Regional climate model emulator based on deep learning: concept and first evaluation of a novel hybrid downscaling approach

Antoine Doury[1] · Samuel Somot[1] · Sebastien Gadat[2] · Aurélien Ribes[1] · Lola Corre[3]

Abstract

Providing reliable information on climate change at local scale remains a challenge of first importance for impact studies and policymakers. Here, we propose a novel hybrid downscaling method combining the strengths of both empirical statistical downscaling methods and Regional Climate Models (RCMs). In the longer term, the final aim of this tool is to enlarge the high-resolution RCM simulation ensembles at low cost to explore better the various sources of projection uncertainty at local scale. Using a neural network, we build a statistical RCM-emulator by estimating the downscaling function included in the RCM. This framework allows us to learn the relationship between large-scale predictors and a local surface variable of interest over the RCM domain in present and future climate. The RCM-emulator developed in this study is trained to produce daily maps of the near-surface temperature at the RCM resolution (12 km). The emulator demonstrates an excellent ability to reproduce the complex spatial structure and daily variability simulated by the RCM, particularly how the RCM refines the low-resolution climate patterns. Training in future climate appears to be a key feature of our emulator. Moreover, there is a substantial computational benefit of running the emulator rather than the RCM, since training the emulator takes about 2 h on GPU, and the prediction takes less than a minute. However, further work is needed to improve the reproduction of some temperature extremes, the climate change intensity and extend the proposed methodology to different regions, GCMs, RCMs, and variables of interest.

Keywords Emulator · Hybrid downscaling · Regional climate modeling · Statistical downscaling · Deep neural network · Machine learning · EURO-CORDEX · FPS convection · CORDEX

1 Introduction

Climate models are an essential tool to study possible evolutions of the climate according to different scenarios of greenhouse gas emissions. These numerical models represent the physical and dynamical processes present in the atmosphere and their interactions with other components of the Earth System. The complexity of these models involves compromises between the computational costs, the horizontal resolution and, in some cases, the domain size.

Global climate models (GCMs) produce simulations covering the whole planet at reasonable cost thanks to a low spatial resolution (from 50 to 300 km). The large number of different GCMs developed worldwide allows to build large and coordinated ensembles of simulations, thanks to a strong international cooperation. These big ensembles (CMIP3/5/6, Meehl et al. 2007; Taylor et al. 2012; Eyring et al. 2016) are necessary to correctly explore the different sources of variability and uncertainties in order to deliver reliable information about future climate change at large spatial scales. However, the resolution of these models is too coarse to derive any fine scale information, which is of primary importance for impact studies and adaptation policies. Consequently, it is crucial to downscale the GCM outputs to a higher resolution. Two families of downscaling have emerged: empirical-statistical downscaling and dynamical downscaling. Both approaches have their own strengths and weaknesses.

Empirical Statistical Downscaling methods (ESD) estimate functions to link large scale atmosphere fields with

✉ Antoine Doury
    antoine.doury@meteo.fr

[1] CNRM, Université de Toulouse, Météo-France, CNRS, Toulouse, France

[2] Toulouse School of Economics, Université Toulouse 1 Capitole, Institut Universitaire de France, Toulouse, France

[3] Météo-France, Toulouse, France

local scale variables using observational data. Local implications of future climate changes are then obtained by applying these functions to GCM outputs. Gutiérrez et al. (2019) present an overview of ESD methods and evaluate their ability to downscale historical GCM simulations. The great advantage of these statistical methods is their computational efficiency, which makes the downscaling of large GCM ensembles possible. On the other hand, they have two main limitations due to their dependency on observational data. First of all, they are applicable only for regions and variables for which local long-term observations are available. Secondly, they rely on the stationary assumption of the large-scale/local-scale relationship, which implies that a statistical model calibrated in the past and present climate remains reliable in the forthcoming climate. Studies tend to show that the calibration period has a non-negligible impact on the results (Wilby et al. 1998; Schmith 2008; Dayon et al. 2015; Erlandsen et al. 2020).

Dynamical downscaling (DD) is based on Regional Climate Models (RCMs). These models have higher resolution than GCMs (1–50 km) but are restricted to a limited area domain to keep their computational costs affordable. They are nested in a GCM, e.g., they receive at small and regular time intervals dynamical information from this GCM at their lateral boundaries. One key advantage of RCMs is to rely on the same physical hypotheses as the one involved in GCMs. They provide a complete description of the state of the atmosphere over their domain through a large set of variables at high temporal and spatial resolution. The added value of RCMs has been demonstrated in several studies (Giorgi et al. 2016; Torma et al. 2015; Prein et al. 2016; Fantini et al. 2018; Kotlarski et al. 2015, for examples). In order to deliver robust information about future local responses to climate change, it is necessary to explore the uncertainty associated with RCM simulations. Déqué et al. (2007) and Evin et al. (2021) assess that four sources of uncertainty are at play in a regional climate simulation: the choice of the driving GCM, the greenhouse gas scenario, the choice of the RCM itself and the internal variability. Their relative importance depends on the considered variables, spatial scale, and timeline. According to these results, it is important (Déqué et al. 2012; Evin et al. 2021; Fernández et al. 2019) to complete 4D matrices [SCENARIO, GCM, RCM, MEMBER] to deliver robust messages, where members are several simulations of each (*SCENARIO*, *GCM*, *RCM*) triplet. However, the main limitation of RCM is their high computational costs which makes the completion of such matrices impossible.

This study proposes a novel hybrid downscaling method to enlarge the size of RCM simulation ensembles. The idea is to combine the advantages of both dynamical and statistical downscaling to tackle their respective limits. This statistical RCM-emulator uses Machine Learning methods to learn the relationship between large scale fields and local

scale variables inside regional climate simulations. It aims to estimate the downscaling function included in the RCM in order to apply it to new GCM simulations. This framework will allow to learn this function on the entire RCM domain, in past and future climate, under different scenarios. Besides, the emulator relies on Machine Learning algorithms with low computational costs, which will enable to increase RCM simulation ensembles and to better explore the uncertainties associated with these high resolution simulations.

Hybrid statistical-dynamical downscaling methods have already been proposed. They are methods which combine, in different ways, regional climate models and statistical approaches to obtain local climate information. Several studies such as Pryor and Barthelmie (2014), Vrac et al. (2012) or Turco et al. (2011) perform 2-step downscaling by applying ESD methods to RCM simulations. Colette et al. (2012) apply bias correction methods to GCM outputs before downscaling with RCMs. Maraun and Widmann (2018) are among the first to mention the concept of emulators. Few studies have combined ESD and DD for the same purpose as in this study. For instance, Walton et al. (2015) propose a statistical model which estimates from GCM outputs, the high resolution warming pattern for each month in California. It is calibrated using RCM simulations and relies on a simple linear combination of two predictors from the GCM (the monthly mean over the domain and an indicator for the land/sea contrast) plus an indicator for the spatial variance (obtained thanks to PCA). Berg et al. (2015) adapt the same protocol for monthly changes in precipitation over California. With respect to those pioneer studies, we propose here to further develop this approach by using a neural network based method and by emulating the full time series at the input time scale, allowing to explore daily climate extremes.

In recent years, climate science has taken advantage of the recent strides in performances of Deep Learning algorithms (see Lecun et al. 2015, for an overview). Indeed, thanks to their capacity to deal with large amounts of data and the strong ability of Convolutional Neural Network (CNN) (LeCun et al. 1998) to extract high level features from images, these algorithms are particularly adapted to climate and weather sciences. Reichstein et al. (2019) present a complete overview of the climate studies applying Deep Learning and future possibilities. In particular, Vandal et al. (2017, 2019) and Baño-Medina et al. (2020, 2021) showed the good ability of Convolutional Neural Network (CNN) architecture to learn the transfer function between large scale fields and local variables in statistical downscaling applications. Baño-Medina et al. (2021) confirms the suitability of CNN for ESD in an inter-comparison study, while Vandal et al. (2017) demonstrates the good performances of CNN in front of a state-of-the-art bias correction method. The concept of emulator is mentioned in Reichstein et al. (2019) as surrogate models trained to replace a complex and computationally expensive

physical model (entirely or only parts of it). Once trained, this emulator should be able to produce simulations much faster than the original model. In this context, the RCM-emulator proposed here is based on a different fully convolutional neural network architecture known as UNet (Ronneberger et al. 2015). Wang et al. (2021) propose a different emulator following a different strategy than ours as they train it using a low and high-resolution version of the same RCM and another type of neural network (namely CGAN).

This study presents and validates the concept of statistical RCM-emulator. We will focus on emulating the near-surface temperature in a RCM over a specific domain, including high mountains, shore areas, and islands in Western Europe. This domain regroups areas where the RCM presents added value compared to GCM but remains small enough to perform quick sensitivity tests. This paper is organized as follows: Sect. 2 presents the whole framework to define, train and evaluate the emulator, while Sect. 3 shows the emulator results. Finally, Sects. 4 and 5 discuss the results of the emulator and provide conclusions.

2 Methodology

This section provides a complete description of the UNet-based RCM-emulator used in this paper. The notations are summarised in Table 1. The RCM-emulator uses a neural network architecture to learn the relationship between large-scale fields and local-scale variables inside regional climate simulations. RCMs include a downscaling function (F) which transforms large scale information (X,Z) into high resolution surface variables (Y). The statistical RCM-emulator aims to solve a common Machine Learning problem

$$Y = F(X, Z)$$

which is to estimate F by $\hat{F}$ in order to apply it to new GCM simulations. The following paragraphs describe the list of predictors used as inputs and their domain, the predictand (or target) and its domain, the neural network architecture, the framework used to train the emulator and the metrics used to evaluate its performances.

2.1 Models and simulations

This study focuses on the emulation of the daily near-surface temperature from EURO-CORDEX simulations based on the CNRM-ALADIN63 regional climate model (Nabat et al. 2020) driven by the CNRM-CM5 global climate model used in CMIP5 (Voldoire et al. 2013). The latter provides lateral boundary conditions, namely 3D atmospheric forcing at 6-hourly frequency, as well as sea surface temperature, sea ice cover and aerosol optical depth at monthly frequency. The simulations use a Lambert Conformal grid covering the European domain (EUR-11 CORDEX) at the 0.11° (about 12.5 km) scale (Jacob et al. 2014). The historical period runs from 1951 to 2005. The scenarios (2006–2100) are based on two Representative Concentration Pathways from the fifth phase of the Coupled Model Intercomparison Project (CMIP5): RCP4.5 and RCP8.5 (Moss et al. 2010). The monthly aerosol forcing evolves according to the chosen RCP and the driving GCM.

2.2 Predictors

Neural Networks can deal with large datasets at low computational time. During their self optimisation, they are able to select the important variables and regions for the prediction. In this way, a large number of raw predictors can be given to the learning algorithm, with minimum prior selection (which could introduce some bias) or statistical pre-work (which might delete some of the information). Several ESD studies (Lemus-Canovas and Brands 2020; Erlandsen et al. 2020) show that the right combination of predictors depends strongly on the target region and the season. The RCM domains are often composed of very different regions in terms of orography, land types, distance to the sea, etc.

Table 1 Notations

Notation	Description	Dimensions
i, j	Spatial indexes over input grid	$\{i,j\} \in [\![1, I]\!] \times [\![1, J]\!]$
k, l	Spatial indexes over target grid	$\{k,l\} \in [\![1, K]\!] \times [\![1, L]\!]$
t	Temporal index, daily	$t \in \mathbb{N}$
x	2-D variables index	List of 2D variables : $V2D$
z	1-D variables index	List of 1D variables : $V1D$
X	Set of 2D input for the emulator	$X_{t,i,j,x} \in \{\mathbb{N} \times [\![1, I]\!] \times [\![1, J]\!] \times V2D\}$
Z	Set of 1D input for the emulator	$Z_{t,z} \{\mathbb{N} \times V1D\}$
Y	Target: daily near-surface temperature	$Y_{t,k,l} \in \{\mathbb{N} \times [\![1, K]\!] \times [\![1, L]\!]\}$
F	Downscaling function of the RCM	
$\hat{F}$	Emulator: estimation of F	

For these reasons, we decided to give all potentially needed predictors to the emulator and leave the algorithm to determine the right combination to be used to predict each RCM grid point.

The set of predictors (X, Z) used as input in the emulator is composed of 2 dimensional variables X, and 1D predictors Z (Table 2). The set of 2D variables includes atmospheric fields commonly used in ESD (Baño-Medina et al. 2020; Gutiérrez et al. 2019) at different pressure levels. We also added the total aerosol optical depth present in the atmosphere since it constitutes a key regional driver of the regional climate change over Europe (Boé et al. 2020). It leads to 19 2D predictors. These variables are normalised (see Eq. 1) according to their daily spatial mean and standard deviation so that they all have the same importance for the neural network before the training.

The set of 1D variables includes external forcing also given to the RCM: the total concentration of greenhouse gases and the solar and ozone forcings. It also includes a cosinus, sinus vector to encode the information about the day of the year. Given that the 2D variables are normalised at each time step by their spatial mean, they don't carry any temporal information. For this reason, the daily spatial means and standard deviations time series for each 2D variable are included in the 1D input, bringing the size of this vector to 43. In order to always give normalised inputs to the emulator, Z is normalized (see Eq. 2) according to the means and standard deviations over a reference period (1971–2000 here) chosen during the emulator training. The same set of means and standard deviations will be used to normalise any low resolution data to be downscaled by the emulator.

This decomposition of the large scale information consists in giving separately the spatial structure of the atmosphere (X) and the temporal information (Z) to the emulator. Thanks to the neural network architecture described in Sect. 2.4.3, we force the emulator to consider both items of information equally.

Table 2 List of predictors

| Field	Altitude levels	Variables notation	Units	Temporal aggregation	Dimension
2D variables					
Geopotential	850, 700, 500 hPa	zg500 zg700 zg850	m	Daily mean	$[i, j]$
Specific humidity	850, 700, 500 hPa	hus500 hus700 hus850		Daily mean	$[i, j]$
Temperature	850, 700, 500 hPa	ta500 ta700 ta850	K	Daily mean	$[i, j]$
Eastward wind	850, 700, 500 hPa + Surface	ua500 ua700 ua850 uas	m/s	Daily mean	$[i, j]$
Northward wind	850, 700, 500 hPa + Surface	va500 va700 va850 vas	m/s	Daily mean	$[i, j]$
Sea level pressure	Surface	psl	Pa	Daily mean	$[i, j]$
Total aerosol optical depth forcing		TAOD		Monthly mean	$[i, j]$
1-D variables					
Daily spatial means of 2D variables		$\bar{x}\ with\ x \in V2D$		Daily	$[\#V2D]$
Daily spatial standard deviation of 2D variables		$\dot{x}\ with\ x \in V2D$		Daily	$[\#V2D]$
Total anthropogenic greenhouses gas forcings		ant_ghg		Yearly	$[1]$
Solar and ozone forcings		sol, oz		Yearly	$[2]$
Seasonal indicators $Cos(\frac{2\pi t}{365})$; $Sin(\frac{2\pi t}{365})$		cos,sin		Daily	$[2]$
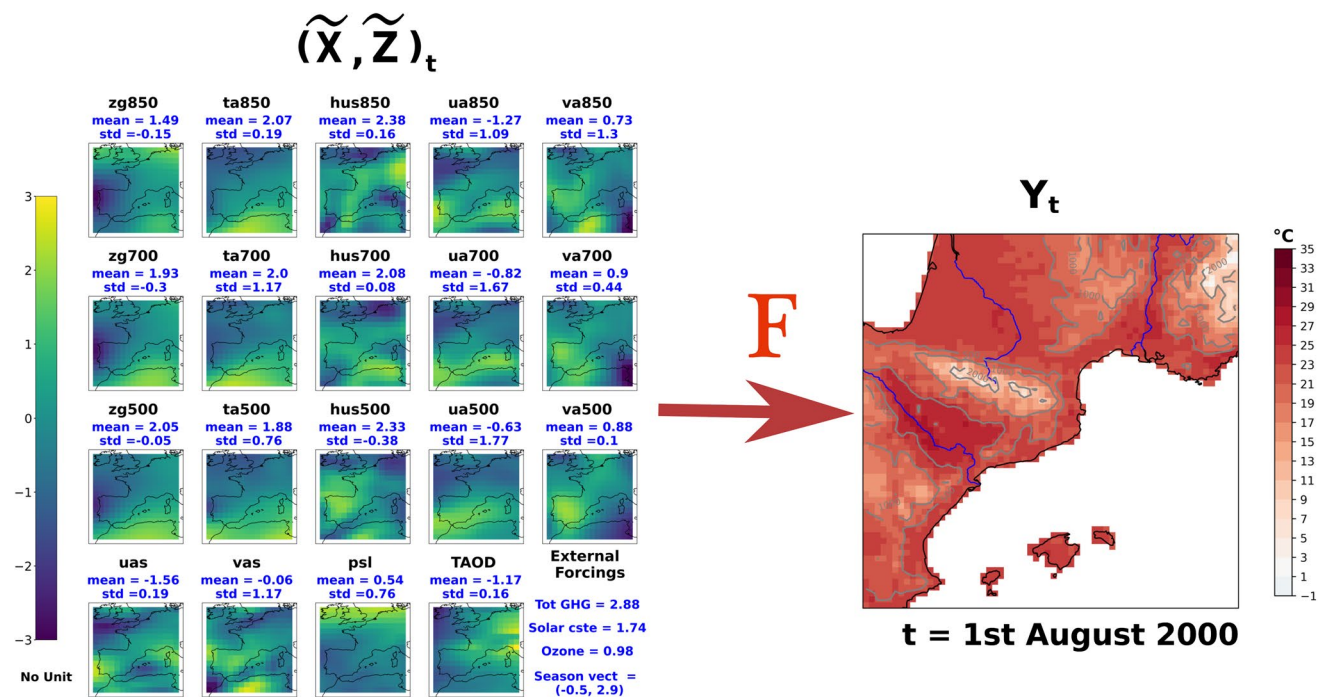
Fig. 1 Illustration of an observation for a randomly-chosen day. Left: each map represents a 2D input variables (X), on the input domain, and the blue numbers correspond to the 1D variables (Z). Right: an example of Y, the near surface temperature on the target domain

$$\widetilde{X}_{t,i,j,x} = \frac{X_{t,i,j,x} - \overline{X}_{t,x}}{\overset{\bullet}{X}_{t,x}}$$

$$\text{with } \overline{X}_{t,x} = \frac{\sum_{i=1}^{I}\sum_{j=1}^{J} X_{t,i,j,x}}{I \times J} \qquad (1)$$

$$\text{and } \overset{\bullet}{X}_{t,x} = \sqrt{\frac{\sum_{i=1}^{I}\sum_{j=1}^{J}(X_{t,i,j,x} - \overline{X}_{t,x})^2}{I \times J}}$$

$$\widetilde{Z}_{t,z} = \frac{Z_{t,z} - \overline{Z}_{ref,z}}{\overset{\bullet}{Z}_{ref,z}}$$

$$\text{with } \overline{Z}_{ref,z} = \frac{\sum_{t \in R} Z_{t,z}}{nb \ days \ in \ R} \qquad (2)$$

$$\text{and } \overset{\bullet}{Z}_{ref,z} = \sqrt{\frac{\sum_{t \in R}(Z_{t,z} - \overline{Z}_{ref,z})^2}{nb \ days \ in \ R}}$$

where R is the reference period.

The aim of the emulator is to downscale GCM simulations. Klaver et al. (2020) shows that the effective resolution of climate models is often larger (about 3 times) than its nominal resolution. For instance, CNRM-CM5 is more reliable at a coarser resolution, probably about 450–600 km, than at its own horizontal resolution (≈ 150 km). For this reason, the set of 2D predictors are smoothed with a 3×3 moving average filter. The grid of the GCM is conserved, with each

point containing a smoother information than the raw model outputs.

For this study the input domain is defined around the target domain (described in Sect. 2.3). It is a 16×16 ($J = I = 16$ in Table 1) CNRM-CM5 grid box visible on Fig. 1. Each observation given to the emulator (see Fig. 1 for an illustration) is a day t and it is composed of a 3D array ($X_{i,j,x}$), where the two first dimensions are the spatial coordinates and the third dimension lists the different variables chosen as predictors, and a 1D array (Z_z) regrouping all the 1 dimensional inputs.

2.3 Predictands

In this study, to assess the ability of the RCM-emulator to reproduce the RCM downscaling function, we focused on the emulation of the daily near surface temperature over a small but complex domain. The target domain for this study is a box of 64×64 RCM grid points at 12 km resolution (about 600,000 km^2) centred over the south of France (Fig. 1). It gathers different areas of interest for the regional climate models. It includes three mountain massif (Pyrenees, Massif Central and French Alps) which are almost invisible at the GCM scale (specially the Pyrenees). The domain also includes coastlines on the Mediterranean side and on the Atlantic side. Thanks to a better representation of the coastline at the RCM resolution, it takes better into account the sea effect on the shore climate. It was also important for us to add small islands on our evaluation

domain, such as the Baleares (Spain), since they are invisible on the GCM grid and the RCM brings important information. Finally, three major rivers (plotted in blue in Fig. 1) are on the domain with interesting impacts on climate (commented in Sect. 3): the Ebro in Spain, the Garonne in southwest of France and the Rhone on the east of the domain. This domain should therefore illustrate the added value brought by a RCM at local scale and be a good test-bed on the feasibility of emulating high resolution models.

2.4 Deep learning with UNet

2.4.1 Neural network model as a black-box regression model

The problem of statistical downscaling and of emulation of daily near-surface variables may be seen as a statistical regression problem where we need to build the best relationship between the output response Y and the input variables (X, Z). When looking at the L^2 loss between the prediction $\hat{Y}$ and the true Y, the optimal link (denoted by F below) is theoretically known as the conditional expectation:

$$F(X, Z) = E[Y \,|(X, Z)]. \tag{3}$$

Unfortunately, since we will only have access to a limited amount of observations collected over a finite number of days, we shall work with a training set formed by the collected observations $((X_t, Z_t), Y_t)_{1 \le t \le T}$ and try to build from the data an empirical estimation $\widehat{F}$ of the unknown F.

For this purpose, we consider a family of relationship between (X, Z) and Y generated by a parametric deep neural network, whose architecture and main parameters are described later on. We use the symbol θ to refer to the values that describe the mechanism of one deep neural network, and Θ as the set of all possible values of θ. Hence, the family of possible relationships described by a collection of neural networks correspond to a large set $(F_\theta)_{\theta \in \Theta}$. Deep learning then corresponds to the minimization of a L^2 data-fidelity term associated to the collected observations:

$$\hat{F} = \arg \min_{F_\theta, \theta \in \Theta} \sum_{t=1}^{T} \|Y_t - F_\theta(X_t, Z_t)\|^2. \tag{4}$$

2.4.2 Training a deep neural network

To train our emulator between low resolution fields and one high resolution target variable, we used a neural network architecture called UNet whose architecture is described below. As usual in neural networks, the neurons of UNet are organised in layers. Given a set $E_{\mathbf{n}}$ of input variables denoted by $(x_i)_{i \in E_{\mathbf{n}}}$ of an individual neuron $\mathbf{n}$, the output of $\mathbf{n}$

corresponds to a non-linear transformation ϕ (called activation function) of a weighted sum of its inputs:

$$\mathbf{n}\big((x_i)_{i \in E_{\mathbf{n}}}\big) = \phi\big(\langle (w_i)_{i \in E_{\mathbf{n}}}, (x_i)_{i \in E_{\mathbf{n}}} \rangle\big) = \phi\left(\sum_{i \in E_{\mathbf{n}}} w_i x_i\right). \tag{5}$$

The connection between the different layers and their neurons then depends on the architecture of the network. In a fully connected network (multilayer perceptron) all the neurons of a hidden layer are connected to all the neurons of the previous layer. The deepness of a network then depends on the number of layers.

As indicated in the previous paragraph, the machine learning procedure corresponds to the choice of a particular set of weights over each neuron to optimise a data fidelity term. Given a training set, a deep learning algorithm then solves a difficult multimodal minimisation problem as the one stated in (4) with the help of gradient descent strategies with stochastic algorithms. The weights associated with each neuron and each connection are then re-evaluated according to the evolution of the loss function, following the back-propagation algorithm (Rumelhart and Hinton 1986). This operation is repeated over all the examples until the error is stabilized. Once the neural network is trained, it may be used for prediction, i.e., to infer the value Y from new inputs (X, Z).

We emphasize that the bigger the dimension of the inputs and outputs, the larger the number of the parameters to be estimated and so the bigger the training set must be. Therefore, the quality of the training set is crucial: missing or wrong values will generate some additional fluctuation and errors in the training process. Moreover, we also need to cover a sufficiently large variety of scenarios in the input variables to ensure that our training set covers a wide range of possible inputs. For all these reasons, climate simulation datasets are ideal to train deep learning networks.

2.4.3 UNet architecture

The emulator proposed in this study relies on a specific architecture called UNet introduced by Ronneberger et al. (2015). It is a fully convolutional architecture, i.e. all layers are convolutional layers known for their strength to deal efficiently with image data. UNet is known for its good ability to identify different objects and areas in an image. It involves gathering pixels that correspond to the same object. This key feature is naturally interesting for meteorological maps. The emulator needs to identify the different meteorological structures present in the low resolution predictors for a given day in order to predict the corresponding high resolution near-surface temperature. Moreover, the good performances of UNet have been
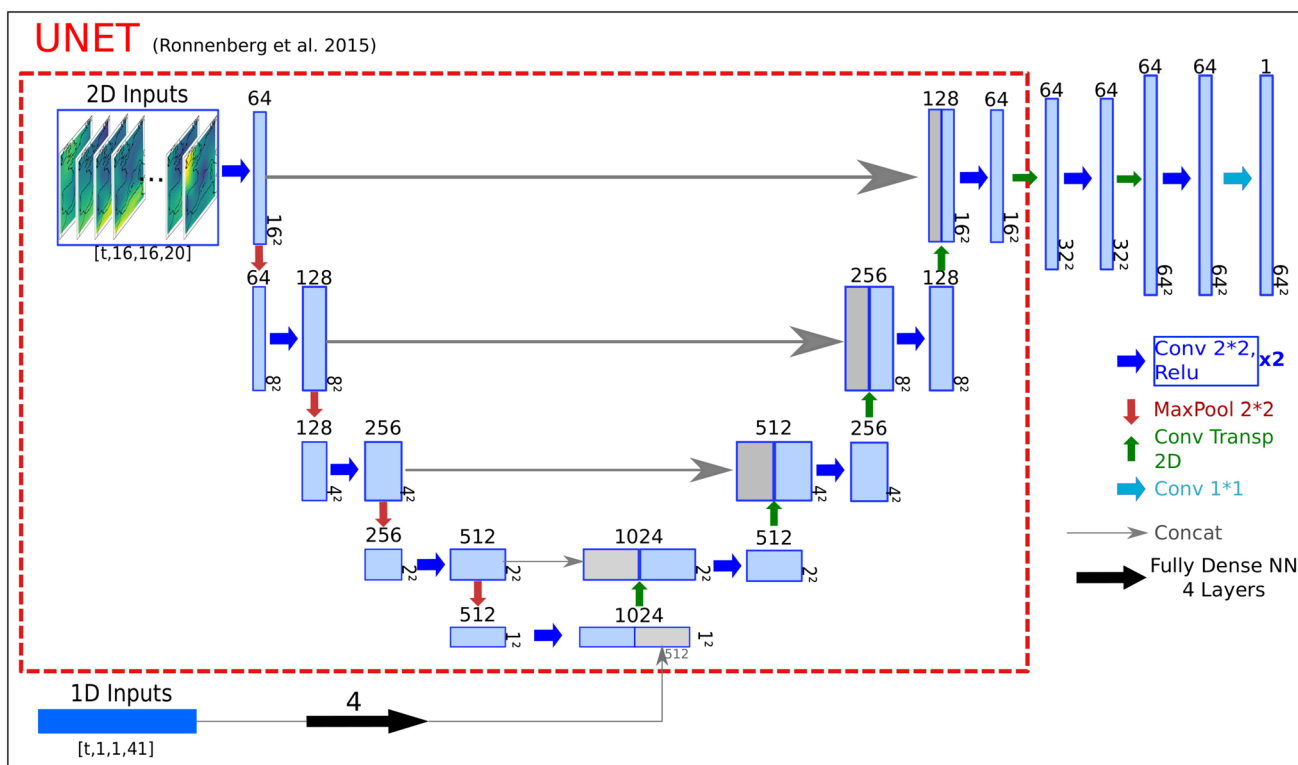
Fig. 2 Scheme of the neural network architecture used for UNet emulator. The part of the network in the red frame corresponds to the original UNet defined in Ronneberger et al. (2015)

demonstrated in problems such as pixel-wise regression (Yao et al. 2018) or super-resolution (Hu et al. 2019), which are comparable to ours.

The original UNet architecture is illustrated in the red frame in Fig. 2 and more precisely described in Appendice. It is composed of an encoding part on the left side, which reduces the spatial dimension of the input image while the pixel dimension gets deeper. On the right side, the expansive path decodes the encoded information to reconstruct the target image. Each encoding step is recalled in the decoding part, allowing the network to find the best way to the target image through different levels.

The architecture we propose is adapted from the original UNet to suit our problem. Firstly, the input object is a description of the climate conditions leading to the target image. It is then a 3D object while the classical UNet takes a single image as input. This aspect differs from the natural use of UNet but does not modify its construction. Secondly, we added a second source of input, the set of 1D predictors (see Table 2) which is given at the bottom of the UNet (see Fig. 2). This 1D object is encoded by a dense network and then concatenated with the encoded 2D inputs. The "U" shape of the UNet architecture allowed us to give this information such that the network treats equally the two sources of input. Finally, we extended the expansive part of the UNet

to reach the higher resolution of the output as illustrated in Fig. 2.

We chose the mean square error (MSE) as the loss function to train the network, as we have a regression problem. Moreover, it is well adapted for variables following Gaussian-like distribution, such as temperature. The neural network was built and trained using the Keras Tensorflow environment in Python (https://keras.io). The network trained for this study has about 25 million parameters to fit.

2.5 Training of the emulator: perfect model framework

As any statistical downscaling and any machine learning method, the emulator needs to be calibrated on a training set. It consists in showing the emulator many examples of predictors and their corresponding target such that the parameters of the network can be fitted as mentioned in Sect. 2.4.2. The emulator is trained in a perfect model framework, with both predictors and predictands coming from the same RCM simulation. The intuitive path to train the emulator is to use GCM outputs as predictors and its driven RCM outputs as predictands, but there are many reasons for our choice. First of all, it guarantees a perfect temporal correlation between large scale predictors and a local scale predictand. Indeed,

Fig. 3 shows that GCM and RCM large scales are not always well correlated, with an average correlation of 0.9 and 10% of the days with a coefficient of correlation lower than 0.75. These mismatches are quite well known and often due to internal variability as explained by Sanchez-Gomez et al. (2009); Sanchez-Gomez and Somot (2018). Moreover, there are more consistent biases (discussed in Sect. 4.1) between GCM and RCM large scales. It is of primary importance that the inputs and outputs used to calibrate the model are perfectly consistent, otherwise, the emulator will try to learn a non-existing or non-exact relationship. In this context, the perfect model framework allows us to focus on the downscaling function of the RCM, specifically. This approach is similar to the "super-resolution downscaling" mentioned by Wang et al. (2021) and deployed by Vandal et al. (2017) using observational data in a empirical statistical downscaling framework.

The training protocol is summarised in Fig. 4. In a first step, the RCM simulation outputs are upscaled to the GCM resolution (about 150 km) thanks to a conservative interpolation. This first step transforms the RCM outputs into GCM-like outputs. This upscaled RCM is called UPRCM in the rest of this paper. In the second step, these UPRCM outputs are smoothed by a 3×3 moving average filter to respect the protocol described in Sect. 2.2. This smoothing

also targets to delete any local scale information which might persist through the upscaling step (as discussed in Sect. 4.3).

The near-surface temperature on the target domain is extracted from the same RCM simulation. Following this procedure, the emulator is trained using the ALADIN63 simulation forced by CNRM-CM5, covering the 1950–2100 period with the RCP8.5 scenario from 2006. As shown in Fig. 4, we will use a different simulation driven by a different RCP scenario for the evaluation. We chose the two most extreme simulations (historical and RCP8.5) for the training in order to most effectively cover the range of possible climate states since the emulator does not target to extrapolate any information. Future studies could explore the best combination of simulations to calibrate the emulator.

2.6 Evaluation metrics

The emulated temperature series ($\hat{Y}$) will be compared to the original RCM series (Y) (mentioned as "RCM truth" in the rest of this paper) through statistical scores described below. Each of these metrics will be computed in each point over the complete series:
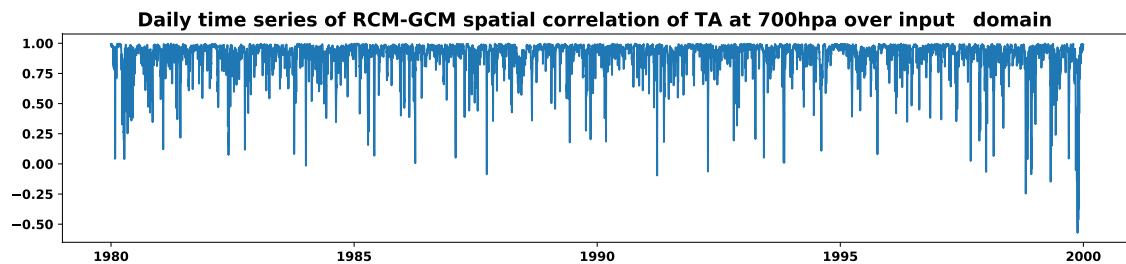

Fig. 3 Time series of spatial correlation of the atmospheric temperature at 700 hpa between ALADIN63 and its driving GCM, CNRM-CM5, over the input domain
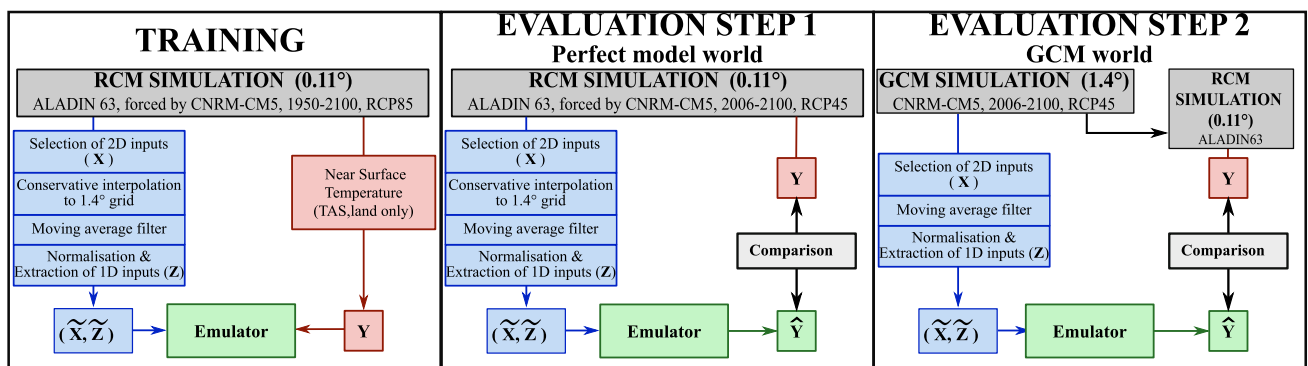

Fig. 4 Scheme of the protocols for the training (left) and the two steps of evaluations (center and right)

– **RMSE**: The root mean squared error measures the standard deviation of the prediction error (in °C):

$$RMSE(Y, \widehat{Y}) = \sqrt{\frac{1}{T} \sum_t (Y_t - \widehat{Y}_t)^2} \qquad (6)$$

– **Temporal anomalies correlation**: This is the Pearson correlation coefficient after removing the seasonal cycle:

$$ACC(Y, \widehat{Y}) = \rho(Y_a, \widehat{Y}_a), \qquad (7)$$

with ρ the Pearson correlation coefficient and Y_a and $\widehat{Y}_a$ are the anomaly series after removing a seasonal cycle computed on the whole series.

– **Ratio of variance**: It indicates the performance of the emulator in reproducing the local daily variability. We provide this score as a percentage.

$$RoV(Y, \widehat{Y}) = \frac{Var(\widehat{Y})}{Var(Y)} * 100 \qquad (8)$$

– **Wasserstein distance**: It measures the distance between two probability density functions (P, Q). It relies on the optimal transport theory (Villani 2009) and measures the minimum required "energy" to transform P into Q. The energy here corresponds to the amount of distribution weight that is needed to be moved multiplied by the distance it has to be moved. In this study we use the 1-d Wasserstein distance, and its formulation between two samples becomes a rather simple function of ordered statistics:

$$W_1(f(Y), f(\widehat{Y})) = \sum_{i=1}^{T} |Y_{(i)} - \widehat{Y_{(i)}}|, \qquad (9)$$

with $f(\bullet)$ the probability density function associated with the sample $\bullet$.

– **Climatology**: We compare the climatology maps over present (2006–2025) and future (2081–2100, not shown in Sect. 3) climate. The RCM truth and emulator maps are shown with their spatial correlation and RMSE. The error (emulator minus RCM) map is also computed.

– **Number of days over** 30°C: Same as climatology for the maps showing the number of days over 30 °C.

– **99th Percentile**: Same as climatology for the maps showing the 99th percentile of the daily distribution.

– **Climate Change**: Climate change maps for the climatology, the number of days over 30 °C and the 99th percentile (delta between future (2080–2100) and present (2006–2025) period).

These metrics are at the grid point scale and are presented as maps. However, to summarise these maps with few numbers we can compute their means and their super-quantile of order 0.05 (SQ05) and 0.95 (SQ95). The super-quantile α is defined as the mean of all the values larger (resp. smaller) than the quantile of order α, when α is larger (resp. smaller) than 0.5. These values are shown in the Figures of Sect. 3 and Tables SM.T1 and SM.T2 in supplementary material.

2.7 Benchmark

For this study, we propose as benchmark the near surface temperature from the input simulation (before the moving average smoothing), interpolated on the target grid by bilinear interpolation. As this study is the first to propose such an emulator, there is no already established benchmark. The one proposed here is a naive high-resolution prediction given available predictors (low-resolution), it is used as a basic reference and not a potential competitor for the emulator. It allows the reader to locate the emulator somewhere in between the simplest possible downscaling (simple interpolation of the original low resolution simulation) and the most complex one (RCM simulation). All the metrics introduced in Sect. 2.6 will be applied to our benchmark.

3 Results

This section presents the emulator performances in terms of its computational costs (in Sect. 3.1) and its ability to reproduce the near surface temperature time series at high resolution. As illustrated in Fig. 4, we will evaluate the emulator in two steps, (1) in the perfect model world (Sect. 3.2) and (2) when the emulator inputs come from a GCM simulation (Sect. 3.3). The RCM simulation used to evaluate the model (also called target simulation) is the ALADIN63, RCP45, 2006–2100 forced by the GCM simulation CNRM-CM5, RCP45, 2006–2100. Note that the emulator never saw the target simulation during the training phase. This evaluation exercise illustrates a potential use of the emulator: downscaling a scenario that the RCM has not previously downscaled.

3.1 Computational efficiency

The emulator is trained on a GPU (Nvidia, GeForce GTX 1080 Ti). About 60 epochs are necessary to train the network, and each epoch takes about 130 s with a batch size of 100 observations. The training of the emulator takes about 2 h. Once the emulator is trained, the downscaling of a new low resolution simulation is almost instantaneous (less than a minute). It is a significant gain in time compared to RCM, even if these time lengths do not include the preparation of the inputs, which depends mainly on the size of the series to downscale and on the data accessibility. It would, when including the input preparation, take only a few hours on a simple CPU or GPU to produce a simulation with the trained

emulator, while it takes several weeks to perform a RCM simulation on a super-computer.

3.2 Evaluation step 1: perfect model world

In a first step, the emulator is evaluated in the perfect model world, meaning that the inputs come from the UPRCM simulation. This first evaluation step is necessary to control the performances of the emulator in similar conditions as during its training. Moreover, the perfect model framework guarantees perfect temporal matches between the large scale low resolution fields (the inputs) and the local scale high resolution temperature (the target). The emulator should then be able to reproduce perfectly the temperature series that is simulated by the RCM. This first evaluation of the emulator is divided in two parts. In the Sect. 3.2.1 we analyse the ability of the emulator to reproduce the RCM simulation. The Sect. 3.2.2 compares the specific emulator proposed in this study with two others emulators relying on standard empirical statistical downscaling methods.

3.2.1 Ability to reproduce the RCM simulation

The emulator aims to learn and reproduce the downscaling function included in the RCM, i.e., to transform the low resolution daily information about the state of the atmosphere into high-resolution daily surface temperature. We dedicate this first evaluation section to comparing the prediction of the emulator with the RCM truth in perfect model. The benchmark for this first evaluation is the UPRCM near surface temperature re-interpolated on the RCM grid. It is referred as "I-UPRCM". We are aware that the "I-UPRCM" field is a very simple benchmark and can not compete with the emulator in terms of realism but it allows to simply measure the action of the emulator regarding the input simulation.

Figure 5a illustrates the production of the emulator for a random day regarding the target and the benchmark. The RCM truth map presents a refined and complex spatial structure largely missing in the UPRCM map. Moreover, it is evident on the I-UPRCM map that the simple bilinear interpolation does not recreate these high resolution spatial patterns. The emulator shows for this given day an excellent ability to reproduce the spatial structure of the RCM truth. It has very accurate spatial correlation and RMSE and estimates the right temperature range. On Fig. 5b, we show the daily time series for four specific points shown on the RCM truth map (Marseille, Toulouse, a high Pyrenees grid point and a point in Majorca) during a random year. The RCM transforms the large scale temperature (visible on I-UPRCM) differently over the four points. In the Pyrenees, the RCM shifts the series and seems to increase the variance. In Marseille, it appears to produce a warmer summer

without strongly impacting winter characteristics, and it seems to be the exact opposite in Majorca. On the contrary, in Toulouse, I-UPRCM and RCM are close. For each of these 4 cases, the emulator reproduces the sophistication of the RCM series almost perfectly.

Figure 5a gives the impression that the emulator has a good ability to reproduce the complex spatial structure brought by the RCM, and we can generalise this result with the other Figures. First of all, the performance scores (Fig. 6b) of the emulator confirm that this good representation of the temperature's spatial structure is robust over the whole series. The spatial homogeneity of the score maps tends to show that the emulator does not have particular difficulties over complex areas. The spatial correlation (equal to 1) and the very low spatial RMSE (0.07 °C) of the climatology maps in Fig. 7 in the present climate support this result. In particular, it is worth noting on the climatology maps that the altitude dependency is well reproduced as well as the warmer patterns in the Ebro and Rhone valley or along the coastlines. Moreover, the comparison with the interpolated UPRCM shows the added-value of the emulator and in particular its ability to reinvent the fine-scale spatial pattern of the RCM truth from the large-scale field. Indeed, the score maps of the I-UPRCM (Fig. 6c) or the present climatology error map (Fig. 7a) shows strong spatial structures, highlighting the regions where the RCM brings added value and that the emulator reproduces successfully.

The RCM resolution also allows to have a better representation of the daily variability at the local scales over critical regions. The difference in variance between the I-UPRCM and the RCM is visible on Fig. 6c. The I-UPRCM underestimates the variability and is poorly correlated over the higher reliefs, the coastlines and the river valley. In contrast, the emulator reproduces more than 90% of the RCM variance over the whole domain. The RMSE and temporal correlation maps of the emulator confirm the impression given by Fig. 5b that it sticks almost perfectly to the RCM truth series. Moreover, the RCM daily variability is strongly dependent on the region. Indeed, the RCM transforms the "I-UPRCM" pdfs in different ways across the domain (visible on Fig. 6a, c). Figure 6a, b show that the emulator succeeds particularly well in filling these gaps.

The emulator's good representation of the daily variability and temporal correlation involves a good representation of the extreme values. The probability density functions of the four specific points on Fig. 6a show that the entire pdfs are fully recreated, including the tails. The Wasserstein distance map extends this result to the whole target domain. The two extreme scores computed for the present climate on Fig. 7b, c confirm these results. The 99th percentile emulated map is almost identical to the target one verified by the difference map, with a maximum difference of less than 1 °C for values over 35 °C. The spatial pattern
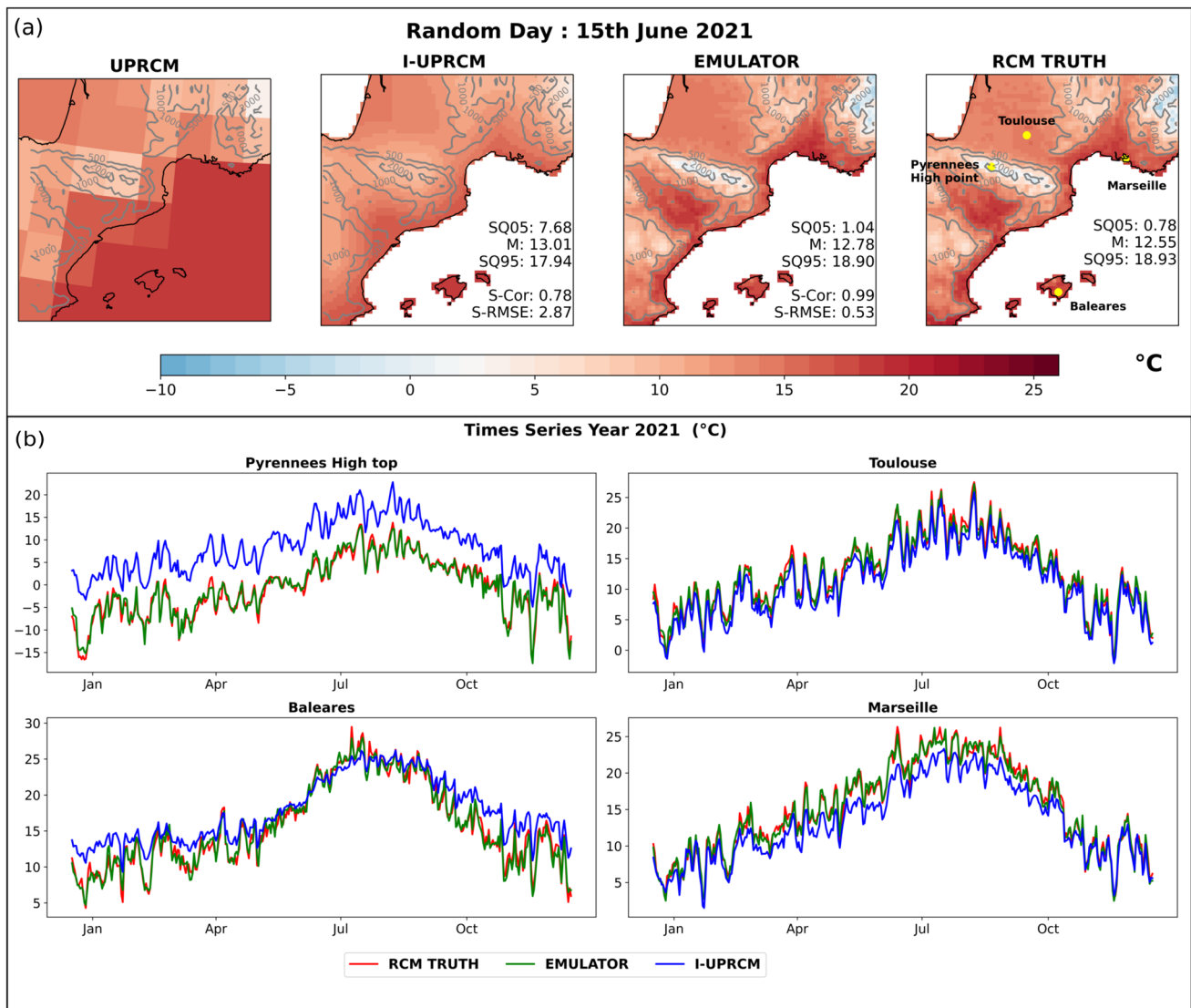
Fig. 5 Randomly chosen illustration of the production of the emulator (in evaluation step 1) with inputs coming from UPRCM: **a** temperature (°C) at a random day over the target domain for the raw UPRCM, the interpolated UPRCM, the emulator and the RCM truth and **b** random year time series (°C) for 4 particular grid points

of the 99th percentile map is here again correctly captured by the emulator, particularly along the Garonne river that concentrates high extremes. The number of days over 30 °C is a relatively more complicated score to reproduce since it involves an arbitrary threshold. The emulator keeps performing well with a high spatial correlation between the emulator and the RCM truth. However, it appears that the emulator misses some extreme days, involving a lack in the intensity of some extremes metrics.

Finally, the high-resolution RCM produces relevant small-scale structures in the climate change maps. In particular, RCMs simulate an elevation-dependent warming (see the Pyrenées and Alps areas Kotlarski et al. 2015), a weaker warming near the coasts (see the Spanish or Atlantic

coast) and a specific signal over the islands as shown in the second lines of Fig. 7a, c. It can be asked if the emulator can reproduce these local specificities for the climate change signal. The emulator is able to capture this spatial structure of the warming but with a slight lack of intensity which is general over the whole domain. The reproduction of the climate change in the extremes suffers the same underestimation of the warming but also offers the same good ability to reproduce the spatial structure, with high spatial correlation.

This first evaluation step shows that if the emulator is still perfectible, in particular when looking at extremes or climate change intensity, it is able to almost perfectly reproduce the spatial structure and daily variability of the near surface temperature in the perfect model world.
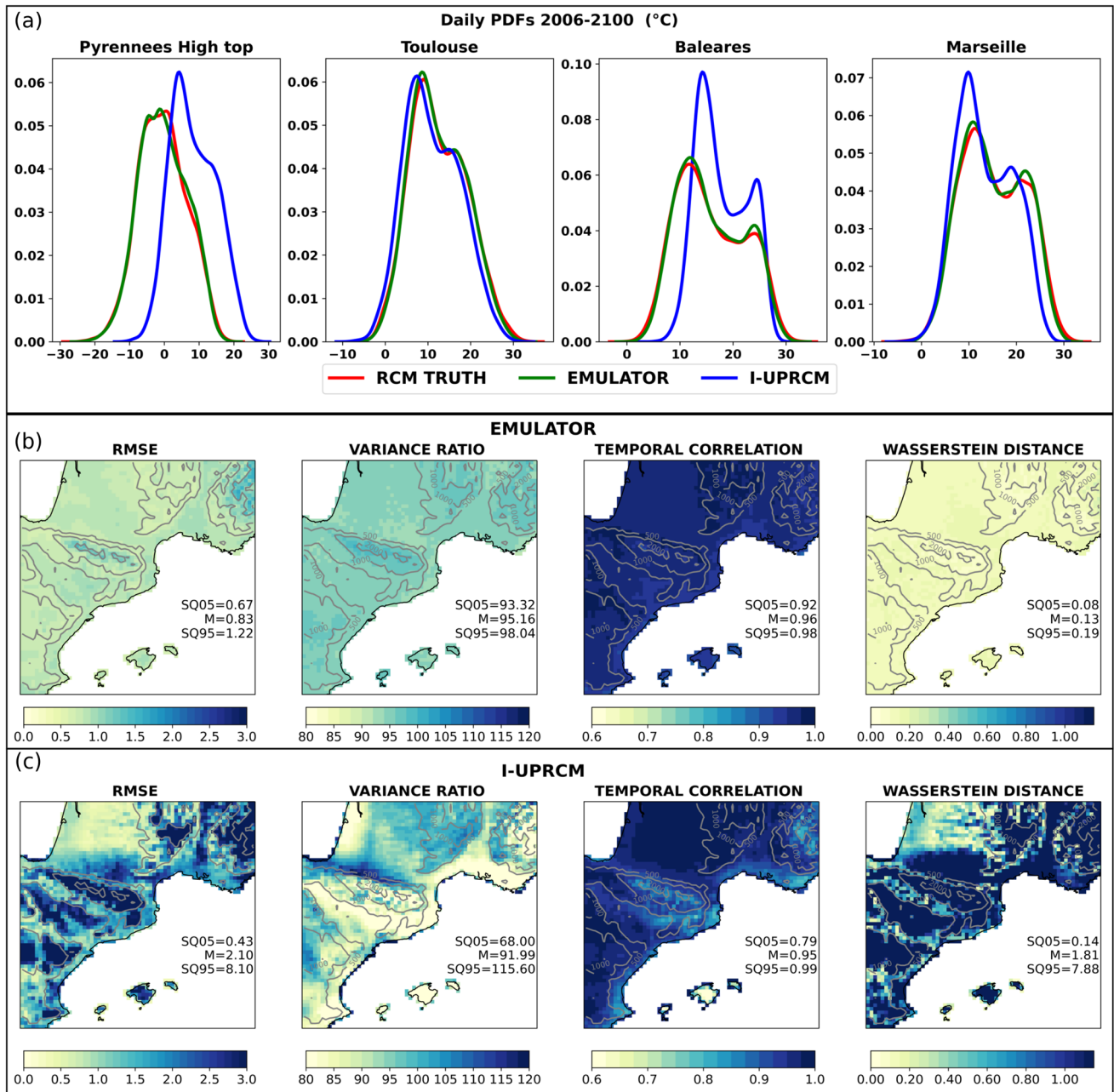
Fig. 6 a Daily probability density functions from the RCM truth, the emulator (in evaluation step 1) and the I-UPRCM at 4 particular grid points over the whole simulation period. **b** (Resp. **c**) Maps of performance scores of the emulator (resp. of the I-UPRCM) with respect to the RCM truth computed over the whole simulation period. For each map, the values of the spatial mean and super-quantiles (SQ05 and SQ95) are added

3.2.2 Comparison to simpler emulators

This section validates the UNet-based emulator by comparing its performances to two other emulators based on standard empirical statistical downscaling methods. The perfect model framework allows comparing different emulators since the RCM truth is the ideal reference when downscaling the UPRCM simulation.

The two comparison emulators proposed here rely on very classical ESD methods. The first one is CDFt (Michelangeli et al. 2009; Vrac et al. 2012), which belongs to the Model Output Statistic family of methods. It can be considered as an extended "quantile-quantile" method. It transforms the low resolution temperature cumulative distribution function into the high resolution one. The second method is a simple Multiple Linear Regression (MLR Huth 2002; Huth

et al. 2015), which belongs to the Perfect Prognosis family of method. The inputs used for the MLR are the same as for the UNet emulator, but only the closest low resolution point is used for the 2D variables. The methods and the way we used them as RCM-emulator are more precisely described in Appendice. Both methods are evaluated in the VALUE project (Gutiérrez et al. 2019), showing reasonable results. They are trained in the same conditions as the UNet emulator, following the perfect model framework described in Sect. 2.5.

The results are illustrated on Fig. 8 and a more complete evaluation can be found in the Appendice. As for the neural network emulator, both CDFt and MLR emulators present a good ability to reproduce the small scale information brought by the RCM. They present a good variance ratio and temporal correlation leading to a good RMSE. The climate change maps (Fig. 8c) show that both methods are also able to reproduce the RCM climate change maps, even if the MLR method seems to have more difficulties and particularly over the relief. These results are expected as these methods showed good performances in statistical downscaling and both methods could constitute reasonable emulators. However, the neural network emulator performs better in most metrics. Indeed, the UNet Emulator presents a lower RMSE showing a better accuracy to fit the original RCM series. It also better captures the spatial structure of the RCM truth, as shown by the spatial correlation scores on the climatology, and extremes maps (Figs. 7, 8 and Appendice).

According to this comparison, we conclude that the three approaches may constitute good RCM-emulators but the one based on the UNet approach is far better, especially for the correlation with the RCM truth. As the UNet-approach is not significantly more complex to apply and not costly to train, we decide to select it as our main emulator for the following of the article, that is to say to illustrate the way RCM-emulators can be applied in practical to downscale GCM simulations. Furthermore, the UNet based emulator, as well as the two simpler emulators, are possible emulator among many other options. We invite the statistical downscaling community to propose other emulators, relying on different statistical methods, trained in different ways, to find the best tool to emulate RCMs.

3.3 Evaluation step 2: GCM world

In this second evaluation step, we directly downscale a GCM simulation. The benchmark for this evaluation is the near surface temperature from the GCM, interpolated on the target grid. It will be referred to as I-GCM. The emulated series and the benchmark are compared to the RCM simulation driven by the same GCM simulation.

Figure 9a illustrates the production of the emulator regarding the benchmark and RCM truth for the same day as Fig. 5a. First of all, as for the I-UPRCM, the I-GCM map does not show any of the complex RCM spatial structures. The I-GCM is less correlated with the RCM and warmer than the I-UPRCM. In contrast, the emulator reproduces the complex spatial structure of the RCM very well with a spatial correlation of 0.98 but appears to have a warm bias with respect to the RCM truth. The four time series are consistent with the previous section, with fundamental differences between I-GCM and RCM, which the emulator captured very well. However, the correlation between the emulator and the RCM is not as good as in the perfect model framework.

Figure 9b is a very good illustration of the RCM-GCM large scale de-correlation issue presented in Sect. 2.4.2. Indeed the less good correlation of the emulator with the RCM is probably due to mismatches between GCM and RCM large scales. For instance, in the beginning of November, on the time series shown on Fig. 9b, the RCM seems to simulate a cold extreme on the whole domain, which appears neither in the interpolated GCM nor in the emulator. The same kind of phenomenon occurs regularly along the series and is confirmed by lower temporal correlations between the RCM truth and the I-GCM (Fig. 10c) than with the I-UPRCM (Fig. 6c). According to this, the emulated series can not present a good temporal correlation with the RCM truth since it is a daily downscaling of the GCM large scale. Keeping in mind these inconsistencies, it is still possible to analyse the performances of the emulator if we leave aside these scores which are influenced by the poor temporal correlation (RMSE, ACC).

As in the first step of the evaluation (Sect. 3.2), the spatial structure of the RCM truth is well reproduced by the emulator. The present climatology map (Fig. 11a) has a perfect spatial correlation with the RCM. The added value from the emulator is clear if compared to the interpolated GCM. The spatial temperature gradients simulated by the GCM seem to be mainly driven by the distance to the sea. On the other hand, the emulator manages to recreate the complex structures created by the RCM, related to relief and coastlines. The emulator capacity to reproduce the RCM spatial structure seems as good as in Sect. 3.2. The scores in Fig. 10b present a good spatial homogeneity, exactly like in the previous Sect. 6b.

The error map in Fig. 11a shows that the emulator is warmer in the present climate than the RCM truth ($+0.96\,°$C). This bias presents a North-South gradient with greater differences over the North of the target domain, which is consistent with the Wasserstein distance map on Fig. 10b. The Wasserstein metric shows that the density probability functions from the emulated series are further away from the RCM truth with GCM-inputs than in UPRCM mode. The similarities between the Wasserstein scores and the present climatology difference map indicate that the emulator shifts the mean.
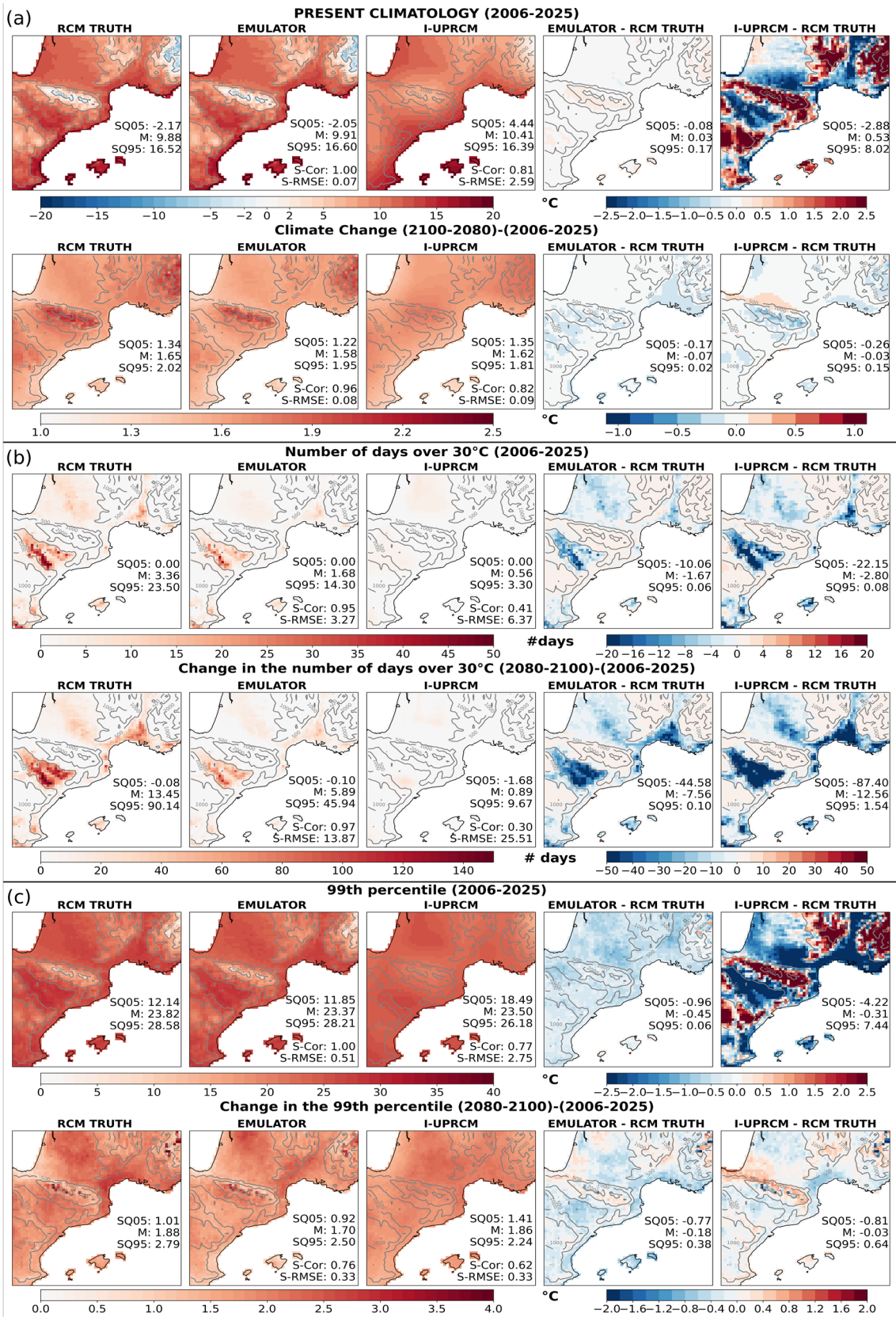
◀**Fig. 7 a** Maps of long-term mean climatologies, **b** number of days over 30 °C and **c** 99th percentile of daily near-surface temperature for a present-climate period (2006–2025) and for the climate change signal (2080–2100 minus 2006–2025), for the RCM truth, the emulator (in evaluation step 1) and the I-UPRCM. On each line, the two last maps show the error map of the emulator and the I-UPRCM. For each map, the spatial mean and super-quantiles (SQ05 and SQ95) are added, as well as the spatial correlation and spatial RMSE for the emulator and I-UPRCM maps

The daily variability is well reproduced by the emulator. As mentioned before, the weaker RMSE (Fig. 10b) is mainly due to the lower correlation between GCM and RCM. But the ratio of variance demonstrates that the emulator manages to reproduce the daily variability over the whole domain. The RCM brings a complex structure of this variability (higher variability in the mountains than in plains, for example), and the emulator, as in the first evaluation step, recreates this fine scale. Moreover, the daily pdfs of the emulator (Fig. 10a) are very consistent with the RCM ones, and the same range of values is covered for each of the four particular points.

This good representation of daily variability tends to suggest that the emulator can reproduce the local extremes. Fig. 11b, c confirm these results, with a very high spatial correlation between the emulator and the RCM truth maps in present climate. The warmer extremes along the three river valleys are present in both RCM and emulator maps, while they are absent from the I-GCM maps. The warm bias observed in the present climatology map also impacts these scores. The emulator map of the number of days over 30 °C in the present climate shows more hot days than the RCM but the same spatial structure. The map of the 99th percentile over the 2006–2025 period shows the same observation, with a warm bias (+ 0.82) slightly lower than the climatology bias.

Finally, the climate change signal is also well captured by the emulator. The different spatial patterns that bring the high resolution of the RCM in the Fig. 11a–c are also visible in the emulator climate change Figures. The emulator represents a weaker warming than the RCM, observable in average warming but also on the map of extremes. This underestimated warming is mainly due to the warm bias between GCM and RCM, which is less intense in the future. For instance, the warming from the emulator is 0.27 °C weaker on average over the domain (with almost no spatial variation) than in the RCM. This number corresponds approximately to the cold bias from the I-GCM (0.19 °C) plus the missed warming by the emulator in the perfect model framework (0.07). This tends to show that the emulator performs well in the GCM world but reproduces the GCM-RCM biases.

This section shows that the emulator remains robust when applied to GCM inputs since it provides a realistic high-resolution simulation. As in the first step, the emulator exhibits several desirable features with an outstanding ability to reproduce the complex spatial structure of the daily variability and climatology of the RCM. We also showed that the emulator remains consistent with its driving large scale, which leads to inconsistencies with the RCM. In the next section, we will develop this discussion further.

4 Discussion

4.1 On the inconsistencies between GCM and RCM

Several recent studies (Sørland et al. 2018; Bartók et al. 2017; Boé et al. 2020) have highlighted the existence of large scale biases for various variables between RCMs and their driving GCM, and have discussed the reasons behind these inconsistencies. From a theoretical point of view, it is still controversial as to whether these inconsistencies are for good or bad reasons (Laprise et al. 2008) and therefore if the emulator should or should not reproduce them. In our study, the emulator is trained in such a way that it focuses only on learning the downscaling function of the RCM, i.e., from the RCM large scale to the RCM small scale. Within this learning framework, the emulator can not learn GCM-RCM large-scale inconsistencies, if there should be any. Therefore, when GCM inputs are given to the emulator, the estimated RCM downscaling function is applied to the GCM large scales fields, and any GCM-RCM bias is conserved between the emulated serie and the RCM one. Figure 12 shows the biases for the present-climate climatology between the GCM and the UPRCM over the input domain for TA700 and ZG700, at the GCM resolution. The GCM seems generally warmer than the UPRCM, which could partly explain the warm bias observed between the emulator results and the RCM truth in present climate (e.g., Fig. 11). These large scale biases between GCM and RCM raise the question of using the RCM to evaluate the emulator when applied to GCM data. Indeed, if these inconsistencies are for bad reasons (e.g., inconsistent atmospheric physics or inconsistent forcings), the emulator somehow corrects the GCM-RCM bias for the emulated variable. In this case, the RCM simulation cannot be considered as the targeted truth. However, if the RCM revises the large scale signal for good reasons (e.g., upscaling of the local added-value due to refined representation of physical processes), then the design of the emulator should probably be adapted.

In future studies, we plan to use RCM runs with spectral nudging (Colin et al. 2010), two-way nested GCM-RCM runs or global high-resolution simulations for testing other modelling frameworks to further develop and evaluate the emulator.
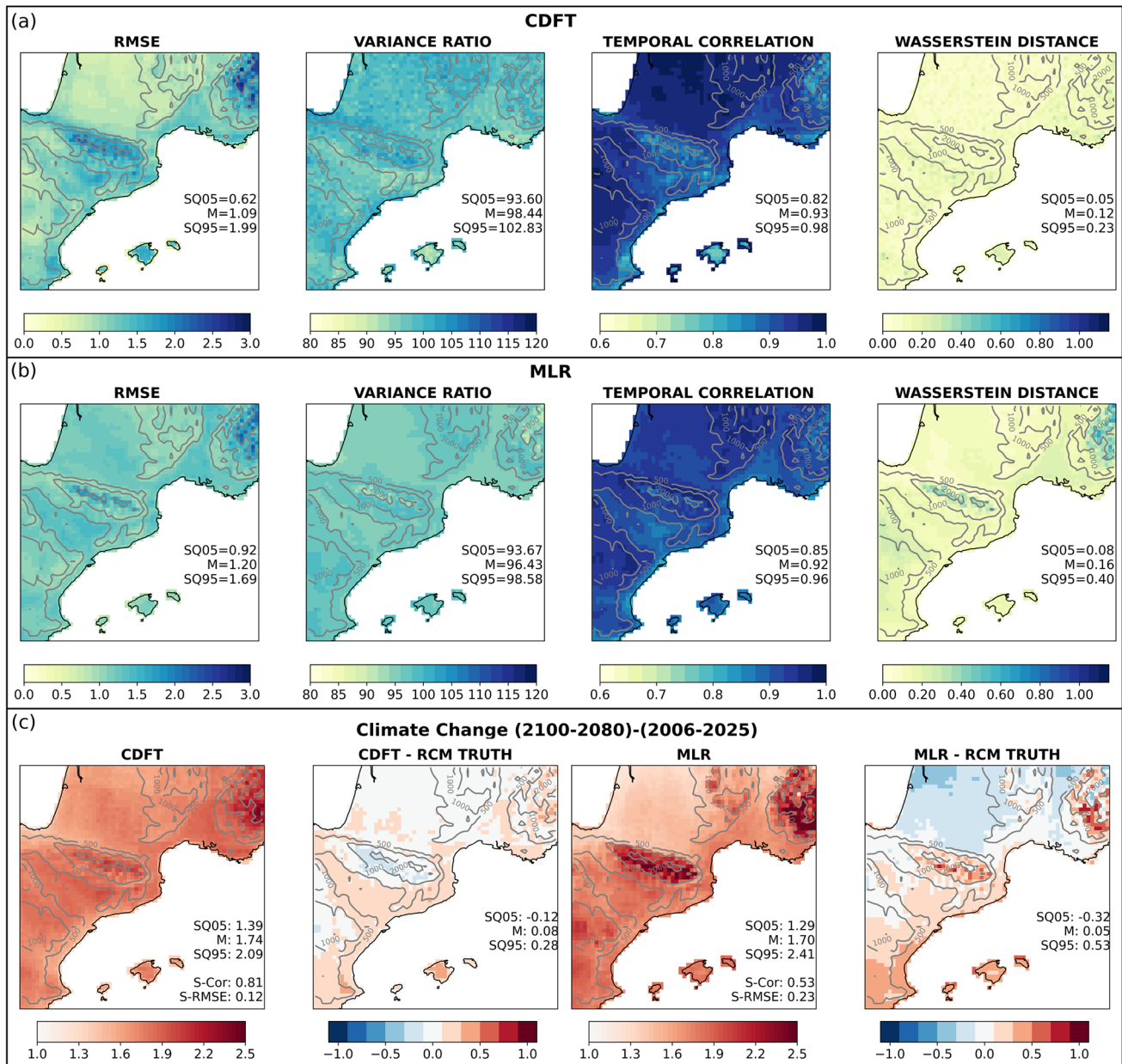
Fig. 8 a (Resp (**b**)) Maps of performance scores of the CDFt emulator (resp. of the MLR emulator) with respect to the RCM truth computed over the whole simulation period in perfect model evaluation. For each map, the values of the spatial mean and super-quantiles (SQ05 and SQ95) are added. **c** Climate change maps and the difference maps with respect to the RCM truth for the CDFt emulator and the MLR emulator. As on Figs. 6 and 7 the spatial scores are added on the maps

4.2 On the stationary assumption

In the introduction of this paper, we state that the stationary assumption is one of the main limitations of empirical statistical downscaling. The emulator proposed here is similar in many ways to a classical ESD method, the main difference being that the downscaling function is learnt in a RCM simulation. The framework used to train the emulator is a good opportunity to test the stationary assumption for the RCM-emulator. We train

the same emulator, with the same neural network architecture and same predictor set, but on the historical period (1951–2005) only. Results are reported in Tables SM.T1 and SM.T2 in the supplementary material, this version being named 'Emul-Hist'.

The perfect model (Table SM.T1) evaluation constitutes the best way to evaluate the validity of this assumption properly. Emul-Hist has a cold bias over the whole simulation regarding the RCM truth and the range of emulators described in Sect. 4.4. Moreover, this bias
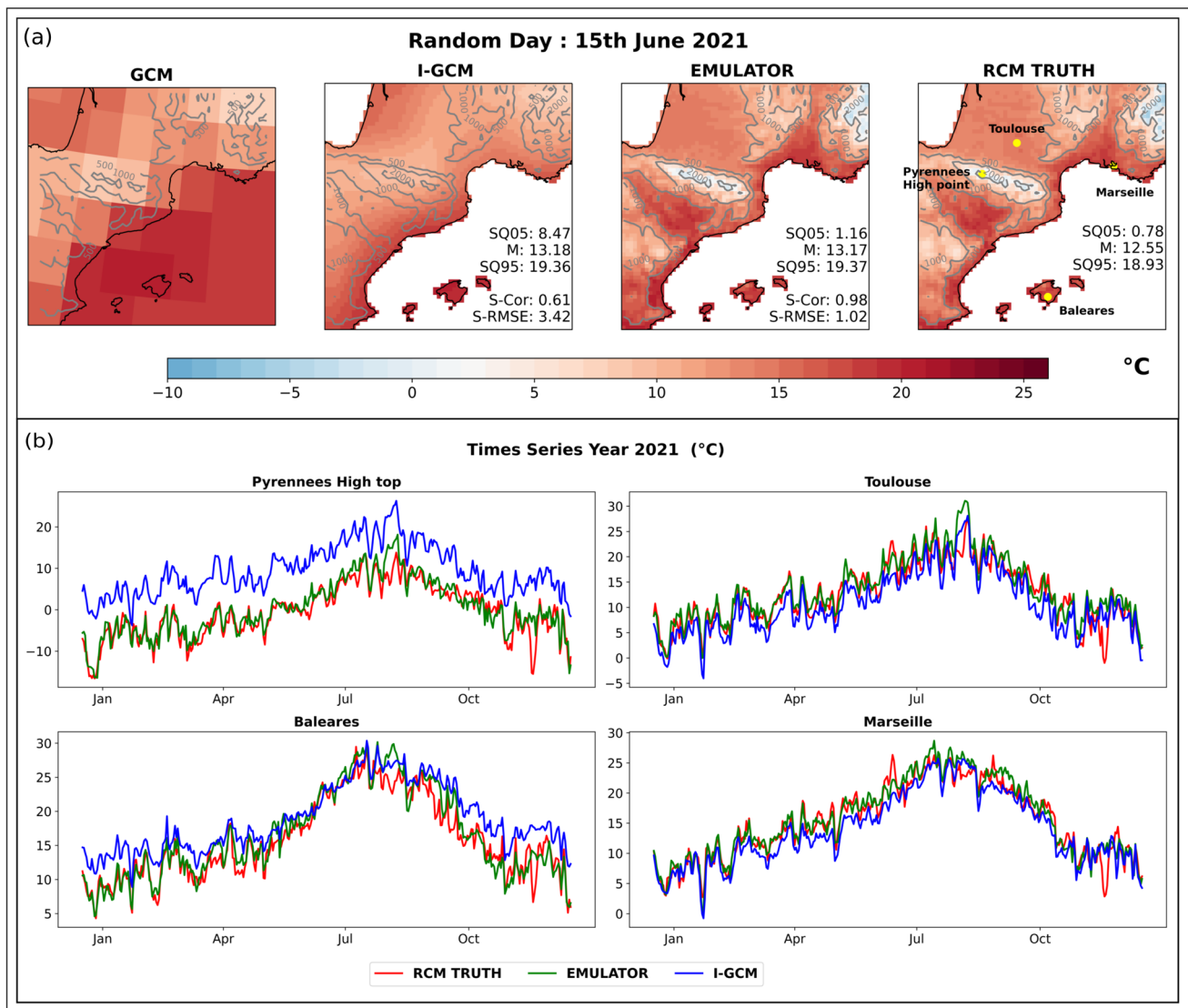
Fig. 9 Randomly chosen illustration of the production of the emulator (in evaluation step 2) with inputs coming from the GCM: **a** near-surface temperature (°C) at a random day over the target domain for the the GCM, the interpolated GCM, the emulator, and RCM truth and **b** random year time series (°C) for 4 particular grid points

is much stronger for the future period (from 0.3 °C in 2006–2025 to 1.3 °C in 2080–2100). Emul-Hist manages to reproduce only 30% of the climate change simulated by the RCM. It also fails to capture most of the spatial structure of the warming since the spatial correlation between the Emul-hist and RCM climate change maps (0.86) is closer to the I-UPRCM (0.82) and than the main emulator (0.95) (see Fig. 13). The Emul-hist average RMSE (1.35 °C) over the whole series is also out of emulator range ([0.8;0.86]). Results in GCM evaluation are also presented (Table SM.T2), but due to the lack of proper reference, it is difficult to use them to assess the stationary assumption. However, it presents the same cold bias regarding the ensemble of emulators. These results demonstrate the

importance of training the emulator in the wider range of possible climate states.

We underline that not all ESD methods are expected to behave that poorly with respect to projected warming. However learning in the future is one of the main differences between our RCM emulator approach and the standard ESD approach that relies on past observations.

4.3 On the selection of the predictors

For this study, we chose to use a large number of inputs with almost no prior selection, leaving the emulator to select the right combination of inputs for each grid point. However, we are aware that it involves a lot of data, which
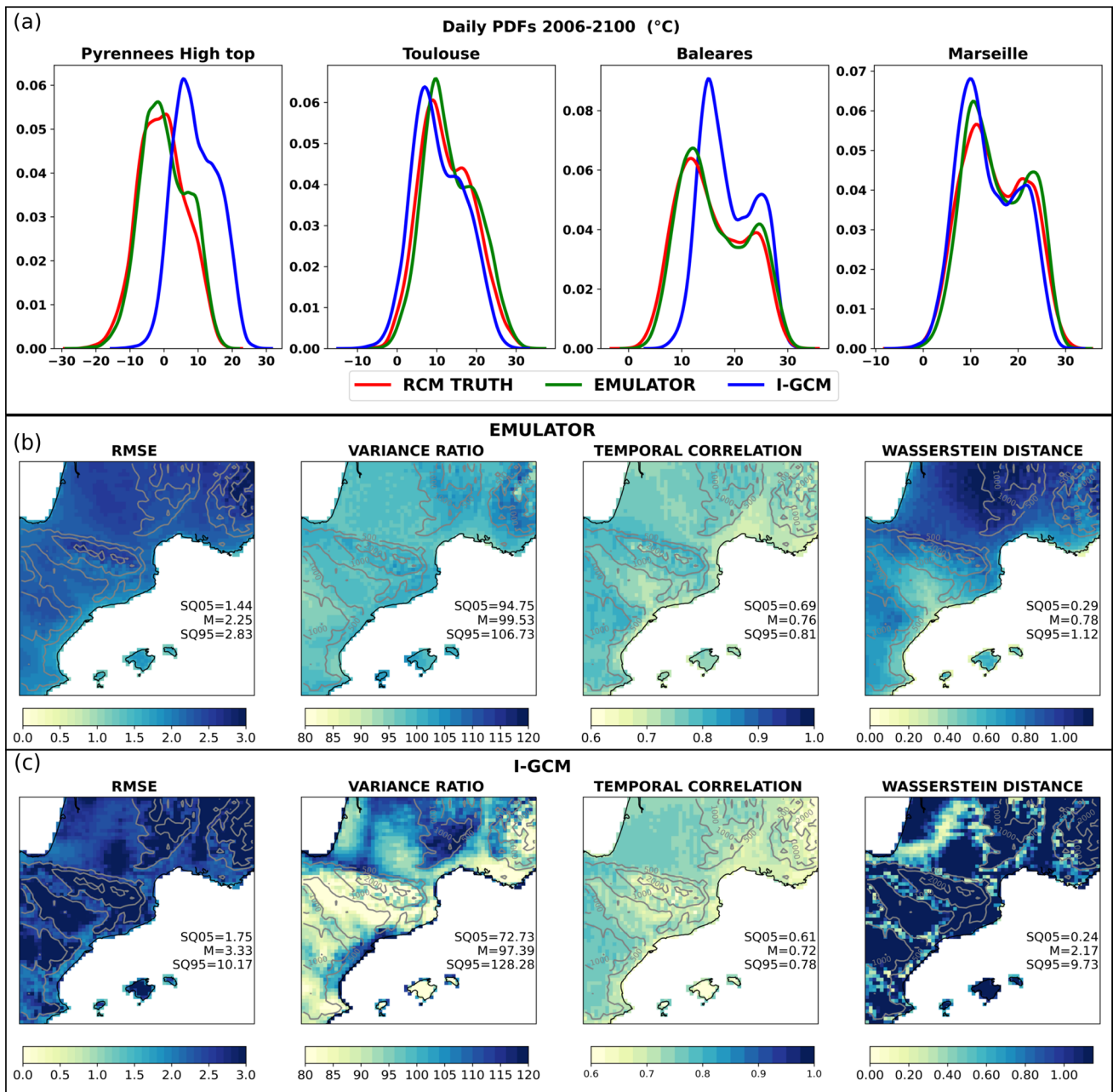
Fig. 10 a Daily probability density functions from the RCM truth, the emulator (in evaluation step 2) and the I-GCM at 4 particular grid points over the whole simulation period. **b** (Resp. **c**) Maps of performance scores of the emulator (resp. of the I-GCM) with respect to the RCM truth computed over the whole simulation period. For each map, the spatial mean and super-quantiles (SQ05 and SQ95) are added

is not always available, and leads to several computations due to the different preprocessing steps described in Sect. 2. For this reason, we tried to build an emulator with fewer inputs in *X* including only the variables from Table 2 at 700 hpa and removing the solar and ozone forcings from *Z*. The results are reported in Tables SM.T1 and SM.T2. They show that having more inputs increases the quality of the emulated series, but the "cheap" emulator

presents satisfying results and can be considered as a serious option. For instance, in the first step of evaluation (Table SM.T1), the cheap emulator provides a good representation of the spatial structure (S-Cor $= 0.94$ vs 0.96 for the main emulator) but is more biased than the main emulator (-0.3 versus $-0.18\,°C$). Having a specific selection of inputs for specific areas would probably increase the performance of the cheap emulator. However, this is

not in the spirit of the tool proposed here, which aims to be as simple and as general as possible.

Moreover, we also wanted to discuss the smoothing of the 2D inputs mentioned in Sects. 2.2 and 2.5. There are two reasons behind it. First of all, it allows getting closer to the effective resolution of the GCM models (Klaver et al. 2020). Secondly, it deletes any local scale information that could remain after the upscaling step when constructing the UPRCM inputs for the training (see Sect. 2.5). To verify the usefulness of this step, we trained an emulator without prior smoothing of the inputs. The results are reported under the name "No_Smoothing" in Tables SM.T1 and SM.T2, in Online Appendix. The "No_smoothing" emulator performs slightly better than the main emulator in perfect model world. It has for example a slightly better RMSE (0.76 °C versus 0.82 °C for the main emulator) and a better ratio of variance (96.2% versus 95.2%). However, in the GCM world, the "No_smoothing" emulator shows less good results than the original emulator. The ratio of variance and the spatial correlation over the different maps are the most adapted scores to compare two emulators in GCM world (see Sects. 3.3 and 4.1). The ratio of variance of the "No_smoothing" emulator falls to 83% for the worst points, while it is 95% for the main emulator. The spatial correlation also confirms that it fails to reproduce the entire spatial structure of the RCM. For example, when looking at the climate change map, the main emulator's spatial correlation is equal to 98% while the one of the No_smoothing emulator is 72%.

Finally, the last test on the inputs concerns the non-inclusion of the low resolution surface temperature in the predictors. The motivation behind this choice comes from the perfect model training detailed in Sect. 2.5. The intuition is that too much high resolution information remains in TAS_UPRCM. The emulator uses undoubtedly this information, leading to a less accurate downscaling of GCM fields. To test this hypothesis, we trained an emulator including TAS, and the results are referred as "TAS_included" in Tables SM.T1 and SM.T2. The "TAS_included" emulator performs better in the perfect model world but not in the GCM world. The most striking score is probably the variance ratio. In GCM world, it ranges from 83 to 120% which is closer to our very basic benchmark ([72;128]%) than to the main emulator ([95;106]%). Moreover, the spatial correlations are also much less good than for the main emulator. These results confirm clearly our hypothesis.

4.4 On the non reproducibility of neural network training

While neural networks have experienced considerable success over the last decades and the number of applications is constantly increasing, they have also been largely criticised for their lack of transparency due to an excessive number

of parameters. Several studies (see Guidotti et al. 2018, for review) have tried to provide the keys to "open the black box". However, users of deep neural networks should also be aware of their non-reproducibility. Indeed the training of deep neural networks with GPUs involves several sources of randomness (initialisation, operation ordering, etc.). A few recent studies have raised the issue for medical applications with a UNet (Marrone et al. 2019; Bhojanapalli et al. 2021), but to the best of our knowledge, no study has addressed it.

In order to document this issue for the RCM emulator developed in this paper and assess its robustness, we propose a Monte Carlo experiment where the same configuration of the emulator (as described in Sect. 2) is trained 31 times, resulting in 31 emulators. The results of this experiment are illustrated in Fig. 14 and summarised in Tables SM.T1 and SM.T2. The results of Sect. 3 are based on a randomly picked emulator, mentioned as the main emulator in appendix and plotted in darker green in Fig. 14. All the emulators from the Monte Carlo experiment show consistent results with the ones presented in Sect. 3. The RMSEs (and correlated scores such as variance) are really close to each other ($[0.8;0.86]°C$), which is expected since it is the loss function used to fit the neural network. However, the path taken to minimise the loss during training might vary from one emulator to another, leading to bigger differences in climatological metrics (see tables). For instance, in the UPRCM evaluation step, the average error in the future climatology varies from 0.003 to 0.139 °C. The results from the GCM world evaluation step (Table SM.T2) are similar to those in the perfect model framework. This stability shows again the robustness of the emulator when using GCM large scale fields.

We believe that the readers of this study and any potential emulator users should be aware of this characteristic of deep learning neural networks. However, it is worth mentioning that this randomness in the training of the network does not impact the key conclusions of the results (Sect. 3), which prove their robustness.

5 Conclusion

This study aims to explore a novel hybrid downscaling approach that emulates the downscaling function of a RCM. That is to say learning the transformation of the large-scale climate information into a local climate information performed by a regional climate model. Here, we develop this approach for the near surface temperature and a Southwest European domain. This new method, called RCM-emulator, is designed to help increase the size of the high-resolution regional simulation ensembles at a lower cost.

To achieve this overall goal, we develop a specific conceptual framework. The emulator is trained using existing
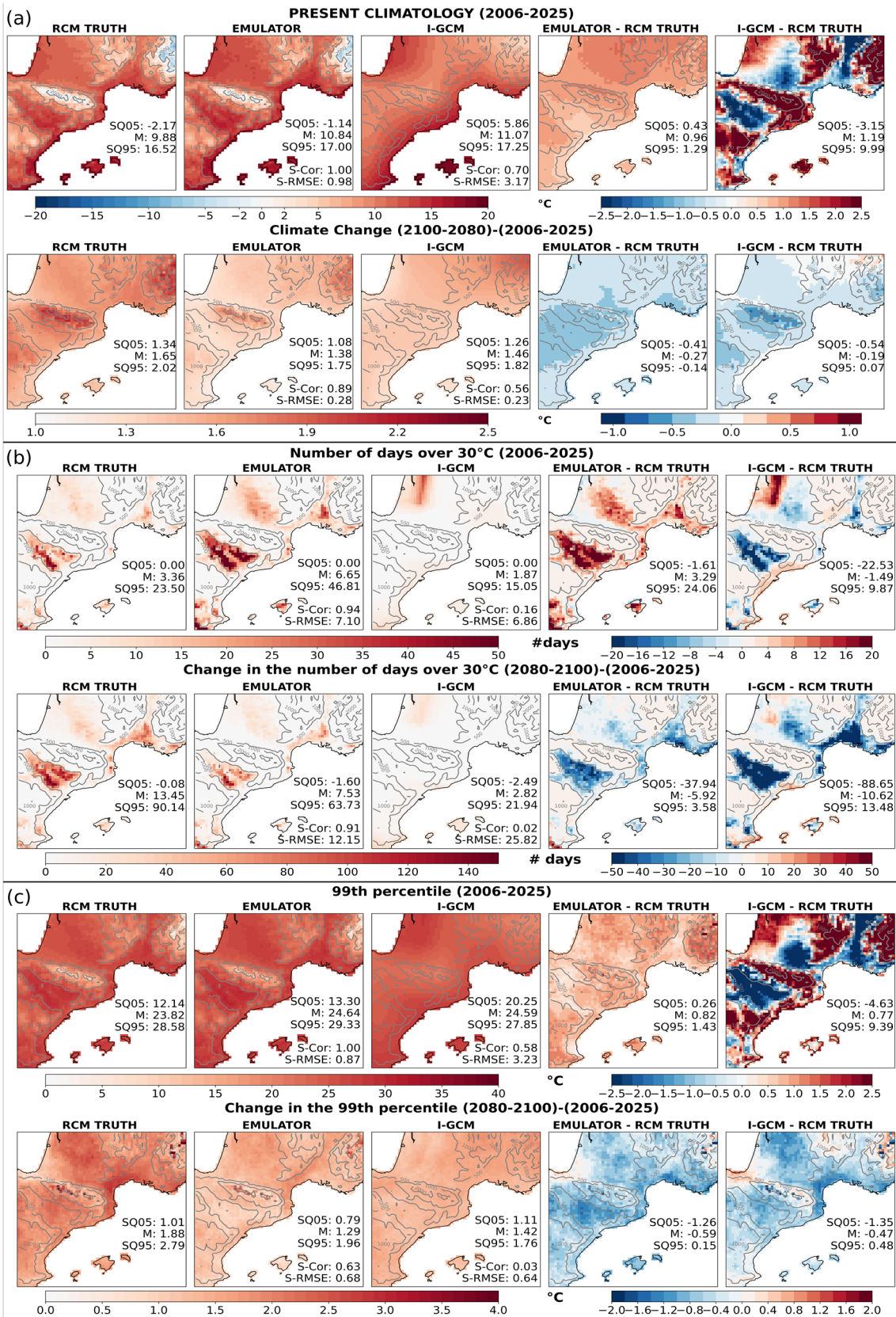
Fig. 11 **a** Maps of long-term mean climatologies, **b** number of days over 30 °C and **c** 99th percentile of daily near-surface temperature for a present-climate period (2006–2025) and for the climate change signal (2080–2100 minus 2006–2025), for the RCM truth, the emulator (in evaluation step 2) and the I-GCM. On each line, the two last maps show the error map of the emulator and the I-GCM. For each map, the spatial mean and super-quantiles (SQ05 and SQ95) are added, as well as the spatial correlation and spatial RMSE for the emulator and I-GCM maps

RCM simulations which allows it to learn the large scale/local scale relationship in different climates and in particular in future climate. Simply speaking, the general functioning of a RCM can be broken down into a large scale transformation and a downscaling function. To focus on the downscaling function, the emulator is trained in a perfect model framework, where both predictors and predictand come from the same RCM simulation. This framework implies to carefully prepare and select the predictors, to ensure that no unwanted high resolution information remains in the training predictors. The emulator takes daily large-scale and low-resolution information as input and produces daily maps of the near surface temperature at the RCM resolution. It is worth noting that the downscaling function likely depends on the RCM choice. So the emulator developed in this study is RCM-dependent.

Technically speaking, the RCM-emulator is based on a fully convolutional neural network algorithm, called UNet. A key point of the emulator is the substantial computational

gain regarding RCM computational cost. Training the emulator used in this study took two hours on a GPU. While this time depends on the target domain size it will never exceed some hours even for much bigger domains. Once the emulator is trained, the downscaling of a new low resolution simulation takes less than a minute. These time lengths do not include the preparation of the inputs; nevertheless the gain remains evident when a RCM simulation involves weeks of computation on a super-computer.

The emulator is evaluated in both perfect model and GCM worlds. The results show that the emulator generally fulfils its mission by capturing very well the transformation from low resolution information to the high resolution near surface temperature. Firstly, the emulator is robust to different sources of input, which validates our conceptual framework. Secondly, the emulator succeeds very well in reproducing the high resolution spatial structure and daily variability of the RCM. The perfect model evaluation shows that the emulator is able to reproduce the original series almost perfectly. Moreover, it appears clearly that training the emulator in future climate improves its ability to reproduce warmer climate. Nevertheless the emulator shows some limitations in accurately simulating extreme events and of the complete climate change magnitude. Future work should focus on these two aspects to further improve the emulator ability. It is worth noting that similar emulators could easily be built over different domains, for different RCMs and for surface variables other than near-surface temperature.
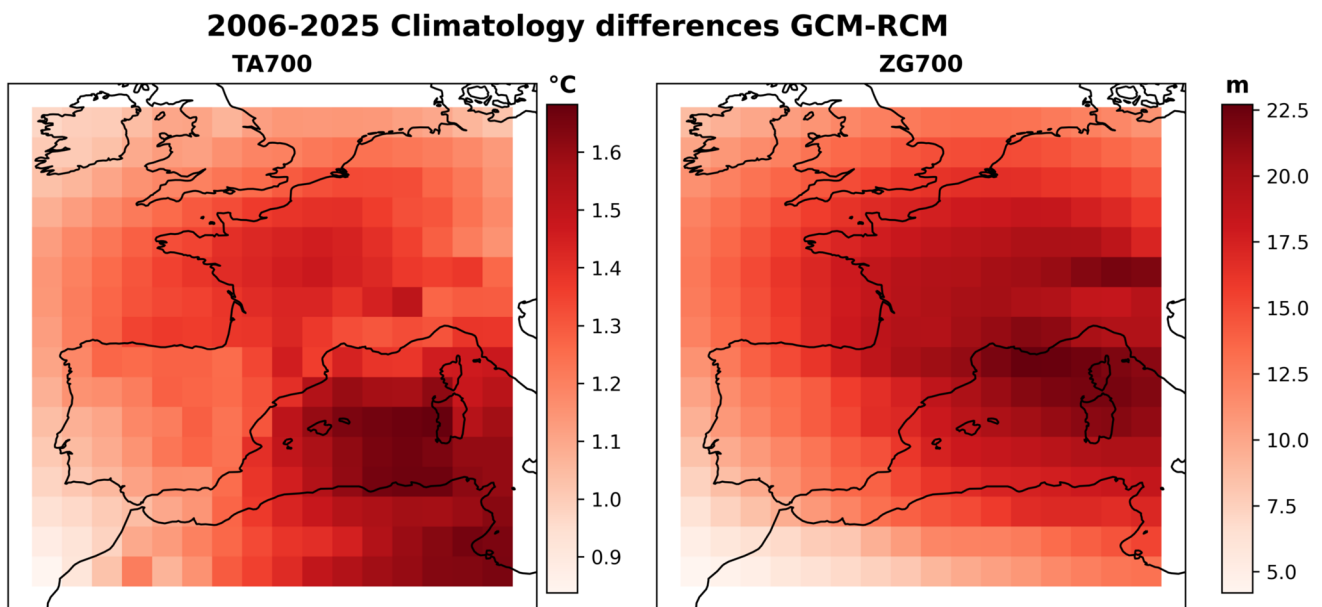

Fig. 12 Present (2006–2025) climatology differences for the atmospheric temperature and geopotential at 700 hpa: CNRM-CM5 RCP45 minus ALADIN63 driven by CNRM-CM5 RCP45 upscaled on the GCM grid
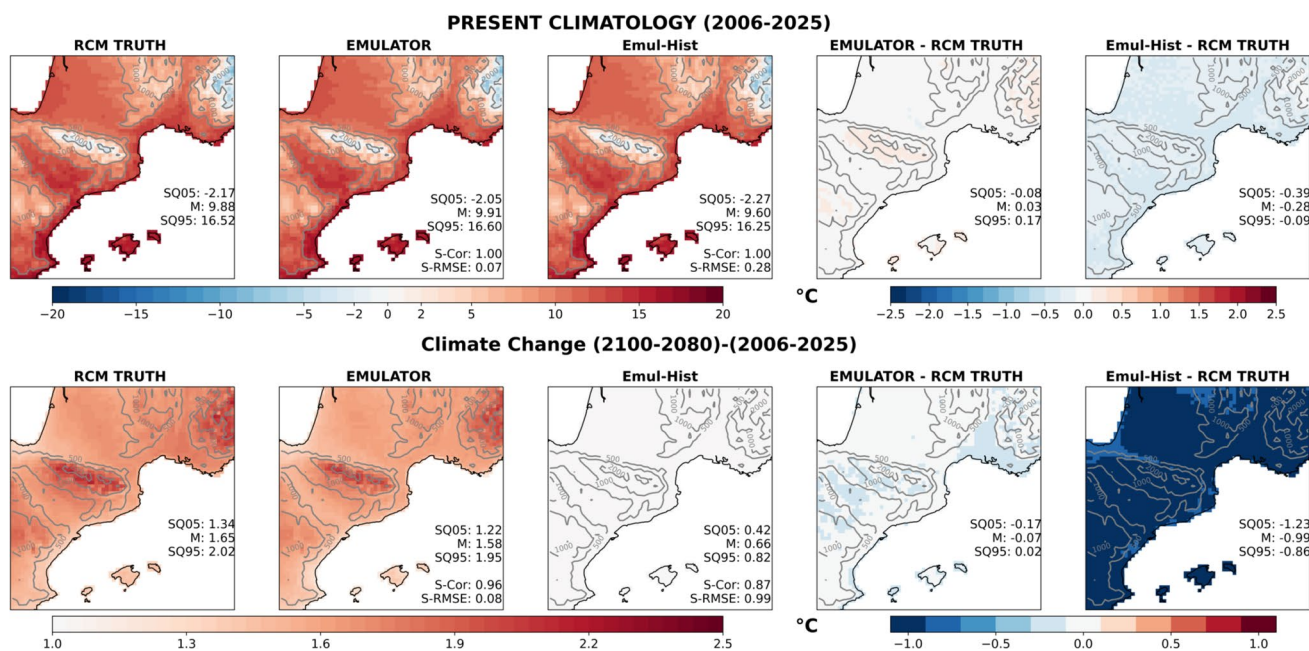
Fig. 13 Maps of climatologies over present period (2006–2025) and climate change signal (2080–2100 versus 2006–2025), for the RCM truth, the Emulator presented in Sect. 3 and the Emul-Hist. The two last columns correspond to the error maps of the Emulator and Emul-Hist with respect to the RCM truth. For each map, the spatial mean and super-quantiles (SQ05 and SQ95) are added, as well as the spatial correlation and spatial RMSE for the Emul-Hist and I-UPRCM maps

This UNet emulator has been compared to two other possible emulators based on more classical ESD methods using the well adapted perfect model framework. The exercise allows us to put into perspective the result of the UNet emulator and validate our choice as it outperforms these simpler emulators. Moreover, it is worth mentioning that this study does not target to identify the best possible emulator but aims to shed light on the feasibility to emulate the RCM complexity at high-frequency and high-resolution, and to propose one good technical solution to do it. We are fully aware that our method will certainly be one among others and we hope that the scientific community will propose new and probably better solutions for RCM emulators in the next years.

Finally, and even if it was not our original goal, this study highlights, as others before, the RCM-GCM inconsistencies at large scales. As the emulator focuses on the downscaling function of the RCM, it does not learn to reproduce these large scale transformations. This raises the question of how to evaluate the emulator when downscaling GCM simulation. An RCM run might not be the correct reference in such a case. Secondly, it also puts into question the final role of a RCM-emulator. Under the strong hypothesis that the large scale transformation carried out by the RCM results from physics or forcings inconsistencies—which will require further investigation—the emulator provides a high-resolution simulation that is corrected from the GCM-RCM large-scale inconsistencies.

Appendices

Emulators inter-comparison

In order to validate the neural network based emulator presented in this study, we proposed two other emulators relying on standard empirical statistical downscaling methods: CDFt and a Multiple Linear Regression. The idea is not to identify the best emulator, but to give a baseline and put into perspective the performance of the UNet architecture. The methods were chosen because both are well established and have been widely applied. Moreover they are good example of to two different families of ESD methods, Model Output Statistic (MOS, for CDFt) and Perfect Prognosis (PP, for MLR). Both methods are part of the large ESD method intercomparison project, VALUE (Gutiérrez et al. 2019).

CDFt was introduced by Michelangeli et al. (2009) and used in several studies as Vrac et al. (2012), Lavaysse et al. (2012), Vigaud et al. (2013) and Famien et al. (2018). It is based on the transformation of the Cumulative Distribution

Function (CDF) of the low resolution variable into the CDF of the high resolution variable. In our case the low resolution variable is the smoothed surface temperature from the UPRCM conservatively interpolated on the RCM grid. We took the smooth TAS in order to be as coherent as possible with the UNet emulator inputs. The high resolution variable we try to reproduce is then the RCM near surface temperature TAS.

To understand the theory behind CDFT, let $F_{T,Lr}$ be the CDF of the low resolution surface temperature from the training set, and $F_{T,Hr}$ the CDF of the high resolution surface temperature from the training set. Then CDFt relies on the assumption that there exists a transformation T such that

$$T(F_{T,Lr}(x)) = F_{T,Hr}(x). \tag{10}$$

Then by replacing x by $F_{T,Lr}^{-1}(u)$, with u any probability in [0, 1] we get a simple definition of T :

$$T(u) = F_{T,Hr}(F_{T,Lr}^{-1}(u)). \tag{11}$$

And under the assumption that T is the same in the training set than in the evaluation set it comes that:

$$T(F_{E,Lr}(x)) = F_{E,Hr}(x), \tag{12}$$
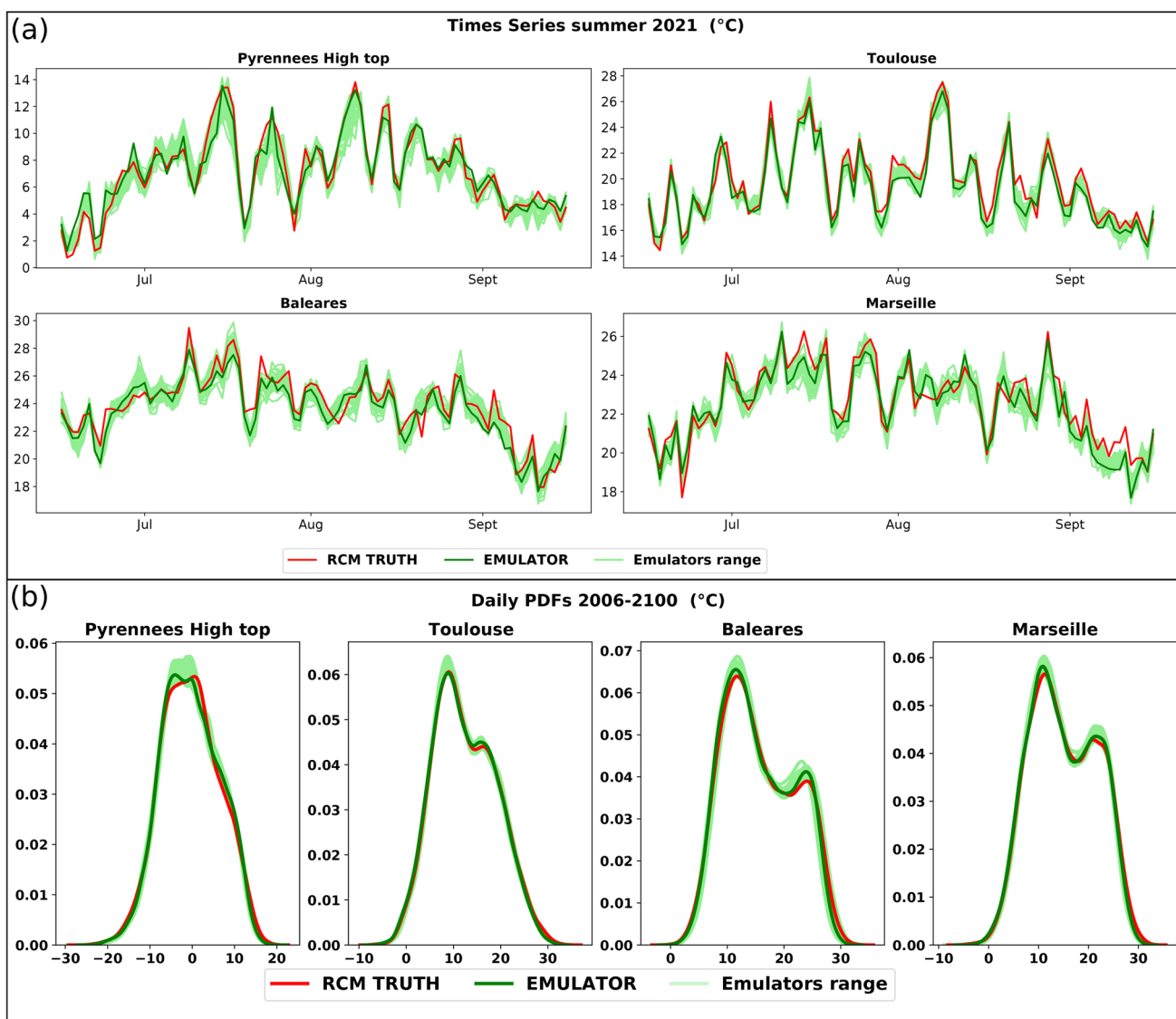
Fig. 14 Illustration of the results for the Monte Carlo experiment. **a** Present a year time series for 4 given points and **b** the pdfs on the whole serie. The red line refers to the RCM truth, the dark green line is the main emulator, and the light green lines are the 30 emulators from the Monte Carlo experiment

with $F_{E,Lr}$ (resp $F_{E,Hr}$) the CDF of the low (resp high) resolution surface temperature in the evaluation set. Then $F_{E,Hr}$ can be determined as :

$$F_{E,Hr}(x) = F_{T,Hr}(F_{T,Lr}^{-1}(F_{E,Lr})). \qquad (13)$$

It is then possible to reconstruct the high resolution series from the low resolution one, following a Quantile-Quantile approach between $F_{E,Hr}$ and $F_{E,Lr}$.

The second method is a Multiple Linear Regression (MLR), which has also been used in multiple studies (Huth 2002; Huth et al. 2015; Manzanas et al. 2018; Joshi et al. 2015). It takes exactly the same list of inputs as the UNet emulator (Table 2), but by taking a single grid point for the 2D variables. The predicted high resolution temperature is then:

$$\hat{Y} = \alpha_0 + \sum_{i=0}^{M} \alpha_i P_i, \qquad (14)$$

with M the number of predictors.

Both CDFt and MLR emulators are trained in the same conditions as the UNet emulator. The simulation used for training is the ALADIN63 simulation forced by CNRM-CM5, covering the 1950–2100 period with the RCP8.5 scenario from 2006. The ALADIN63 simulation forced by CNRM-CM5, covering the 2006–2100 period with the RCP4.5, is used to evaluate and compare the emulators. The evaluation is made in perfect model framework as it gives the best conditions to properly compare the results. The results are reported in Figs. 15 and 16, where the two simpler emulators are compared to the UNet Emulator used in the study. Figure 15 is showing the performances of the three emulators regarding the performance metrics defined in Sect. 2.6. Both MLR and CDFt show a good reproduction of the pdfs, the daily variance of the original RCM truth, as well as a good temporal correlation. However regarding all these scores the UNet emulator performs better, and this is well summarized by the RMSE maps, where the UNet Emulator shows much more accurate results. Figure 16 presents the ability of the emulators to reproduce some climatological statistics that are usually studied in climate studies. The conclusions are very similar, the MLR and CDFt emulators have good results. However the UNet emulator captures better the complexity of the high resolution temperature simulated by the RCM. This results is clear when watching at the spatial correlation of the Fig. 16.

To conclude, both MLR and CDFt are reasonable candidates for the emulator, as they were for statistical downscaling, however the UNet emulator provides better results, specially in the more complex regions. Moreover, as a comparison, the very simple multiple

linear regression involves 233 thousand parameters over the whole target domain when it uses a single grid point as input for each of the 2D variables used. However, if we want to fit a MLR with the full 2D grid of input as for UNet, the model would imply about 250 times more parameters (16*16), so more than the UNet. For these reasons we decided to mainly present the UNet emulator.

Description of the UNet-emulator architecture

We describe here with more details the Fig. 2, which schematize the UNet architecture used for the emulator. On the figure, the different blocks are represented by colorful arrows. Each arrow is described below :

– Each blue arrow $\Rightarrow$ corresponds to a convolution block composed of a layer built with a set of convolutional 2×2 filters. The number of filters increases all along the contracting part (the number of filters is respectively 64, 128, 256, 512 and 1024 and is given on the top of each block in Fig. 2). The outputs of this layer are then normalised with a batch normalisation layer (see e.g. Ioffe and Szegedy 2015) to improve the statistical robustness of the layer. Finally the ReLu activation function completes this block.

$$ReLu : R(z) = max(0, z)$$

– Each red arrow $\Downarrow$ is a Maxpooling layer. It performs 2×2 pooling on each feature map, which simply divides by 2 the spatial dimension by taking the maximum of each 2×2 block. It is applied through all convolution block outputs in the encoding path. The size of the images is indicated on the side of each block on Fig. 2.
– Each green arrow $\Uparrow$ is a transpose convolution layer. It allows to perform up-sampling in the expansive part of the algorithm. It multiplies the spatial dimension by 2 by applying the same connection as the classical convolution but in the backward direction.
– The black arrow $\Longrightarrow$ represents a fully connected dense network of 4 layers which is applied on the 1-dimensional inputs (Z).
– The grey arrow $\Rightarrow$ represents a simple concatenation layer which recalls the layers from the encoding path in the decoding one.
– Finally, the light blue arrow $\Rightarrow$ is the output layer, which is a simple convolutional layer with a single filter and a linear activation function.
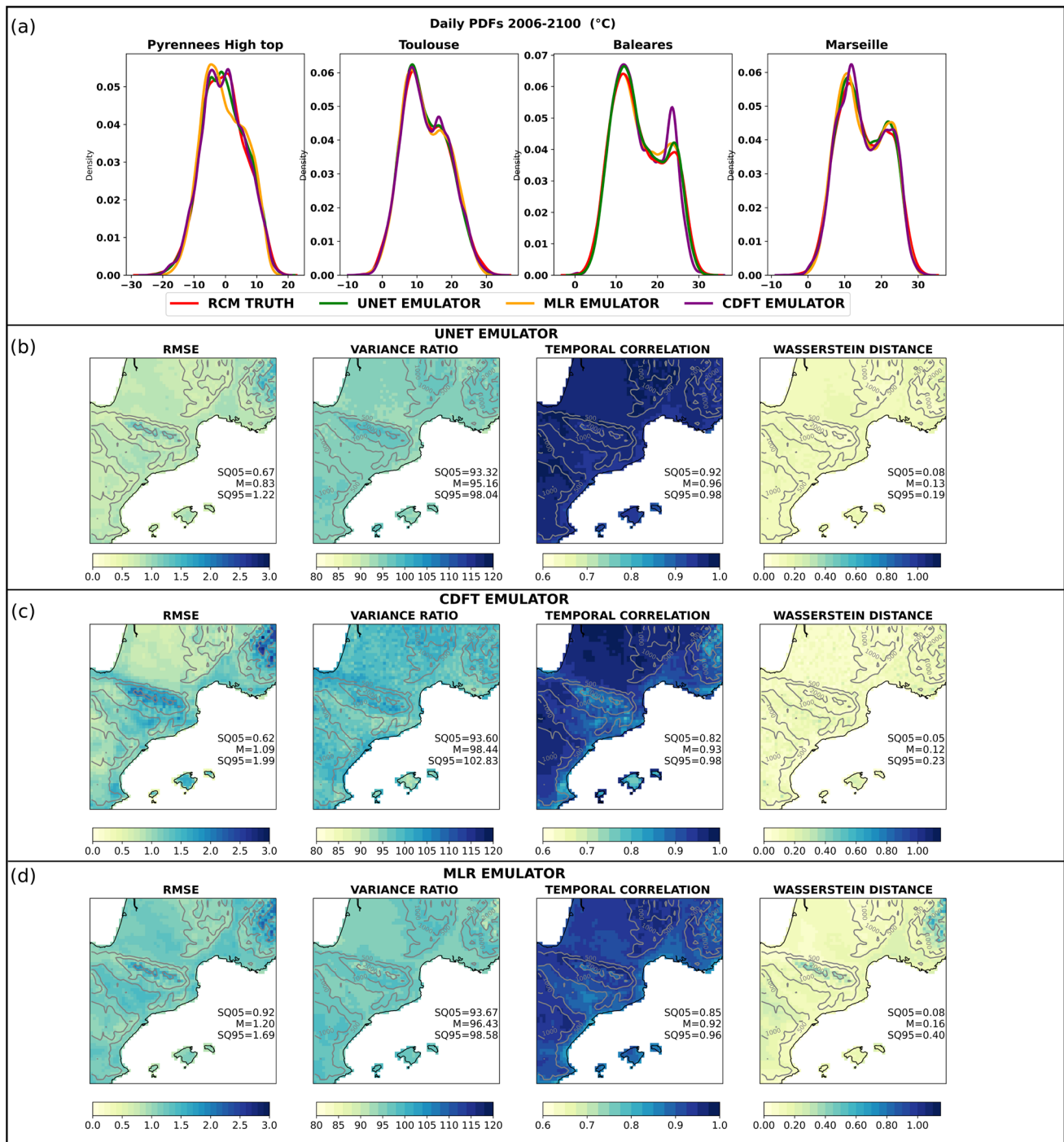
Fig. 15 **a** Daily probability density functions from the RCM truth, the UNet, the CDFt and the MLR emulators at 4 particular grid points over the whole simulation period. **b** (resp. (c and d)) Maps of performance scores of the UNet emulator (resp. the CDFt and MLR) with respect to the RCM truth computed over the whole simulation period. For each map, the spatial mean and super-quantiles (SQ05 and SQ95) are added
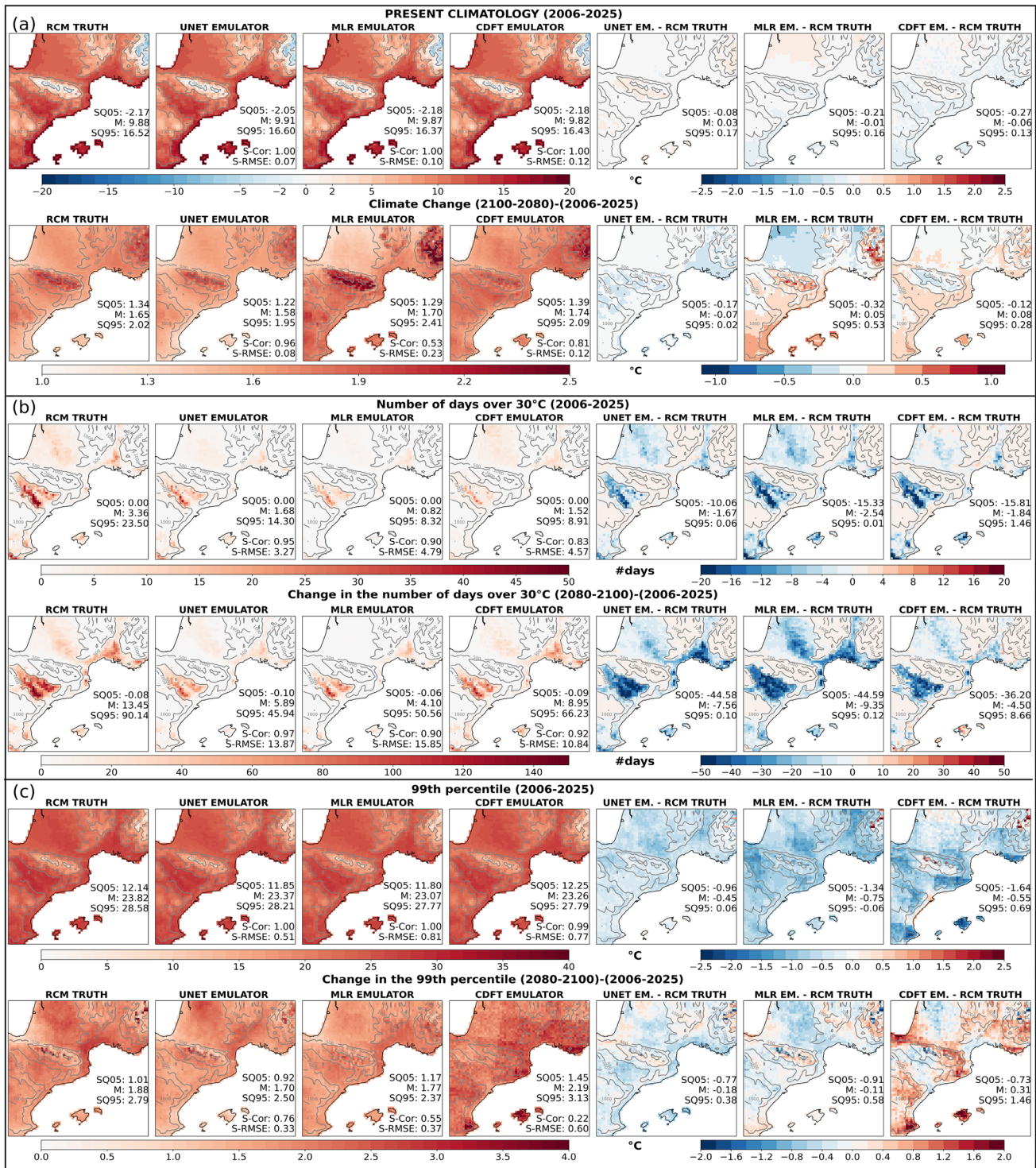
Fig. 16 **a** Maps of climatologies, **b** number of days over 30 °C and **c** the 99th percentile of daily temperature, for the present-climate period (2006–2025) and for the climate change signal (2080–2100 minus 2006–2025), for the RCM truth, the UNet, the MLR and CDFt emulators in perfect model mode. On each line, the two last maps show the error maps of the emulators. For each map, the spatial mean and super-quantiles (SQ05 and SQ95) are added, as well as the spatial correlation and spatial RMSE with respect to the RCM truth maps

Declarations

Conflict of interest Nothing to mention

References

Baño-Medina J, Manzanas R, Gutierrez JM, Gutiérrez JM (2020) Configuration and intercomparison of deep learning neural models for statistical downscaling. Geosci Model Dev 13(4):2109–2124. https://doi.org/10.5194/gmd-13-2109-2020

Baño-Medina J, Manzanas R, Gutiérrez JM (2021) On the suitability of deep convolutional neural networks for continental-wide downscaling of climate change projections. Clim Dyn. https://doi.org/10.1007/s00382-021-05847-0

Bartók B, Wild M, Folini D, Lüthi D, Kotlarski S, Schär C, Vautard R, Jerez S, Imecs Z (2017) Projected changes in surface solar radiation in CMIP5 global climate models and in EURO-CORDEX regional climate models for Europe. Clim Dyn 49(7–8):2665–2683. https://doi.org/10.1007/s00382-016-3471-2

Berg N, Hall A, Sun F, Capps S, Walton D, Langenbrunner B, Neelin D (2015) Twenty-first-century precipitation changes over the los angeles region. J Clim 28(2):401–421. https://doi.org/10.1175/JCLI-D-14-00316.1

Bhojanapalli S, Wilber K, Veit A, Rawat AS, Kim S, Menon A, Kumar S (2021) On the reproducibility of neural network predictions. pp 1–19. arXiv preprint arXiv:2102.03349

Boé J, Somot S, Corre L, Nabat P (2020) Large discrepancies in summer climate change over Europe as projected by global and regional climate models: causes and consequences. Clim Dyn. https://doi.org/10.1007/s00382-020-05153-1

Colette A, Vautard R, Vrac M (2012) Regional climate downscaling with prior statistical correction of the global climate forcing. Geophys Res Lett 39(13):1–5. https://doi.org/10.1029/2012GL052258

Colin J, DéQué M, Radu R, Somot S (2010) Sensitivity study of heavy precipitation in limited area model climate simulations: influence of the size of the domain and the use of the spectral nudging technique. Tellus Ser Dyn Meteorol Oceanogr 62(5):591–604. https://doi.org/10.1111/j.1600-0870.2010.00467.x

Dayon G, Boé J, Martin E (2015) Transferability in the future climate of a statistical downscaling method for precipitation in France. J Geophys Res Atmos 120:1023–1043. https://doi.org/10.1002/2014JD022236

Déqué M, Rowell DP, Lüthi D, Giorgi F, Christensen JH, Rockel B, Jacob D, Kjellström E, De Castro M, Van Den Hurk B (2007) An intercomparison of regional climate simulations for Europe: assessing uncertainties in model projections. Clim Change 81(SUPPL. 1):53–70. https://doi.org/10.1007/s10584-006-9228-x

Déqué M, Somot S, Sanchez-Gomez E, Goodess CM, Jacob D, Lenderink G, Christensen OB (2012) The spread amongst ENSEMBLES regional scenarios: Regional climate models, driving general circulation models and interannual variability. Clim Dyn 38(5–6):951–964. https://doi.org/10.1007/s00382-011-1053-x

Erlandsen HB, Parding KM, Benestad R, Mezghani A, Pontoppidan M (2020) A hybrid downscaling approach for future temperature and precipitation change. J Appl Meteorol Climatol 59(11):1–46. https://doi.org/10.1175/jamc-d-20-0013.1

Evin G, Somot S, Hingray B (2021) Balanced estimate and uncertainty assessment of European climate change using the large EURO-CORDEX regional climate model ensemble. Earth System Dyn Discuss. https://doi.org/10.5194/esd-2021-8

Eyring V, Bony S, Meehl GA, Senior CA, Stevens B, Stouffer RJ, Taylor KE (2016) Overview of the Coupled Model Intercomparison Project Phase 6 (CMIP6) experimental design and organization. Geosci Model Dev 9(5):1937–1958. https://doi.org/10.5194/gmd-9-1937-2016

Famien AM, Janicot S, Delfin Ochou A, Vrac M, Defrance D, Sultan B, Noël T (2018) A bias-corrected CMIP5 dataset for Africa using the CDF-t method: a contribution to agricultural impact studies. Earth Syst Dyn 9(1):313–338. https://doi.org/10.5194/esd-9-313-2018

Fantini A, Raffaele F, Torma C, Bacer S, Coppola E, Giorgi F, Ahrens B, Dubois C, Sanchez E, Verdecchia M (2018) Assessment of multiple daily precipitation statistics in ERA-Interim driven Med-CORDEX and EURO-CORDEX experiments against high resolution observations. Clim Dyn 51(3):877–900. https://doi.org/10.1007/s00382-016-3453-4

Fernández J, Frías MD, Cabos WD, Cofiño AS, Domínguez M, Fita L, Gaertner MA, García-Díez M, Gutiérrez JM, Jiménez-Guerrero P, Liguori G, Montávez JP, Romera R, Sánchez E (2019) Consistency of climate change projections from multiple global and regional model intercomparison projects. Clim Dyn 52(1–2):1139–1156. https://doi.org/10.1007/s00382-018-4181-8

Giorgi F, Torma C, Coppola E, Ban N, Schär C, Somot S (2016) Enhanced summer convective rainfall at Alpine high elevations in response to climate warming. Nat Geosci 9(8):584–589. https://doi.org/10.1038/ngeo2761

Guidotti R, Monreale A, Ruggieri S, Turini F, Giannotti F, Pedreschi D (2018) A survey of methods for explaining black box models. ACM Comput Surv. https://doi.org/10.1145/3236009

Gutiérrez JM, Maraun D, Widmann M, Huth R, Hertig E, Benestad R, Roessler O, Wibig J, Wilcke R, Kotlarski S, San Martín D, Herrera S, Bedia J, Casanueva A, Manzanas R, Iturbide M, Vrac M, Dubrovsky M, Ribalaygua J, Pórtoles J, Räty O, Räisänen J, Hingray B, Raynaud D, Casado MJ, Ramos P, Zerenner T, Turco M,

Bosshard T, Štěpánek P, Bartholy J, Pongracz R, Keller DE, Fischer AM, Cardoso RM, Soares PM, Czernecki B, Pagé C (2019) An intercomparison of a large ensemble of statistical downscaling methods over Europe: results from the VALUE perfect predictor cross-validation experiment. Int J Climatol 39(9):3750–3785. https://doi.org/10.1002/joc.5462

Hu X, Naiel MA, Wong A, Lamm M, Fieguth P (2019) RUNet: a robust UNet architecture for image super-resolution. IEEE computer society conference on computer vision and pattern recognition workshops 2019-June. pp 505–507. https://doi.org/10.1109/CVPRW.2019.00073

Huth R (2002) Statistical downscaling of daily temperature in Central Europe. J Clim 15(13):1731–1742. https://doi.org/10.1175/1520-0442(2002)015<1731:SDODTI>2.0.CO;2

Huth R, Mikšovský J, Štěpánek P, Belda M, Farda A, Chládová Z, Pišoft P (2015) Comparative validation of statistical and dynamical downscaling models on a dense grid in central Europe: temperature. Theor Appl Climatol 120(3–4):533–553. https://doi.org/10.1007/s00704-014-1190-3

Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. Proceedings of the 32nd international conference on machine learning, PMLR 1(37). pp 448–456

Jacob D, Petersen J, Eggert B, Alias A, Christensen OB, Bouwer LM, Braun A, Colette A, Déqué M, Georgievski G, Georgopoulou E, Gobiet A, Menut L, Nikulin G, Haensler A, Hempelmann N, Jones C, Keuler K, Kovats S, Kröner N, Kotlarski S, Kriegsmann A, Martin E, van Meijgaard E, Moseley C, Pfeifer S, Preuschmann S, Radermacher C, Radtke K, Rechid D, Rounsevell M, Samuelsson P, Somot S, Soussana JF, Teichmann C, Valentini R, Vautard R, Weber B, Yiou P (2014) EURO-CORDEX: new high-resolution climate change projections for European impact research. Reg Environ Change 14(2):563–578. https://doi.org/10.1007/s10113-013-0499-2

Joshi D, St-Hilaire A, Ouarda T, Daigle A (2015) Statistical downscaling of precipitation and temperature using sparse Bayesian learning, multiple linear regression and genetic programming frameworks. Can Water Resour J 40(4):392–408. https://doi.org/10.1080/07011784.2015.1089191

Klaver R, Haarsma R, Vidale PL, Hazeleger W (2020) Effective resolution in high resolution global atmospheric models for climate studies. Atmos Sci Lett 21(4):1–8. https://doi.org/10.1002/asl.952

Kotlarski S, Lüthi D, Schär C (2015) The elevation dependency of 21st century European climate change: an RCM ensemble perspective. Int J Climatol 35(13):3902–3920. https://doi.org/10.1002/joc.4254

Laprise R, de Elía R, Caya D, Biner S, Lucas-Picher P, Diaconescu E, Leduc M, Alexandru A, Separovic L (2008) Challenging some tenets of regional climate modelling network for regional climate modelling and diagnostics. Meteorol Atmos Phys 100:3–22. https://doi.org/10.1007/s00703-008-0292-9

Lavaysse C, Vrac M, Drobinski P, Lengaigne M, Vischel T (2012) Statistical downscaling of the French Mediterranean climate: assessment for present and projection in an anthropogenic scenario. Nat Hazards Earth Syst Sci 12(3):651–670. https://doi.org/10.5194/nhess-12-651-2012

LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2323. https://doi.org/10.1109/5.726791

Lecun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436–444. https://doi.org/10.1038/nature14539

Lemus-Canovas M, Brands S (2020) Assessing several downscaling methods for daily minimum and maximum temperature in a mountainous area. Are we able to statistically simulate a warmer climate in the Pyrenees? In: EGU General Assembly Conference Abstracts, EGU General Assembly Conference Abstracts, pp 11389. https://doi.org/10.5194/egusphere-egu2020-11389

Manzanas R, Gutiérrez JM, Fernández J, van Meijgaard E, Calmanti S, Magariño ME, Cofiño AS, Herrera S (2018) Dynamical and statistical downscaling of seasonal temperature forecasts in Europe: added value for user applications. Clim Serv 9:44–56. https://doi.org/10.1016/j.cliser.2017.06.004

Maraun D, Widmann M (2018) Statistical downscaling and bias correction for climate research. Cambridge University Press, Cambridge. https://doi.org/10.1017/9781107588783

Marrone S, Piantadosi G, Olivieri S, Sansone C (2019) Reproducibility of deep CNN for biomedical image processing across frameworks and architectures. European Signal Processing Conference 2019-Septe. https://doi.org/10.23919/EUSIPCO.2019.8902690

Meehl GA, Covey C, Delworth T, Latif M, McAvaney B, Mitchell JF, Stouffer RJ, Taylor KE (2007) The WCRP CMIP3 multimodel dataset: a new era in climatic change research. Bull Am Meteorol Soc 88(9):1383–1394. https://doi.org/10.1175/BAMS-88-9-1383

Michelangeli PA, Vrac M, Loukos H (2009) Probabilistic downscaling approaches: application to wind cumulative distribution functions. Geophys Res Lett 36(11):2–7. https://doi.org/10.1029/2009GL038401

Moss RH, Edmonds JA, Hibbard KA, Manning MR, Rose SK, Van Vuuren DP, Carter TR, Emori S, Kainuma M, Kram T, Meehl GA, Mitchell JF, Nakicenovic N, Riahi K, Smith SJ, Stouffer RJ, Thomson AM, Weyant JP, Wilbanks TJ (2010) The next generation of scenarios for climate change research and assessment. Nature 463(7282):747–756. https://doi.org/10.1038/nature08823

Nabat P, Somot S, Cassou C, Mallet M, Michou M, Bouniol D, Decharme B, Drugé T, Roehrig R, Saint-Martin D (2020) Modulation of radiative aerosols effects by atmospheric circulation over the Euro-Mediterranean region. Atmos Chem Phys 20(14):8315–8349. https://doi.org/10.5194/acp-20-8315-2020

Prein AF, Gobiet A, Truhetz H, Keuler K, Goergen K, Teichmann C, FoxMaule C, van Meijgaard E, Déqué M, Nikulin G, Vautard R, Colette A, Kjellström E, Jacob D (2016) Precipitation in the EURO-CORDEX 0.11 and 0.44 simulations: high resolution, high benefits? Clim Dyn 46(1–2):383–412. https://doi.org/10.1007/s00382-015-2589-y

Pryor SC, Barthelmie RJ (2014) Hybrid downscaling of wind climates over the eastern USA. Environ Res Lett. https://doi.org/10.1088/1748-9326/9/2/024013

Reichstein M, Camps-Valls G, Stevens B, Jung M, Denzler J, Carvalhais N, Prabhat (2019) Deep learning and process understanding for data-driven Earth system science. Nature 566(7743):195–204. https://doi.org/10.1038/s41586-019-0912-1

Ronneberger O, Fischer P, Brox T (2015) U-net: Convolutional networks for biomedical image segmentation. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 9351. pp 234–241. https://doi.org/10.1007/978-3-319-24574-4_28

Rumelhart DE, Hinton GE (1986) Learning representations by back-propagating errors. Cogn Model 2:3–6. https://doi.org/10.7551/mitpress/1888.003.0013

Sanchez-Gomez E, Somot S (2018) Impact of the internal variability on the cyclone tracks simulated by a regional climate model over the Med-CORDEX domain. Clim Dyn 51(3):1005–1021. https://doi.org/10.1007/s00382-016-3394-y

Sanchez-Gomez E, Somot S, Déqué M (2009) Ability of an ensemble of regional climate models to reproduce weather regimes over Europe-Atlantic during the period 1961–2000. Clim Dyn 33(5):723–736. https://doi.org/10.1007/s00382-008-0502-7

Schmith T (2008) Stationarity of regression relationships: Application to empirical downscaling. J Clim 21(17):4529–4537. https://doi.org/10.1175/2008JCLI1910.1

Sørland SL, Schär C, Lüthi D, Kjellström E (2018) Bias patterns and climate change signals in GCM-RCM model chains. Environ Res Lett. https://doi.org/10.1088/1748-9326/aacc77

Taylor KE, Stouffer RJ, Meehl GA (2012) An overview of CMIP5 and the experiment design. Bull Am Meteorol Soc 93(4):485–498. https://doi.org/10.1175/BAMS-D-11-00094.1

Torma C, Giorgi F, Coppola E (2015) Added value of regional climate modeling over areas characterized by complex terrain-precipitation over the Alps. J Geophys Res 120(9):3957–3972. https://doi.org/10.1002/2014JD022781

Turco M, Quintana-Seguí P, Llasat MC, Herrera S, Gutiérrez JM (2011) Testing MOS precipitation downscaling for ENSEM-BLES regional climate models over Spain. J Geophys Res Atmos 116(18):1–14. https://doi.org/10.1029/2011JD016166

Vandal T, Kodra E, Ganguly AR (2019) Intercomparison of machine learning methods for statistical downscaling: the case of daily and extreme precipitation. Theor Appl Climatol 137(1–2):557–570. https://doi.org/10.1007/s00704-018-2613-3

Vandal T, Kodra E, Ganguly S, Michaelis A, Nemani R, Ganguly AR (2017) DeepSD: generating high resolution climate change projections through single image super-resolution. Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining part F1296. pp 1663–1672. https://doi.org/10.1145/3097983.3098004

Vigaud N, Vrac M, Caballero Y (2013) Probabilistic downscaling of GCM scenarios over southern India. Int J Climatol 33(5):1248–1263. https://doi.org/10.1002/joc.3509

Villani C (2009) Optimal transport, vol 338. https://doi.org/10.1007/978-3-540-71050-9

Voldoire A, Sanchez-Gomez E, Salas y Mélia D, Decharme B, Cassou C, Sénési S, Valcke S, Beau I, Alias A, Chevallier M, Déqué M, Deshayes J, Douville H, Fernandez E, Madec G, Maisonnave E, Moine MP, Planton S, Saint-Martin D, Szopa S, Tyteca S, Alkama R, Belamari S, Braun A, Coquart L, Chauvin F (2013) The CNRM-CM5.1 global climate model: description and basic evaluation. Clim Dyn 40(9–10):2091–2121. https://doi.org/10.1007/s00382-011-1259-y

Vrac M, Drobinski P, Merlo A, Herrmann M, Lavaysse C, Li L, Somot S (2012) Dynamical and statistical downscaling of the French Mediterranean climate: uncertainty assessment. Nat Hazards Earth Syst Sci 12(9):2769–2784. https://doi.org/10.5194/nhess-12-2769-2012

Walton DB, Sun F, Hall A, Capps S (2015) A hybrid dynamical-statistical downscaling technique. Part I: development and validation of the technique. J Clim 28(12):4597–4617. https://doi.org/10.1175/JCLI-D-14-00196.1

Wang J, Liu Z, Foster I, Chang W, Kettimuthu R, Kotamarthi VR (2021) Fast and accurate learned multiresolution dynamical downscaling for precipitation. Geosci Model Dev 14(10):6355–6372. https://doi.org/10.5194/gmd-14-6355-2021

Wilby RL, Hassan H, Hanaki K (1998) Statistical downscaling of hydrometeorological variables using general circulation model output. J Hydrol 205(1–2):1–19. https://doi.org/10.1016/S0022-1694(97)00130-3

Yao W, Zeng Z, Lian C, Tang H (2018) Pixel-wise regression using U-Net and its application on pansharpening. Neurocomputing 312:364–371. https://doi.org/10.1016/j.neucom.2018.05.103