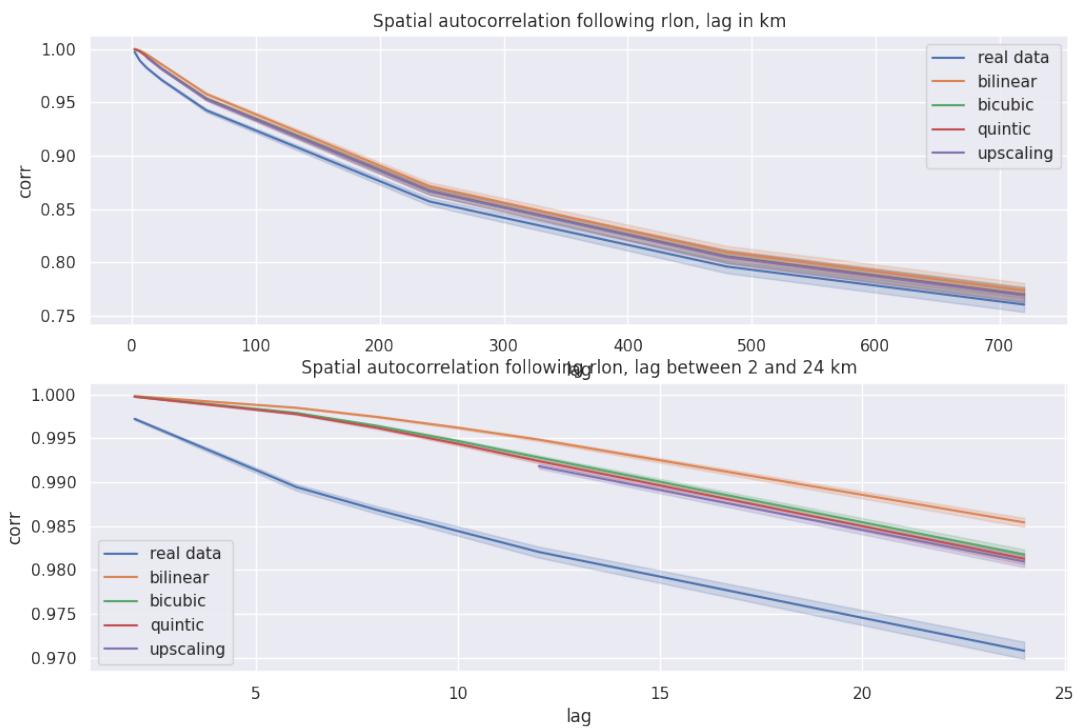


Figure III.2: Spatial autocorrelation following longitude (Temperature)

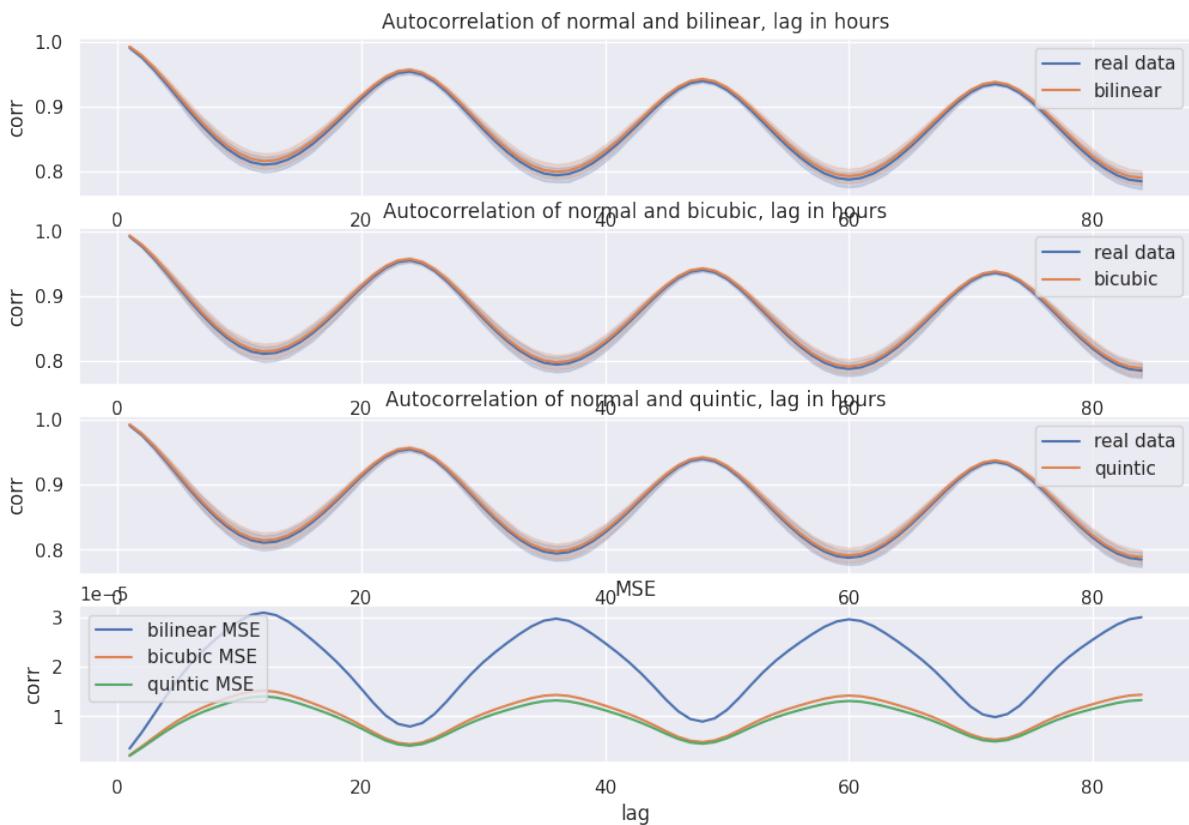


Figure III.3: Temporal autocorrelation for each method. The last plot represents the MSE between the original week autocorrelation and each method

that the bilinear interpolation is less accurate than the other methods.

III.1.2 Relative humidity downscaling

The relative humidity, as the temperature, seems to have high and low values that are smoothed during the upscaling process.

The RMSE and MAE results are unexpected. Once again, the quintic interpolation is not the most effective compared to the other methods according to the SSIM metric.

| method | RMSE | MAE | SSIM | Hellinger | Perkins |
|----------|------|------|----------|-----------|----------|
| bilinear | 3.46 | 1.97 | 0.380925 | 0.034 | 0.973364 |
| bicubic | 3.48 | 1.97 | 0.401112 | 0.028 | 0.977290 |
| quintic | 3.5 | 1.98 | 0.398616 | 0.028 | 0.977597 |

Table III.2: Baselines results for the relative humidity downscaling

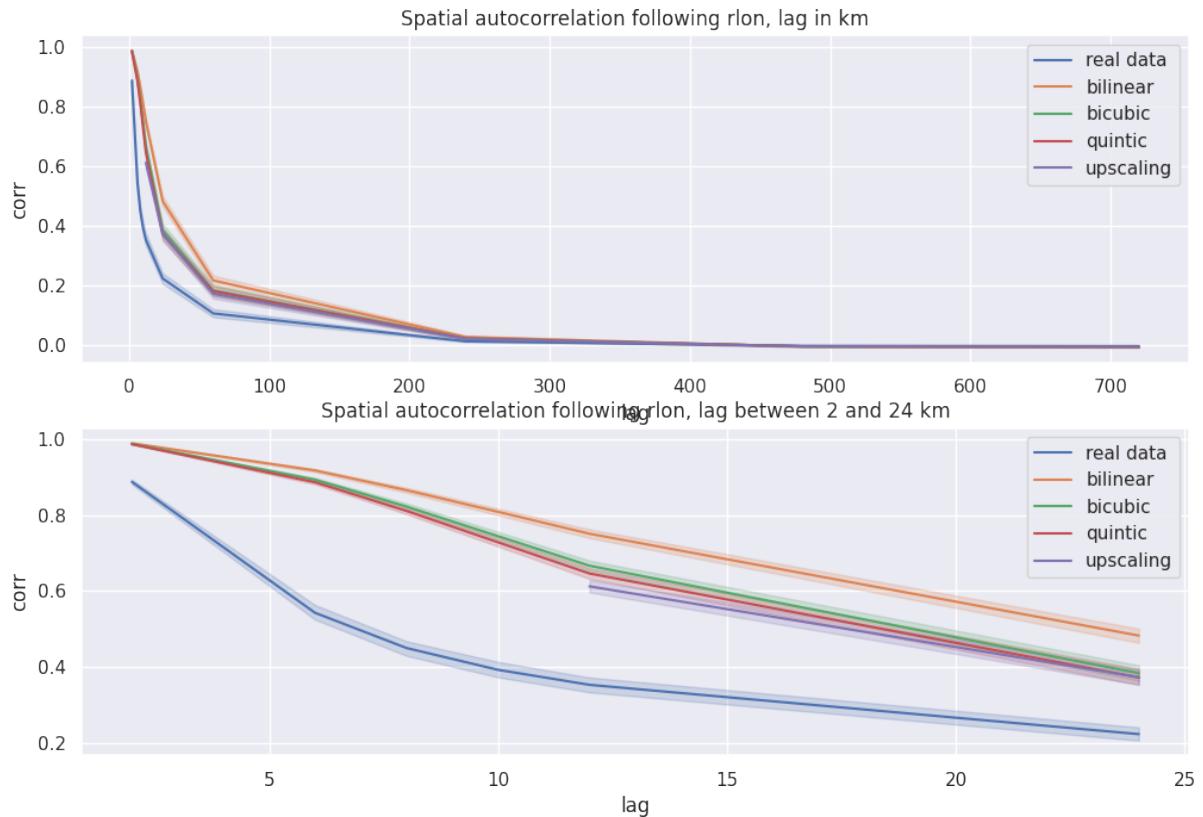


Figure III.4: Spatial autocorrelation following longitude (RELHUM)

Here, on the spatial autocorrelation figure, the evolution of the relative humidity autocorrelation is reaching zero after approximately 240km. Contrarily to the temperature, there is not a gradient increasing following the latitude. The spatial autocorrelation on the latitude is following the same trend, and so didn't add any information. The interpolation methods are once again following the trend from the upscaled data.

Finally, a 24h cycle is still present with the temporal autocorrelation in figure III.5. However, compared to the figure III.3, the autocorrelation is around 0.7 in average, which is less than the temperature temporal autocorrelation. The MSE plot shows that the bilinear temporal autocorrelation is once again less accurate.

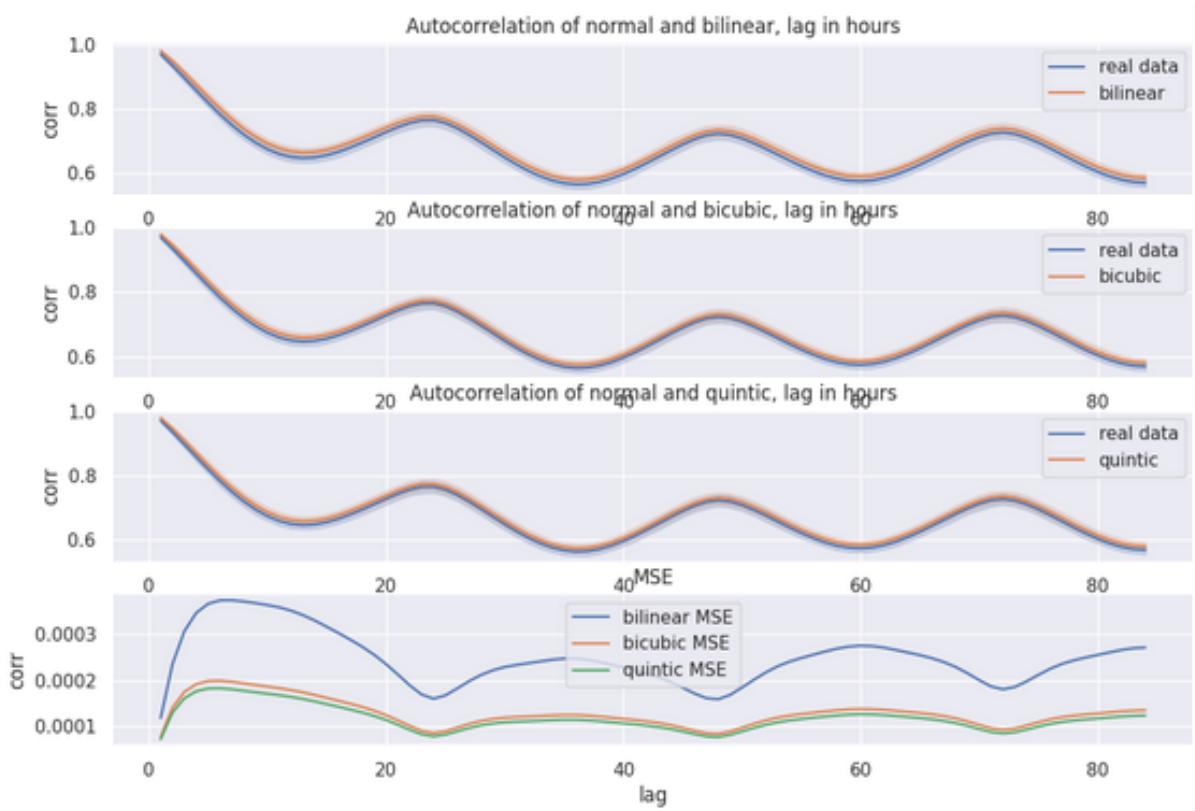


Figure III.5: Histogram of the total precipitation in the original dataset

III.1.3 Total precipitation downscaling

The total precipitation is really present on small parts of Europe at a time, with the major part of the studied area without precipitation. The upscaled data loses the precision, and so larger zones, of 12km at least, are under precipitation with value lower than the original data. The large part of zero values can lead to a difficulty to interpret the pixel-wise and the autocorrelation metrics. However, the metrics focusing on the total precipitation density can be more useful for information.

| method | RMSE | MAE | SSIM | Hellinger | Perkins |
|----------|--------|----------|----------|-----------|----------|
| bilinear | 0.0002 | 0.000013 | 0.999974 | 0.04 | 0.514047 |
| bicubic | 0.0002 | 0.000014 | 0.999973 | 0.094 | 0.488452 |
| quintic | 0.0002 | 0.000015 | 0.999973 | 0.105 | 0.483243 |

Table III.3: Baselines results for the total precipitation downscaling

Indeed, the RMSE and MAE are really bad a getting information on the downscaling for the baselines. There is no precipitations on the major part of the studied area, the SSIM will not be useful.

It is the same issue with the spatial autocorrelation, but we can at least say that the variability of this variable is really low. The temporal autocorrelation is also really low, and after less than 10 hours, it reaches a zero value.

Hellinger will not give useful insights on the downscaling results with the baselines, because as shows the figure III.6, the densities are somewhat similar. The Perkins skill score can be more useful in that case. The distributions are really localised in small zones, it is the reason why the Perkins skill score is only around 0.5 for each method.

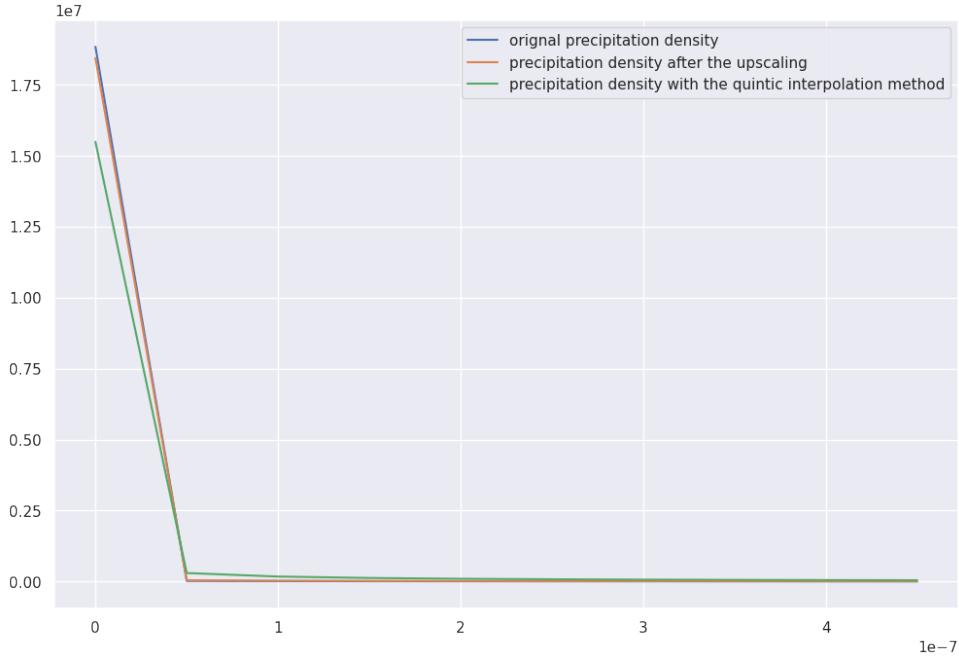


Figure III.6: Zoomed histogram of the total precipitation for an entire week

The figure III.6 illustrates the small variability of the distribution. The lower Perkins score for the bicubic and quintic methods could be explained by the fact that some negative values are given during the interpolation. It is also clear with the quintic method that the interpolation tends to give higher values, similar to the bicubic interpolation, that will decrease the Perkins score.

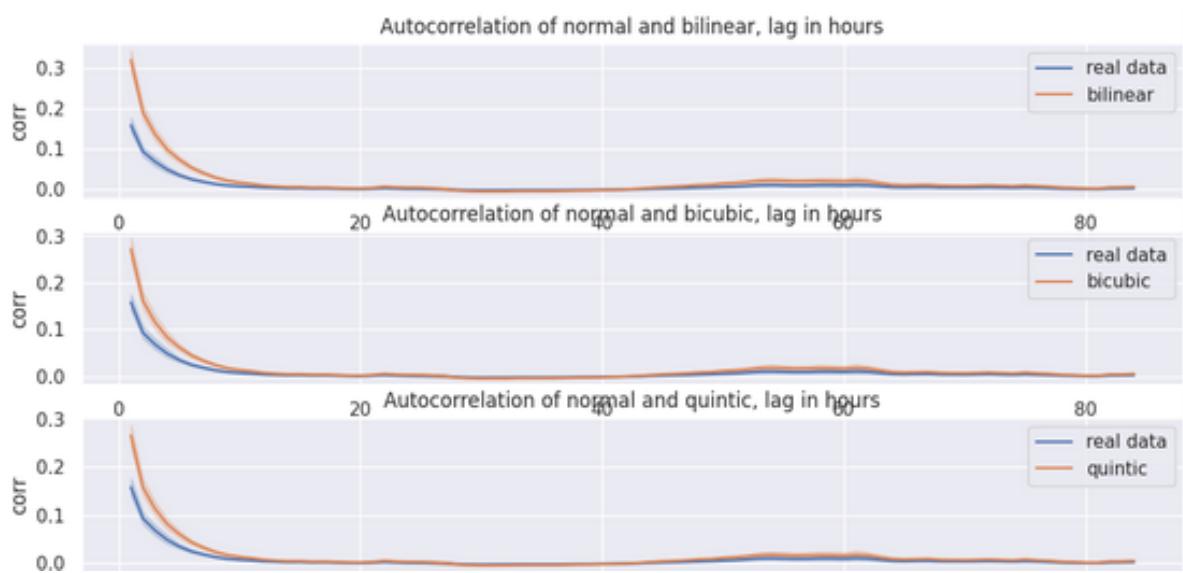


Figure III.7: Temporal autocorrelation for each method (TOT_PR)

III.1.4 Future climate models

| method | RMSE | MAE | SSIM | Hellinger | Perkins |
|----------|-------|----------|----------|-----------|----------|
| bilinear | 0.79 | 0.611421 | 0.550487 | 0.972787 | 0.975109 |
| bicubic | 0.795 | 0.611609 | 0.559947 | 0.976849 | 0.978556 |
| quintic | 0.799 | 0.609172 | 0.556110 | 0.977219 | 0.978766 |

Table III.4: Baselines results for the temperature downscaling

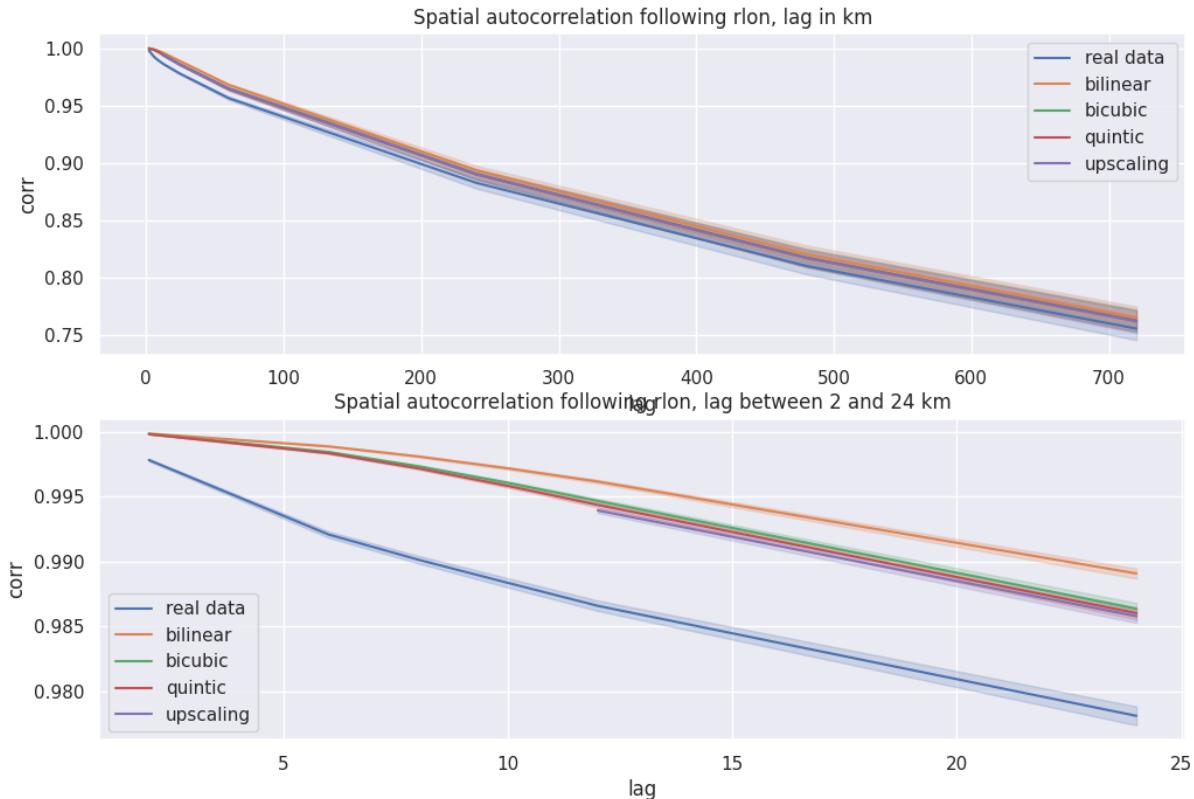


Figure III.8: Temporal autocorrelation for each method (Temperature), with the 2083 model

The results obtained by downscaling the 2083 dataset with the baselines is really similar to the 2003 results. It can be explained by the fact that the interpolation works without any training image, and so the efficiency of this method is not affected by the increase or decrease in some variables. The variables are different, this explains of course why the values for the different metrics are slightly different.

It can be more interesting to see how the downscaling methods that takes training images to see if the performance decreases.

III.2 Other downscaling algorithms

The metrics and some baselines are implemented. The different GitHub repositories from past articles that already used SRGAN, CNN and QS are cloned on the cluster. The QS algorithm already has the code to try the downscaling of one image only, and can be

used on a local machine.

The training, testing and validation datasets in .netCDF format are already in the cluster, but they needs to be converted to .tfrecord in order to use the deep learning methods. There is already a function, called generate_TFRecords, in the PhIRE GitHub can do this conversion from a numpy array dataset. The metrics are already implemented, the only addition will be to convert the output files into .netCDF datasets to obtain the results.

Conclusion

This internship was a great opportunity to be introduced to climate models and climate change prediction problems. I learnt a lot about getting information and useful articles when introduced to new algorithms, and the preliminary work necessary before using new methods. It was a real chance to work in the same office as PhD students, and to be truly immersed into a research environment.

To finalize, I have been able to get into the data and implement the baselines. To begin with, working from a cluster was challenging since I didn't have access to the project data directly. The metrics implementation has also been longer than expected, having to work with new data types was a challenge for me. The cluster was an issue for me to start the real downscaling methods, I was unable to modify the environment to have the correct packages before the report writing.

The next phase for the current work will be to try the QS method locally on my computer, and start the writing of a small paper if I manage to get some results with the downscaling method to attend a Climate Change AI workshop.

The following stages for the project will be to train the SRGAN and CNN methods with the converted training dataset, and to obtain more results over a larger dataset. Statistical downscaling will then be possible to do with multiple iteration of one method.

Bibliography

- [1] Tom Beucler. Robust statistical downscaling in a changing climate. Unpublished proposal, 2022.
- [2] Climate models and their evaluation. climate change 2007: The physical science basis. contribution of working group i to the fourth assessment report of the intergovernmental panel on climate change.
- [3] Bjorn STEVENS, Claudia ACQUISTAPACE, Akio HANSEN, Rieke HEINZE, Carolin KLINGER, Daniel KLOCKE, Harald RYBKA, Wiebke SCHUBOTZ, Julia WINDMILLER, Panagiotis ADAMIDIS, Ioanna ARKA, Vasileios BAR-LAKAS, Joachim BIERCAMP, Matthias BRUECK, Sebastian BRUNE, Stefan A. BUEHLER, Ulrike BURKHARDT, Guido CIONI, Montserrat COSTA-SURÓS, Susanne CREWELL, Traute CRÜGER, Hartwig DENEKE, Petra FRIEDERICH, Cintia Carbajal HENKEN, CATHY Hohenegger, Marek JACOB, Fabian JAKUB, Norbert KALTHOFF, Martin KÖHLER, Thirza W. van LAAR, Puxi LI, Ulrich LÖHNERT, Andreas MACKE, Nils MADENACH, Bernhard MAYER, Christine NAM, Ann Kristin NAUMANN, Karsten PETERS, Stefan POLL, Johannes QUAAS, Niklas RÖBER, Nicolas ROCHEVIN, Leonhard SCHECK, Vera SCHEMANN, Sabrina SCHNITT, Axel SEIFERT, Fabian SENF, Metodija SHAPKALIJEVSKI, Clemens SIMMER, Shweta SINGH, Odran SOURDEVAL, Dela SPICKERMANN, Johan STRANDGREN, Octave TESSIOT, Nikki VERCAUTEREN, Jessica VIAL, Aiko VOIGT, and Günter ZÄNGL. The added value of large-eddy and storm-resolving models for simulating clouds and precipitation. *Journal of the Meteorological Society of Japan. Ser. II*, 98(2):395–435, 2020.
- [4] Jordis S. Tradowsky et al. Frank Kreienkamp, Sjoukje Y. Philip. Rapid attribution of heavy rainfall events leading to the severe flooding in western europe during july 2021. *World Weather Attribution*, 2021.
- [5] R. Brogli, C. Heim, S. L. Sørland, and C. Schär. The pseudo-global-warming (pgw) approach: Methodology, software package pgw4era5 v1.1, validation and sensitivity analyses. *Geoscientific Model Development Discussions*, 2022:1–28, 2022.
- [6] Karen Stengel, Andrew Glaws, Dylan Hettinger, and Ryan N. King. Adversarial super-resolution of climatological wind and solar data. *Proceedings of the National Academy of Sciences*, 117(29):16805–16815, 2020.
- [7] Sanjeev Jha, Gregoire Mariethoz, Jason Evans, Matthew McCabe, and Ashish Sharma. A space and time scale-dependent nonlinear geostatistical approach for downscaling daily precipitation and temperature. *Water Resources Research*, 51, 08 2015.

- [8] Kevin Höhlein, Michael Kern, Timothy Hewson, and Rüdiger Westermann. A comparative study of convolutional neural network models for wind field downscaling. *Meteorological Applications*, 27(6):e1961, 2020.
- [9] Fabio Oriani, Matthew F. McCabe, and Gregoire Mariethoz. Downscaling multispectral satellite images without colocated high-resolution data: A stochastic approach based on training images. *IEEE Transactions on Geoscience and Remote Sensing*, 59(4):3209–3225, 2021.
- [10] Gregoire Mariethoz. *When Should We Use Multiple-Point Geostatistics?*, pages 645–653. Springer International Publishing, 2018.
- [11] M. Gravey and G. Mariethoz. Quicksampling v1.0: a robust and simplified pixel-based multiple-point simulation approach. *Geoscientific Model Development*, 13(6):2611–2630, 2020.
- [12] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network, 2016.
- [13] Philip W. Jones. First- and second-order conservative remapping schemes for grids in spherical coordinates. *Monthly Weather Review*, 127(9):2204 – 2210, 1999.
- [14] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.
- [15] S. E. Perkins, A. J. Pitman, N. J. Holbrook, and J. McAneney. Evaluation of the ar4 climate models' simulated daily maximum temperature, minimum temperature, and precipitation over australia using probability density functions. *Journal of Climate*, 20(17):4356 – 4376, 2007.

List of Tables

| | |
|---|----|
| III.1 Baselines results for the temperature downscaling | 27 |
| III.2 Baselines results for the relative humidity downscaling | 31 |
| III.3 Baselines results for the total precipitation downscaling | 32 |
| III.4 Baselines results for the temperature downscaling | 34 |

List of Figures

| | | |
|-------|--|----|
| I.1 | Representation of the three variables (respectively T_2M, RELHUM_2M and TOT_PR) on May 22, 2003 | 14 |
| I.2 | Representation of T_2M on May 22, 2003 and May 22, 2083 | 14 |
| II.1 | Structure of a random hourly dataset and structure of a modified dataset (values of an entire day) | 20 |
| II.2 | Representation of T_2M in the original model and after the upscaling in a mountainous area | 21 |
| II.3 | Diagram of how the SIM works, from the original article [14] | 22 |
| II.4 | Bidimensional probability density function between temperature and relative humidity, for the original data the quintic interpolation method with 5% of the values | 23 |
| II.5 | Temporal autocorrelation for a random day, with the 95% confidence interval | 24 |
| III.1 | Temperature in a mountainous region, with the original data and the three methods | 28 |
| III.2 | Spatial autocorrelation following longitude (Temperature) | 29 |
| III.3 | Temporal autocorrelation for each method. The last plot represents the MSE between the original week autocorrelation and each method | 29 |
| III.4 | Spatial autocorrelation following longitude (RELHUM) | 31 |
| III.5 | Histogram of the total precipitation in the original dataset | 32 |
| III.6 | Zoomed histogram of the total precipitation for an entire week | 33 |
| III.7 | Temporal autocorrelation for each method (TOT_PR) | 33 |
| III.8 | Temporal autocorrelation for each method (Temperature), with the 2083 model | 34 |
| A.1 | Temporal autocorrelation for each method | 47 |
| A.2 | Spatial autocorrelation following rlon for each method | 48 |
| A.3 | Spatial autocorrelation following rlat for each method | 48 |
| A.4 | Temporal autocorrelation for each method | 49 |
| A.5 | Spatial autocorrelation following rlon for each method | 50 |
| A.6 | Spatial autocorrelation following rlat for each method | 50 |
| A.7 | Temporal autocorrelation for each method | 51 |
| A.8 | Spatial autocorrelation following rlon for each method | 52 |
| A.9 | Spatial autocorrelation following rlat for each method | 52 |
| A.10 | Temporal autocorrelation for each method | 53 |
| A.11 | Spatial autocorrelation following rlon for each method | 54 |
| A.12 | Spatial autocorrelation following rlat for each method | 54 |

Appendices

Appendix A

Autocorrelation plots for the baselines methods

A.1 Temperature autocorrelation plots

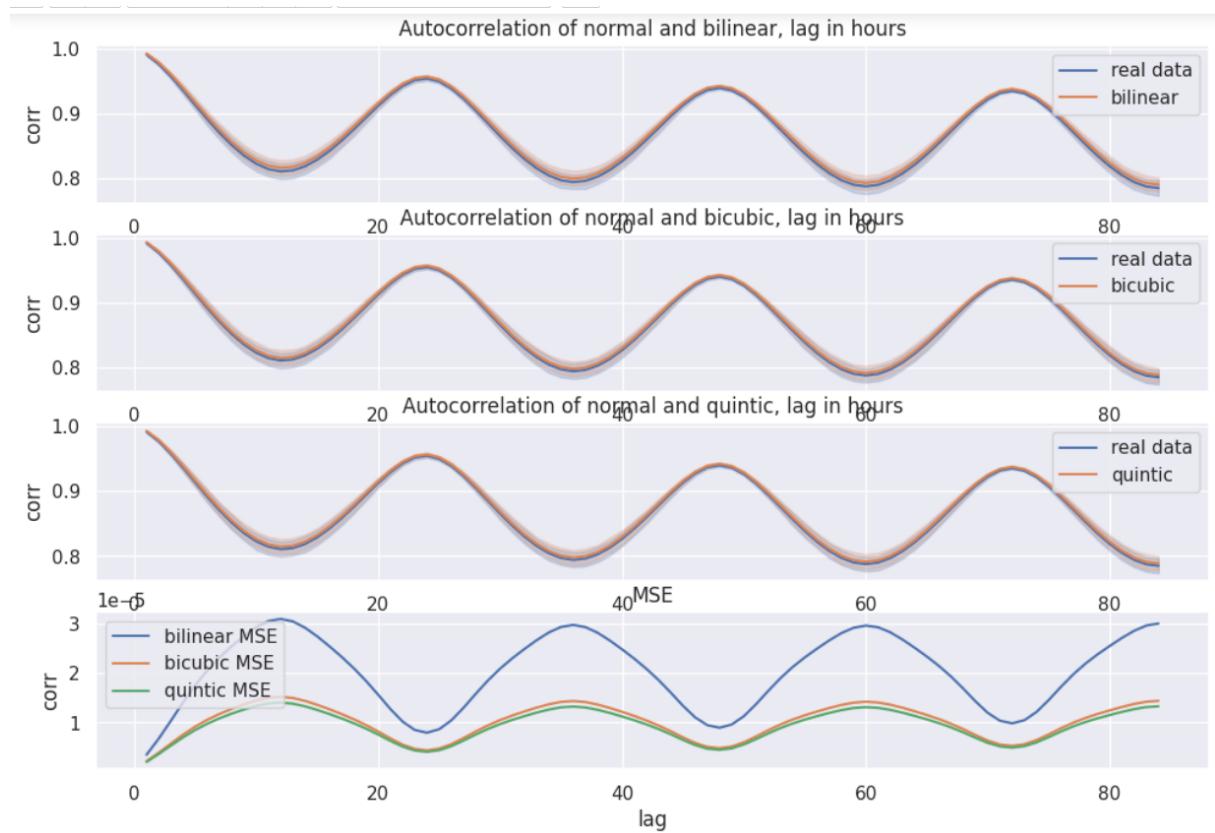


Figure A.1: Temporal autocorrelation for each method

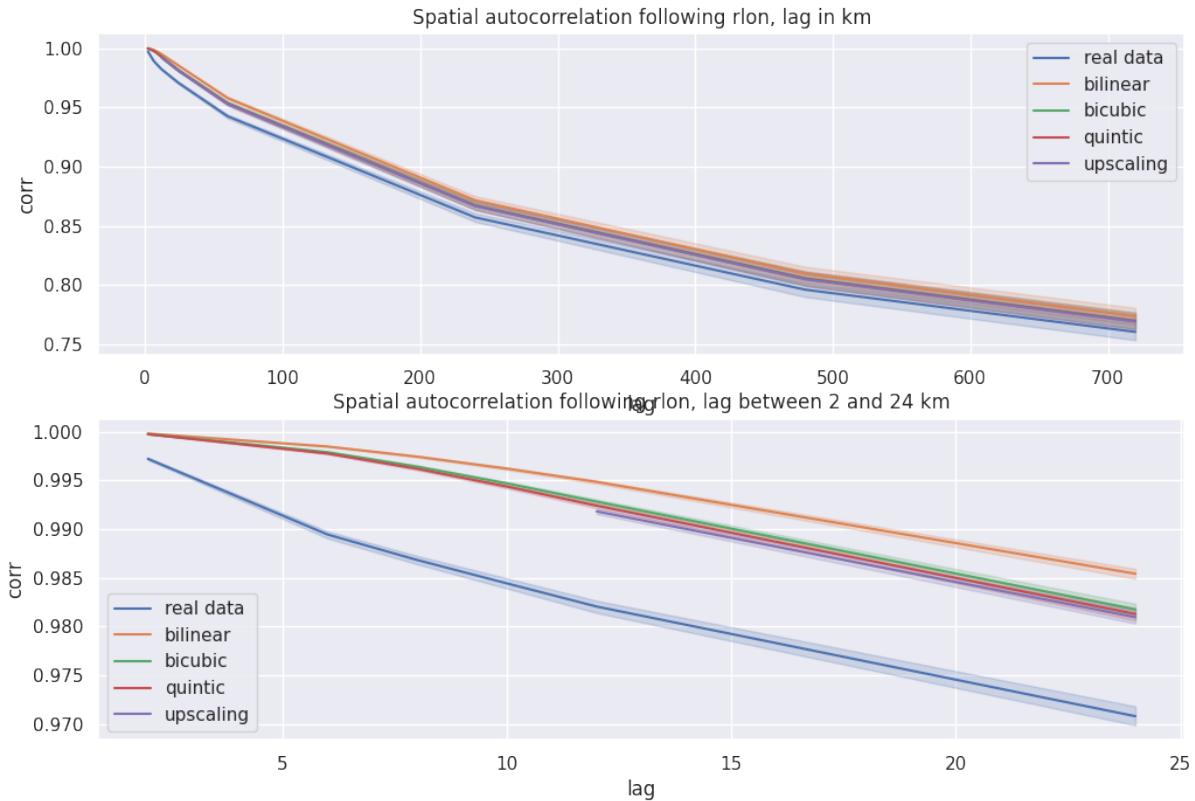


Figure A.2: Spatial autocorrelation following rlon for each method

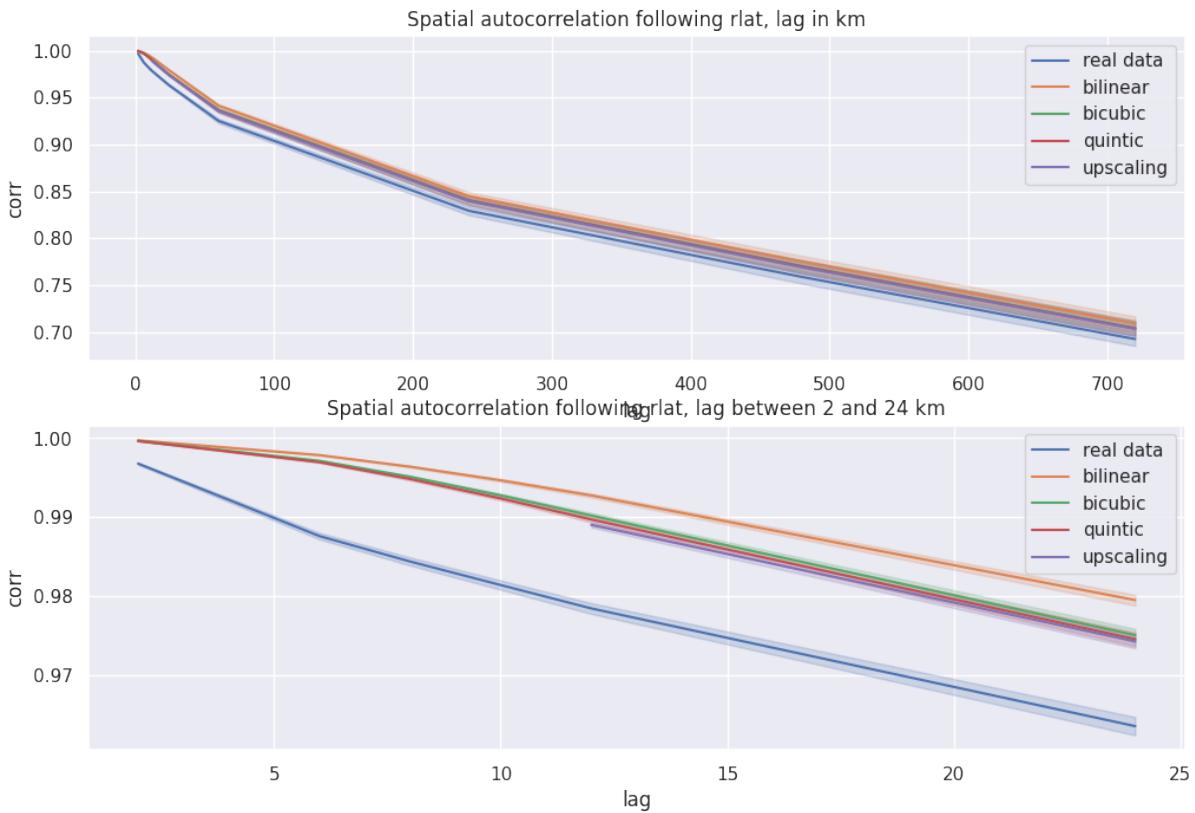


Figure A.3: Spatial autocorrelation following rlat for each method

A.2 Relative humidity autocorrelation plots

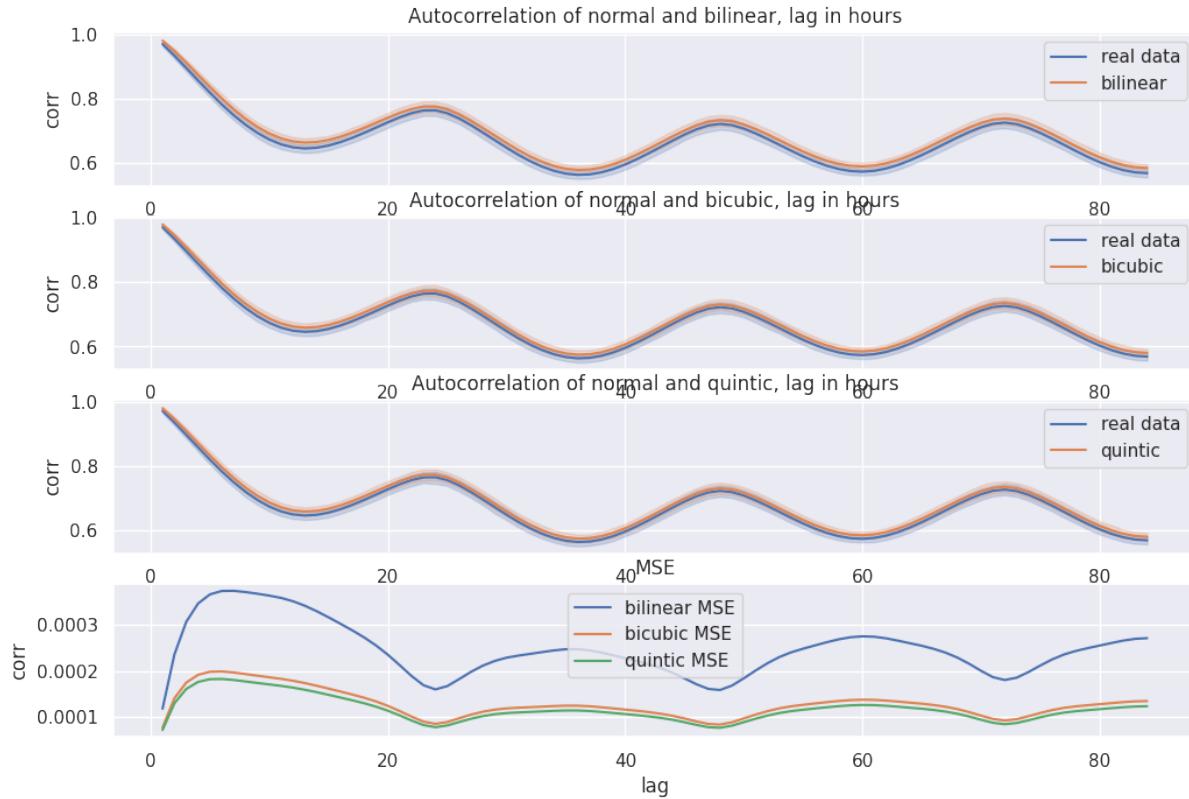


Figure A.4: Temporal autocorrelation for each method

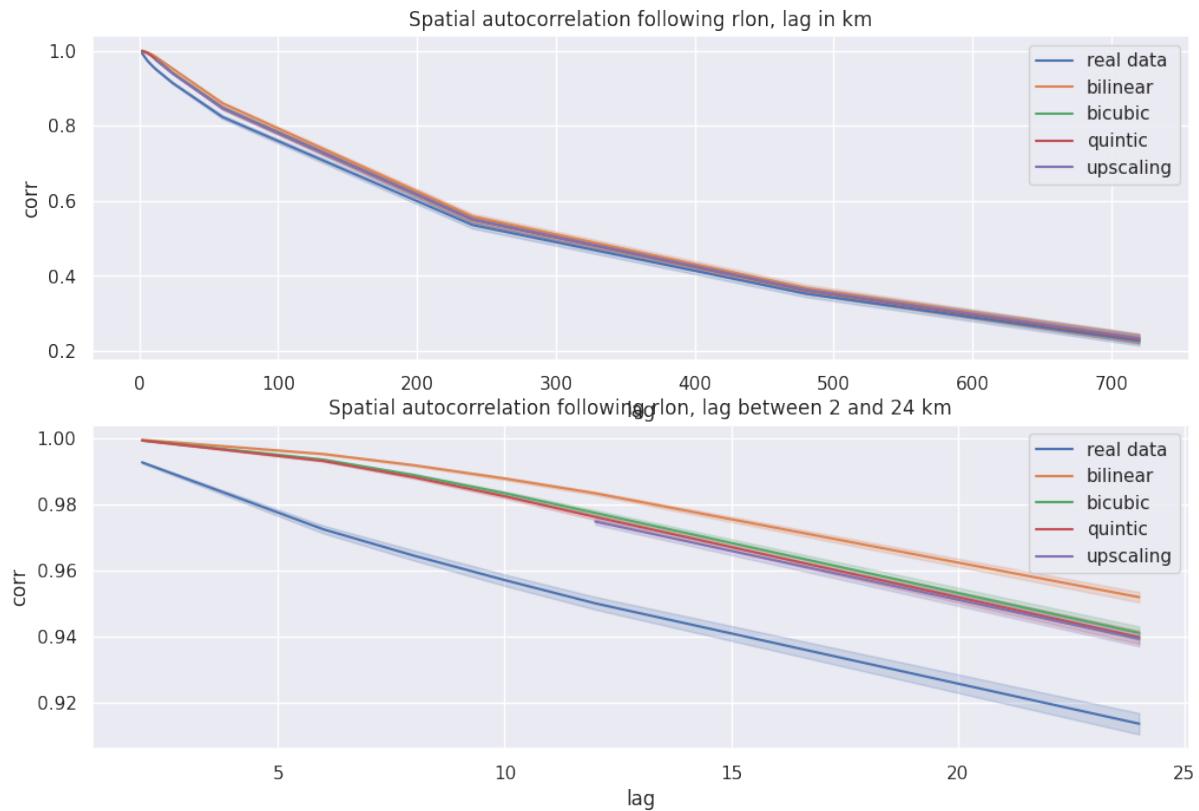


Figure A.5: Spatial autocorrelation following rlon for each method

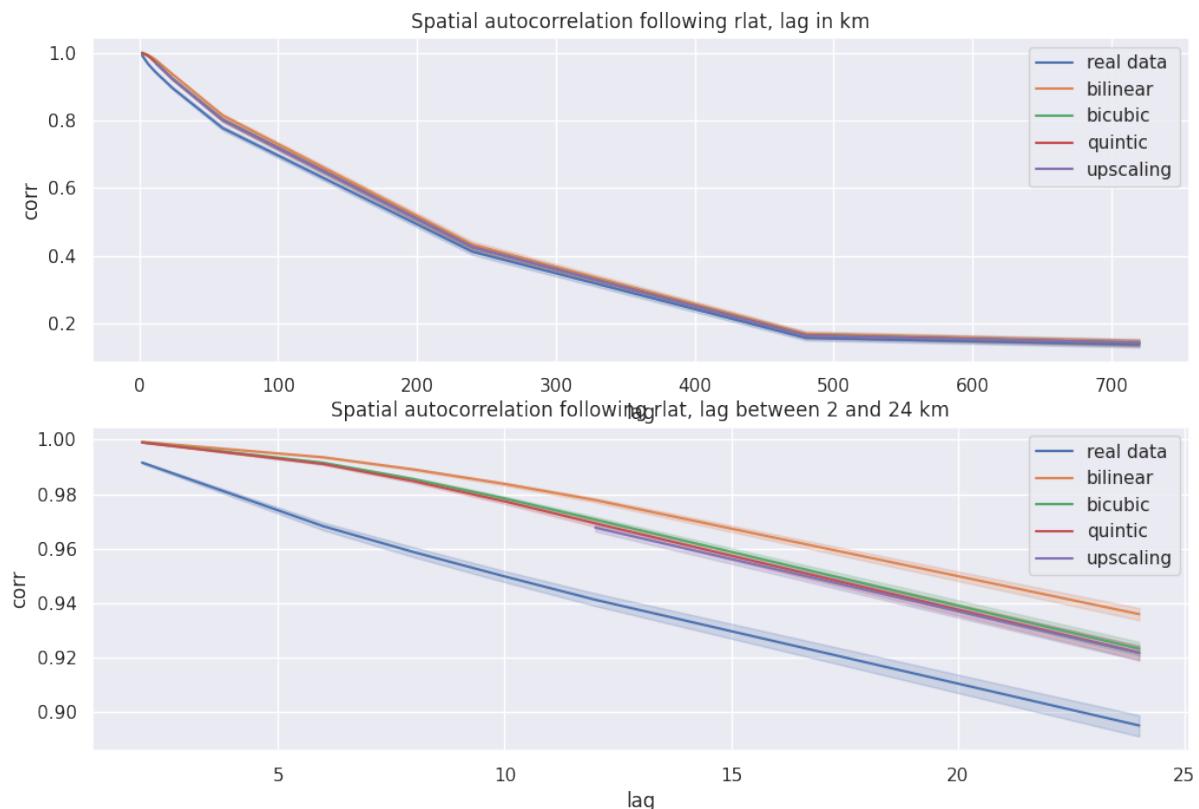


Figure A.6: Spatial autocorrelation following rlat for each method

A.3 Total precipitation autocorrelation plots

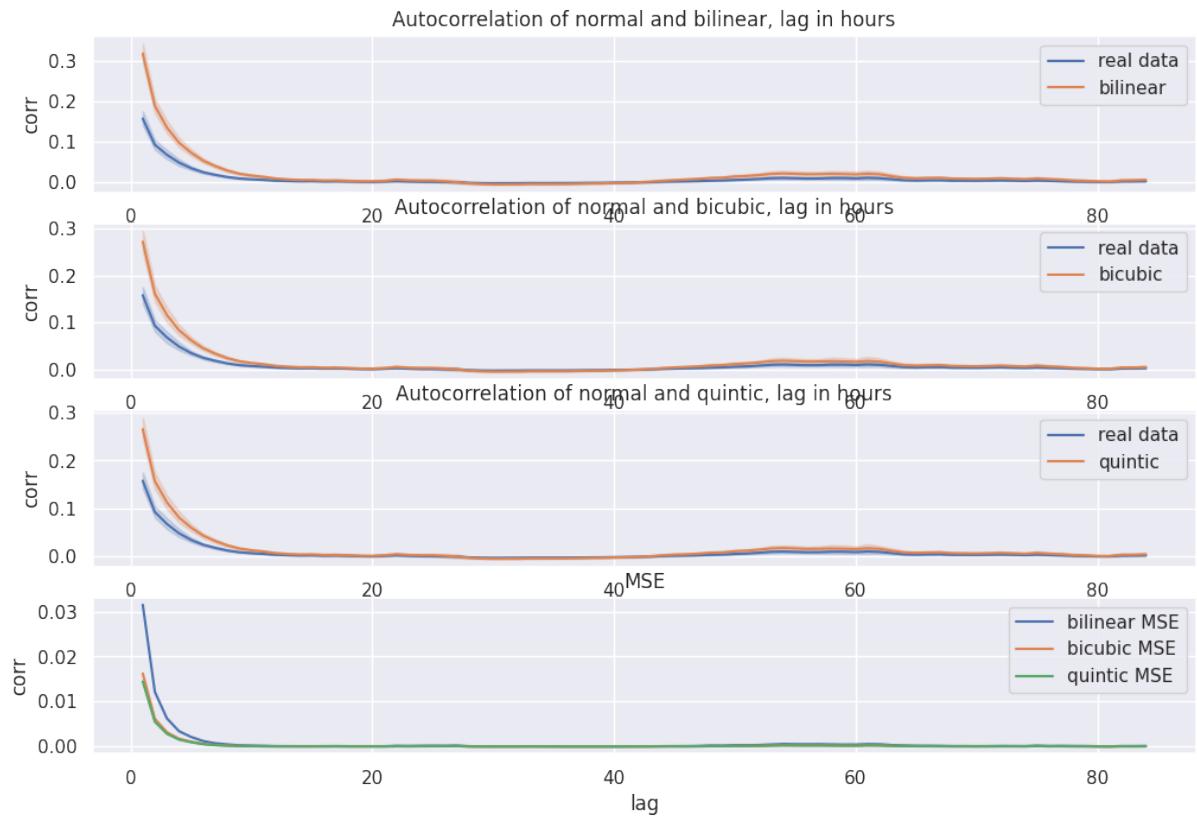


Figure A.7: Temporal autocorrelation for each method

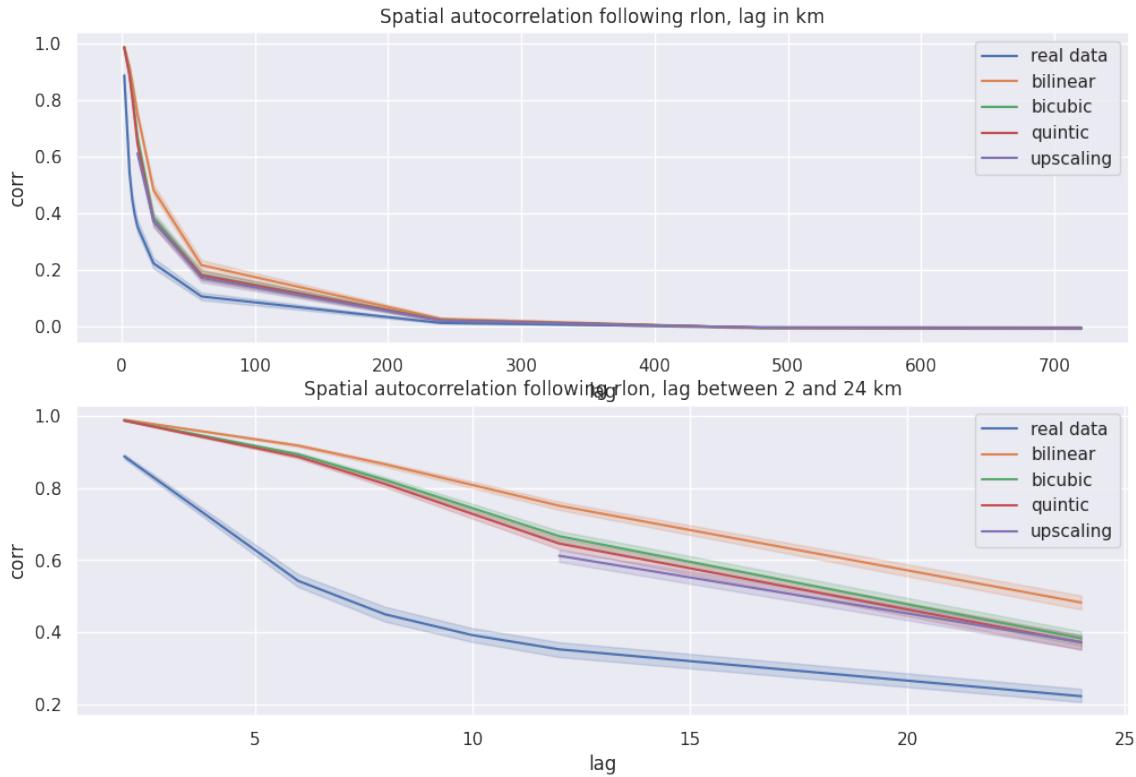


Figure A.8: Spatial autocorrelation following rlon for each method

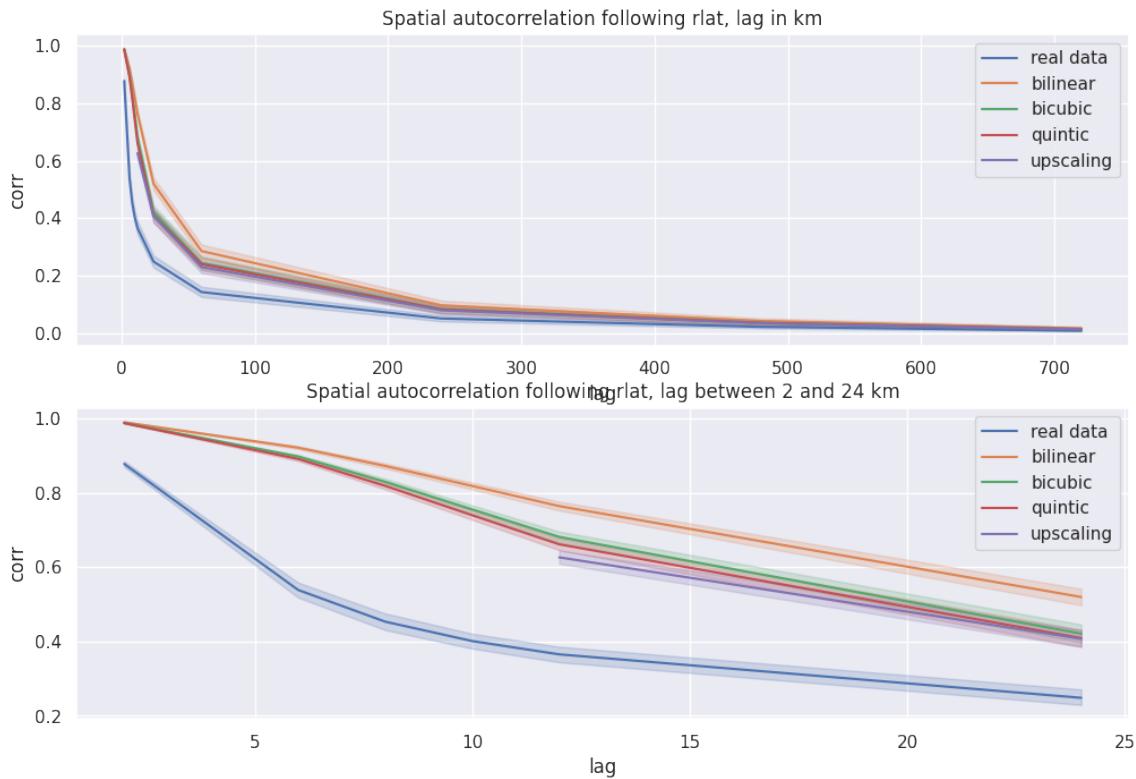


Figure A.9: Spatial autocorrelation following rlat for each method

A.4 Future temperature autocorrelation plots

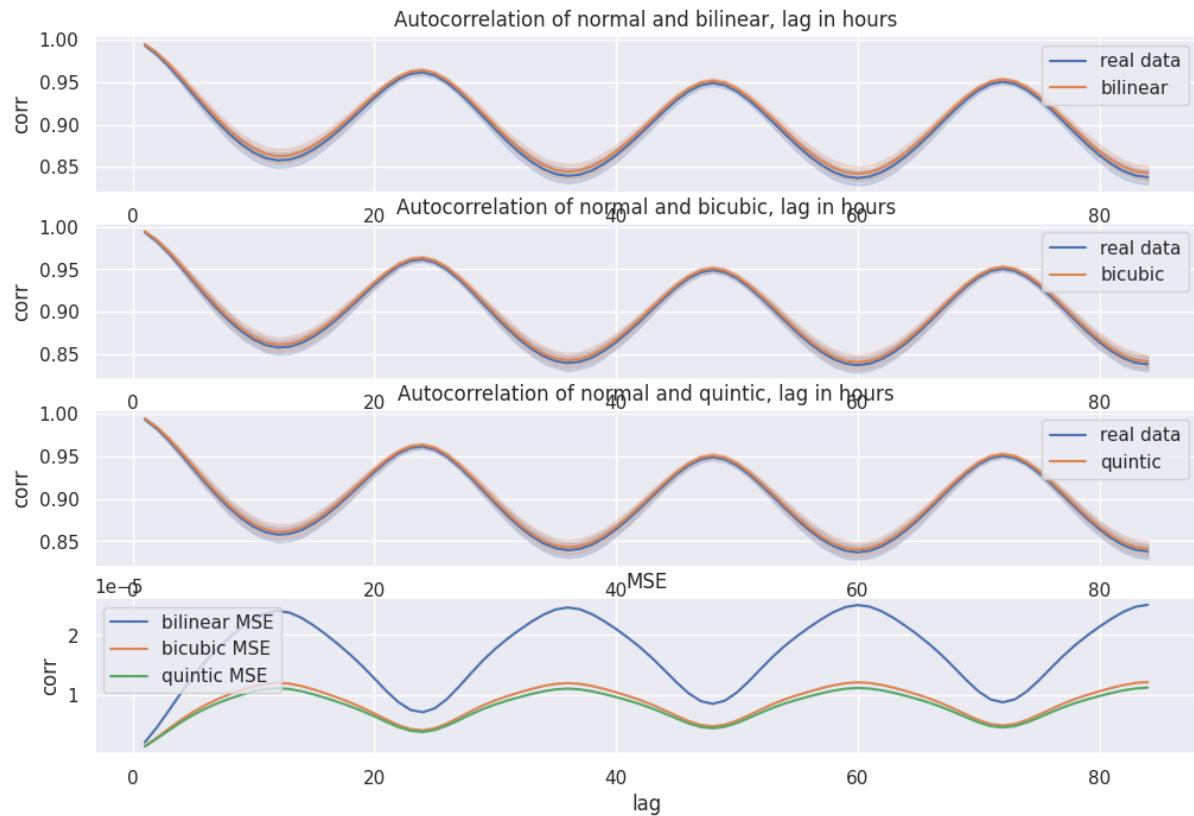


Figure A.10: Temporal autocorrelation for each method

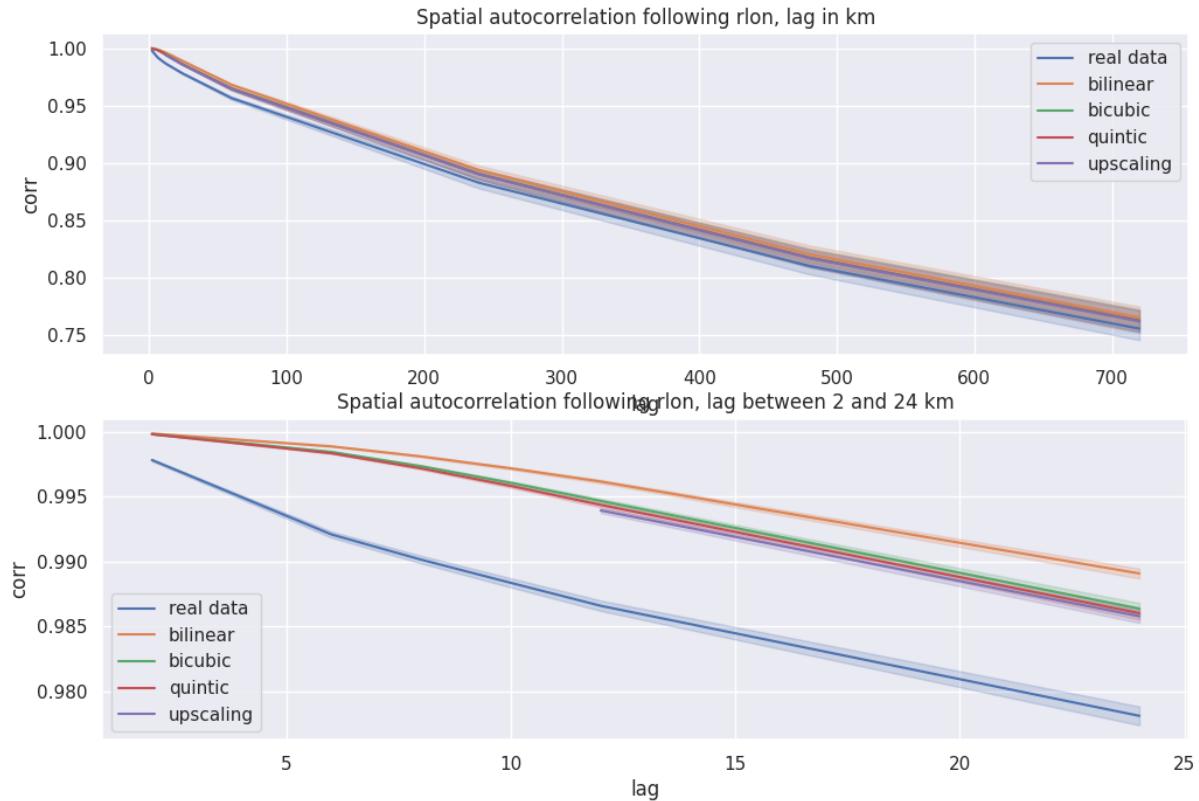


Figure A.11: Spatial autocorrelation following rlon for each method

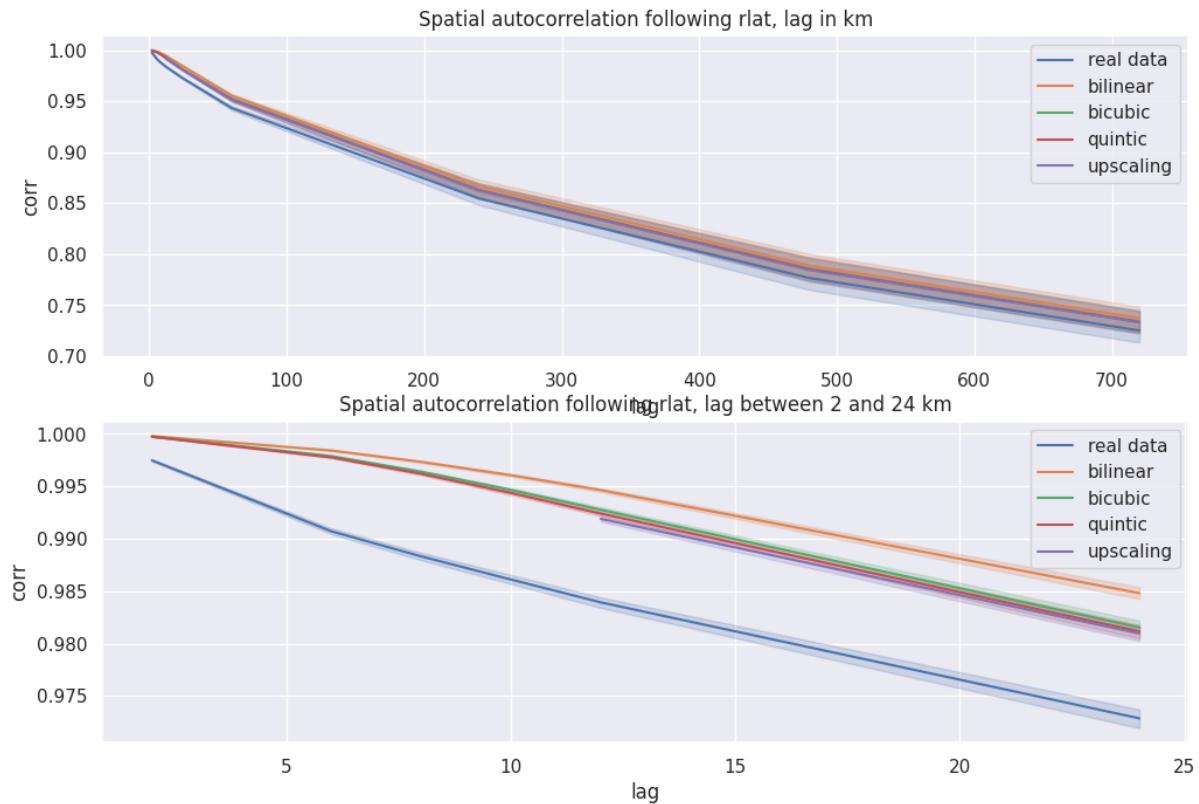


Figure A.12: Spatial autocorrelation following rlat for each method