

Warning

This document is made available to the wider academic community.

However, it is subject to the author's copyright and therefore, proper citation protocols must be observed.

Any plagiarism or illicit reproduction of this work could result in criminal or civil proceedings.

Contact : portail-publi@ut-capitole.fr

Liens

Code la Propriété Intellectuelle – Articles L. 122-4 et L. 335-1 à L. 335-10

Loi n° 92-597 du 1^{er} juillet 1992, publiée au *Journal Officiel* du 2 juillet 1992

<http://www.cfcopies.com/V2/leg/leg-droi.php>

<http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm>

The University neither endorses nor condemns opinions expressed in this thesis.



THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Université Toulouse 1 Capitole (UT1 Capitole)*

Présentée et soutenue le *16/12/2022* par :

Antoine DOURY

**Robust estimation of regional climate change: construction
of an hybrid approach between deep neural networks and
climate models.**

JURY

MATHIEU VRAC
PATRICK GALLINARI
ERIKA COPPOLA
ERWAN LEPENNEC
JOSÉ MANUEL
GUTIÉRREZ
SEBASTIEN GADAT
SAMUEL SOMOT

Rapporteur
Rapporteur
Examinatrice
Examinateur
Examinateur

Directeur de Thèse
Co-Directeur de Thèse

École doctorale et spécialité :

MITT : Domaine Mathématiques : Mathématiques appliquées

Unité de Recherche :

Centre National de Recherche Météorologiques (UMR 3589)

Directeur(s) de Thèse :

Sebastien GADAT et Samuel SOMOT

Rapporteurs :

Mathieu VRAC et Patrick GALLINARI

Abstract

An essential challenge for the climate science community is to provide trustful information about the local impacts of global warming. Climate models are the main tool to study climate evolution according to human activities and greenhouse gas emissions scenarios. They are a numerical representation of the Earth System. Global climate models (GCM) produce worldwide simulations at too low resolution to correctly represent extreme meteorological events that strongly impact our societies. Today we use dedicated regional climate models (RCM) that transform a global low-resolution simulation into a high-resolution one over an area of interest. Nevertheless, the high resolution of those models implies a (much) higher cost that strongly limits the number of those climate simulations and, thus, the necessary exploration of the different sources of uncertainties.

The present work aims to propose a strategy to recreate, at low-cost, high-resolution simulations from low-resolution ones. The RCM-emulator introduced here aims to estimate the downscaling function included in a RCM using the recent development of neural networks. This study introduces the concept of RCM-emulator and presents a framework to build, train, and evaluate it. The main result of this study is that the RCM emulator is a credible approach to take up this challenge. Indeed it shows an excellent ability to create realistic high-resolution temperature and precipitation fields, consistent with the low-resolution simulation it downscales. We also study the applicability of the tool the various low-resolution simulations. Moreover, this work also highlights the decisive advantage of using RCM simulations to *learn* this relationship as it allows to explore future climates and poorly known regions.

The conclusions of this study open the door to further development and various promising applications. Indeed, the RCM-Emulator makes possible the production of robust messages about the local impacts of climate change. Moreover, another significant result of this work is that the emulator performance relies strongly on the calibration set. It is then essential to design the best simulation set to have the most robust emulator implying maybe to revisit the way of choosing which simulation to make with a RCM.

Résumé

Un des grands défis pour la communauté scientifique du climat est de produire une information fiable concernant les impacts locaux du changement climatique global. Les modèles de climat sont les principaux outils pour étudier son évolution en fonction de scénarios liés aux activités humaines et émissions de gaz à effet de serre. Les modèles de climat globaux (GCM) produisent des simulations couvrant l'ensemble du globe à des résolutions trop faibles pour représenter correctement certains événements météorologiques extrêmes qui impactent fortement nos sociétés. Aujourd'hui, nous utilisons des modèles de climat régionaux (RCM) pour transformer une simulation globale à basse résolution en simulation à haute résolution sur une zone géographique d'intérêt. Cependant, la haute résolution de ces modèles implique des coûts de calcul très élevés limitant le nombre de simulations réalisables et donc l'exploration nécessaire des différentes sources d'incertitudes.

Le présent travail a pour objectif de proposer une stratégie pour créer à moindre coût, des simulations haute résolution à partir d'autres à basse résolution. Le RCM-émulateur, introduit ici, cherche à estimer la fonction de descente d'échelle incluse dans un RCM en utilisant notamment les récents développements concernant les réseaux de neurones. Cette étude introduit le concept de l'émulateur et présente un cadre pour le construire, l'entraîner et l'évaluer. Le résultat principal de cette étude est que l'émulateur est une approche crédible pour relever ce défi. En effet, il montre une excellente capacité à créer des champs de température et précipitation à haute résolution cohérent avec la simulation basse résolution d'origine. Nous étudions aussi son applicabilité à différentes simulations basse résolution. Ce travail met aussi en évidence l'avantage décisif d'utiliser des simulations RCM pour *apprendre* cette relation puisque cela permet d'explorer des climats futurs et des régions mal observées.

Les résultats de cette étude ouvrent la porte à de nouveaux développements mais aussi à diverses applications prometteuses. En effet, le RCM-émulateur rend possible la production de messages robustes concernant les impacts locaux du changement climatique. De plus, un autre résultat important de ce travail est l'importance du jeu de données d'apprentissage sur les performances de l'émulateur. Il est essentiel de définir le meilleur ensemble de simulation pour construire un émulateur robuste ce qui implique peut-être de revoir la manière de choisir quelle simulation faire avec un RCM.

Acknowledgements

I would like to apologize any non-french reader as the acknowledgements will be in french, it was complicated enough like that. . . Je n'aurai jamais pensé que cette page eut été la plus difficile à écrire de cette thèse. En fait si je dois retenir une chose de ces années, c'est certainement la richesse des rencontres qui ont toutes contribué à faire de moi qui je suis et donc de cette thèse ce qu'elle est. Ces remerciements sont un exercice difficile car à l'opposé du reste de ce manuscrit, alors on va essayer de faire les choses bien et dans l'ordre.

Je souhaiterais tout d'abord remercier mes directeurs de these. Un grand merci a Samuel, qui il y a 4 ans maintenant a accepté de travailler avec un presque economiste presque statisticien (moi-même du coup) pour faire du machine learning avec des modèles de climat. Le pari était risqué et je l'en remercie le plus sincèrement du monde de l'avoir pris. Pendant ces 4 années il a eu la patience de me former aux enjeux de la modélisation climatique, de me transmettre un bout de son experience sur les modèles regionaux et la curiosité de se confronter aux obscures réseaux de neurones. Je le remercie aussi d'avoir pris le temps comprendre ce qu'il se passait dans ma tête (ou du moins de faire semblant) et de m'avoir fait confiance pour qu'au final on arrive à quelque chose de pas trop mal. Enfin merci d'avoir su gerer les différentes sollicitations parce que je n'aurais pas su le faire et cette thèse n'aurait jamais fini.

Merci aussi à Sébastien qui a accepté de diriger cette thèse alors que le sujet était encore plus loin de son champ d'expertise. Au delà de son soutien et ses conseils au combien importants, il se sera attaché durant ces 3 années à me transmettre la rigueur mathématique à laquelle l' impatient que je suis aura eu du mal à se plier.

Dans l'ordre de ceux qui m'ont mis sur ce chemin périlleux de la thèse et de la recherche climatique je remercie Bastien qui m'a montré qu'après la TSE une autre vie était possible. Je tiens à remercier tres chaleureusement Hervé qui m'a ouvert les portes du CNRM au printemps 2018 et qui a poussé pour faire de moi le premier alternant du labo. Vient ensuite Aurélien, qui après 9 mois de collaboration que je trouvais interessante a décidé d'aller encadrer et former ailleurs. Je le remercie, quand même, d'avoir arde un œil sur nos avancées et du temps pour des discussions et conseils toujours très utiles.

Je tiens à remercier tout paticulièrement Antoinette, Lola, Cécile, Pierre, Florence qui ont contribué à faciliter cette thèse en étant toujours disponibles et d'une aide précieuse face à mes récurrentes (et souvent redondantes) questions techniques. J'en profite ici pour remercier tou.te.s les membres de la MOSCA-team pour leur

accueil et leur bonne humeur permanente.

Plus généralement je tiens à remercier l'ensemble des collègues grâce à qui il est plutôt agréable de venir au labo. Ici je pourrai faire une longue liste de noms en espérant n'oublier personne mais je vais en fait essayer d'être efficace parce que c'est suffisamment long comme ça et dire en une phrase (longue certes) un grand merci à tout GMGEC pour ces moments partagés à la pause café (j'en profite ici pour adresser un merci particulier à tou.te.s les faiseuses et faiseurs de café, Christine et Florence en tête), autour d'un délicieux repas à Eurest ou encore au bar à partager de la boisson alcoolisé (et là je remercierai spécialement les plus fidèles car ils ont largement contribué au bon déroulé de cette thèse: Robin, Erwan, Raphael, Nicolas, Maya, William, Romain, Nils, Chloé, Gildas, Thomas...).

Enfin, last but not least, je remercie tous mes proches qui ont été un soutien indispensable à cette thèse. Mes parents qui m'ont toujours permis d'avancer dans les meilleures conditions et toujours soutenus dans mes différents projets. Mes frères qui ont bien de la chance de m'avoir comme frère. Mes amis, qui ne liront probablement jamais ces lignes et ils ont bien raison, mais heureusement qu'ils étaient là pour me sortir la tête du guidon. Et je terminerai par celle que cette thèse a le plus impacté car elle m'a supporté au quotidien pendant 3 ans, et je suis bien placé pour savoir que (notamment ces derniers mois) c'était pas facile, un très grand merci à Anja d'être toujours et pour longtemps encore inshallah à mes cotés.

Contents

| | | |
|----------|--|-----------|
| 1 | Introduction | 10 |
| 1.1 | What is Climate? | 10 |
| 1.1.1 | Definition | 10 |
| 1.1.2 | Climate System | 11 |
| 1.1.3 | Natural variability and climate change | 12 |
| 1.2 | Climate modelling | 14 |
| 1.2.1 | What is a climate model? | 14 |
| 1.2.2 | Climate simulations | 15 |
| 1.2.3 | Different sources of uncertainties in climate projections. | 16 |
| 1.3 | Regional climate change information | 20 |
| 1.3.1 | Dynamical downscaling | 20 |
| 1.3.2 | Statistical Downscaling | 30 |
| 1.4 | Objectives and scientific questions of this study. | 35 |
| 2 | Why deep neural networks are a credible strategy to build the RCM-emulator ? | 38 |
| 2.1 | A neural network | 38 |
| 2.2 | Training the network: the backpropagation algorithm | 40 |
| 2.3 | The last milestones: the ReLu function and the computational resources | 44 |
| 2.4 | Convolutional neural network (CNN) | 44 |
| 2.5 | Other great families of architecture | 48 |
| 2.6 | Deep Learning and climate applications | 49 |
| 2.7 | Link with this work | 52 |
| 3 | Regional Climate Model emulator based on deep learning: concept and first evaluation of a novel hybrid downscaling approach | 53 |
| 3.1 | Introduction | 54 |
| 3.2 | Methodology | 56 |
| 3.2.1 | Models and simulations | 56 |
| 3.2.2 | Predictors | 57 |
| 3.2.3 | Predictands | 59 |
| 3.2.4 | Deep learning with UNet | 60 |
| 3.2.5 | Training of the emulator : Perfect Model Framework | 63 |
| 3.2.6 | Evaluation metrics | 64 |
| 3.2.7 | Benchmark | 65 |
| 3.3 | Results | 66 |
| 3.3.1 | Computational efficiency | 66 |
| 3.3.2 | Evaluation step 1 : Perfect model world | 66 |

| | | |
|----------|--|------------|
| 3.3.3 | Evaluation step 2 : GCM world | 73 |
| 3.4 | Discussion | 78 |
| 3.4.1 | On the inconsistencies between GCM and RCM | 78 |
| 3.4.2 | On the stationary assumption | 79 |
| 3.4.3 | On the selection of the predictors | 80 |
| 3.4.4 | On the non reproducibility of neural network training | 81 |
| 3.5 | Conclusion | 83 |
| 3.6 | Appendices | 85 |
| 3.6.1 | Emulators inter-comparison | 85 |
| 3.6.2 | Description of the UNet-emulator architecture | 89 |
| 3.6.3 | Supplementary Material: Summary tables | 89 |
| 4 | Study of the RCM-Emulator applicability to GCM ensembles | 92 |
| 4.1 | Introduction | 93 |
| 4.2 | Methodology | 95 |
| 4.2.1 | The ALADIN63 matrix | 95 |
| 4.2.2 | Recall on the main emulator and Experiment design | 96 |
| 4.2.3 | Justification of the perfect model framework | 98 |
| 4.2.4 | Metrics, Benchmark and results presentation | 99 |
| 4.3 | Perfect Model evaluation | 101 |
| 4.3.1 | Main emulator | 101 |
| 4.3.2 | Cross-Validation on other training sets | 111 |
| 4.4 | Application on GCM data | 117 |
| 4.5 | Discussion | 120 |
| 4.5.1 | On the standardization method | 120 |
| 4.5.2 | Some tests on inputs choices | 121 |
| 4.6 | Conclusion | 122 |
| 5 | Precipitation downscaling with the RCM-Emulator : The role of the loss function | 124 |
| 5.1 | Introduction | 125 |
| 5.2 | Methodology | 127 |
| 5.2.1 | Datas : the ALADIN63's matrix | 128 |
| 5.2.2 | RCM-Emulator conception : Predictors, architecture and target. | 128 |
| 5.2.3 | Loss function for the neural network training | 130 |
| 5.2.4 | Evaluation Metrics | 133 |
| 5.3 | Perfect model Evaluation | 138 |
| 5.3.1 | Comparison of the three emulators | 138 |
| 5.3.2 | Deeper evaluation of Emul-ASYM | 146 |
| 5.4 | GCM data application | 155 |
| 5.4.1 | Illustration of the daily GCM/RCM differences | 155 |
| 5.4.2 | Present climate analysis | 157 |
| 5.4.3 | Climate change analysis | 157 |
| 5.5 | Discussion | 160 |
| 5.5.1 | On the asymmetric loss | 160 |
| 5.5.2 | On the computational efficiency | 162 |
| 5.6 | Conclusion | 163 |

| | | |
|----------|--|------------|
| 6 | Conclusion & Perspectives | 166 |
| 6.1 | RCM-Emulator concept | 167 |
| 6.2 | Efficiency | 168 |
| 6.3 | Trustability | 169 |
| 6.4 | Applicability step 1: different sources of input | 171 |
| 6.5 | Applicability step 2: GCM world | 172 |
| 6.6 | More general perspectives | 173 |

Acronyms

| | |
|---------------|--|
| IPCC | Intergovernmental Panel on Climate Change |
| CMIP | Coupled Model Intercomparison Project, different phases exist, the last is CMIP6 |
| AR | Assessment Report, they are the IPCC reports 6 have published by now. |
| GCM | Global Climate (or Circulation) Model |
| RCM | Regional Climate Model |
| CPM | Convection Permitting Model |
| RCP | Representative Concentration Pathway (CMIP5 scenarios of greenhouse gas emissions) |
| SSP | Shared Socio-economic Pathway (CMIP6 scenarios of greenhouse gas emissions) |
| CORDEX | COordinated Regional climate Downscaling EXperiment |
| DD | Dynamical Downscaling |
| ESD | Empirical Statistical Downscaling |
| PP | Prefect Prognosis |
| MOS | Model Output Statistic |
| WG | Weather Generator |
| GHG | GreenHouse Gas |
| TAS | Surface Air Temperature |
| PR | Precipitations |
| UA,VA | Wind vector components |
| HUS | Specific Humidity (sometimes Q) |
| Z | Geopotential altitude |
| SST | Sea Surface Temperature |
| NN | Neural Network |
| CNN | Convolutional Neural Network |
| RNN | Recurent Neural Network |
| LSTM | Long Short-Term Memory |
| GAN | Generative Adversarial Network |

Chapter 1

Introduction

1.1 What is Climate?

1.1.1 Definition

Climate is what you expect, weather is what you get.
Andrew John Herbertson

Defining climate can be delicate. The best way to understand it is maybe to differentiate it from the weather. Climatology and meteorology are two different disciplines looking at the same object but at different time scales. The object is the Earth system and, more specifically, its atmosphere component. Meteorology uses the current knowledge of the atmospheric processes and the observation of the previous and actual atmospheric conditions to eventually predict the weather for the next hours and days. The system is by nature chaotic (Lorenz, 1963). The smallest differences in the *initial conditions*¹ can lead to a radically different evolution of the atmospheric circulation. The chaotic nature of the atmospheric system limits the weather prediction to a maximum of two weeks.

The climatology goes beyond these short time variations. Climatology is directly linked to statistics and probabilities. It aims to comprehensively describe the distribution of the possible weather that can be encountered in a given region and period of time. It does not look at the weather of a given day but at all weathers observed over decades in a region to understand the mean state of the local climate, its variability, its seasonal evolution or the extremes of climate events, for example. More practically, the climate, such as the weather, is described through variables and quantities like temperature, humidity, wind or cumulative precipitations. The typical period length to study the climate is 30 years (as defined by the World Meteorological Organization). It allows having enough samples of weather to obtain some robust statistical properties. The climate of a region depends on many factors like the latitude, surrounding topography, distance to the sea, presence of a lake or forest etc..

In this context, climate change is the transformation of the statistical distribution of the variables describing the climate. Proving it is establishing the statistical

¹i.e. the atmospheric conditions at a given time t marking the beginning of the experiment.

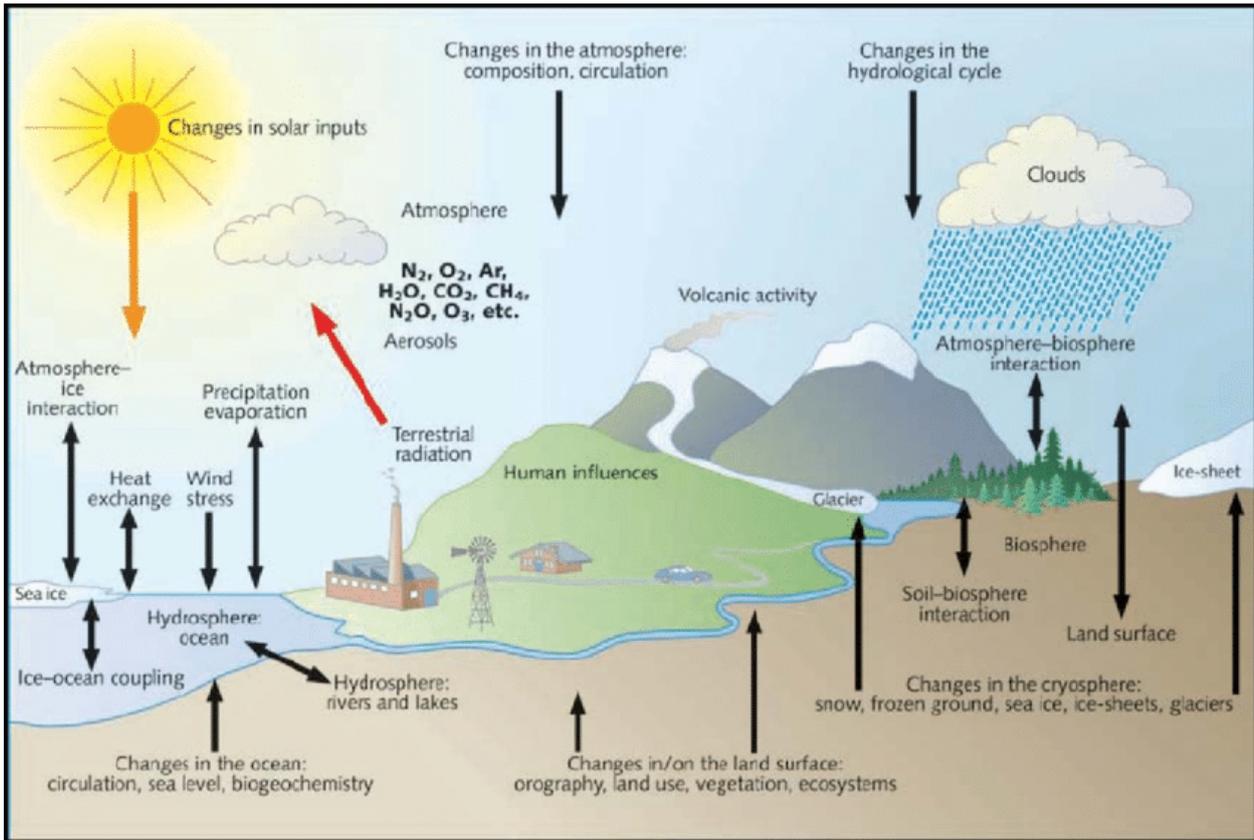


Figure 1.1: Climate system representation, extracted from IPCC 4th Assessment Report²(AR4 Solomon et al, 2007).

difference between two periods of time, taking into account the internal variability of the climate due to its chaotic nature.

1.1.2 Climate System

The climate system is the interactive system of 5 major components: the atmosphere, the hydrosphere, the cryosphere, the lithosphere and the biosphere. This whole system is directly dependent on external forcings³ like the sun, which is the most influential. It is the primary source of energy on Earth. Because it affects the planet unequally, it creates temperature gradients, putting the atmosphere and the ocean in movement to redistribute this heat. The poles are cooler because they receive less radiation but also because the ice reflects the radiation from the sun. This significant gradient between the poles and the tropics creates atmospheric circulation. The land surfaces (lithosphere) also play an essential role as they evaporate water differently, absorb or reflect solar radiation, or because of the topography, all these interactions impact atmospheric circulation.

²IPCC Assessment Reports are the well known reports about knowledge on climate change and its impacts. IPCC is an intergovernmental organisation to which thousands of people contribute. The volunteer authors of those reports review the thousands of scientific papers published each year to provide a comprehensive summary of what is known about the drivers of climate change, its impacts and future risks, and how adaptation and mitigation can reduce those risks.

³External agent that impacts the system while being outside.

The global ocean covers 71% of the Earth. It is the primary climate regulator through permanent interactions with the atmosphere. It receives, stocks and transports the sun's energy. The oceanic circulation transports heat from the tropical regions to the higher latitudes. The cold water from these regions gets denser and dives into the deep ocean. Those mechanisms combined with the surface wind are the start of the reasons behind ocean circulation. The time scale of these transfers varies from season or year for the tropical regions to centuries for the profound water changes. Oceans also play a significant role in the water cycle. Heated by the sun, the surface water evaporates continuously, is transported by atmospheric currents, condensates in clouds, and precipitates over land or ocean. Freshwater exchanges with the atmospheres, land surfaces or the cryosphere create a gradient of density in the ocean, also contributing to the ocean circulation. Because of its much slower circulation, the ocean generally impacts the atmospheric circulation at longer times scales.

The atmosphere contains a mix of gases. Among them, some *trace gases* interact with the sun's radiations. This is the greenhouse effect. The atmosphere directly reflects a third of the sun's radiations in space. The system absorbs the rest, which is then re-emitted by the atmosphere and the land surface. The greenhouse gases (GHG, e.g. carbon dioxide, methane, ozone or water vapour) absorb the infrared radiation from inside and outside the system and re-emit them in all directions. These radiations are then trapped in the system and warm the planet to a temperature adapted to support life.

1.1.3 Natural variability and climate change

Since the work of Lorenz (1963), it is common knowledge that the atmosphere is a chaotic system. It is sensitive to any change in the initial conditions making it the most unstable component of the climate system. The climate research community often refers to this chaotic aspect of the system as the natural or internal variability of the system in opposition to the external forcing.

Natural climate variability encompasses various processes driving the variability at different time scales. Williams et al (2017) retraces the different kinds of mechanisms behind internal atmospheric variability. It goes from the turbulence induced by the diurnal cycle or rapid air-sea interactions, which have immediate impacts on the atmosphere, to multiyear oscillations coupling atmosphere and ocean circulations such as the El-Nino phenomena. The critical point here is that naturally, the climate system greatly varies, following different modes impacting local climates differently and at different time scales. This chaotic nature has strong impact on the way to predict weather and to model climate requiring probabilist or ensembles-based approach.

On top of the atmospheric variability, some natural external forcing might strongly influence the system. Volcanoes' eruptions emit gases absorbing the sun's radiations and cooling the atmosphere. For example, The Mount Pinatubo eruption in 1991 caused a cooling of the global temperature of half a degree for the following months. The variations in the sun radiations received by the Earth due to the sun itself or

changes in the Earth's orbit also impact climate as they modify the quantity of energy received by the system.

Human activity also influences the climate system and plays the role of anthropogenic external forcing. Modifying land surfaces for agriculture or creating large urban areas has apparent consequences on the local climate but can also have repercussions at larger spatial scales. Since the industrial revolution, human activities have emitted a considerable amount of greenhouse gases. The concentration of carbon dioxide in the atmosphere increased by 150% compared to the levels of 1750, methane has more than doubled (+ 260 %) and the nitrous oxide increased by 123% (cf. Figure 1.2). The increase in greenhouse gas concentration modifies the composition of the atmosphere. It warms the planet as it reinforces the greenhouse effect⁴. The speed of the atmospheric changes induced by human activities is unprecedented in Earth's history and has alarmed the scientific community for decades. The international community decided to merge efforts to evaluate the impact of this GHG increase on our climate. The main tools used to do so are climate models.

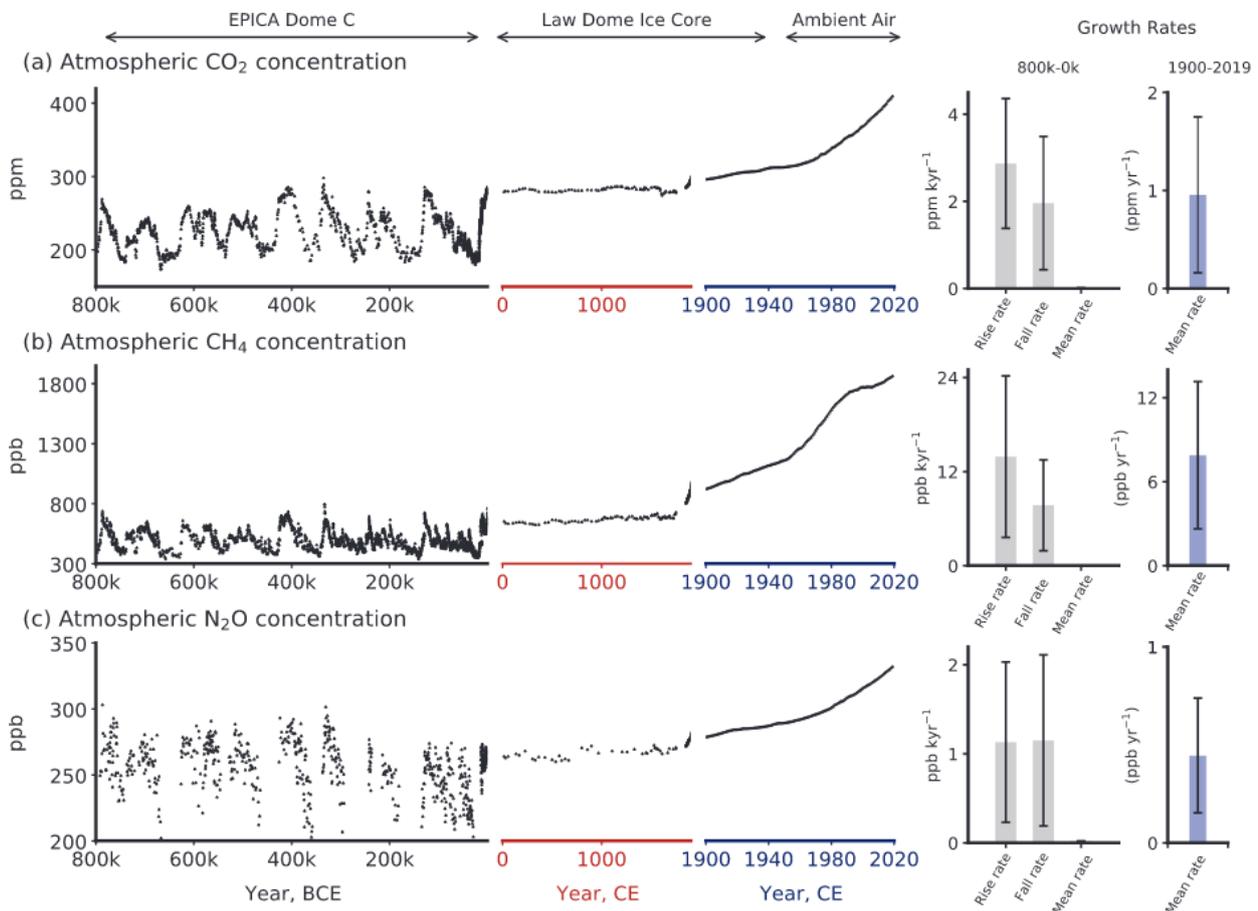


Figure 1.2: Evolution of the main greenhouse gas concentration over million year. IPCC AR6, Masson-Delmotte et al (2021), chap5. The reader should notice here that the x-axis is not linear.

⁴The warming process due to GHG emissions is more complex, in particular including positive feedback loops, but we keep it simple here as it is not the purpose of this manuscript.

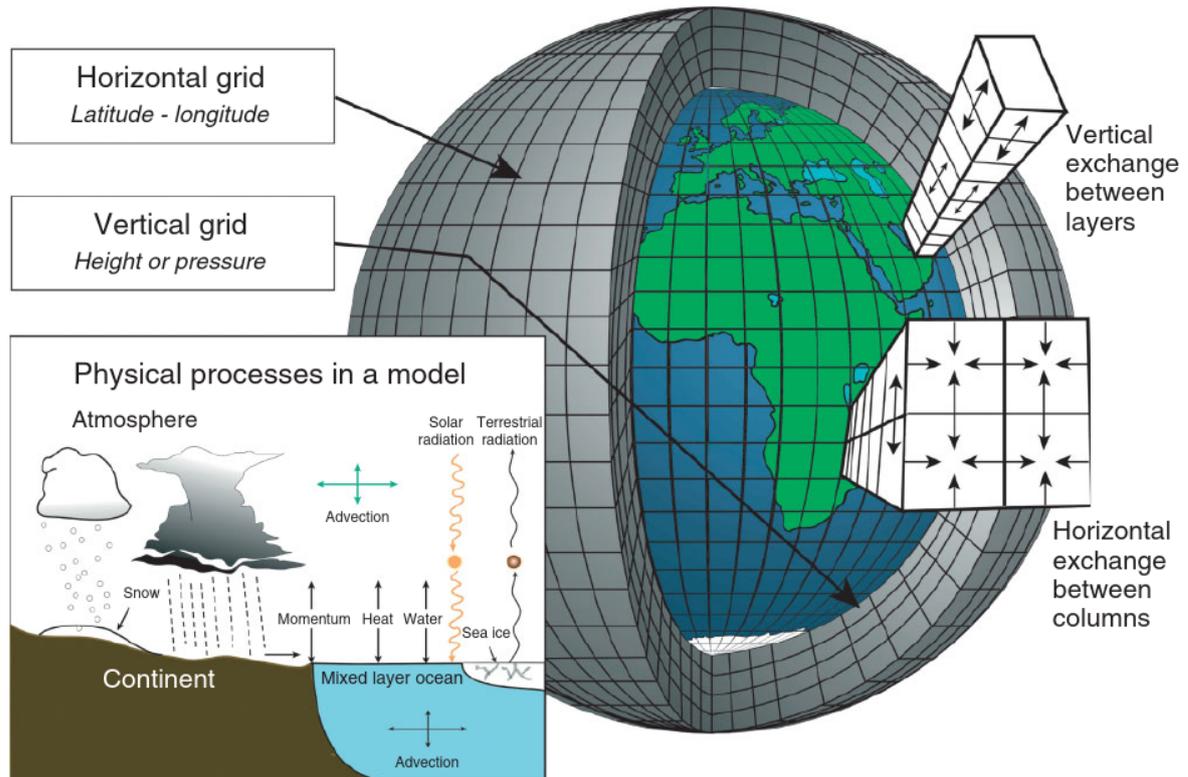


Figure 1.3: Scheme of a GCM.

1.2 Climate modelling

Climate models are numerical tools representing the climate system. The interest in such models is double. They allow a better understanding of the system by realising numerical experiments impossible to run in the real world. Furthermore, they permit the exploration of other climates by modifying some components to reconstruct past periods or to make projections about future climates.

1.2.1 What is a climate model?

There is a wide variety of climate models. Simple climate models represent part of the system at the global scale, like the system's energy balance. Such models are used to assess the global temperature according to the CO₂ concentration in the atmosphere, for example. In this work, I will focus only on more complex models that aim to represent the entire system. The starting idea of such models is to represent the time evolution of the atmosphere.

The atmospheric circulation is represented by the equations of fluid mechanic theory, respecting the physical laws of mass and energy conservation. The atmosphere is divided into small cubes (or meshes or grid points) in which the equations are solved from one time step to the other as well as interactions with the surrounding cubes. The grid is three-dimensional in order to represent both horizontal and vertical exchanges. At each time step and in each mesh, the model returns climatic variables like temperature, humidity or wind. A similar numerical representation of

oceanic circulation is done with a dedicated model. The atmospheric and oceanic models are then coupled to represent the interactions between the two components.

Following the same principle, land surfaces, continental hydrology cycle and sea ice exchanges with the atmosphere and ocean are represented by dedicated models and coupled with the atmosphere and ocean components. These are the main components of the so-called General Circulation Models or Global Climate Models (GCMs). Besides, the model receives external forcing, like the truth climate system. The sun radiations entering the system or the greenhouse gas concentrations or emissions resulting from anthropogenic activities are given as input to the model.

The complexity of GCMs is constantly increasing to represent better each physical or chemical process playing in each system component. The most recent and complex models are called Earth System Models. They target to represent the entire carbon cycle. For example, they represent the life of aerosols (suspended particles in the atmosphere): their formation, transportation by wind, and interaction with radiations, clouds or precipitation. The representation of land and biosphere is also more accurate: different types of vegetation are represented as they play a significant role in the system through photosynthesis or evapotranspiration processes.

The cube size is a determinant parameter in a climate model. As mentioned earlier, some climate processes can be highly local such as the formation of clouds or the occurrence of rain. Therefore, the smaller the cube, the finer is the representation of the system since the fluids mechanic equations are more precisely resolved. However, small cubes imply more cubes, so more equations to solve, more variables to estimate and save, and more computational resources. The model resolution (the grid point size) results from a compromise between the precision of the model, its complexity⁵ and the available computing resources. GCMs have mostly a horizontal resolution of about one to two hundred kilometres. It implies that some mechanisms cannot be simulated. *Empirical* or *physical parametrisations* are set to represent the effects of not represented processes on the model variable at the meshes scale. The resolution of the model grid and the choices of those parameters explain a large part of the differences between climate models.

1.2.2 Climate simulations

Climate models can be used for different purposes. If we put aside a better understanding of the processes, there are two main applications of climate models: exploring the past or the future. First, it is used to create climate *reanalysis*. They reconstruct the recent climate history by combining climate models and all available observations (in-situ records and satellite). They the best estimate of the true atmospheric circulation of the last decades. They are widely used in the scientific community since they are continuous and homogeneous in space and time and match local observations. They are notably used to evaluate climate models, especially for non-observed variables. In a further extent, Global Climate Models are used to reconstruct past climates for which no records exist. This is paleoclimatology,

⁵i.e. the number of climate system components that are represented.

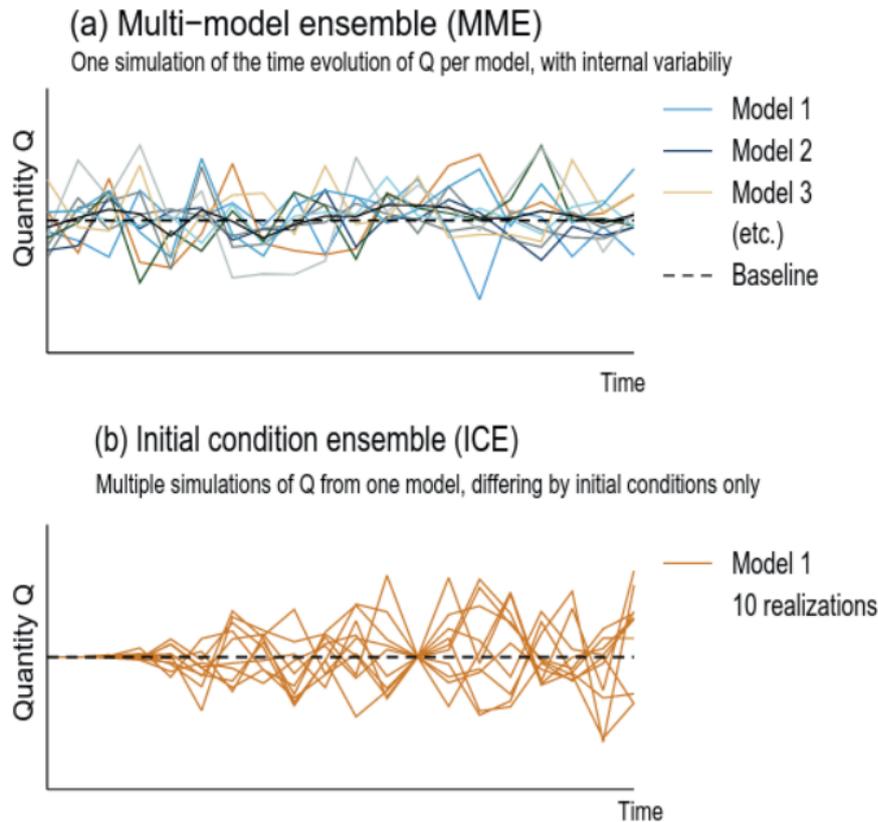


Figure 1.4: Illustration of two types of models ensembles exploring two different sources of uncertainty.

combining climate models and the study of geological sediments. Studying ancient climates is necessary to understand the situation where we are now.

Second, climate models are widely used to create future climate projections. The idea is to study the effects of human activities and related evolving emissions on the system. Such simulations generally start at the beginning of the industrial revolution (1850). They are pushed to a few centuries after today, even if the most studied period is the end of the current century. The study of climate change compares a future climate with either pre-industrial or more recent climates (generally 1981-2010). Nevertheless, there is considerable uncertainty from multiple sources associated with such simulations.

1.2.3 Different sources of uncertainties in climate projections.

Climate models are far from being perfect. Even in the most complex model, it is impossible to represent all processes, bringing some uncertainty to the accuracy of the representation. We also mentioned above the role of resolution or parameters choices in the differences between models. The answer to this first source of uncertainty is the multiplication of the number of models. Many global climate models are developed in many research centres worldwide. Each of them is wrong in its way. Since the mid-'90s, the scientific community decided to merge the efforts and build

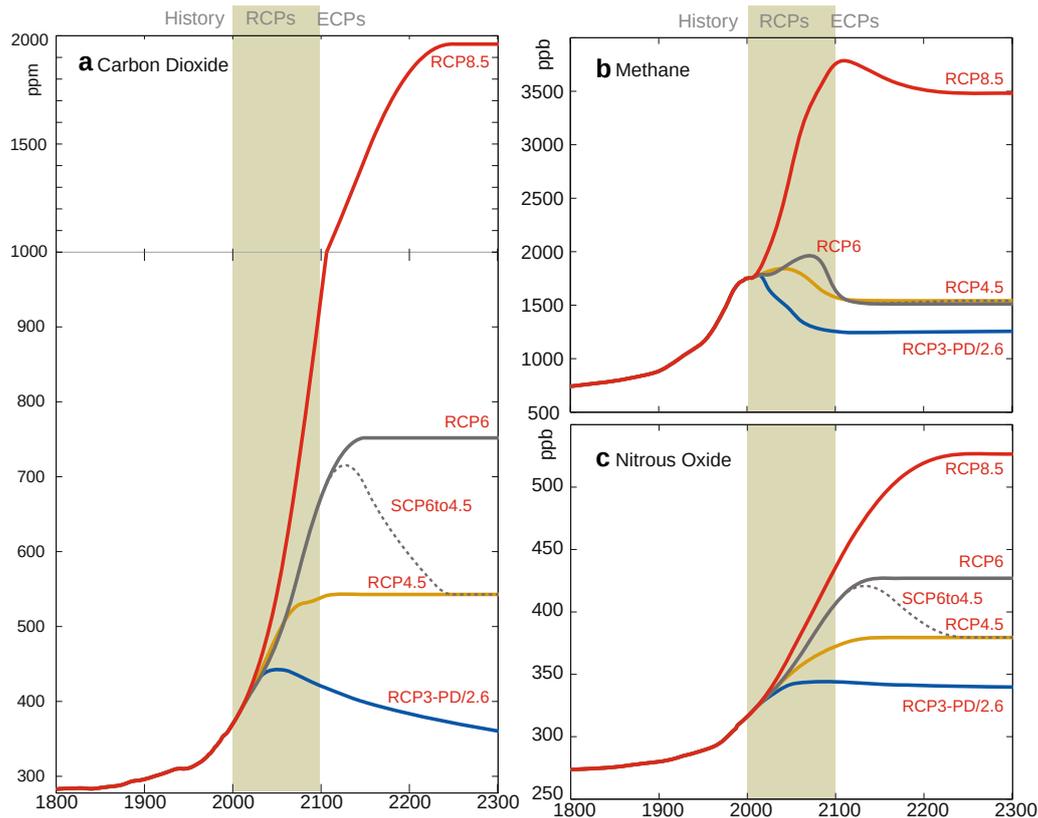


Figure 1.5: Time evolution of the main greenhouse gas concentration in the 4 RCP scenarios. From AR5, Tignor (2013).

great models intercomparison exercises called CMIP (Coupled Model Intercomparison Project). The aim is to build large ensembles of climate simulations from the existing models. Comparing models built by different teams who made different choices brings robustness. It helps to identify the weaknesses and the strengths of each model. Moreover, if all models agree on a given message, it gets more trustable. Figure 1.4 illustrates the time series of a multi-model ensemble. In Figures 1.6 and 1.7, commented later on, the straight line shows the multi-model mean and the shaded area represents a measure of the model ensemble spread.

The second source of uncertainty is brought by the internal (or natural) variability of the climate and so of the models. Indeed, climate models aim to reproduce the chaotic dynamic of the climate system, so the simulations are highly dependent on the initial conditions fixed at the beginning of the simulation. In order to take into account the possible impact of the internal variability, the same climate models are run multiple times with the same configuration and external forcing but different initial conditions. These multiple simulations are members of the model ensemble. To a larger extent, SMILEs (Single Model Initial-condition Large Ensemble) are vast ensembles of climate model members (more than 30). The number of studies using such ensembles has primarily increased over the last decade, and they help to quantify the model's internal variability. Maher et al (2021) proposes an overview of the application of SMILEs.

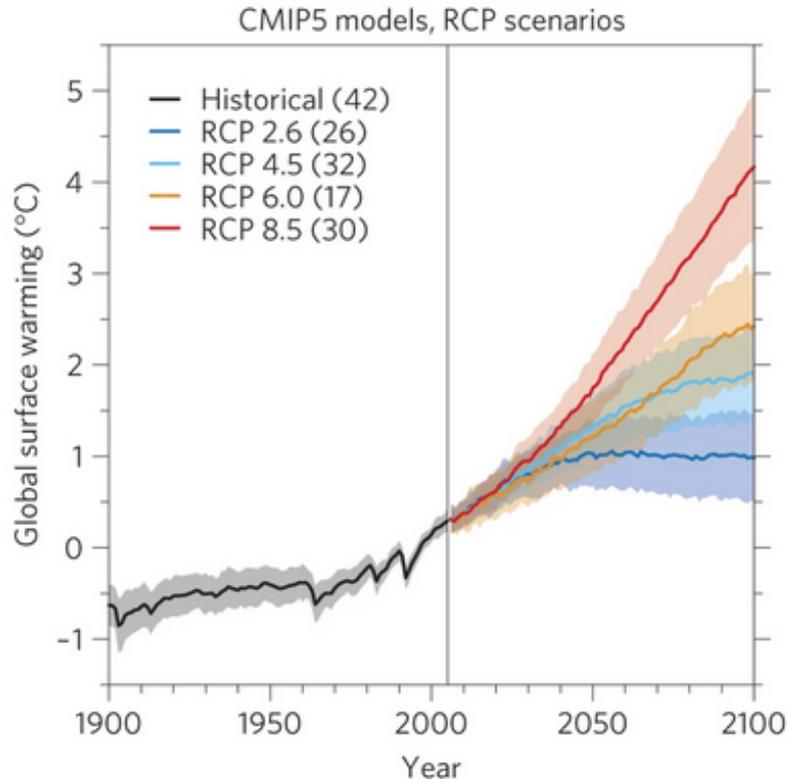


Figure 1.6: Global surface temperature evolution in the CMIP5 ensemble. From Knutti and Sedláček (2013).

The last source of uncertainty concerns the future evolution of human activities. In order to answer this uncertainty, the climate community established different scenarios of greenhouse gas emissions. The scenarios are determined according to key drivers such as demography, economy trajectories etc. Following these scenarios, tables of greenhouse gas concentrations are used to drive the climate models.

CMIP exercises comport several members of several GCMs driven by different scenarios of human emissions. CMIP5⁶ is the fifth generation of global climate models ensemble used for the fifth report of the IPCC published in 2014, Tignor (2013). CMIP5 regroups about 50 climate models, resulting into about 1000 long-term simulations (i.e. at least 20th and 21st centuries). Four main scenarios were chosen to drive CMIP5 models after 2006, while the historical part is driven by a reconstruction of historical greenhouse gas emissions. The RCP2.6, RCP4.5, RCP6.0 and RCP8.5 scenarios have been introduced in Meinshausen et al (2011) and are named accordingly to the level of global radiative forcing induced by the GHG concentrations they reach in 2100. The four scenarios have similar trajectories until 2030 and then diverge, as shown in Figure 1.5. The RCP2.6 is the most “optimistic” one, as it peaks around 2030 and then projects a slow reduction of the CO₂ concentration while the methane concentration diminishes drastically to reach mid-20th century levels before 2100. The RCP8.5 corresponds to an exponential increase in GHG until the end of the 21st century. The two others are intermediate scenarios. Figure

⁶The models used in this work are from CMIP, which is why CMIP5 scenarios and figures are presented here.

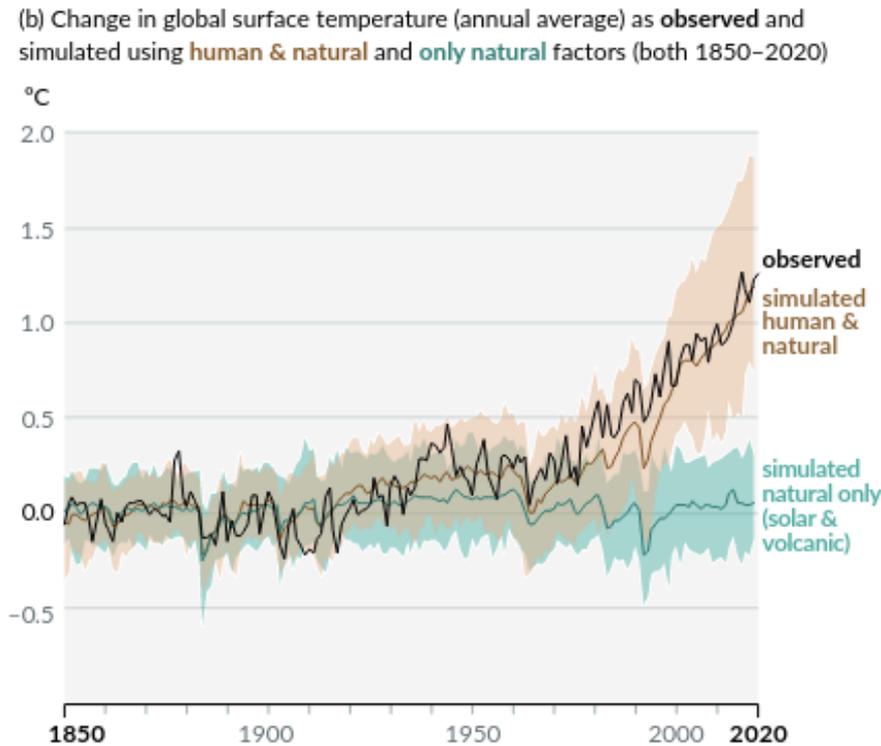


Figure 1.7: Illustration of the time evolution of the global surface temperature in CMIP6 model with and without anthropogenic greenhouse gas emissions. From IPCC AR6, Masson-Delmotte et al (2021).

1.6 shows the global surface warming projected for the different scenarios. The solid line shows the model’s mean, and the shaded area shows the model spread. CMIP6 is the most recent CMIP exercise and had been used in the sixth IPCC assessment report (AR6,2021/22, Masson-Delmotte et al, 2021). It regroups about 70 models and more than a thousand simulations. New scenarios have been defined under “Shared Socioeconomic Pathways” (SSPs).

Besides historical and scenario-based projections, CMIP exercises include other kinds of experiments. For example, one consists in running the model on the historical period (1850–2006 for CMIP5, -2014 for CMIP6), removing the GHG emissions due to human activities. Figure 1.7, extracted from the AR6 summary for policy-makers, shows the experiment’s results on the CMIP6 models. This figure is a good illustration of the use of large ensembles. Here all models agree that the observed global temperature warming is not due to the climate’s internal variability or natural forcings. The observed temperature time series fits in the range of simulations as a single realisation among the possible ones. The variety of models ensembles establishes the range of possibilities.

The repartition of the total uncertainty among the three sources we defined before is an essential topic with multiple studies trying to establish it. Lehner et al (2020), following the framework set in Hawkins and Sutton (2009), assessed this question using seven SMILEs models and both CMIP5 and six ensembles. Figure 1.8 shows the study’s results for the global scale temperature. Over the first decades, internal variability is the primary source of uncertainty. It is followed by the model

uncertainty and scenario uncertainty, which becomes the leading one from the mid-century. We can notably see that internal variability plays no role in the global temperature uncertainty after some decades.

1.3 Regional climate change information

*Various mechanisms operating at different time scales can modify the amplitude of the regional-scale response of temperature, and both the amplitude and sign of the response of precipitation, to human influence (high confidence).
IPCC AR6, Working Group I, (Masson-Delmotte et al, 2021), Chapter 10.*

If climate change is a global phenomenon, its impacts vary from one region to the other. Worldwide, the consequences of global warming are already impacting societies. In order to conduct studies on the impacts on agriculture or hydrological cycle, for example, it is crucial to produce climate change information at regional scales. The regional scale varies from a few thousand kilometres for sub-continental regions such as the Mediterranean basin to some kilometres representing cities, valleys or coastal climates. In the rest of this manuscript, we will also talk about local scales to refer to those small areas that share the same climate and where climate change information is needed to set up adaptation strategies.

Global climate models have a resolution of about 100 to 200 km. Several studies have demonstrated the difference between nominal resolution and effective resolution. The nominal resolution is the one we defined earlier; it corresponds to the actual size of the grid points. The effective resolution of the model is the resolution where the model is reliable. Klaver et al (2020) concludes that the effective resolution of a model is about 3 to 5 times its nominal resolution. This is too coarse to account for local drivers of regional climate correctly. Thus, GCMs provide, at best subcontinental information on climate change. For example, in the last CMIP exercises, models agree on more precipitation over Northern Europe in winter and a drier summer over southern Europe at the end of the century for the RCP4.5 scenario and above.

Nevertheless, impact studies and adaptation policies necessitate much higher resolution. The higher resolution brings a better representation of processes parametrised in GCMs, and a better representation of local drivers such as mountains, coastlines, towns, great lakes or islands. The solution brought by the community is to downscale the GCM simulations. Two great families of downscaling have appeared: dynamical and statistical downscaling.

1.3.1 Dynamical downscaling

Dynamical downscaling relies on another kind of climate model called Regional Climate Models (RCMs), which focus only on a regional domain. Regional climate modelling is quite a recent activity as it appeared about 30 years ago. Dickinson et al (1989) and Giorgi and Bates (1989) introduced the first RCM in the late 80s. Since

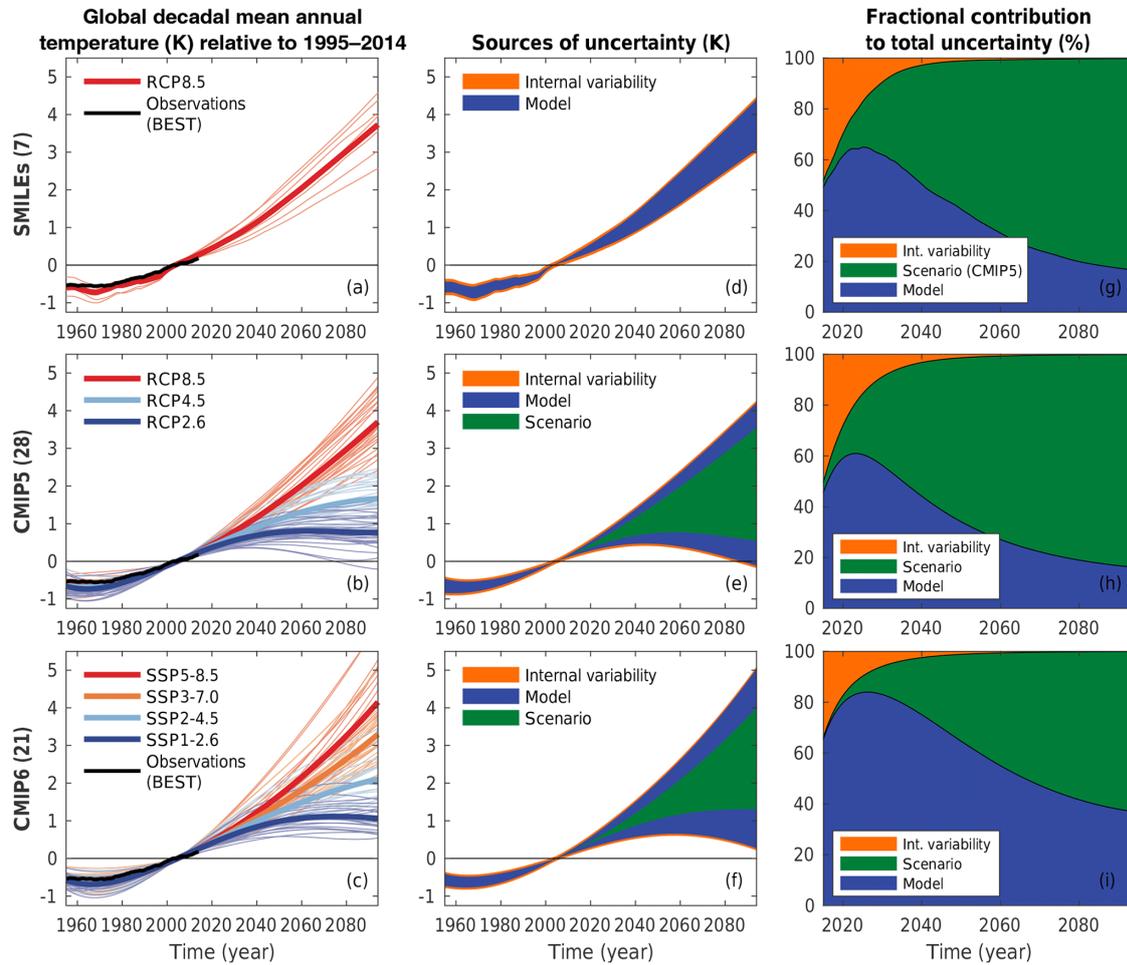


Figure 1.8: Figure and legend from Lehner et al (2020): (a–c) 10-year running means of global annual mean temperature time series from (a) SMILES, (b) CMIP5 and (c) CMIP6, with observations (Rohde et al., 2013) superimposed in black, all relative to 1995–2014. For SMILES, the ensemble mean of each model and the multimodel average of those ensemble means are shown; for CMIP the polynomial fit for each model and the multimodel average of those fits are shown. (d–f) Illustration of the sources of uncertainty in the multimodel multi-scenario mean projection. (g–i) Fractional contribution of individual sources to total uncertainty. Scenario uncertainty in SMILES in (g) is taken from CMIP5, since not all SMILES offer simulations with multiple scenarios. (d–i) In all cases, the respective multimodel mean estimate of internal variability (I_{mean}) is used.

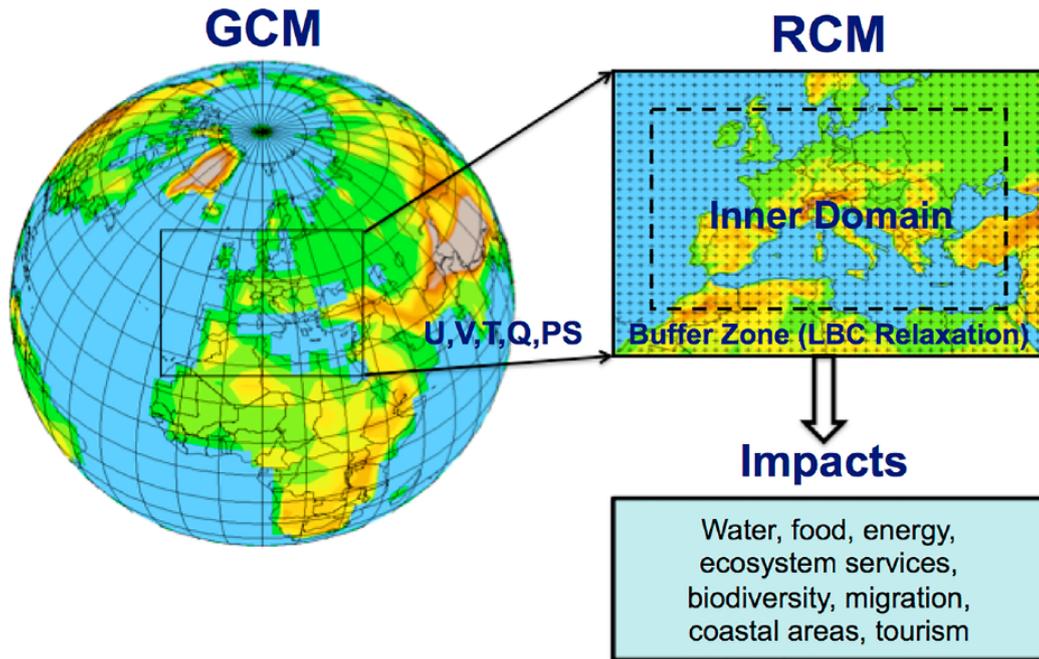


Figure 1.9: Scheme of dynamical downscaling process, from Giorgi (2019)

then, the number of studies including an RCM never stopped increasing. Hereafter, we review some key conceptual and technical points of RCMs.

Concept and technique

Regional climate models are numerical models based on the same concept as global models, except that they run on a limited area domain such as Europe (cf Figure 1.9)⁷. It allows them to run at higher resolution, between 10 and 50 km. The regional climate model is embedded or nested in a GCM simulation to ensure the transfer of information with the global scale. We say that the GCM drives or forces the RCM. Figure 1.9 illustrates the nesting scheme. RCMs generally have a temporal resolution of less than half an hour. At each time step, the RCM receives information about general circulation outside its domain from the GCM simulation thanks to a set of lateral boundary conditions (LBC). LBCs contain the main atmospheric variables: the vertical distribution of moisture, temperature, zonal and meridional wind components in the vertical, and surface pressure. They are extracted from the GCM driving simulation and transmitted to the RCM in the so-called buffer zone (cf Fig. 1.9) thanks to some relaxation technique (Davies and Turner, 1977). LBCs are generally stored every 6hrs in the GCM simulation and interpolated to match the RCM temporal and spatial resolution. The sea surface temperature (SST), the sea ice cover, the aerosols and greenhouse gas forcings from the driving GCM simulation are also given as input to the RCM.

RCMs have a higher resolution than the driving GCM. The first RCM was introduced to downscale a 500km resolution GCM over the Yucca mountains in the

⁷An other family of RCMs are global models with a stretched grid allowing a higher resolution over a specific domain. However, for simplicity we do not consider them here

United States of America in the late 90s. It had a horizontal resolution of 70km. Nowadays, thanks to a significant improvement in computational efficiency, most GCMs have a resolution of about 100 to 200km, with some reaching higher resolutions close the 50km, with computational costs still allowing to run numerous long-term simulations. RCMs simultaneously increased their resolution to 10 to 25km. The ratio between the GCM and RCM resolution is an important parameter to ensure the accuracy of the downscaling, and it should stay below 12 times (Denis et al, 2003). Moreover, RCMs include their land surface scheme at a higher resolution than the GCM to take better account of local land surface characteristics (reliefs, littoral, land-use).

The nesting technique of the RCM is a widely debated topic, and various methods have been developed. The one described above is called one-way nesting. The RCM receives at its boundary regular information from the GCM but does not feed back to it. Inside its domain, the RCM develops its internal physics and dynamics. However, at large scales⁸ the RCM should not deviate from the GCM simulation. It is designed to add realistic sub-GCM grid scale details but not to correct or modify the GCM global circulation. It implies using good GCMs in order to get good high-resolution RCM simulations. However, it depends strongly on the domain size. In a large domain, the boundary control is less strong, and the RCM is more susceptible to developing its own large scales (Laprise et al, 2008; Miguez-Macho et al, 2004).

A variant of the classical one-way nesting allowing controlling the eventual RCM deviation is called spectral nudging (Waldron et al, 1996; Von Storch et al, 2000). Such a method constrains the RCM stronger, ensuring a better consistency with the GCM large scale. It is the same as one-way nesting but also controls the large scales on top of the domain. Above a given altitude, the RCM atmosphere large scales are controlled to follow the GCM information for some key variables (e.g. temperature, moisture or wind components). The soundness of spectral nudging has been widely debated in the literature. Some studies argue that it limits the RCM internal processes too much, creating inconsistent dynamics or failing to represent some local extremes (Alexandru et al, 2009; Separovic et al, 2012; Rummukainen, 2010; Castro, 2005). Even if multiple studies have shown that with the appropriate settings, the spectral nudging technique increases the consistency with the driving GCM simulation without evident deterioration of the local processes (Sanchez-Gomez et al, 2009; Lucas-Picher et al, 2016; Colin et al, 2010), most RCM simulations are performed following the classical one-way nesting methodology. Two-way nesting simulations have also been performed with the RCM feeding back to the GCM simulation. Such experiments did not show a strong regional to global impact, except for specific regions like the poles (Giorgi, 2019) and the tropics (Lorenz and Jacob, 2005). Besides, their extremely high costs are a severe brake on their use.

Such as GCMs, the development of RCM goes in three directions: a better

⁸The *larges scales* are challenging to define. We consider them in this manuscript to match the effective resolution of the GCM, where the GCM is doing a good job. However, the closer we go to the surface, the more the resolution impacts the quality of the representation. So we consider the large scales in opposition to the local scales where the RCM aims to add value. When we refer to large scales in this manuscript, we mean the high altitude general circulation of the atmosphere represented by variables at low resolution and low-pressure levels.

model content (improved parametrisation), a better consideration of the different components of the climate system and an increase in resolution to represent better the most local processes. There has been a strong development of Atmosphere-Ocean coupled RCM with some including biogeochemistry representation in both land and ocean (called RCSM or RESM, Somot et al, 2008; Sevault et al, 2014; Sitz et al, 2017; Sein et al, 2015). Interactive models for the representation of aerosols have also been developed as these particles have a substantial impact on local climates and global warming (Nabat et al, 2014). On the other hand, the last decade has seen the apparition of Convection Permitting Models (CPMs, or CP-RCMs) (Prein et al, 2015; Ban et al, 2021; Lucas-Picher et al, 2021; Caillaud et al, 2021). They are regional climate models running at a resolution between 2 and 5km, and allowing the numerical solving of the deep convection, responsible for local storms. In order to respect a reasonable resolution ratio, CPMs are most of the time nested into RCMs simulations. It is then a two-step nesting.

RCMs ensembles

Regional Climate Models constitute a novel source of uncertainty. They are a modelling step more with different choices in the nesting method, parameters settings or the complexity of the model. Such as GCMs, the choice of the RCM is then a source of uncertainty. In order to take into account all sources of uncertainties, it is necessary to create RCMs ensembles or matrices, where several GCMs simulations drive several RCMs over the same regional domain and period (Déqué et al, 2012). Because of the GCM uncertainties, it is necessary to take GCM simulations driven by different scenarios and several members of the same Scenario-GCM pairs. It leads to 4 dimensional [Scenario, GCMs, Member, RCM] matrices. Moreover, internal variability plays a preponderant role at regional scales and regarding some local and extreme events. Ideally, hundreds to thousands of RCMs simulations would be needed in a given region to cover those uncertainties at the maximum and therefore provide a reliable local climate information. However, here comes the most significant limitation of dynamical downscaling. Even if they have a limited area domain, the high resolution implies much more grid points and much higher computational costs. For example, a global grid at 150km resolution includes about 30 thousand points. In comparison, a large European domain at 12km includes about 200 thousand of them, which must be multiplied by the altitude levels (between 50 to 100). Moreover, RCM have a higher temporal resolution which also increase their costs. The number of long-term RCM simulations is much more limited than the GCMs with the existing computational resources.

Facing this issue, the scientific community, especially in Europe, created ensemble projects similar in spirit to the CMIP exercises. PRUDENCE (Déqué et al, 2007) and ENSEMBLES (Déqué et al, 2012) projects successively contributed to the creation of the first European RCM-GCM matrices at 50km and 25km resolution. Similar exercises have been raised for the different continents leading to the CORDEX initiative (Giorgi and Gutowski, 2015). It aims to coordinate the Regional Downscaling activities worldwide to provide a coherent and reliable source of downscaled information. The different regional communities regroup around some well-defined domains. For each domain, the participants agree on a simulation protocol to coordinate the production of downscaled simulations. For example, GCMs

| | | ERA-I | GCM | | | | | | | | | | | |
|-----|--------------------|-------|-------------------------|--------------------------|------------------|----------------|----------------|-----------------|--------------|--------------|--|--|----|--------------|
| | | | EC-EARTH r12, r1, r3 | MPI-ESM-LR r1, r2, r3 | HadGEM2-ES R1 | CNRM-CM5 R1 | IPSL-CM5 R1 | NorESM1-M R1 | CanESM R1 | MIROC5 R1 | | | | |
| RCM | CCLM4-8-17 | | | | | | | b | | | | | | |
| | crCLIM-v1-1 | | 3m | 3m | | | | | | | | | | |
| | RACMO22E | | 3m | | | | | | | | | | | |
| | RCA4 | | 3m | 3m | | | b | | | | | | | |
| | HIRHAM5 | | 3m | | | | | | | | | | | |
| | REMO | | | 3m | | | | | | | | | | |
| | WRF361H | | | | | | | | | | | | | |
| | WRF381P | | | | | | | | | | | | | |
| | ALADIN53 | | | | | | b | | | | | | 3m | # of members |
| | ALADIN63 | | | | | | | | | | | | | RCP 8.5 |
| | RegCM4-6 | | | | | | | | | | | | | RCP 4.5 |
| | ALARO-0 | | | | | | b | | | | | | | RCP 2.6 |
| | HadREM3 -GA7-05 | | | | | | | | | | | | | HIST |

Figure 1.10: EURO-CORDEX matrix, October 2022.

are selected according to their performance and spread of future projections. Each participant picks some GCMs to downscale among this list. The idea is to explore at maximum among the available and realistic GCMs. A minimum number of scenarios to downscale or the allowed nesting technique are also defined in advance. The aim of such exercises is, for a given region, to coordinate the filling of RCM-GCM-Scenario matrices to explore the possible evolution of local climates.

The production of CORDEX simulations is done in three steps. First, all participating RCMs perform a perfect condition simulation with LBCs from reanalysis over a recent period (1980-present). This simulation, called evaluation simulation, is essential to evaluate the RCMs against observations. Second, the RCMs are used to downscale the historical (1950-present) simulations from the selected GCMs. It helps validate the RCM driven by GCM simulations by comparing the climatology and variability with the evaluation run. It also serves as a reference to assess the eventual changes induced by global warming, as the third steps consist of downscaling GCM projections under various scenarios over a future period running until 2100.

EURO-CORDEX is the most collaborative community and filled up the biggest matrix (cf Fig 1.10 Vautard et al, 2020; Coppola et al, 2021). It includes 13 RCMs, 8 GCMs from CMIP5 and 3 RCP scenarios (2.6, 4.5 and 8.5). After more than ten years of coordinated efforts on the potential 312 projection simulations, only a hundred have been performed. On this number, we should also account for the historical and evaluation simulations, which lead to a number of about 170 simulations. This is largely more than any other CORDEX domain, but still, it is insufficient. Indeed the matrix is highly unbalanced, and only a fifth of the existing CMIP5 GCMs are considered. Evin et al (2019) showed that an unbalanced or too weak matrix could lead to a wrong estimation of future changes.

Moreover, in these exercises, internal variability is rarely explored. In the EURO-

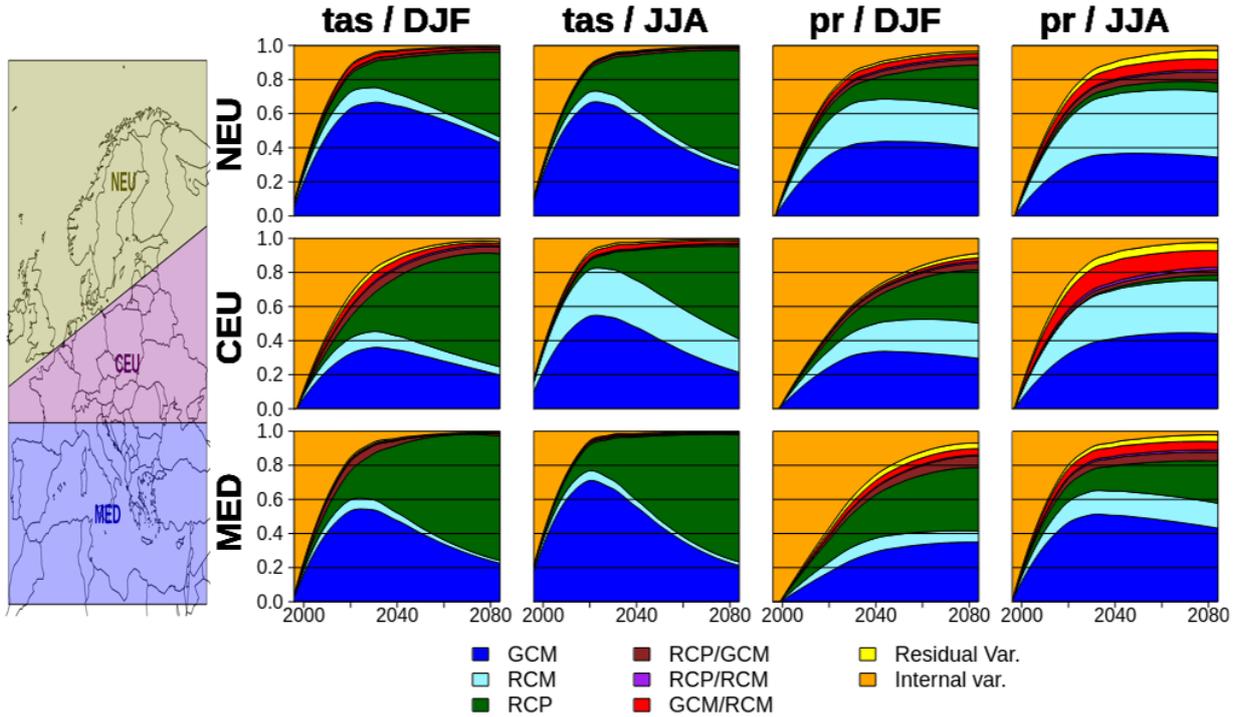


Figure 1.11: Sub-continental uncertainty decomposition in RCM ensemble, from Evin et al (2021).

CORDEX matrix, in most cases, only one GCM member is downscaled. Leduc et al (2019); von Trentini et al (2019) use a SMILE ensemble to assess the role of the internal variability in future projection. Both studies confirm that the internal variability is more substantial at local scales, and CORDEX ensembles should better explore this aspect. The constitution of high resolution (2 to 25 km) large ensembles accounting for all sources of uncertainties is one of the biggest challenges of the regional climate community (Giorgi, 2019; Rummukainen et al, 2015).

Various studies proposed statistical approaches to fill up Scenario-GCM-RCM matrices (Déqué et al, 2007, 2012; Christensen and Kjellström, 2022; Evin et al, 2019). These methods are based on variance decomposition (ANOVA) and allow us to estimate the proportion of the different sources of uncertainties. Among the others, Evin et al (2019, 2021) propose a Bayesian ANOVA model to deal with an unbalanced matrix. Figure 1.11 illustrates the variance decomposition results for temperature and precipitation in winter and summer over three large European regions. Compared with the fractional uncertainty decomposition from Figure 1.8, the RCM is added, as well as the interaction terms between GCM, RCM and RCP scenarios. The results are comparable with the GCM uncertainty analysis, the internal variability role is decreasing through time, conversely to the scenario uncertainty. Regarding specific variables/seasons/regions, the uncertainty brought by the RCM can be equivalent to the one from the GCM. The internal variability seems to be more critical when looking at smaller regions. It is confirmed in Evin et al (2019), which conducts the same analysis at very local scales, in 4 French mountain massifs of the Pyrenees and Alps. The results show that internal variability is the largest source of uncertainty for temperature until the second half of the century, while for

Orography of the EURO-CORDEX domain at typical GCM, RCM and CPM resolution

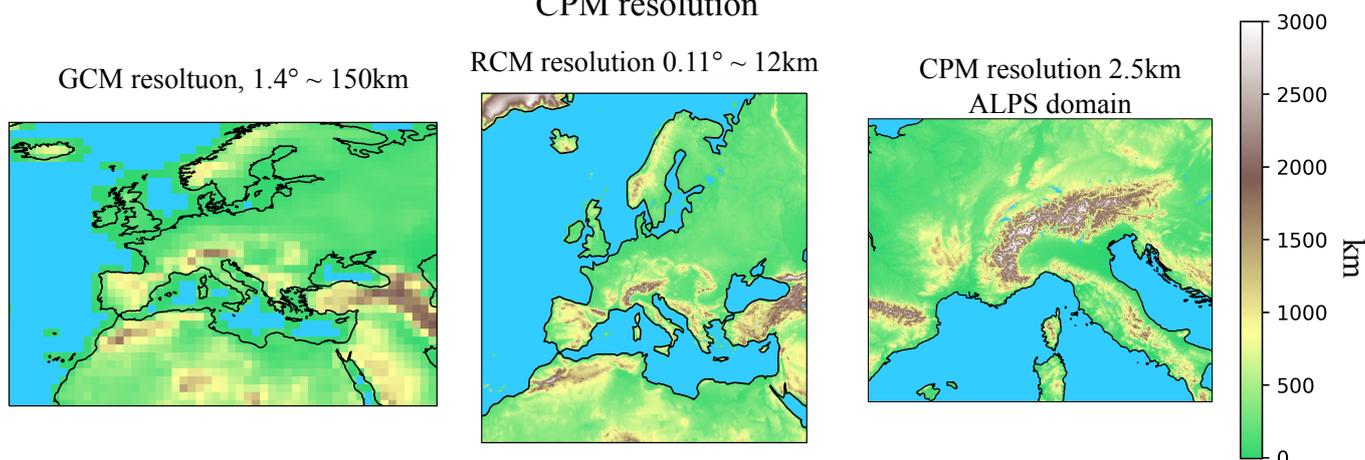


Figure 1.12: Illustration of typical resolutions for GCM, RCM and the very high resolution CPMs.

precipitation, it vastly overcomes all other sources all along the period.

A well-known example of RCM added value: Extreme precipitation over the Alps

The better representation of complex terrains such as reliefs, coastlines or islands due to the higher resolution implies a better simulation of the local climate as these are important drivers, and their interaction with the atmospheric circulation is better represented. It is the first source of added value that benefits all variables. Careto et al (2022) showed, for example, the added value of RCM over the Iberian Peninsula regarding temperature. Nevertheless, the substantial and most recognised added value of RCM concerns the representation spatial and temporal behaviour of precipitation.

Multiple studies have shown that RCMs with higher resolution represent better the extreme precipitations over the Alpine region. Torma et al (2015) and Fantini et al (2018) notably showed that the CORDEX simulations containing the Alps recreate well the entire distribution of precipitation, both studies based on different scores of distribution similarity (Kolmogorov-Smirnoff and Kullback-Leiber). Figure 1.13 (a) illustrates the better representation of the mean daily rainfall by RCM. It is not shown here, but Torma et al (2015) shows that the added value of RCM is still visible at the GCM scale when interpolating (upscaling) the high-resolution simulation on the GCM grid. The second part of the figure shows that this better representation can modify the local climate change response signal. Indeed Giorgi et al (2016) showed that the RCM ensemble agreed on an increase of precipitation over the top of the Alps while the GCM ensemble simulated a strong uniform decrease of prediction. The authors found the RCMs projection more accurate, as they match with recent trends, and the underlying process checked with other variables is credible. The evaluation of climate models and RCMs in a climate change context cannot be empirical. Here is the strength of RCMs, the complete analysis of underlying physical processes is possible, and the credibility of the scenario can

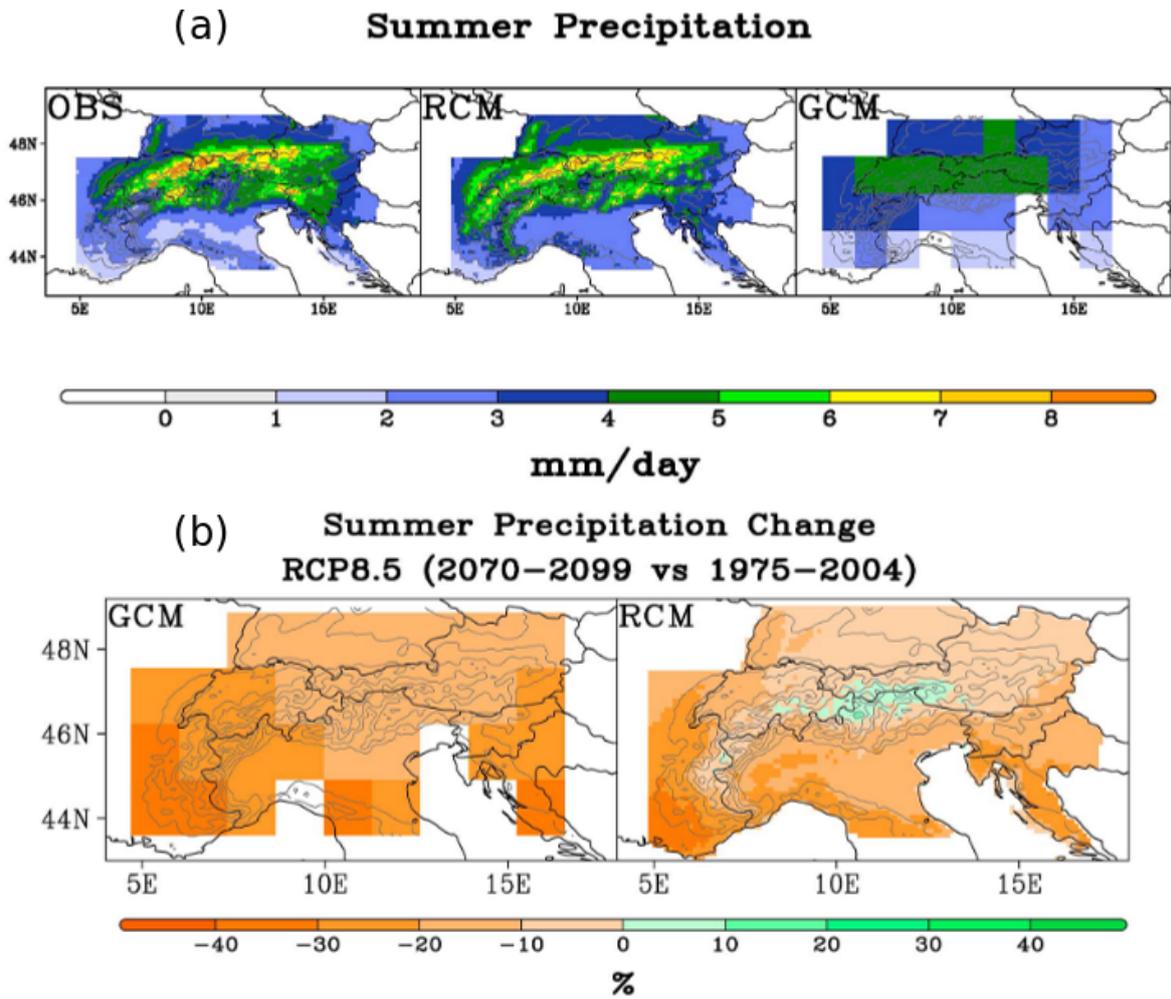


Figure 1.13: (a) Summer mean precipitation over the Alps, comparison of RCM and GCM ensembles with versus the observed mean, from Torma et al (2015); Giorgi (2019). (b) Projection of summer precipitation changes in GCM and RCM ensembles, extracted from Giorgi et al (2016); Giorgi (2019).

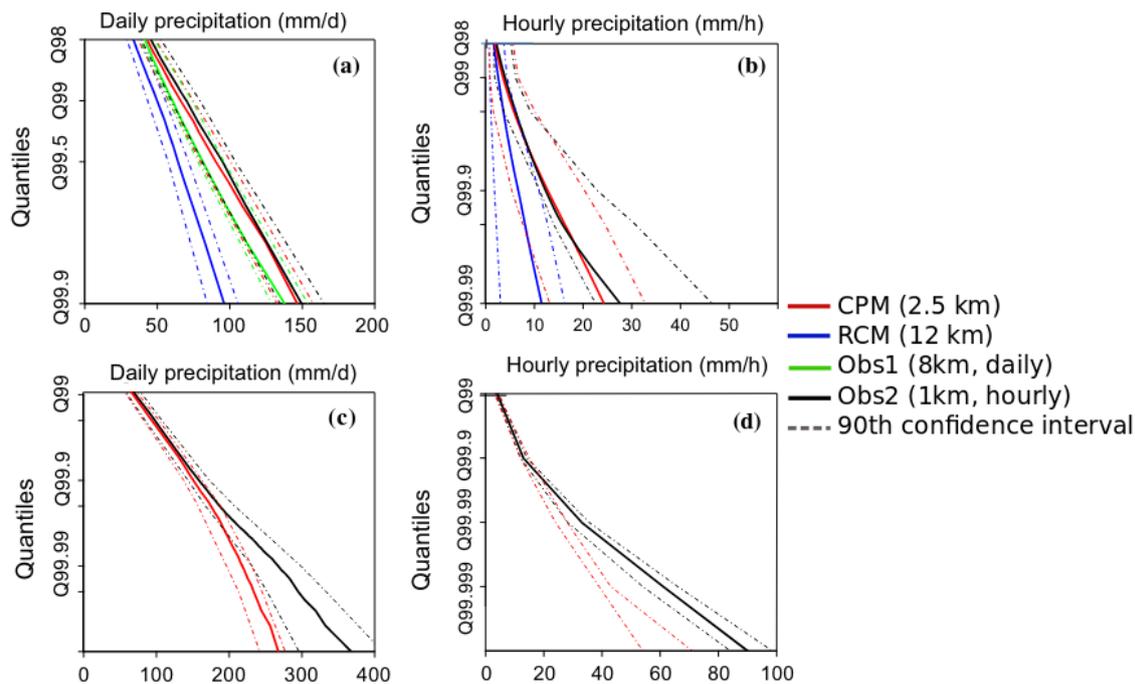


Figure 1.14: Extreme precipitation distribution. The plots show the returned CDF zooming on the extreme quantiles. It compares a CPM (AROME, 2.5km) and its driving RCM (ALADIN63, 12km) with two observations set, at 8km resolution and at 1km.

be addressed. Besides, like for GCMs, the agreement of multiple GCM-RCM pairs bring robustness to the result.

Recent literature (Fumière et al, 2020; Berthou et al, 2020; Ban et al, 2021) have shown that the use of higher resolution CPMs largely improves the representation of extreme precipitations compared with 12km resolution RCMs. Figure 1.14, extracted from Fumière et al (2020) compares the distribution of extreme precipitation for a CPM model and its driving RCM versus two sets of observation. We can see that the CPM reproduces until the highest quantiles of the high-resolution observations. Caillaud et al (2021) shows that CPMs can recreate realistic objects of extreme precipitation thanks to a good representation of hourly rainfall. The time evolution and movements, the intensity and the frequency of the objects are well represented by the evaluated model in the evaluation running mode. The future projections indicate an intensification of such extreme events, but such findings need to be confirmed by ensemble studies.

More on the RCM vs GCM large scales inconsistencies

Laprise et al (2008) propose an original evaluation of regional climate models following a set of 4 fundamental tenets that represent the initial paradigm of dynamical downscaling:

- Tenet 1: RCMs are capable of generating small-scale features absent in the driving fields supplied as lateral boundary conditions ;
- Tenet 2: The small scales that are generated have the appropriate amplitudes and climate statistics ;

- Tenet 3: The generated small scales accurately represent those that would be present in the driving data if it were not limited by resolution;
- Tenet 4: In performing dynamical downscaling, RCM generated small scales are uniquely defined for a given set of LBC.

The study easily validates the two first tenets, while the two last are mostly invalidated. Indeed, as mentioned earlier, the RCM is not fully deterministic and tends to derive from the large driving scale creating its own climatology. Moreover, several studies (Sanchez-Gomez et al, 2009; Lucas-Picher et al, 2008) have shown the impact of the initial conditions on the internal variability of the model, confirming the non-verification of tenet 4. Nevertheless, in climate studies, the deterministic property is not necessary, as only the statistical representation of the local climate is really of interest. Laprise et al (2008) proposes a last tenet which is declined in 3 versions as there is little consensus in the community. It concerns the modification of large scales by the RCM :

- Tenet 5a: The large scales are unaffected within the RCM domain;
- Tenet 5b: The large scales may be improved owing to reduced truncation and explicit treatment of some mesoscale processes with increased resolution within the RCM domain;
- Tenet 5c: The scales larger than or comparable to the RCM domain are degraded because the limited domain is too small to handle these adequately.

If the study invalidates tenet 5a, it is more difficult to assess whether the RCM improves or degrades the large scales. This is still a hot topic in the regional climate modelling community. The debate is intrinsically linked with the spectral nudging debate mentioned earlier. Various recent studies have shown that the Euro-Cordex ensemble largely underestimates the summer European warming projected by the CMIP5 models (Boé et al, 2020; Ribes et al, 2022). Here again, the opinion diverges on the soundness of this result. Some studies argue that the RCMs projection is more reliable than GCM because they show less bias in the historical period or because of the better representation of local physical processes impacting the large scales (Bartók et al, 2017; Sørland et al, 2018). On the other hand, other studies point out the absence of time-varying anthropogenic aerosols (Boé et al, 2020; Gutiérrez et al, 2020) and CO2 effect on vegetation (Schwingshackl et al, 2019; Boé, 2021) in most EURO-CORDEX simulations which would go in the line of CMIP5 models. Lastly, with a dedicated modelling experiment, (Taranu et al, 2022) concludes that the aerosols play a significant role in these discrepancies but also highlight the inconsistencies in physical parametrisations in GCM-RCM pairs.

1.3.2 Statistical Downscaling

The alternative to dynamical downscaling and RCMs relies on statistical approaches. Empirical/Statistical Downscaling (ESD) aims to estimate the link between high resolution observations (predictands, target, Y) and large scale climate variables (predictors X) (Maraun and Widmann, 2018; Maraun et al, 2010):

$$\mathbb{E}(Y|X) = f_{\Omega}(X) \quad (1.1)$$

with \mathbb{E} the common expectation and Ω represents a set of unknown parameters estimated during the training or calibration phase. These methods focus mainly on surface variables, such as temperature, wind or precipitation, essential for impact studies. The model f will vary according to the variable to downscale. The set of parameters Ω represents the local characteristics influencing the climate and will depend highly on the location. This estimated relationship f is then applied to low resolution global climate model outputs to reconstruct high-resolution climate information. Over the last three decades, many methods have been proposed. Maraun et al (2010) regroups ESD methods into three classes that we will detail below.

Perfect Prognosis

The perfect prognosis (PP) statistical downscaling methods aim to link a set of observed large scale predictors ($X = X_1, X_2, \dots$) and observed local target variable Y at time t . Eq. 1.1 becomes then :

$$\mathbb{E}(Y_t|X_t) = f_{\Omega}(X_t) + \epsilon_t \quad (1.2)$$

With ϵ , the error term represents what is not explained by X . The development of PP approaches includes three steps: selecting and preparing inputs (1), obtaining a *good*⁹ dataset for Y (3), and choosing the statistical model (2). Accordingly, it is necessary to calibrate the model, i.e. estimate the parameters Ω with the observed data. Each X_t has a unique corresponding Y_t , so the calibration is done element-wise. The observed Y_t is generally a meteorological station allowing long period records. Recently, high-resolution gridded observational products have also been used since their period extent starts to be long enough (Baño-Medina et al, 2020). The predictors X_t generally come from reanalysis that ensures homogeneous and continuous data.

Predictors are chosen to describe the general atmospheric circulation. Temperature, altitude of given pressure levels (geopotential), humidity or wind components are the most commonly used predictors (Gutiérrez et al, 2019). They can be considered at the surface level or higher altitudes. Multiple studies have shown that the best set of predictors highly depends on the season and the location (Timbal et al, 2009; Lemus-Canovas and Brands, 2020; Erlandsen et al, 2020). Moreover, those predictors are high-dimensional 2D fields, and it is necessary to reduce their dimension to suit most statistical models. Because of the high correlation between neighbour points and between variables, the application of variance decomposition methods such as Principal Components Analysis (PCA) helps to reduce the dimension drastically while keeping a maximum of information. Other methods taking into account the predictand can be applied for the same purpose. The high spatial variability of climate when it comes to very local scales, especially for precipitation, implies building very local models or choosing common predictors for larger target domain, which can degrade performance.

A wide range of statistical models has been proposed to estimate that relationship. To keep it simple, we can regroup them through two big families: the transfer

⁹i.e. long, homogeneous

functions (TF) and analogues approach (A) (Gutiérrez et al, 2019; Vaittinada Ayar et al, 2015). The transfer functions methods generally rely on regression functions. Multiple Linear function have been widely used to model temperature (Huth et al, 2015; Hertig et al, 2013; Gutiérrez et al, 2013) with various predictor selection techniques. The highly skewed nature of precipitation distribution requires the use of Generalized Linear Models (GLM) as they allow Y to follow a non-Gaussian distribution (San-Martín et al, 2017). The linear model estimates a transformation of the expected outcome (see Eq 1.3). More complex regressions such as General Additive Models (GAMs) that generalize GLMs to nonlinear models, Vector GLM (Vrac et al, 2007a) or non-parametric regressions are existing alternatives.

$$g(\mathbb{E}(Y|X)) = \omega_1 X_1 + \omega_2 X_2 + \dots \quad (1.3)$$

with g the transformation function and $\omega_i \in \Omega$ the model weights.

Recent studies explored the application of machine learning algorithms, notably deep neural network architecture (Vandal et al, 2019; Baño-Medina et al, 2020, 2021; Höhlein et al, 2020). The big advantage of such methods is their ability to identify the appropriate predictors (Baño-Medina et al, 2020) and to deal with high-dimensional inputs and outputs. Indeed, most TF methods are univariate (single-site, single variable), while deep neural networks can easily produce maps and can explore more than one variable. Neural networks are an essential part of this work, and their functioning will be detailed in chapter 2, later in the manuscript. Most methods do not account for ϵ , making them fully deterministic, but some are stochastic since they propose an estimation of the variance or distribution parameters (Cannon, 2008; Baño-Medina et al, 2020)

On the other hand, analog methods use the predictors X_t to find similar (analog) atmospheric situations $A(X_t)$ in the calibration set, such that the $\mathbb{E}(Y_t|X_t)$ is a weighted combinations of $Y_{A(X_t)}$. Those methods have primarily been used in weather prediction (Lorenz, 1969). Different variations on the way to evaluate the distance between predictors or the way to combine the predictands exist (Dayon et al, 2015; Maurer et al, 2010; Gutiérrez et al, 2013; Vaittinada Ayar et al, 2015).

Model Output Statistics (MOS)

MOS approaches aim to link model outputs to observed station records directly. The idea is to find the function that transforms the model output distribution into the station data distribution to fit the local climatology. MOS methods do not imply predictor selection. The target variable low resolution model output is the only predictor and it is taken from the closest grid point. Thus, MOS approaches are quite simple to use. However, the relation is valid only for the calibration model, as the biases are model dependent. Most of these methods are based on quantile mapping algorithms, and a wide range of variants have been developed (Déqué et al, 2007; Michelangeli et al, 2009; Štěpánek et al, 2016; Themeßl et al, 2012). For precipitation, some methods comport two steps: the prediction of occurrence first and then the amounts on wet days. This is, for example, the case of Volosciuk et al (2017), which uses logistic regression to estimate the occurrence or not of rain and a GLM to fit the parameters of a gamma distribution. The great majority of

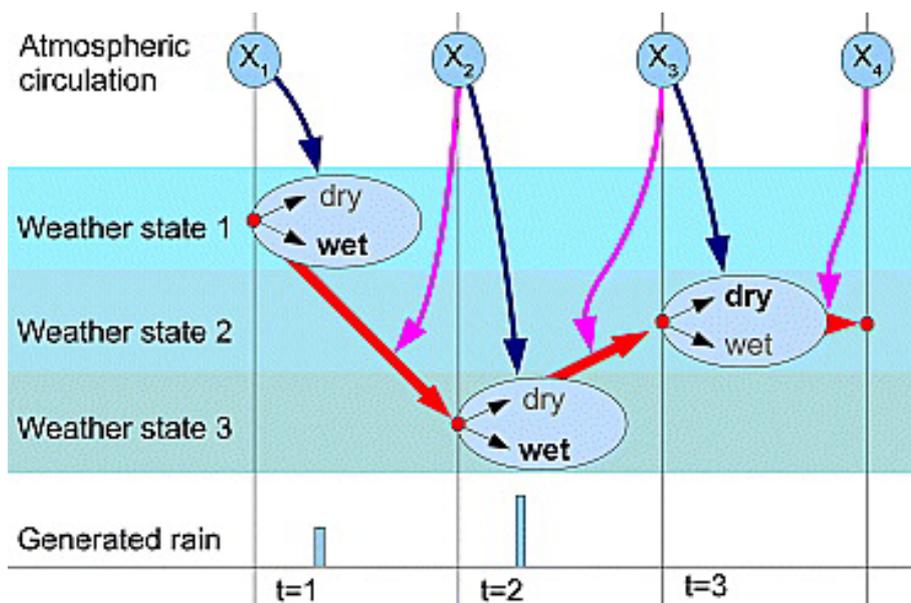


Figure 1.15: Example of weather generator (from Maraun et al, 2010; Vrac and Naveau, 2007) based on non-homogeneous Markov chains. The transition from one state to another depends on the past state (hidden Markov model) and the atmospheric conditions. Moreover, the amount of rain on a wet day is generated from a distribution dependent on the atmospheric conditions.

MOS approaches are single site and single variable. Nevertheless, MOS methods are similar in many points to Bias Correction techniques. Recent developments of multivariate BC techniques (Robin et al, 2019; Vrac, 2018; François et al, 2020, 2021) could eventually be applied in statistical downscaling framework.

Weather Generator (WG)

Weather Generators are defined as *stochastic models of meteorological variables that explicitly model their marginal distribution and temporal dependence* (Maraun and Widmann, 2018). They regroup various models, including Markov chain or auto-regressive models. The model parameters are fitted thanks to observed time series. For instance, first-order Markov chains have been widely used to simulate rain occurrence (Richardson, 1981; Katz, 1977; Wilby et al, 1998; Dubrovský et al, 2004) and are coupled with gamma distribution to model the precipitation amount. Alternatively to the gamma distribution, some studies have used a mixture of exponential distribution (Keller et al, 2017).

In the downscaling framework, statistical changes from one period to the others are evaluated in a climate model at the grid point corresponding to the observed series. Those changes are then applied to the statistical model (Kilsby et al, 2007). Another way is to include covariates from the climate model at different steps of the statistical model. Figure 1.15 (from Maraun et al, 2010; Vrac and Naveau, 2007) illustrates such weather generator which can also be considered as a perfect prognosis method.

Evaluation and intercomparison studies

The fifth assessment report of the IPCC concluded that it is difficult to provide a general assessment of ESD approaches performances because, among other things, of the wide variety of approaches. In most studies mentioned above, ESD methods are evaluated independently on their ability to reproduce a hidden set of observed series. Some intercomparison studies exist, but they mainly focus on a particular family of methods, such as MOS approaches in Gudmundsson et al (2012); Gutmann et al (2014). For this purpose, the VALUE framework (Maraun et al, 2015), proposes a large intercomparison exercise over Europe, where different teams can propose their downscaling methods. The VALUE project includes three steps. A first perfect predictor experiment evaluates the methods versus observations using reanalysis predictors. Then it plans the inter-comparison of the methods with GCM historical (Experiment II) and future (Experiment III) simulations. At the date where this manuscript is written, only the conclusions of the first experiment have been published (Gutiérrez et al, 2019; Hertig et al, 2019; Soares et al, 2019; Maraun et al, 2019a,b). Recently, Vaithinada Ayar et al (2015) proposed also an intercomparison of downscaling methods including both RCMs and ESD over Europe and Bettolli et al (2021) one over South America in the CORDEX framework.

The general result of ESD approaches in perfect predictor experiments is that a well-chosen and constructed model can fit the original data correctly. MOS approaches are generally simple to set up and provide good results. The performance of PP methods relies a lot on predictor choice, and a very local model is preferable to a large one. WG can recreate the correct statistics properties of the original series but are unable (by construction) to create appropriate temporal and spatial variability with, for example, the case of dry or hot/cold spells. More generally, the representation of extreme values is more difficult (Hertig et al, 2019), even if some methods perform better than others (Vaithinada Ayar et al, 2015). The critical point is that no method seems to outperform another in all aspects. The second point is that ESD methods can represent realistic high-resolution surface variables in the present climate. Compared to RCMs, they often have fewer biases with respect to observations, which seems logical since they use them for calibration. Vandal et al (2019) or Baño-Medina et al (2020) propose an intercomparison of methods, including recent machine learning algorithms. Neural Networks showed outstanding accuracy in (1) dealing with gridded inputs and outputs and (2) reproducing well the statistical properties of the target variables.

Because of the evident lack of future observations, it is difficult to assess whether the estimated relationship between large and local scales holds in a future or different climate. Indeed, applying ESD approaches to climate model projections relies on the strong assumption that the relationship is stationary. In order to assess whether this hypothesis holds or not, some studies have used an RCM as observations, following a perfect model framework (Vrac et al, 2007b; Gutiérrez et al, 2013; Dayon et al, 2015; Lanzante et al, 2018; Erlandsen et al, 2020). f is estimated in the historic run, and tested on future projections. All those studies conclude that ESD methods are performing better in the historical period (where the relationship is learnt) than in future, especially regarding extremes (Vrac et al, 2007b). Lanzante et al (2018) found that the ESD method applied could not reproduce the RCM changes over mountains

area and along the littorals, regions where the RCM bring added value thanks to its high resolution. Gutiérrez et al (2013) also concludes that surface predictors are preferable because of a changing relation between altitude and surface temperature. Finally, Erlandsen et al (2020) finds that the calibration period and length are key factors for the method’s accuracy. Besides the stationarity assumption, which seems not to hold, this can be another substantial limitation for ESDs methods. If there are numerous long-term records of observations in Europe and North America, it is not the case for most places worldwide (Bettolli et al, 2021).

1.4 Objectives and scientific questions of this study.

We have seen in this introductory chapter that there is a strong need for reliable information about local climate change to conduct impact studies and prepare adaptive strategies and policies. “Reliable” implies a proper exploration of the different sources of uncertainties to assess what is very likely to happen or what is among the possible. It is also essential to explore the extreme worst-case scenarios among the possibles, as they imply huge societal costs.

Today the primary tool to study climate change is the family of global climate models and earth system models. However, their coarse resolution does not allow evaluation of the impacts at small scales. To answer this issue, two downscaling communities have emerged to fill the gap between local climate and GCM large scales. Empirical Statistical Downscaling methods present the decisive advantage of being relatively not expensive and quite easy to set up. However, they suffer some lack of trustability to reproduce future changes as they rely on the *stationarity assumption* (Dixon et al, 2016). Besides, the calibration requires long-term records of observations which are difficult to find in most places worldwide. On the other hand, dynamical downscaling relies on regional climate models that are recognised for their precise representation of local physical processes and simulate a realistic local climate. For instance, they have good abilities to simulate extreme events, dry/hot spells or heavy precipitations. However, their high expensive costs strongly limit the exploration of the uncertainties.

In this context, this work aims to answer the strong challenge of producing reliable information on local climate change by reconciling these two fields. Indeed, the strengths and limitations of each family of methods match the others pretty well. This work aims to expand the number of high-resolution RCM simulations regarding some specific variables. The idea is to learn, with a machine learning algorithm, the downscaling function included in the RCM to apply it to other GCM simulations. According to the definition given in Reichstein et al (2019)¹⁰, we aim to build an Emulator for the RCM downscaling function. We will name this novel approach RCM-Emulator. Another way of seeing this work’s objective is to build an ESD approach that uses RCM simulations to be calibrated. The advantage of this approach is the possibility of learning the downscaling relationship in different

¹⁰Emulation: Emulation of the full (or specific parts of) a physical model can be useful for computational efficiency and tractability reasons. Machine learning emulators once trained can achieve orders of magnitude faster simulations than the original physical model without sacrificing significant accuracy

climates (including warmer climates) and regions with no observation records. The RCM-emulator proposed here could be categorised as a Perfect Prognosis approach as it aims to learn the relationship between low-resolution large-scale predictors and high resolution surface variables.

Other studies have already combined ESD techniques with RCMs, notably in 2-step downscaling approaches where predictors are taken from RCMs simulations (Vrac et al, 2012; Pryor and Barthelmie, 2014). This is particularly the case for MOS approaches initially designed for this purpose. Chadwick et al (2011) is probably the first study to propose combining ESD and RCM similarly to this work. The authors use neural networks to link GCM and RCM outputs. The study showed promising results and notably concluded the importance of learning in future climates. Walton et al (2015) use RCM simulations to estimate the monthly pattern of climate change over California and then create a small ensemble by emulating five other GCMs. However, it used a complex and domain-specific method to select the predictors, and the model was calibrated concisely (10 years in the past and future). The RCM-Emulator proposed in this work relies on deep neural network algorithm as they have already shown promising results in a perfect prognosis context. Recently, multiple studies have proposed similar approaches also based on neural network architectures (Wang et al, 2021; Serifi et al, 2021; Babausmail et al, 2021). The methods proposed are closer to a MOS approach as they aim to recreate the high resolution temperature or precipitation from the low resolution of the same variables.

Hereafter we introduce the scientific questions that will help us to define, develop and validate the novel hybrid downscaling tool we propose here. A complete understanding of the strengths and weaknesses of the tool is of primary importance in assessing its reliability.

1. Is the RCM-Emulator feasible?

The first essential step is to define the concept of the emulator properly. It implies defining the exact objective and the adapted framework to isolate and estimate the downscaling function carried out in a Regional Climate Model. Moreover, the choice of the statistical method used to estimate it is also part of the conceptualisation of the emulator.

2. How reliable is the emulator?

The second important aspect in developing a novel approach is guaranteeing its trustability. For both downscaling families, numerous studies have been dedicated to proving the reliability of the downscaling method. It implies finding the adapted frameworks to study the downscaling tool's abilities in different steps. It is notably necessary to evaluate the emulator production to reproduce the known added value of RCM. Moreover, the proper identification of strengths and weaknesses is essential to give confidence in the tool.

3. How efficient?

Efficiency is a crucial attribute of the emulator. The tool aims to tackle the high costs of RCMs and expand the number of downscaled simulations. The efficiency objective must be kept in mind throughout the emulator’s development and validation.

4. How transferable?

Finally, the transferability of the emulator to different levels needs to be assessed. The emulator will be defined in a certain framework, and evaluating how it reacts when moving out of this structure is important. The transferability can be considered to other sources of input, to other variables, to other climates or different downscaling problems. Assessing the transferability of the emulator is necessary to define how and to what problems it can be applied.

The RCM we will emulate in this study is ALADIN63 (Nabat et al, 2020). It is the model developed at CNRM, the research centre associated with the French weather service and CNRS. The ALADIN63 physical scheme is the same than the CNRM-CM family of models also developed at CNRM (in collaboration with CERFACS). ALADIN63 has been firstly used to downscale CNRM-CM5 (Voldoire et al, 2013), and then three others GCM: MPI-ESM-LR, developed at the Max Plank Institute (Giorgetta et al, 2013), NorESM1-M, developed at the Norwegian Climate Center (Bentsen et al, 2013) and the HadGEM2-ES, the Met Office climate model (Collins et al, 2011). This small ensemble is a good opportunity to build and evaluate the RCM-Emulator proposed in this study. The better consistency between ALADIN63 and CNRM-CM5 offers the best case for the first tests.

This manuscript is organised into five chapters after this introduction. Chapter 2 is entirely dedicated to the machine learning algorithm we chose for the emulator. As mentioned earlier, neural networks have shown accurate results in statistical downscaling. We will explore their functioning and analyse why those methods are well adapted for climate data and especially downscaling problems. Chapter 3 presents a feasibility study of the emulator. It aims to describe the concept and set the framework to train and evaluate the RCM-emulator for temperature. Chapters 4 and 5 explore then two transferability aspects of the emulator. Chapter 4 assesses its robustness to other input sources, while Chapter 5 explores the application to precipitation which is a more challenging variable. Finally, the last chapter derives some conclusions regarding this work’s achievements and introduces some perspectives that emerge regarding the development and use of RCM-Emulators.

Chapter 2

Why deep neural networks are a credible strategy to build the RCM-emulator ?

This chapter aims to understand the general functioning of artificial neural networks and why they are well adapted to build the RCM-Emulator. We will review the critical aspects of neural networks which led to the recent boom of deep learning approaches. I will specifically focus on convolutional neural networks and the specific UNet architecture that I chose for the Emulator because of its ability to deal with images. Finally, I will review some examples of deep learning applications in climate and weather sciences.

2.1 A neural network

Introduced by McCulloch and Pitts (1943), the artificial neuron is a mathematical representation of the biological neuron. Eq. 2.1 describes the behaviour of such a neuron. It receives an input X of dimension n and returns the evaluation y of dimension 1 through the activation function g of the weighted combination of the inputs. $[\omega_1, \dots, \omega_n]$ is the weights vector associated with the inputs. ω_0 is a constant term called the neuron bias¹. The activation function g is a non-linear function determining the ‘*type*’ of neuron. The different activation functions commonly used are shown in Table 2.1.

$$y = f_{\Omega}(x_1, \dots, x_n) = g\left(\omega_0 + \sum_{i=1}^n \omega_i x_i\right) \quad (2.1)$$

with $\Omega = \omega_0, \dots, \omega_n$

The simple neuron model can be seen as a simple linear binary classifier. It is the perceptron model, introduced by Rosenblatt (1958). The author proposed an algorithm to find the suitable weight vector that correctly classifies each input X . The weights’ vector defines the hyperplane’s coordinates that divide the input space into two distinct regions. The algorithm evaluates a learning set of input-outputs pairs

¹An alternative expression consist to set $x_0 = 1$ and then $y = g(\sum_{i=0}^n \omega_i x_i)$.

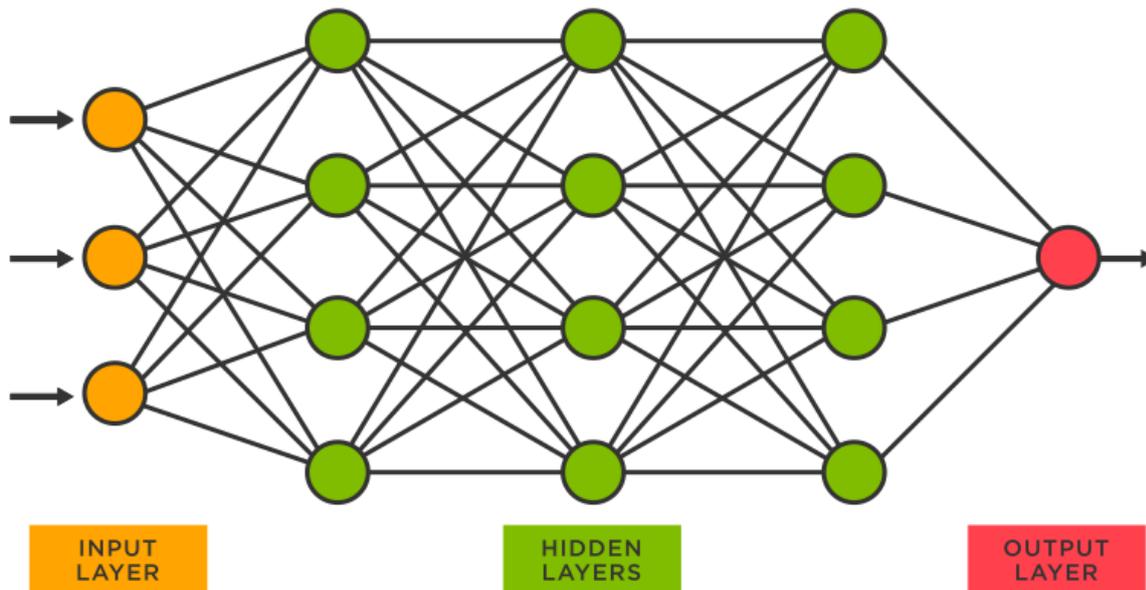


Figure 2.1: Neural Network diagram

$(X_t, Y_t)_{1 \leq t \leq N}$ and updates at each iteration the weight vector in case of misclassification. The algorithm's convergence has been proven in Novikoff (1962). Rosenblatt already imagined that two connected neurons could solve non-linear classification problems, but it was computationally too expensive to be implemented.

A neural network is then a set of neurons organised through layers. A network includes three types of layers illustrated in Figure 2.1: the input and output layers and some hidden layers between them. Hidden layers perform non-linear transformations of the inputs to match the targeted output. A hidden layer includes multiple neurons that are not inter-connected. The organisation of the neurons inside a layer depends on the type of layers and the architecture chosen. For example, in a fully connected network, all neurons from a given layer are connected to all neurons from the previous one. Other architectures exist such as Convolutional Neural Network (CNN) that I will present later on. The idea behind a neural network is to mimic the brain; each neuron can only perform a simple action, but the whole network can solve highly complex non-linear problems.

A neural network is then a set of parametric functions F_θ which links some inputs $X_t \in X$ and outputs $Y_t \in Y$. Θ represents the parameter space that regroups all possible values of θ , i.e. all the weights $\omega_{i,j}^k$ for the i -th input of the j -th neuron of layer k plus the bias $\omega_{0,j}^k$. An error (or loss) function is associated with the neural network. Given a training set $\mathcal{D} = \{(X_t, Y_t), t \in \llbracket 1, N \rrbracket\}$, the error function $L(\mathcal{D}, \theta)$ evaluates the distance between the target outputs Y_t and the actual network outputs \hat{Y}_t for the input X_t . The mean square error (Eq. 2.2) is the most common loss function for regression problems. Training the network involves finding the best set of parameters θ^* to minimise the training set's error:

$$L_{MSE}(\mathcal{D}, \theta) = \frac{1}{N} \sum_{i \in \mathcal{D}} (Y_i - \hat{Y}_i)^2 \quad (2.2)$$

Table 2.1: Commonly used activation functions

| NAME | EXPRESSION |
|------------------------------|---|
| Identity | $g(x) = x$ |
| Heaviside | $g_a(x) = \begin{cases} 0 & \text{if } x < a \\ 1 & \text{otherwise} \end{cases}$ |
| sign | $g_a(x) = \begin{cases} -1 & \text{if } x < a \\ 1 & \text{otherwise} \end{cases}$ |
| sigmoid | $g(x) = \frac{1}{1+\exp(-x)}$ |
| tanh | $g(x) = \frac{\exp(x)-\exp(-x)}{\exp(x)+\exp(-x)}$ |
| Rectified Linear Unit (RELU) | $g(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{otherwise} \end{cases}$ |
| Leaky RELU | $g_\alpha(x) = \begin{cases} \alpha x & \text{if } x < 0 \\ x & \text{otherwise} \end{cases}$ |
| softplus | $g(x) = \ln(1 + \exp(x))$ |

Therefore, the learned network corresponds to:

$$F_{\theta^*} = \arg \min_{F_\theta, \theta \in \Theta} L(\mathcal{D}, \theta) \quad (2.3)$$

To simplify, we include the bias ω_0 in the weight vector ω .

2.2 Training the network: the backpropagation algorithm

The next significant improvement in neural networks is the backpropagation algorithm introduced by Rumelhart and Hinton (1986). The big challenge when calibrating a neural network is to find the proper weights and biases for hidden layers while each neuron does not have a defined target outcome. Each neuron must be calibrated following the entire network error, implying that each neuron error function depends on the parameters from all previous and following layers. A direct update of all parameters, like in the perceptron model, is unimaginable because of the computations it implies. The backpropagation algorithm is an efficient optimisation scheme based on the gradient descent algorithm to address this issue. I briefly describe the gradient descent below.

Table 2.2: Notations for the backpropagation algorithm

| Notations | |
|--|--|
| $X \in \mathbb{R}^n$ | Set of inputs to the network |
| $Y \in \mathbb{R}$ | 1-D (for this example) target variable |
| \hat{Y}_t | Output of the network for sample t |
| $\mathcal{D} = \{(X_t, Y_t), t \in \llbracket 1, N \rrbracket\}$ | Training set |
| L | Loss function |
| θ | Set of parameter for the entire network |
| Ω | Set of weights for one neuron |
| k | Indicator for the layer, the last layer is called z |
| j | Indicator for the neuron in layer k |
| i | Indicator for the input of the j th neuron in layer k |
| $\omega_{i,j}^k$ | weight for the input i of the j th neuron in layer k |
| o_j^k | Output of the j th neuron in layer k |
| O^k | Number of outputs for layer k |
| a_j^k | Activation of the j th neuron in layer k |
| g, g_z | Activation function, (z for the activation function of the last layer) |
| ℓ | loss function output for one given sample (X^*, Y^*) |

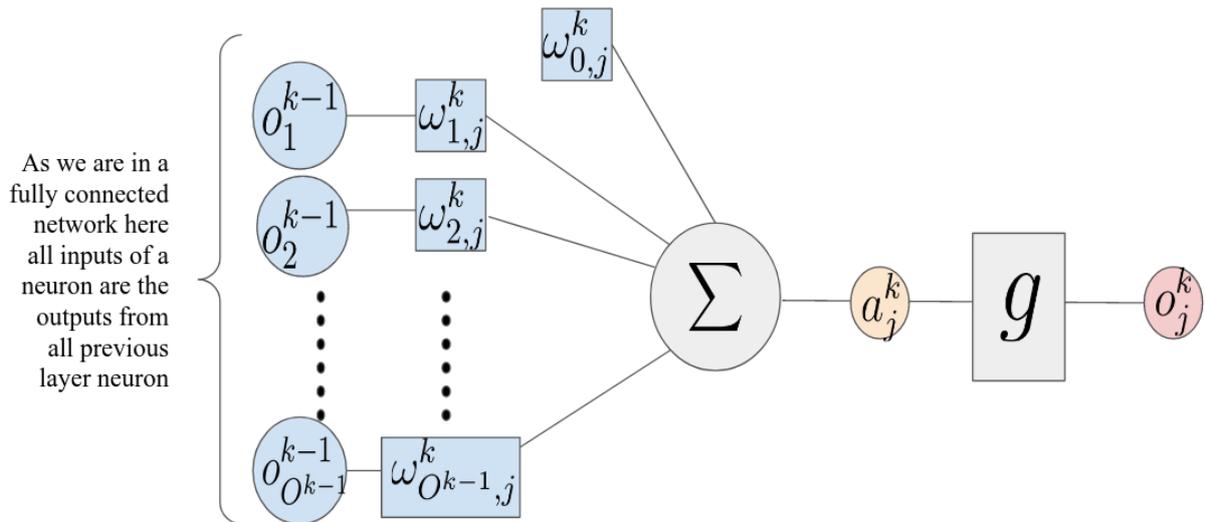


Figure 2.2: Illustration of a neuron with the notation for Section 2.2

The gradient descent is an iterative algorithm that finds a local minimum of a differentiable convex function. The gradient at position p of a n -dimensional function h , written $\nabla h(p)$ is the vector of its n partial derivative evaluated at this point. It gives the direction of the “fastest” increase of h at a given point p . In order to reach a local minimum, the algorithm evaluates the function gradient at a starting point and moves in the gradient’s opposite direction following a pre-defined step-size called “learning rate”. By iterating this process, one should be able to find a local minimum of the given function following Eq 2.4.

$$p_{t+1} = p_t - \lambda \nabla h(p_t) \quad (2.4)$$

In the case of neural networks, the aim is to minimise the loss function L according to the set of parameters θ . The backpropagation algorithm proposes an efficient way to update all the weights and biases of the network by simplifying the gradient descent computation. The idea is to evaluate the gradient in each weight $w_{i,j}^k$. Here we consider the loss function to be MSE which can be decomposed into a sum of individual errors (Eq. 2.5). It allows us to simplify the explanation here and to look at the computation for a single couple (X^*, Y^*) and we write the loss function evaluated at this point ℓ (Eq. 2.6). The notations for this part are summarized in Table 2.2 and illustrated for one neuron in Figure 2.2.

$$\frac{\partial L(\mathcal{D}, \theta)}{\partial \omega_{i,j}^k} = \frac{1}{N} \sum_{t \in \mathcal{D}} \frac{\partial L((X_t, Y_t), \theta)}{\partial \omega_{i,j}^k} \quad (2.5)$$

$$\ell = L((X^*, Y^*), \theta) \quad (2.6)$$

First of all we can re-write the partial derivative using the chain rule and defining the activation a_j^k that is the weighted sum of the neuron inputs plus the bias.

$$\frac{\partial \ell}{\partial \omega_{i,j}^k} = \frac{\partial \ell}{\partial a_j^k} \frac{\partial a_j^k}{\partial \omega_{i,j}^k} \quad (2.7)$$

We set the first term as $\delta_j^k = \frac{\partial \ell}{\partial a_j^k}$. It somehow represents the contribution of the j th neuron of layer k to the total error $(\hat{Y}^* - Y^*)$. The second term $\frac{\partial a_j^k}{\partial \omega_{i,j}^k}$ is the derivative of a sum. It is simply equal to the i th input of the neuron, which is the output o from the i th neuron of the previous layer² (Eq. 2.8). We define O^k as the total number of outputs of layer k .

$$\frac{\partial a_{i,j}^k}{\partial \omega_{i,j}^k} = \frac{\partial \sum_{r=0}^{O^{k-1}} \omega_{r,j}^k o_r^{k-1}}{\partial \omega_{i,j}^k} = o_i^{k-1} \quad (2.8)$$

The missing terms to evaluate are the δ_j^k . For the last layer z , we consider having a single output model so $j = 1$, we have :

$$\delta_1^z = \frac{\partial \ell}{\partial a_1^z} = \frac{\partial \frac{1}{2} (g_z(a_1^z) - Y^*)^2}{\partial a_1^z} = (g_z(a_1^z) - Y^*) g'_z(a_1^z) = (\hat{Y}^* - Y^*) g'_z(a_1^z) \quad (2.9)$$

²We assume here to be in a fully connected network where all neurons from a layer are connected to all neurons from the previous and subsequent layers.

since $\hat{Y}^* = g_z(a_1^z)$ with g_z the activation function of the last neuron.

For the hidden layer, we can write δ_j^k according to δ_j^{k+1} thanks to the backpropagation formula :

$$\begin{aligned}\delta_j^k &= \frac{\partial \ell}{\partial a_j^k} = \sum_{r=1}^{O^{k+1}} \frac{\partial \ell}{\partial a_r^{k+1}} \frac{\partial a_r^{k+1}}{\partial a_j^k} \\ &= \sum_{r=1}^{O^{k+1}} \delta_r^{k+1} \frac{\partial a_r^{k+1}}{\partial a_j^k} \\ &= \sum_{r=1}^{O^{k+1}} \delta_r^{k+1} \left(\frac{\partial}{\partial a_j^k} \sum_{s=1}^{O^k} \omega_{s,r}^{k+1} g(a_s^k) \right)\end{aligned}\tag{2.10}$$

$$\text{Finally, } \delta_j^k = g'(a_j^k) \sum_{r=1}^{O^{k+1}} w_{j,r}^{k+1} \delta_r^{k+1}$$

where O^{k+1} is the number of neurons from the next layer³ and $w_{j,r}^{k+1}$ are the weights given to the output of the j -th neuron from layer k in layer $k+1$. The formula gives that in each layer, the partial derivatives for each neuron's weights and biases can be evaluated from the following layers. Finally, the update formula for each weight, taking into account the entire training set \mathcal{D} and a learning rate λ is given by :

$$\Delta \omega_{i,j}^k = -\lambda \frac{\partial L(\mathcal{D}, \theta)}{\partial \omega_{i,j}^k} = \frac{1}{N} \sum_{t=0}^N \frac{\partial \ell_t}{\partial \omega_{i,j}^k} = \frac{1}{N} \sum_{t=0}^N \delta_j^{t,k} o_{i,j}^{t,k-1}\tag{2.11}$$

$$\text{With } \delta_j^{t,k} = \begin{cases} (\hat{y} - y) g'_z(a_1^z) & \text{if } k \text{ is the output layer} \\ g'(a_j^k) \sum_{r=1}^{O^{k+1}} w_{j,r}^{k+1} \delta_r^{k+1} & \text{else} \end{cases}$$

The optimisation algorithm comports then 2 steps computed for each couple (x_t, y_t) . In the forward step, x_t is given to the network and propagated through all layers to get the prediction. All weights and neurons' outputs are saved during this step. In the backward step, the δ s are computed from the last layer and backpropagated. The partial derivative from layer k are computed after the ones from layer $k+1$ following equation. After doing so for the entire training set, it is possible to calculate all partial derivatives and update θ 2.11.

The backpropagation algorithm was a huge improvement for neural networks since it was finally possible to train a neural network with multiple layers. From this point, there have been many developments with different layers or architectures, as we will see some examples in the next sections. The optimisation algorithm has also evolved to suit novel kinds of layers or to improve the convergence speed, but the principle of backpropagation is still the core of all variants. Most evolutions concern the gradient descent strategy. For example, the stochastic gradient descent works

³I use r here to go through all input for a given layer, it ranges from 1 to leave room for ω_0 which is the bias.

on the same principle but updates θ using only one example from the training set and not the entire dataset at each iteration. In this work, I use the ADAM (Adaptive Moment Estimation) optimiser introduced in Kingma and Lei Ba (2015) and recognised for its efficiency and good performance. It is an adapted stochastic gradient descent based on estimating the gradient's first and second-order momentum to adapt the learning rate for each parameter.

2.3 The last milestones: the ReLu function and the computational resources

In neural networks development, we are missing two important points to fill the 25 years gap between the backpropagation algorithm in the mid-80s and the boom of neural networks in the early 2010s. The first point is related to the vanishing gradient. The sigmoid or tanh functions were the most used activation function until the study of Glorot et al (2011). They showed two strong advantages of using the ReLu function (see Table 2.1) for hidden layers. It is almost linear, making computation easier (so quicker) and avoiding any gradient vanishing effect. Indeed sigmoid or tanh, as they are functions from \mathbf{R} to $[0, 1]$, tend to reduce the gradient after several steps in the backpropagation algorithm. The second advantage concerns the sparsity of the model, thanks to the negative part of the function. Indeed, ReLu can set to 0 some neurons bringing sparsity to the network, which Glorot et al (2011) and other studies found benefit for training the network.

The second point implies computational techniques and the new size of the available training dataset. In 2009, two major publications marked the beginning of neural network expansion. First, Deng et al (2009) created ImageNet, the current largest labeled image dataset. It constituted the largest training set, allowing to train deep neural networks (with numerous hidden layers). Lastly, Raina et al (2009) started to use Graphic Processor Unit (GPU) to train neural networks, which sped up the training of networks at an unprecedented rate (at least 10 times for the smallest networks and up to 70 times for the most time consuming ones).

2.4 Convolutional neural network (CNN)

The neural network presented until now and illustrated in Figure 2.1 is called a Fully Connected Network. It is composed only of Dense layers where each neuron is connected to all outputs from the previous layer. They have shown limitations in dealing with high-dimensional inputs, notably with images. Indeed as all neurons are connected, the number of parameters increases exponentially with the input size. Moreover, too many connections increase the risk of overfitting⁴. LeCun et al (1998), introduces LeNet, the first convolutional neural network specially designed to deal with images (visual recognition, classification, etc.). The originality of the network is the *convolutional layer* that I explain below.

⁴The neural network only learns “by heart” to reproduce the training set instead of estimating the function.

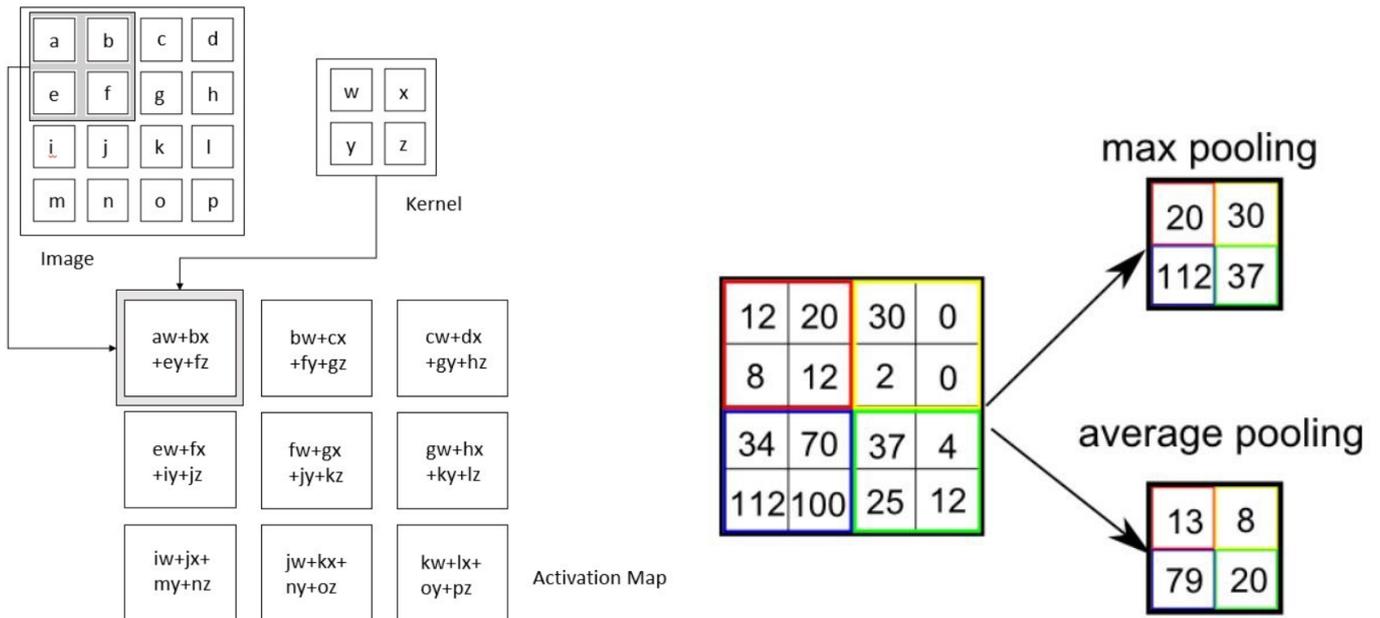


Figure 2.3: Convolution and Pooling principles, Goodfellow et al (2016).

As indicated by its name, the convolutional layer performs a convolution between the input matrix (an image where each matrix coordinate is a pixel) and a kernel matrix. The idea is to look at the image blockwise and extract information from each block. The convolution process is illustrated in Figure 2.3. The kernel moves through the entire image such that the convolution output is another matrix of smaller size. This output matrix, called *Activation Map* on the figure, is then passed to the activation function. The weights ω are then the values of the kernel matrix. Instead of 16 weights to fit on the illustration of Figure 2.3 with a dense layer, there are only 4 in this convolution layer. The number of parameters is defined by the kernel size, not input size. In order to get various decompositions, multiple kernels, called filters, can be passed through the input to isolate different aspects of the input. Moreover, a colourful image is a 3D matrix, where the height and width correspond to the image size, and the third dimension contains the three colour channels (red, green, and blue). Various kernels are then passed on to the different channels.

A strong advantage of a convolutional layer is that it uses the same patch of parameters in the whole image with the reasonable assumption that if a given kernel has been useful in a first position on the image, it is likely to be also useful somewhere else. There are three main parameters to define in a convolutional layer :

- The depth: the number of filters to pass through the image. The size of the kernel also has to be defined.
- The stride: the number of pixels to move the kernel. On Figure 2.3 the stride is 1, with a stride equal to 2, the output would be a 2×2 matrix.
- The Zero-padding: it helps control the output size by adding 0 around the input borders.

The *Pooling layer* is the other specificity of CNNs. It is here to cut the spatial size of its input. It divides the input matrix into separated blocks of the same size and applies a simple function, such as the maximum or the mean, to each of them (cf. Figure 2.3). After a convolutional layer, a pooling window of size 2×2 downsizes each feature matrix by 2. There are two main advantages of pooling. First, it reduces the number of computations, making the network more efficient and avoiding overfitting. Second, it summarises the information extracted by the convolutional layer in a given region. It makes the model more robust to little variations.

Convolutional neural networks are a succession of convolutional and pooling layers for the extraction of features followed by a fully connected network which performs the classification or regression task as illustrated in Figure 2.4. Various CNNs have been developed since LeNet (LeCun et al, 1998): AlexNet introduced by Krizhevsky et al (2012) marked the beginning of neural network and CNNs lead for image recognition task or VGGnet introduced in Simonyan and Zisserman (2015) which is still widely used nowadays.

If CNNs suit image classification or object detection well, they do not suit the semantic segmentation problem. The aim is to assign each pixel to the different objects in the image. The target of such a problem is a mask with the input image size. Long et al (2015) propose a fully convolutional network to address this exercise. It takes the same base as a classical CNN but replaces the dense part with an upsampling layer that enlarges its input's spatial size. There exists multiple upsampling functions, one being the transpose convolution.

A very efficient and famous architecture for image segmentation is the UNet (Ronneberger et al, 2015), shown in Figure 2.5. The idea of this network is to divide the network task into two equal parts. In a first contractive path, the neural network looks like a classical CNN. It is composed of downsampling blocks, including convolutional layers and max pooling. At each step, the spatial dimensions are divided by two while the number of features doubles. It extracts the required features of the input image. The originality of the UNet resides in the expanding or upsampling path, which is almost symmetrical to the downsampling part (making the U-shape). It is composed of transposed convolutions to increase the spatial dimension and recreate the output size. Classical convolutions with multiple features allow the network to propagate information to the high-resolution layers. The downsampling blocks' outputs are concatenated with each upsampling layer, which should help the network to relocate each pixel correctly. The UNet architecture can be seen as an encoder/decoder network. Such networks aim to encode high dimensional and complex information in a latent space of reduced dimension while being able to reconstruct the original data with the decoder. In UNet, the aim is not to reduce dimension, but the important information is extracted and encoded in the space of a smaller dimension and then injected into the expanding part. The UNet will be an essential part of my work, notably because of its superior ability to extract information from a set of inputs and re-inject it to reconstruct a new image.

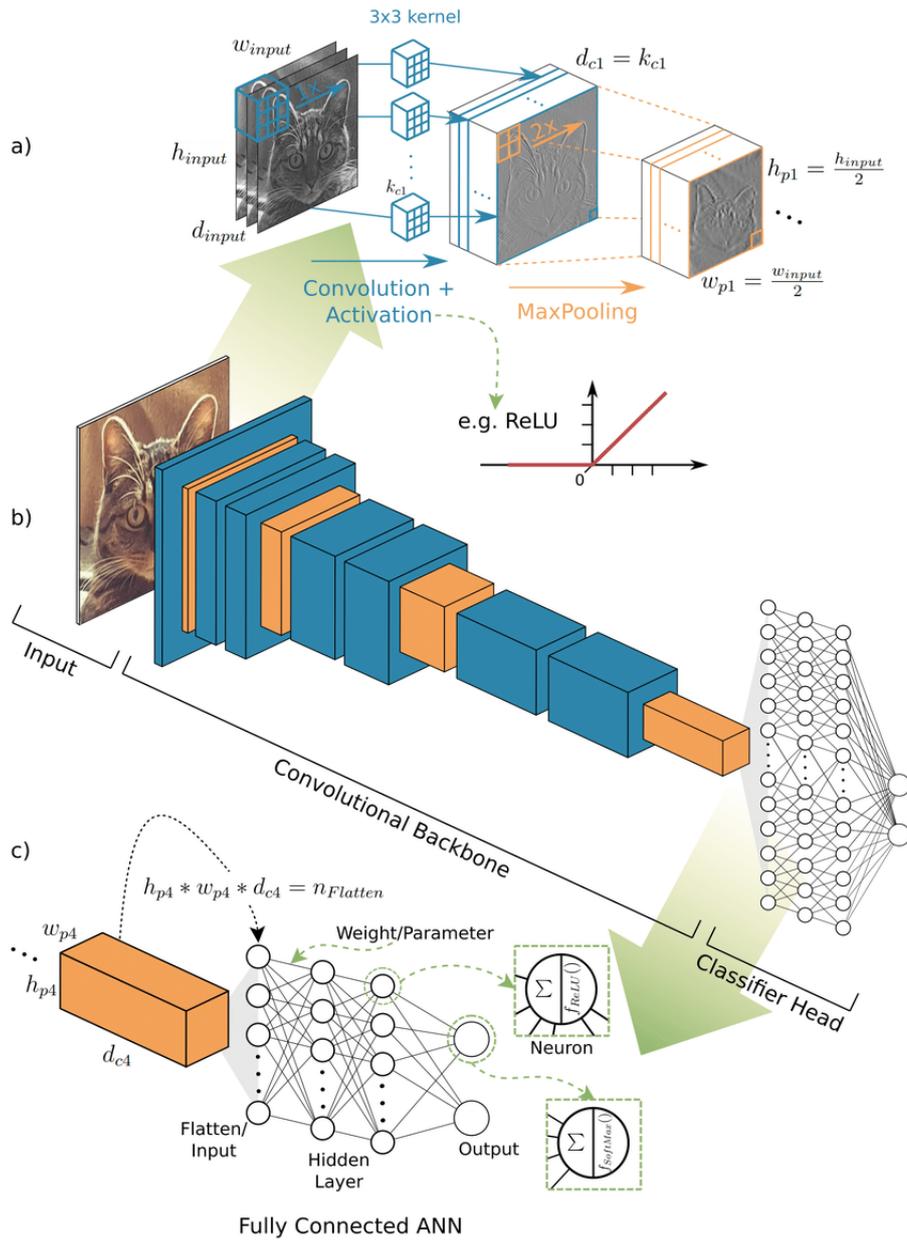


Figure 2.4: Illustration of CNN architecture, from Hoeser and Kuenzer (2020). (b) presents the entire network while (a) and (c) zoom on the beginning and end parts.

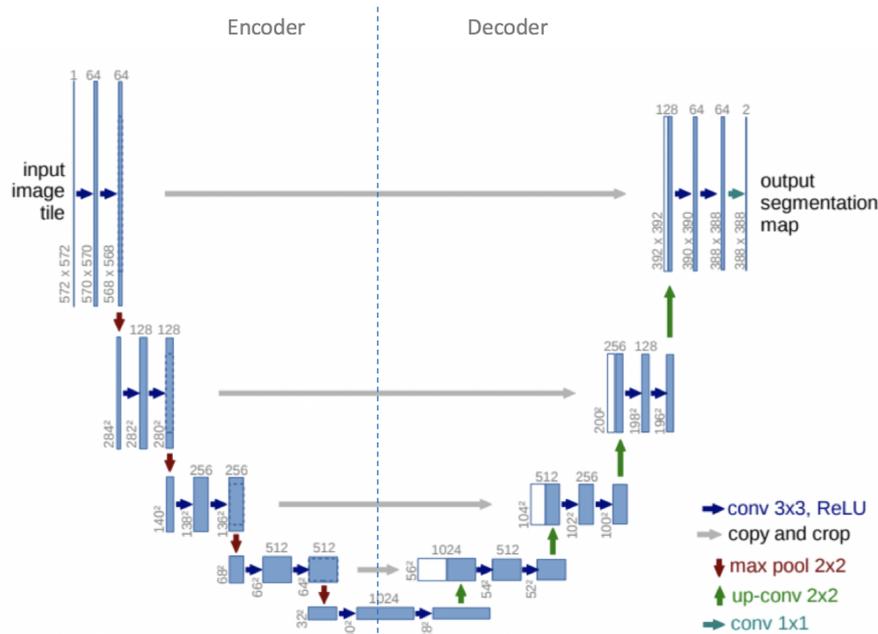


Figure 2.5: Illustration of UNET architecture, from Ronneberger et al (2015).

2.5 Other great families of architecture

On top of the family of convolutional neural networks, other families dedicated to specific tasks exist. I will briefly introduce two other noteworthy kinds of networks. The point here is not to detail their functioning but to understand the main idea behind these two families and for what they are designed.

First, Recurrent Neural Networks (RNNs) help to handle sequential data as time series or sentences. The idea behind RNN is to predict the output at time t thanks to the previous prediction and the corresponding input. Input and output are then considered sequentially. The most potent version of RNN is called LSTM and has shown good abilities to deal with long series. RNN can learn different types of relations as they can map one input to one output, one input to many outputs, many inputs to one output or many to many. Thus, they are widely used for text generation, time series forecasting or video anticipation.

The Generative Adversarial Network (GAN) is the other important family of neural networks. They consist in two neural networks that the user puts in competition. In general, they are used in the context of image generation. The first network is a generator which produces an image. The second network, called Discriminator, is trained to recognise if an image is true or fake, i.e. produced by the Generator. The two networks are then competing: the Generator tries to produce an image as close as possible to the “real” ones (i.e. such that the Discriminator classifies them as true ones). Simultaneously, the Discriminator keeps learning to differentiate between true and fake images. The network is trained when both Generator and Discriminator reach an equilibrium. Thus, the output of a GAN is one possible output among others. Neural networks generally produce what minimises the loss function, which can be a compromise of the less lousy output. A way to

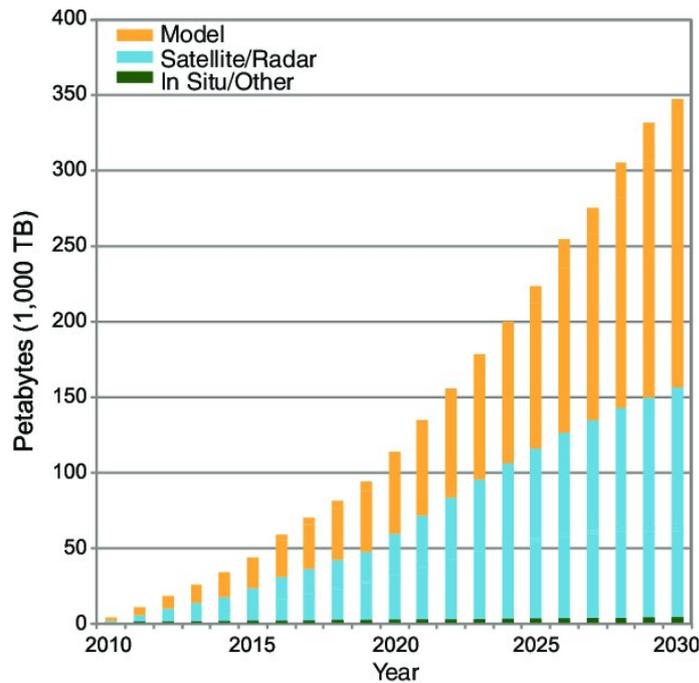


Figure 2.6: Projection of the climate data volume increase, from Overpeck et al (2011).

understand GAN is to see the Discriminator as the loss function of the Generator. Here again, various types of GAN exist, but all rely on the same principle.

2.6 Deep Learning and climate applications

The application of artificial neural networks to meteorological or climate problems is not new. Since the early 90s, several studies have used neural networks for forecasting weather or for statistical downscaling (Gardner and Dorling, 1998). Nevertheless, the recent boom of deep neural networks has also reached meteorology and climate science with an exponential increase in publications. Climate science has generated huge amounts of data from in-situ and satellite observations and climate models. The CMIP5 exercise generated about 5 Petabytes⁵ and CMIP6 about 23PB⁶, so about 5 times more in less than a decade. The projection made by Overpeck et al (2011) shown in Figure 2.6 seems to hold until now, and they forecast a stronger increase for the next decade. From a machine learning point of view, the strong advantage of climate data is that the climate modelling community produces clean, continuous and homogeneous gridded data. The outstanding performance of neural networks to deal with images notably but more generally to estimate non-linear relationships has brought much interest in the climate and weather communities. The convolutional layers are particularly adapted to climate maps since the different variables can be considered as the different channels of an image. In the following application examples, I will focus only on recent studies based on recent architecture.

Numerous studies have explored the capability of neural networks to forecast

⁵1 PB=1 000 TB=1 000 000 GB

⁶source : ESGF, data statistic page, Cinquini et al (2014)

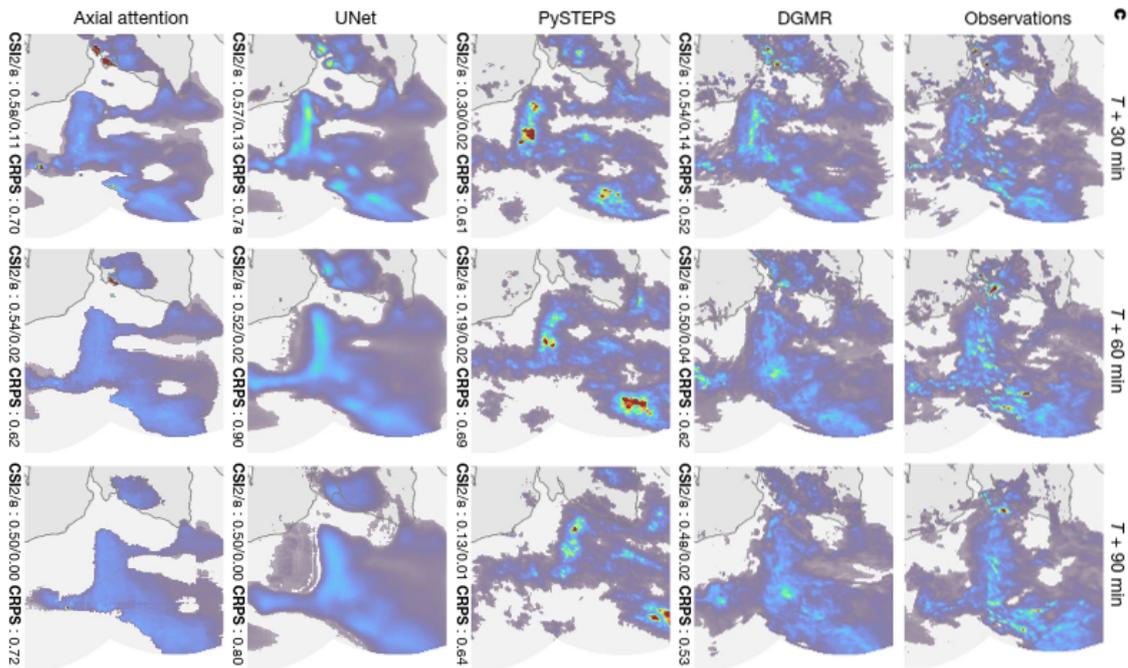


Figure 2.7: Precipitation nowcasting Ravuri et al (2021b).

weather at different time scales. Ravuri et al (2021a) recently presented a complex GAN architecture for precipitation nowcasting. The idea is to use live radar observation of precipitation and project the evolution of the precipitation objects for the following hours. The authors compare their method with a state-of-the-art stochastic approach (Pulkkinen et al, 2019) and two neural networks: one based on LSTM (Ayzel et al, 2020) and a second one based on a UNet model (Sønderby et al, 2020). The GAN network showed promising results and notably produced realistic maps with precise locations for precipitation, while the two other deep learning approaches create blurry images after some time steps. This result, illustrated in Figure 2.7, is typical of what GAN can offer: it is one possible outcome from the starting point while UNet, for example, tries to cover all the possibilities, which are increasing with time. Forecasting at other time scales has also been explored. Rasp et al (2020) proposed a benchmark dataset to evaluate and compare a data-driven medium-range (up to a week) forecasting model, including baseline scores from state-of-the-art numerical models and multiple studies already used it (Rasp and Thuerey, 2021; Scher and Messori, 2021). Among others, Lu et al (2022) or Martin et al (2022) have explored the predictability at sub-seasonal and seasonal scales, notably by focusing on mid-term oscillations. Recently, multiple review studies have tried to analyse the role of recent deep learning approaches in weather forecasting as it is not clear yet how far those methods will go (Schultz et al, 2021; Dewitte et al, 2021). If it seems evident that deep learning methods will help improve some aspects of the forecast, it is more difficult to say if an end-to-end approach will ever replace the current systems.

A second neural network application in the climate field is detecting and classifying meteorological objects. Indeed as mentioned earlier, climate fields can be considered images, and the good properties of convolutional neural networks in ob-

ject detection could help in many cases. For example, Racah et al (2017), introduce a labelled dataset from climate simulations and a baseline CNN architecture to encourage the development of an automatic and efficient method to detect in climate simulation extreme weather patterns. The labelled objects are Tropical Depression, Tropical Cyclones, Extra-Tropical Cyclones and Atmospheric Rivers, all subject to severe cause damage to human societies. Kim et al (2019) or Giffard-Roisin et al (2020) also propose NN-based method to track hurricanes and cyclones. Mounier et al (2022) uses very high resolution simulation to detect bow-echo⁷ in forecast ensembles. Nevertheless, object detection necessitates a large volume of good quality labelled data which limits their development strongly.

In the introductory chapter, I briefly presented the role of parametrisations. They are an empirical representation of processes that occur at finer scales than the grid resolution. These parametrisations are based on expert knowledge and might lead to model biases (Medeiros and Stevens, 2011). Some studies have proposed using neural networks to learn the actual physical processes within high-resolution models that resolve those processes (Rasp et al, 2020; Brajard et al, 2021). The results are auspicious, and most research still has to be done as some stabilisation problems persist (Balogh et al, 2022).

Finally, as mentioned in the first chapter, neural network architecture has also been used in empirical statistical downscaling. Indeed, statistical downscaling aims to learn a generally non-linear relationship between GCM output (or similar large-scale representation) and station data or high-resolution gridded observational products. Convolutional neural networks have proven to deal with high dimensional data in input and output. Their ability to reconstruct images, as in super-resolution exercises, is also very interesting, as spatial downscaling aims to recreate high-resolution maps. Moreover, CNNs manage, by construction, to identify the right combination of inputs for each output (station or grid point). The use of the artificial neural network is not new in statistical downscaling (Cannon, 2008; Chadwick et al, 2011), and the recent performance of CNN has motivated many studies to use them in ESD. Baño-Medina et al (2020) have notably built simple CNN architecture to downscale precipitation and temperature over Europe in a perfect prognosis context with similar or better (depending on the metric) performance than state-of-the-art ESD regression. From the existing literature I see two ways of applying deep neural networks to ESD problems. First, in perfect prognosis approach as Baño-Medina et al (2020) or Höhlelein et al (2020) following a classical regression task. Second, some studies have used them in a MOS approach, trying to link the low-resolution and the high-resolution from the same variables. The task is then similar to super-resolution problems that aim to increase the resolution of images. Wang et al (2021), Serifi et al (2021) and Babaousmail et al (2021) build RCM-emulators following this super resolution approach. In the following chapters, the proposed RCM-emulator is build in a perfect prognosis. Finally, it is remarkable that multiple studies have used a UNet architecture to downscale surface variables with generally promising results (Höhlein et al, 2020; Serifi et al, 2021; Sha et al, 2020a,b).

⁷Convective system responsible for extreme winds or tornados

2.7 Link with this work

This second chapter was somehow an extension of the introductory chapter as it did not aim to present the results of my work but to gently introduce convolutional neural networks and justify why they are adapted for the RCM-emulator. In preliminary work for this PhD, I tried other machine learning methods. The aim was to explore the idea's feasibility, so I concentrated on single grid points. I used classical ESD methods such as linear regression or advanced quantile mapping (CDFt, Michelangeli et al, 2009) and various machine learning algorithms based on decision trees like Random Forest or XGboost. In all cases, the downscaling relationship seemed well estimated, which gave us confidence in the feasibility of the RCM-emulator. Nevertheless, the efficiency of those methods is questioned. Indeed they are all univariate in the sense that they can consider only a single output. Therefore, it is necessary to fit one model for each grid point which can get really heavy as we work with high-resolution models. Convolutional Neural Networks present the nice advantage of dealing efficiently with large images implying a single model for the emulator. Moreover, we saw in this chapter that CNN would ensure some spatial coherence as the convolution layers consider the image blockwise. Multiple studies have shown good results for CNN in similar exercises as the one I propose here. I compared CNN with the other methods over single grid points, and they showed similar or better results. Chapter 3 includes such a comparison over large domain. Furthermore, after some small tests, I decided to base my work on the UNet architecture. Chapter 3 gives more details about the specific architecture I built. However, it is important to keep in mind here that the network architecture does not guarantee anything. The construction of the training set, the choices regarding some parameters or the loss function also have a crucial influence. The following chapters will explore and discuss those choices.

Chapter 3

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

A. Doury, S. Somot, S. Gadat, A. Ribes, L. Corre

This chapter is a published article in *Climate Dynamics*. I did not modify the corpus here. The organisation is the same, with the appendices and supplementary material at the end of the chapter. I simply modified the layout such that it stays coherent with the rest of the document. Some elements of the introduction and methodology section can be redundant with the previous chapters. The original article DOI is [10.1007/s00382-022-06343-9](https://doi.org/10.1007/s00382-022-06343-9).

3.1 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 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 to 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: Section 2 presents the whole framework to define, train and evaluate the emulator, while Section 3 shows the emulator results. Finally, Sections 4 and 5 discuss the results of the emulator and provide conclusions.

3.2 Methodology

This section provides a complete description of the UNet-based RCM-emulator used in this paper. The notations are summarised in Table 3.1. 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.

3.2.1 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

Table 3.1: Notations

| Notation | Description | Dimensions |
|-----------|---|---|
| i, j | Spatial indexes over input grid | $\{i, j\} \in \llbracket 1, I \rrbracket \times \llbracket 1, J \rrbracket$ |
| k, l | Spatial indexes over target grid | $\{k, l\} \in \llbracket 1, K \rrbracket \times \llbracket 1, L \rrbracket$ |
| 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 | $X_{t,i,j,x} \in \{\mathbb{N} \times \llbracket 1, I \rrbracket \times \llbracket 1, J \rrbracket \times V2D\}$ |
| Z | Set of 1D input for the emulator | $Z_{t,z} \{\mathbb{N} \times V1D\}$ |
| Y | Target : Daily Near-Surface Temperature | $Y_{t,k,l} \in \{\mathbb{N} \times \llbracket 1, K \rrbracket \times \llbracket 1, L \rrbracket\}$ |
| F | Downscaling function of the RCM | |
| \hat{F} | Emulator : Estimation of F | |

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 $0,11^\circ$ (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.

3.2.2 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. 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 3.2). 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 pre-

Table 3.2: List of predictors

| 2D Variables | | | | | |
|--|-----------------------------------|--------------------------------|-------|----------------------|-----------|
| Field | Altitude Levels | Variables Notation | Units | Temporal Aggregation | Dimension |
| 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] |

dictors. These variables are normalised (see Equation 3.1) 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 cosine, sine 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 Equation 3.2) 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.

$$\widetilde{X}_{t,i,j,x} = \frac{X_{t,i,j,x} - \overline{X}_{t,x}}{\dot{X}_{t,x}} \quad (3.1)$$

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

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 Section 3.2.4, we force the emulator to consider both items of information equally.

$$\widetilde{Z}_{t,z} = \frac{Z_{t,z} - \overline{Z}_{ref,z}}{\dot{Z}_{ref,z}} \quad (3.2)$$

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

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 to 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 Section 3.2.3). It is a 16×16 ($J = I = 16$ in Table 3.1) CNRM-CM5 grid box visible on Figure 3.1. Each observation given to the emulator (see Figure 3.1 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.

3.2.3 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 12km resolution (about $600\,000 \text{ km}^2$) centred over the south of France (Figure 3.1). 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

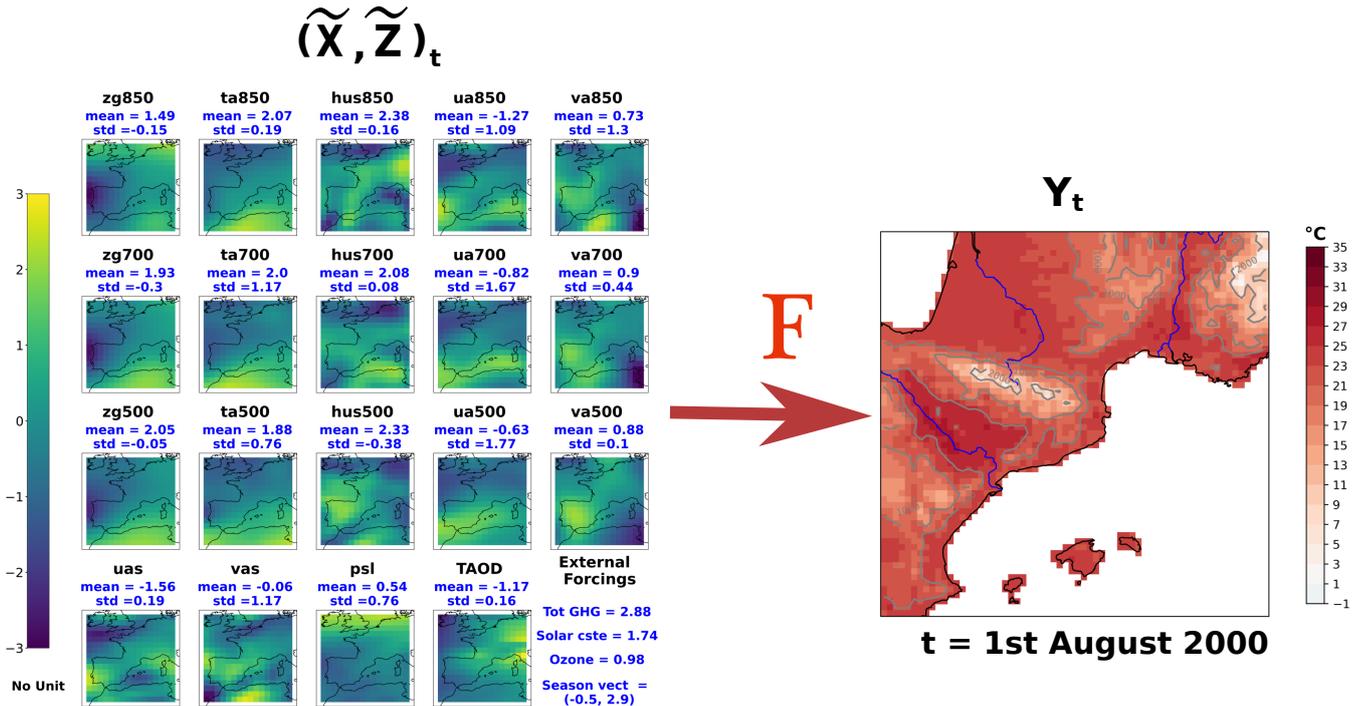


Figure 3.1: Illustration of an observation for a randomly-chosen day. Left: each map represents a 2D input variables (X), on the input domain, and the blue numbers correspond to the 1D variables (Z). Right: an example of Y , the near surface temperature on the target domain.

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 Balears (Spain), since they are invisible on the GCM grid and the RCM brings important information. Finally, three major rivers (plotted in blue in Figure 3.1) are on the domain with interesting impacts on climate (commented in Section 3.3): 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.

3.2.4 Deep learning with UNet

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:

$$F(X, Z) = E[Y | (X, Z)]. \quad (3.3)$$

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 $((X_t, Z_t), Y_t)_{1 \leq t \leq T}$ and try to build from the data an empirical estimation \hat{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:

$$\hat{F} = \arg \min_{F_\theta, \theta \in \Theta} \sum_{t=1}^T \|Y_t - F_\theta(X_t, Z_t)\|^2. \quad (3.4)$$

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:

$$\mathbf{n}((x_i)_{i \in E_{\mathbf{n}}}) = \phi(\langle (w_i)_{i \in E_{\mathbf{n}}}, (x_i)_{i \in E_{\mathbf{n}}} \rangle) = \phi\left(\sum_{i \in E_{\mathbf{n}}} w_i x_i\right). \quad (3.5)$$

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 (3.4) 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 backpropagation 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.

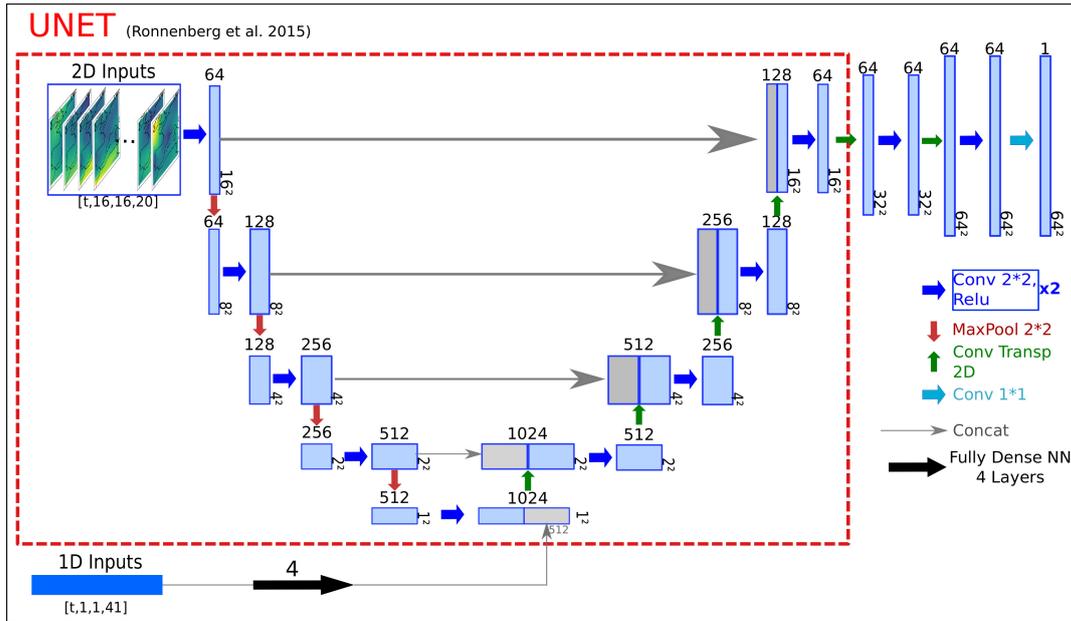


Figure 3.2: Scheme of the neural network architecture used for UNet emulator. The part of the network in the red frame corresponds to the original UNet defined in Ronneberger et al (2015)

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 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 Figure 3.2 and more precisely described in appendix 3.6.2. 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 3.2) which is given at the bottom of the UNet (see Figure 3.2). This 1D

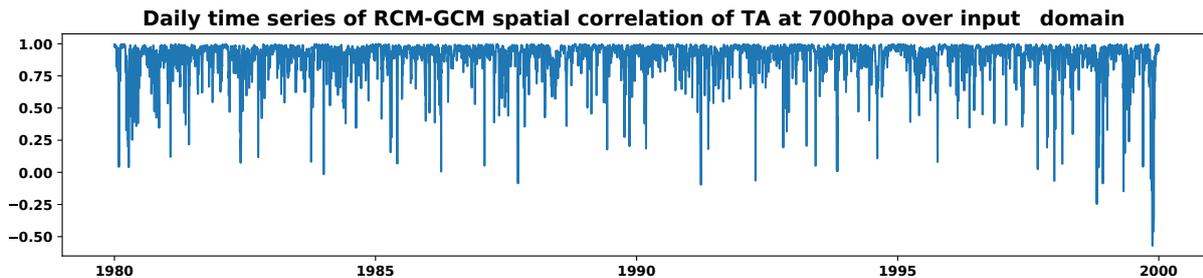


Figure 3.3: Time series of spatial correlation of the atmospheric temperature at 700hpa between ALADIN63 and its driving GCM, CNRM-CM5, over the input domain.

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 Figure 3.2.

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.

3.2.5 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 Section 3.2.4. 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, Figure 3.3 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 Section 3.4.1) 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 an empirical statistical downscaling framework.

The training protocol is summarised in Figure 3.4. In a first step, the RCM

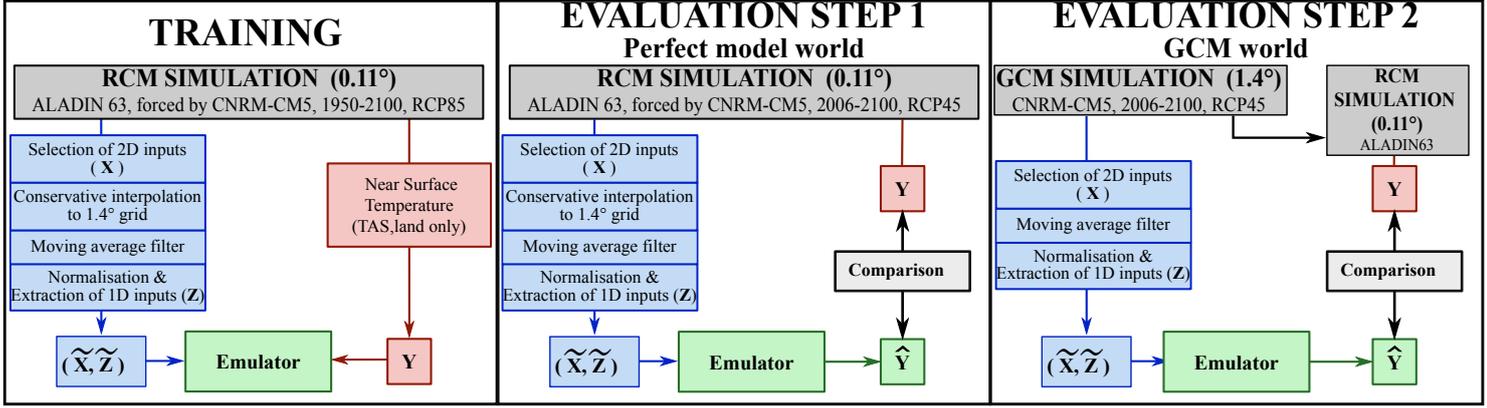


Figure 3.4: Scheme of the protocols for the training (left) and the two steps of evaluations (center and right).

simulation outputs are upscaled to the GCM resolution (about 150km) 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 3x3 moving average filter to respect the protocol described in Section 3.2.2. This smoothing also targets to delete any local scale information which might persist through the upscaling step (as discussed in Section 3.4.3).

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 Figure 3.4, 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.

3.2.6 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:

- **RMSE.** The Root Mean Squared Error measures the standard deviation of the prediction error (in °C):

$$RMSE(Y, \hat{Y}) = \sqrt{\frac{1}{T} \sum_t (Y_t - \hat{Y}_t)^2} \quad (3.6)$$

- **Temporal Anomalies Correlation.** This is the Pearson correlation coefficient after removing the seasonal cycle:

$$ACC(Y, \hat{Y}) = \rho(Y_a, \hat{Y}_a), \quad (3.7)$$

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.

$$RoV(Y, \widehat{Y}) = \frac{Var(\widehat{Y})}{Var(Y)} * 100 \quad (3.8)$$

- **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:

$$W_1(f(Y), f(\widehat{Y})) = \sum_{i=1}^T |Y_{(i)} - \widehat{Y}_{(i)}|, \quad (3.9)$$

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 Section 3.3) 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 (2006-2100) 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 Section 3.3 and Tables 3.3 and 3.4 in supplementary material.

3.2.7 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 Section 3.2.6 will be applied to our benchmark.

3.3 Results

This section presents the emulator performances in terms of its computational costs (in Section 3.3.1) and its ability to reproduce the near surface temperature time series at high resolution. As illustrated in Figure 3.4, we will evaluate the emulator in two steps, (1) in the perfect model world (Section 3.3.2) and (2) when the emulator inputs come from a GCM simulation (Section 3.3.3). 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.

3.3.1 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 seconds with a batch size of 100 observations. The training of the emulator takes about 2 hours. 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.

3.3.2 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 section 3.3.2 we analyse the ability of the emulator to reproduce the RCM simulation. The section 3.7 compares the specific emulator proposed in this study with respect to two others emulators relying on standard empirical statistical downscaling methods.

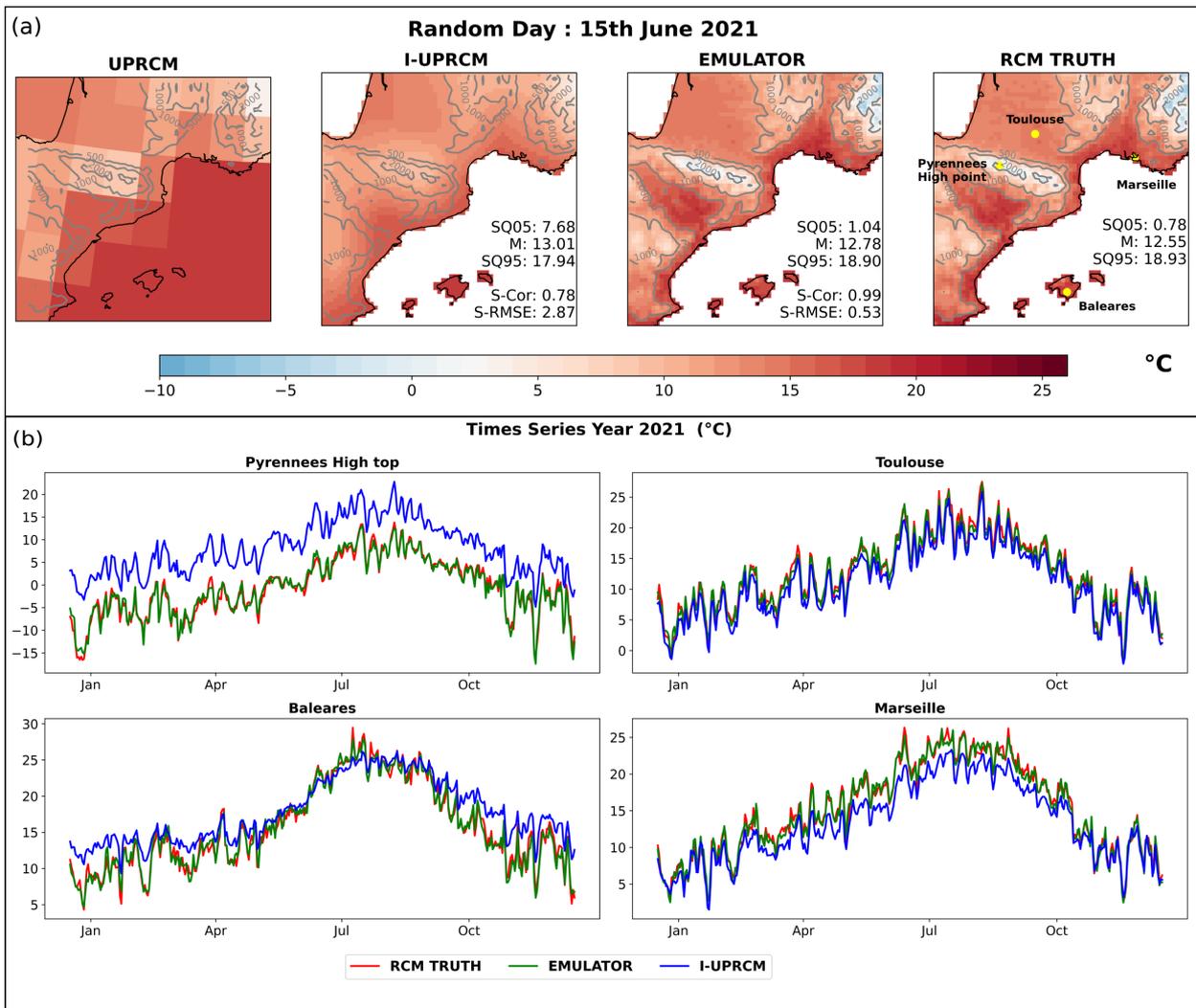


Figure 3.5: Randomly chosen illustration of the production of the emulator (in evaluation step 1) with inputs coming from UPRCM: (a) temperature ($^{\circ}\text{C}$) at a random day over the target domain for the raw UPRCM, the interpolated UPRCM, the emulator and the RCM truth and (b) random year time series ($^{\circ}\text{C}$) for 4 particular grid points.

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 3.5a 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

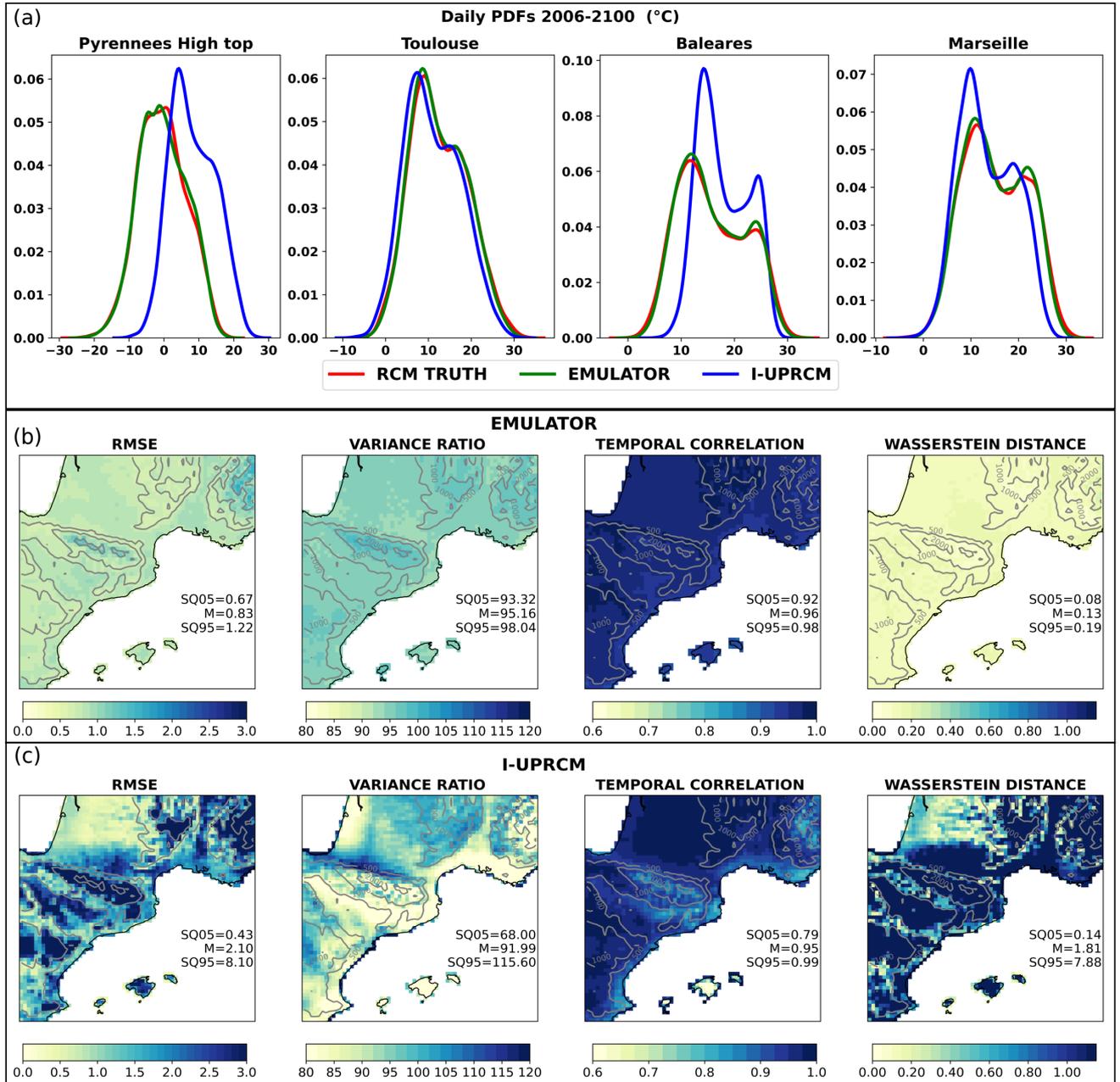


Figure 3.6: (a) Daily probability density functions from the RCM truth, the emulator (in evaluation step 1) and the I-UPRCM at 4 particular grid points over the whole simulation period. (b) (resp. (c)) Maps of performance scores of the emulator (resp. of the I-UPRCM) with respect to the RCM truth computed over the whole simulation period. For each map, the values of the spatial mean and super-quantiles (SQ05 and SQ95) are added.

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 Figure 3.5b, 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 3.5a 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 (Figure 3.6b) 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 Figure 3.7 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 (Figure 3.6c) or the present climatology error map (Figure 3.7a) 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 Figure 3.6c. 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 Figure 3.5b 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 Figures 3.6ac). Figures 3.6ab show that the emulator succeeds particularly well in filling these gaps.

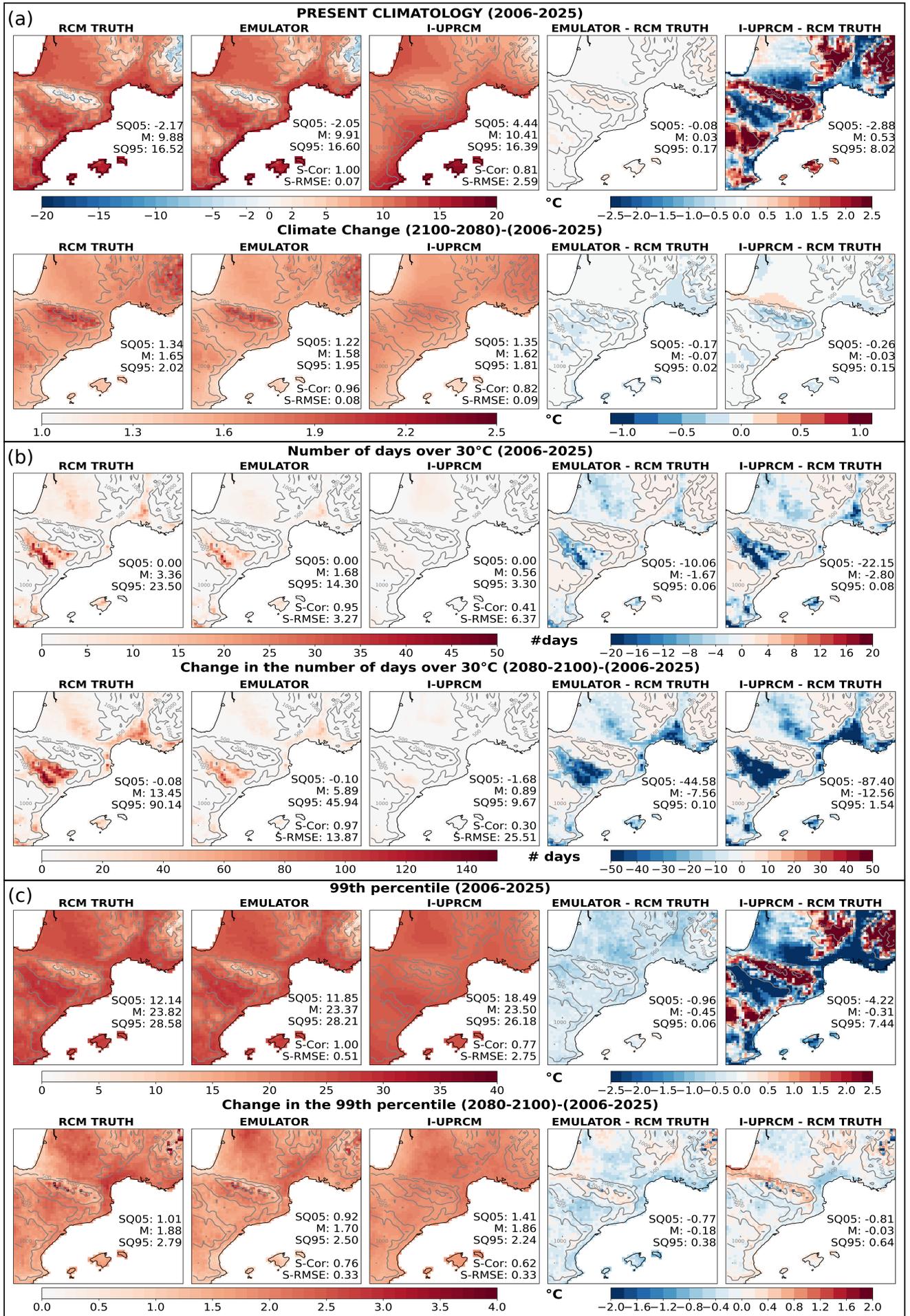


Figure 3.7: (a) Maps of long-term mean climatologies, (b) number of days over 30°C and (c) 99th percentile of daily near-surface temperature for a present-climate period (2006-2025) and for the climate change signal (2080-2100 minus 2006-2025), for the RCM truth, the emulator (in evaluation step 1) and 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 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 Figure 3.6a 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 Figure 3.7bc 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 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 Figures 3.7ac. 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.

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.

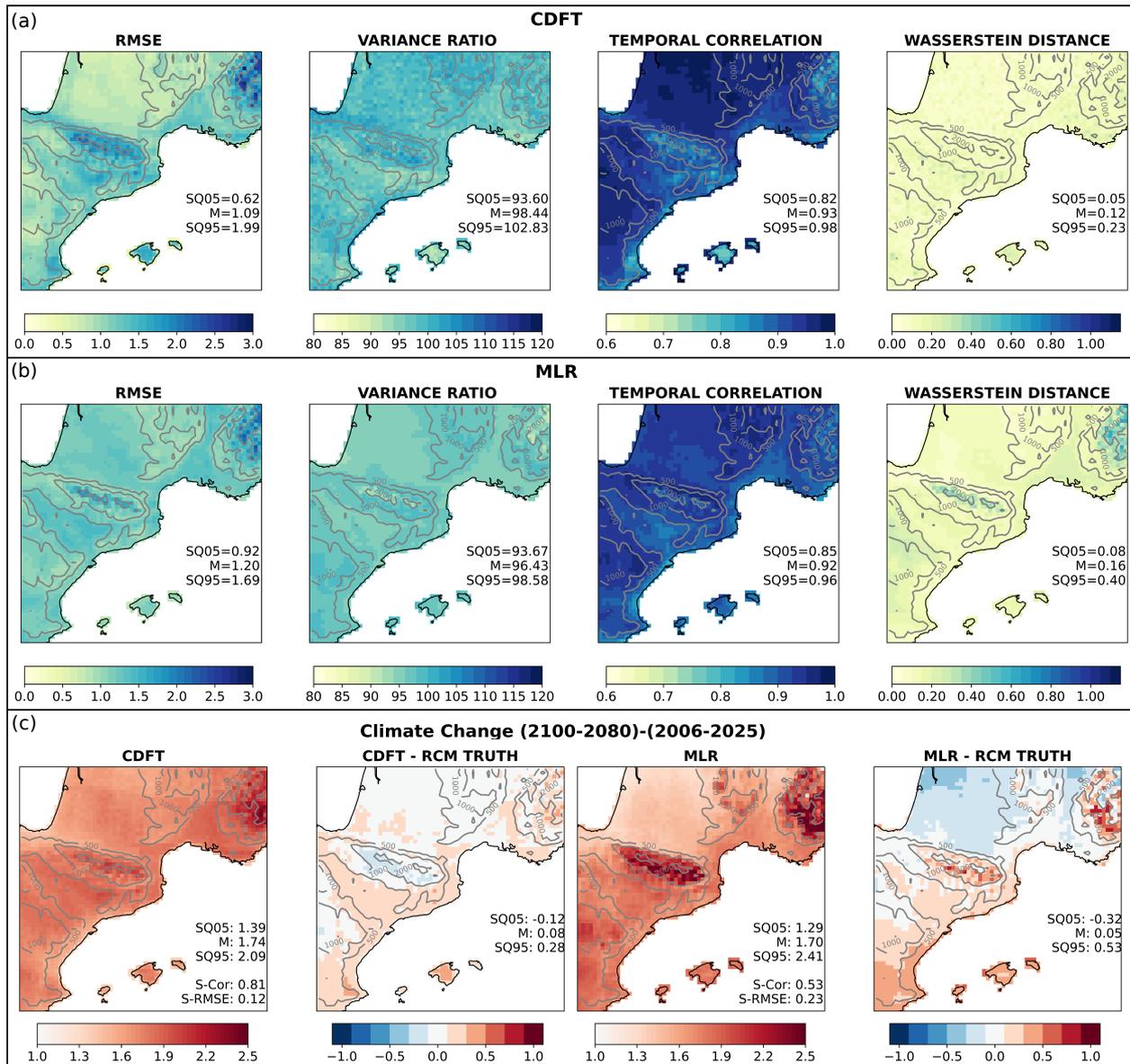


Figure 3.8: (a) (resp (b)) Maps of performance scores of the CDFT emulator (resp. of the MLR emulator) with respect to the RCM truth computed over the whole simulation period in perfect model evaluation. For each map, the values of the spatial mean and super-quantiles (SQ05 and SQ95) are added. (c) Climate change maps and the difference maps with respect to the RCM truth for the CDFT emulator and the MLR emulator. As on figure 3.6 and 3.7 the spatial scores are added on the maps.

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 appendix 3.6.1. 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 Section 3.2.5.

The results are illustrated on Figure 3.8 and a more complete evaluation can be found in the appendix 3.6.1. 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 (Figure 3.8c) 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 (Figures 3.7, 3.8 and appendix 3.6.1).

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.

3.3.3 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 3.9a illustrates the production of the emulator regarding the benchmark and RCM truth for the same day as Figure 3.5a. 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

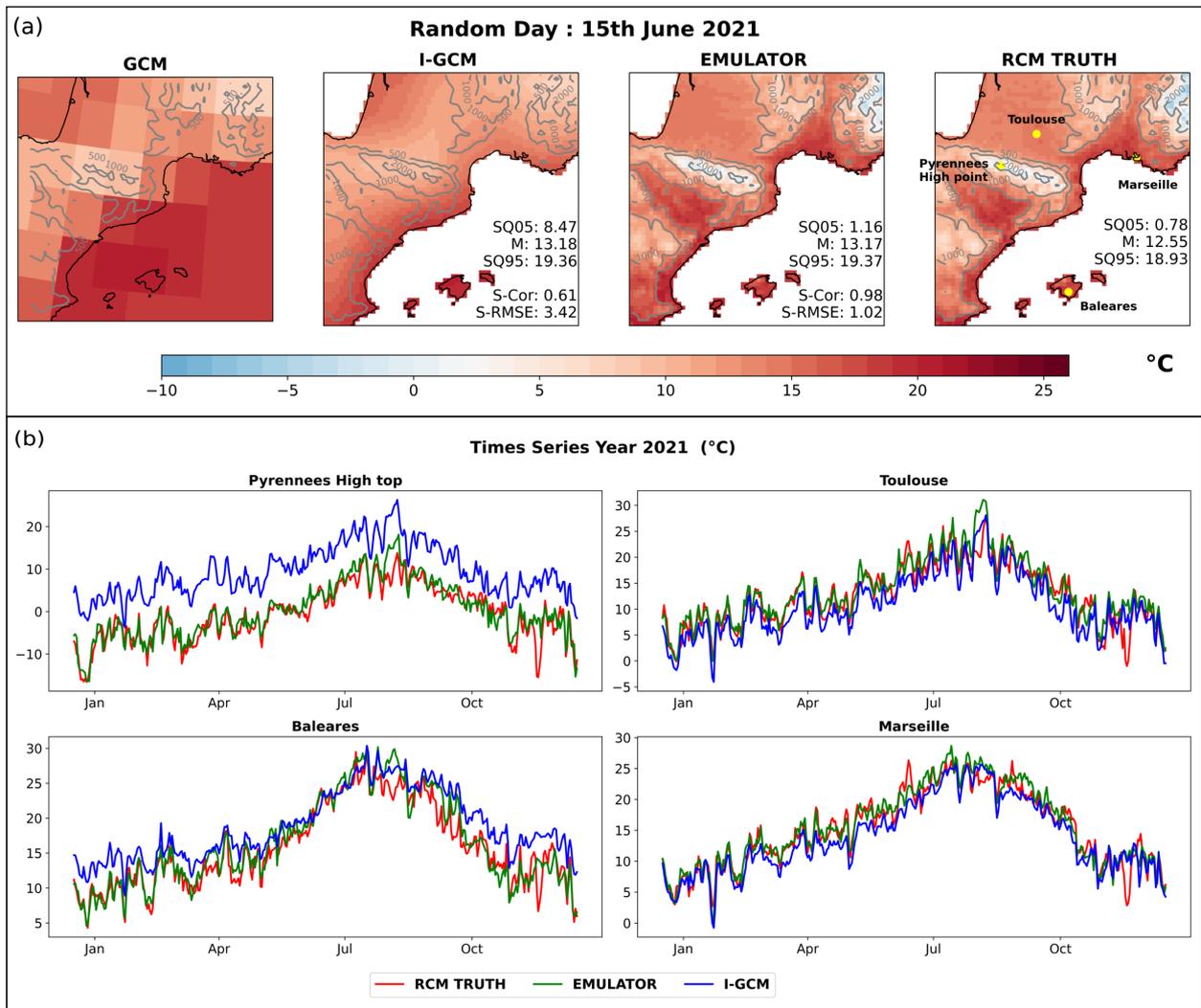


Figure 3.9: Randomly chosen illustration of the production of the emulator (in evaluation step 2) with inputs coming from the GCM: (a) near-surface temperature ($^{\circ}\text{C}$) at a random day over the target domain for the the GCM, the interpolated GCM, the emulator, and RCM truth and (b) random year time series ($^{\circ}\text{C}$) for 4 particular grid points.

fundamental differences between I-GCM and RCM, which the emulator captured very well. However, the correlation between the emulator and the RCM seems to be not as good as in the perfect model framework.

Figure 3.9b is a very good illustration of the RCM-GCM large scale de-correlation issue presented in Section 3.2.4. 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 Figure 3.9b, 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 (Figure 3.10c) than with the I-UPRCM (Figure 3.6c). 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 (Section 3.3.2), the spatial structure of the RCM truth is well reproduced by the emulator. The present climatology map (Figure 3.11a) 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

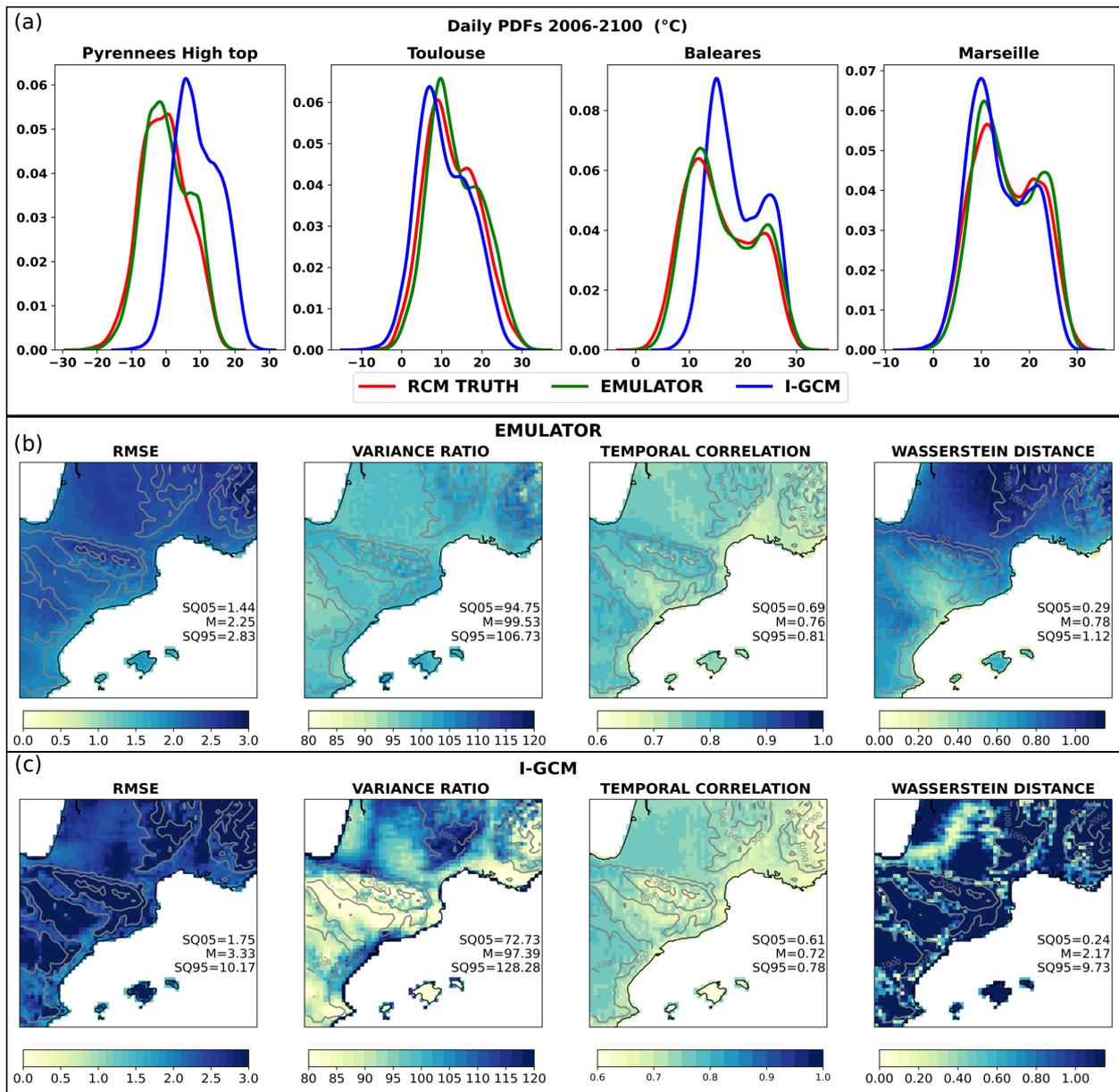


Figure 3.10: (a) Daily probability density functions from the RCM truth, the emulator (in evaluation step 2) and the I-GCM at 4 particular grid points over the whole simulation period. (b) (resp. (c)) Maps of performance scores of the emulator (resp. of the I-GCM) with respect to the RCM truth computed over the whole simulation period. For each map, the spatial mean and super-quantiles (SQ05 and SQ95) are added.

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 Section 3.3.2. The scores in Figure 3.10b present a good spatial homogeneity, exactly like in the previous section 3.6b.

The error map in Figure 3.11(a) shows that the emulator is warmer in the present climate than the RCM truth ($+0.96^{\circ}\text{C}$). 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 Figure 3.10b. 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.

The daily variability is well reproduced by the emulator. As mentioned before, the weaker RMSE (Figure 3.10b) 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 (Figure 3.10a) 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. Figures 3.11bc 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 Figures 3.11abc 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.

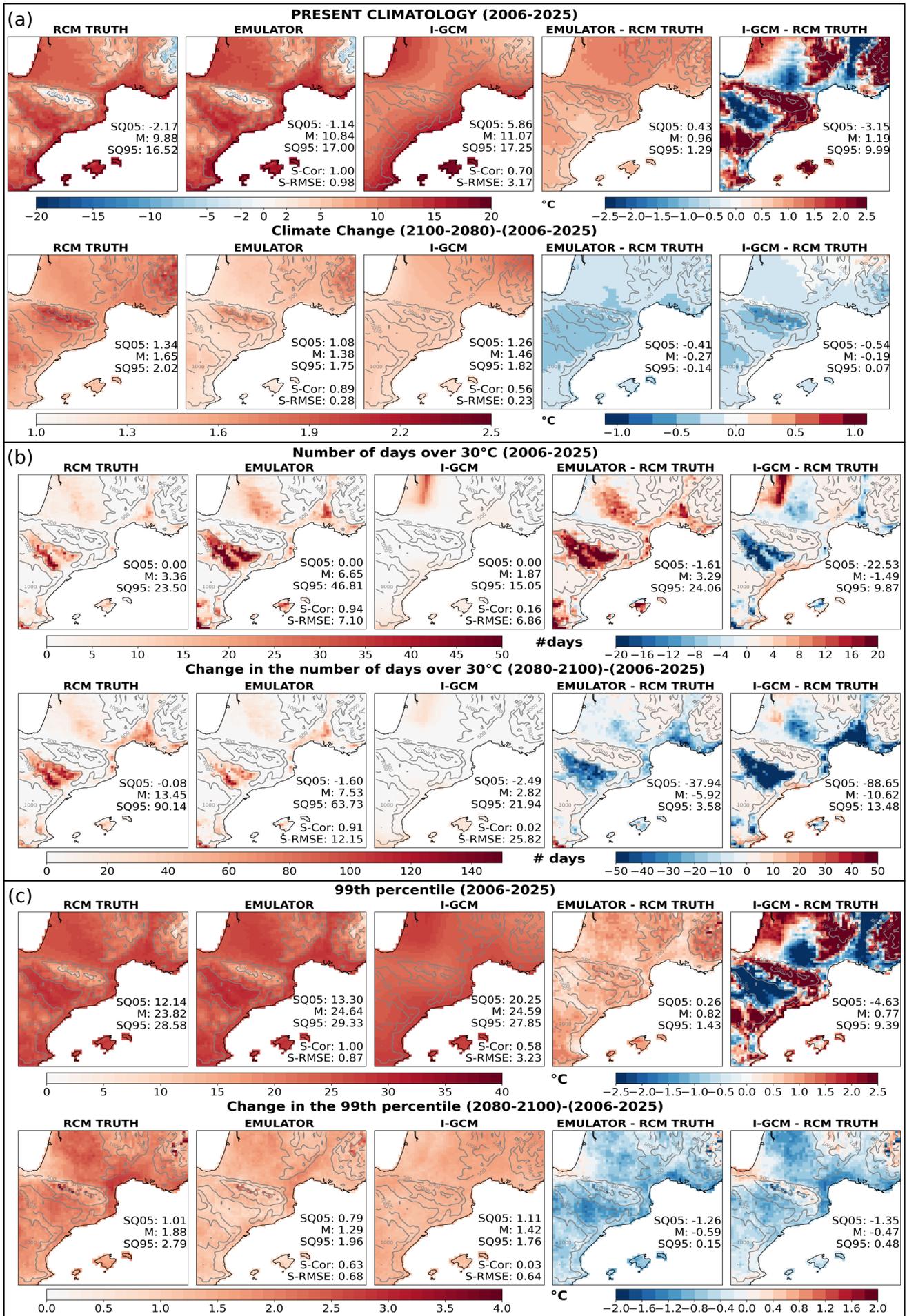


Figure 3.11: (a) Maps of long-term mean climatologies, (b) number of days over 30°C and (c) 99th percentile of daily near-surface temperature for a present-climate period (2006-2025) and for the climate change signal (2080-2100 minus 2006-2025), for the RCM truth, the emulator (in evaluation step 2) 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.

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.

3.4 Discussion

3.4.1 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 3.12 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., Figure 3.11).

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.

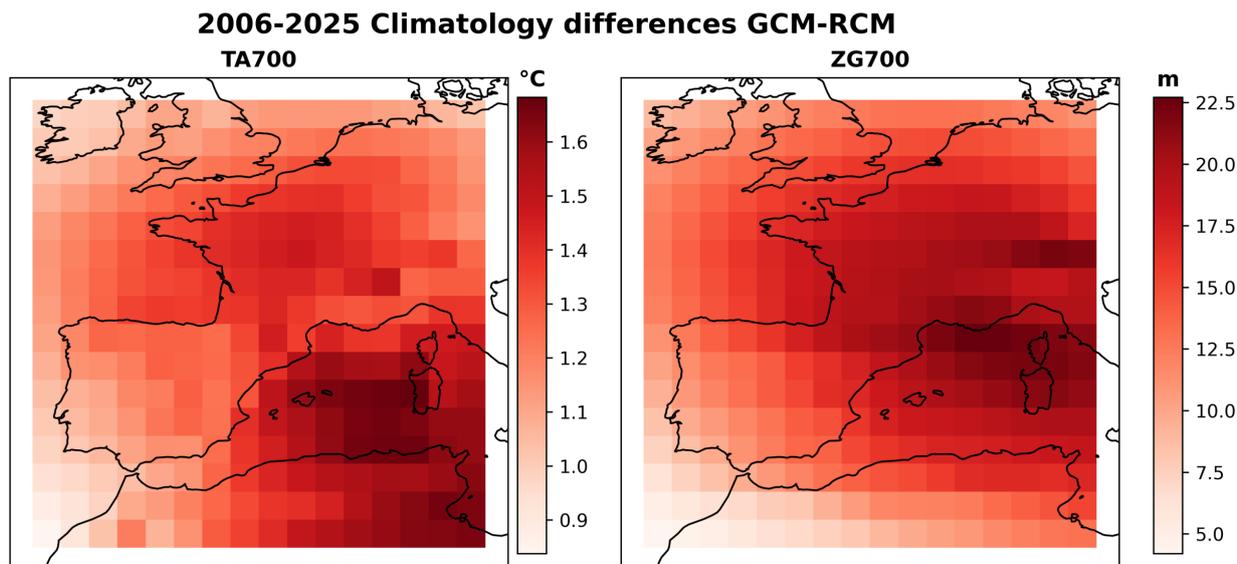


Figure 3.12: Present (2006-2025) climatology differences for the atmospheric temperature and geopotential at 700 hpa: CNRM-CM5 RCP45 minus ALADIN63 driven by CNRM-CM5 RCP45 upscaled on the GCM grid.

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.

3.4.2 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 3.3 and 3.4 in the supplementary material, this version being named 'Emul-Hist'.

The perfect model (Table 3.3) 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 subsection 3.4.4. Moreover, this bias 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 Figure 3.13). 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 3.4), but due to the lack of proper reference, it

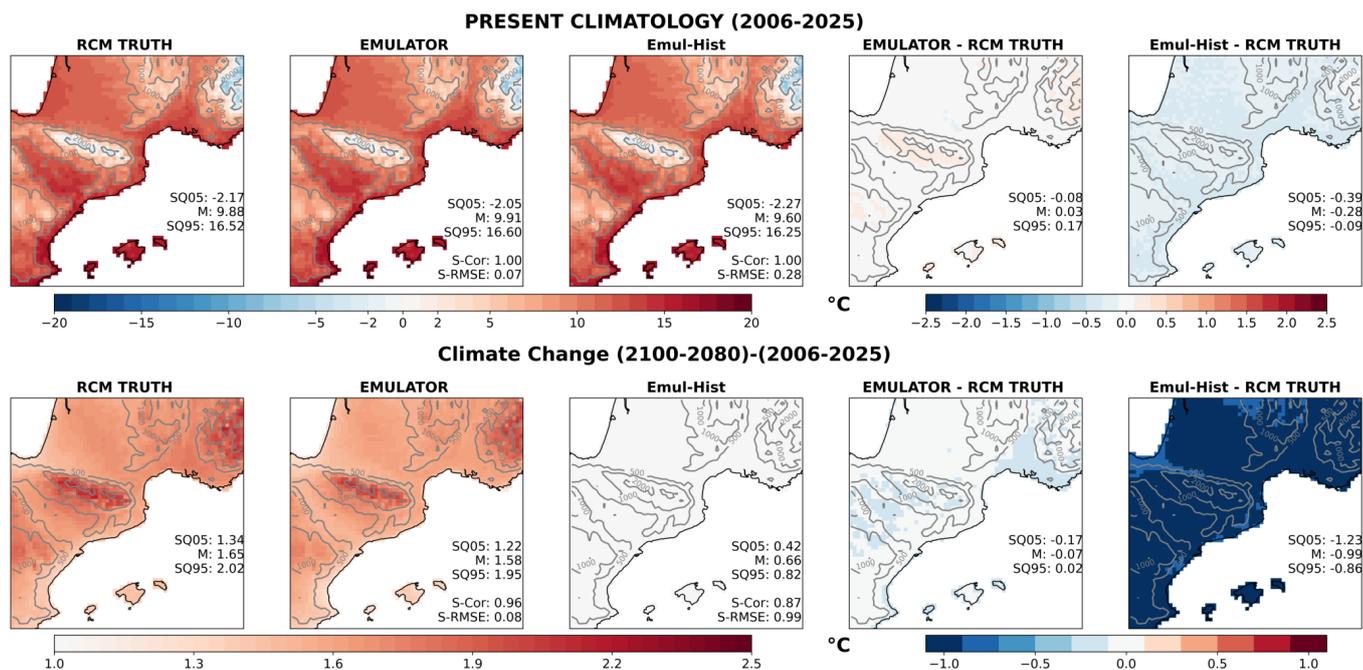


Figure 3.13: Maps of climatologies over present period (2006-2025) and climate change signal (2080-2100 versus 2006-2025), for the RCM truth, the Emulator presented in section 3.3 and the Emul-Hist. The two last columns correspond to the error maps of the Emulator and Emul-Hist with respect to the RCM 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.

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.

3.4.3 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 is not always available, and leads to several computations due to the different preprocessing steps described in Section 3.2. For this reason, we tried to build an emulator with fewer inputs in X including only the variables from Table 3.2 at 700hpa and removing the solar and ozone forcings from Z . The results are reported in tables 3.3 and 3.4. 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 3.3), 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 prob-

ably 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 sections 3.2.2 and 3.2.5. 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. 3.2.5). 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 3.3 and 3.4, in appendix C. 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 sections 3.3.3 and 3.4.1). 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 Section 3.2.5. 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 3.3 and 3.4. 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.

3.4.4 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 Section 3.2) is trained 31 times, resulting in 31 emulators. The results of this experiment are illustrated in Figure 3.14 and summarised in tables 3.3 and 3.4. The results of Section 3.3 are based on a randomly picked emulator, mentioned as the main emulator in appendix and plotted in darker green in Figure 3.14. All the emulators from the Monte Carlo experiment show consistent results with the ones presented in Section 3.3. The RMSEs (and correlated scores such as variance) are really close to each other ($[0.8; 0.86]^{\circ}C$), 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

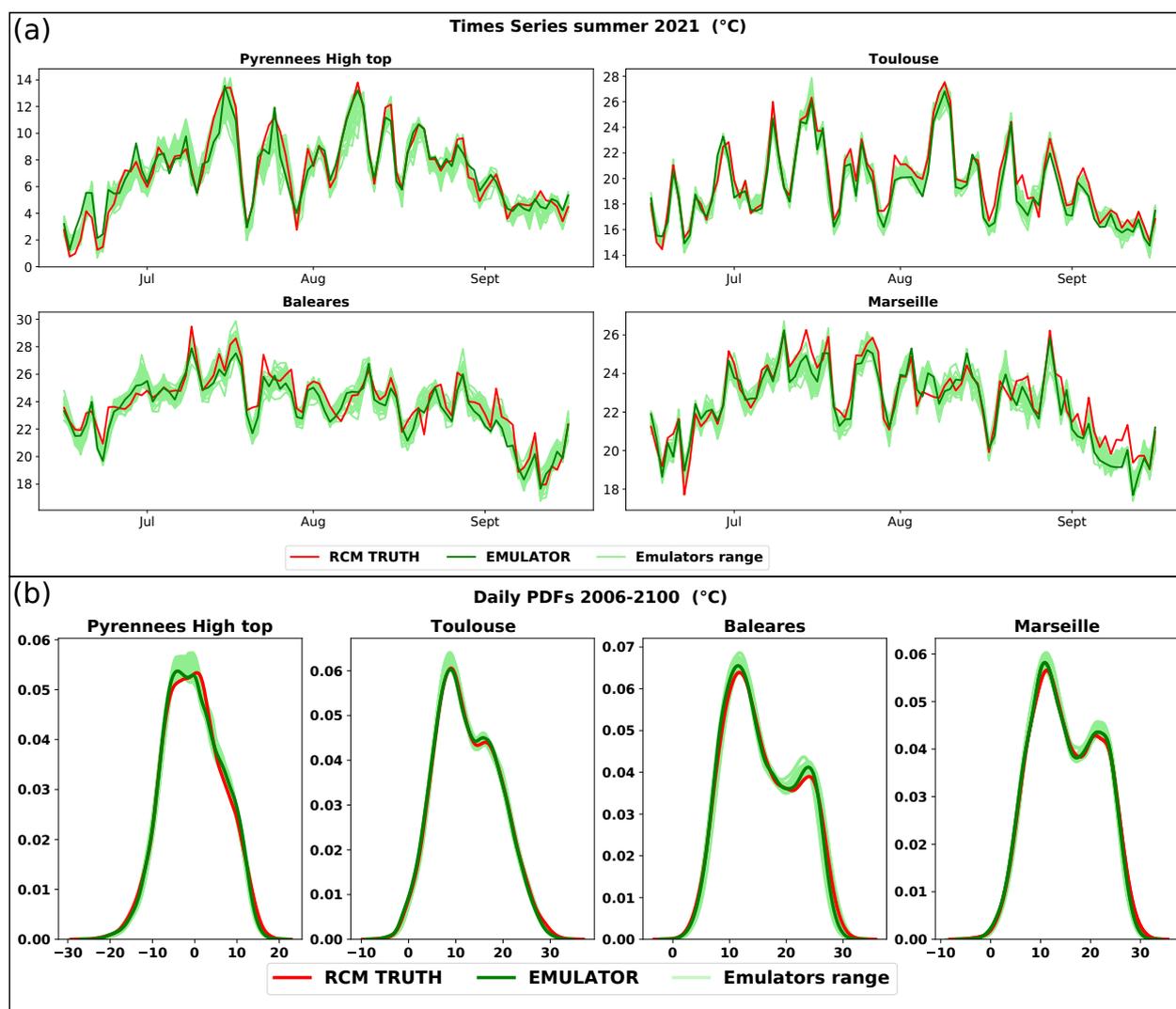


Figure 3.14: Illustration of the results for the Monte Carlo experiment. (a) present a year time series for 4 given points and (b) the pdfs on the whole serie. The red line refers to the RCM truth, the dark green line is the main emulator, and the light green lines are the 30 emulators from the Monte Carlo experiment.

tables). For instance, in the UPRCM evaluation step, the average error in the future climatology varies from .003 to .139°C. The results from the GCM world evaluation step (Table 3.4) 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 (Section 3.3), which prove their robustness.

3.5 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 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.

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.

3.6 Appendices

3.6.1 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 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); 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

$$T(F_{T,Lr}(x)) = F_{T,Hr}(x). \quad (3.10)$$

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

$$T(u) = F_{T,Hr}(F_{T,Lr}^{-1}(u)). \quad (3.11)$$

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

$$T(F_{E,Lr}(x)) = F_{E,Hr}(x), \quad (3.12)$$

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 :

$$F_{E,Hr}(x) = F_{T,Hr}(F_{T,Lr}^{-1}(F_{E,Lr}(x))). \quad (3.13)$$

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 3.2), but by taking a single grid point for the 2D variables. The predicted high resolution temperature is then :

$$\hat{Y} = \alpha_0 + \sum_{i=0}^M \alpha_i P_i, \quad (3.14)$$

with M the number of predictors.

Both CDFt and MLR emulators are trained in the same conditions as the UNet

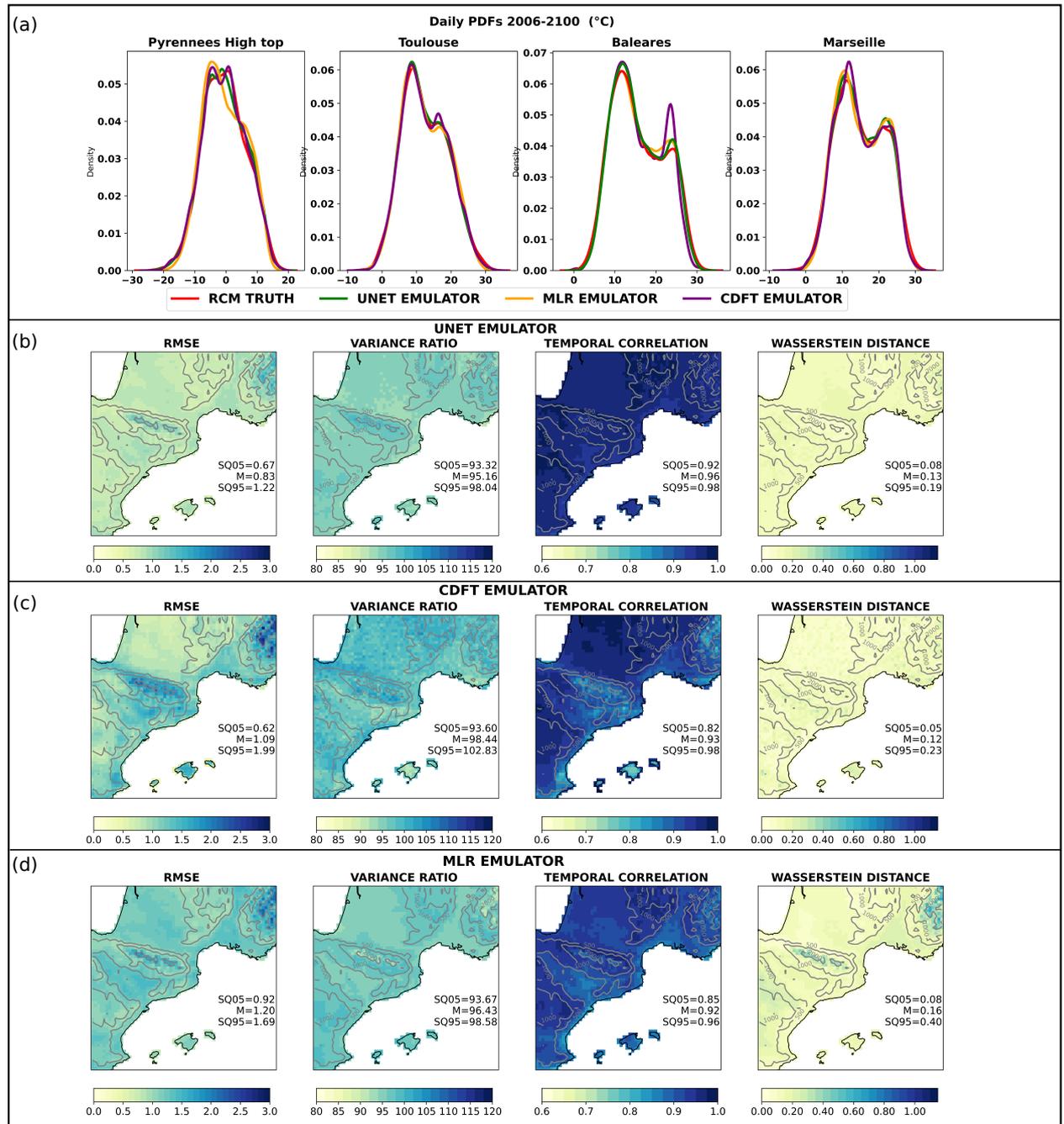


Figure 3.15: (a) Daily probability density functions from the RCM truth, the UNet, the CDFt and the MLR emulators at 4 particular grid points over the whole simulation period. (b) (resp. (c and d)) Maps of performance scores of the UNet emulator (resp. the CDFt and MLR) with respect to the RCM truth computed over the whole simulation period. For each map, the spatial mean and super-quantiles (SQ05 and SQ95) are added.

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 Figures 3.15 and 3.16, where the two simpler emulators are compared to the UNet Emulator used in the study. Figure 3.15 is showing the performances of the three emulators regarding the performance metrics defined in section 3.2.6. 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 3.16 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 Figure 3.16.

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.

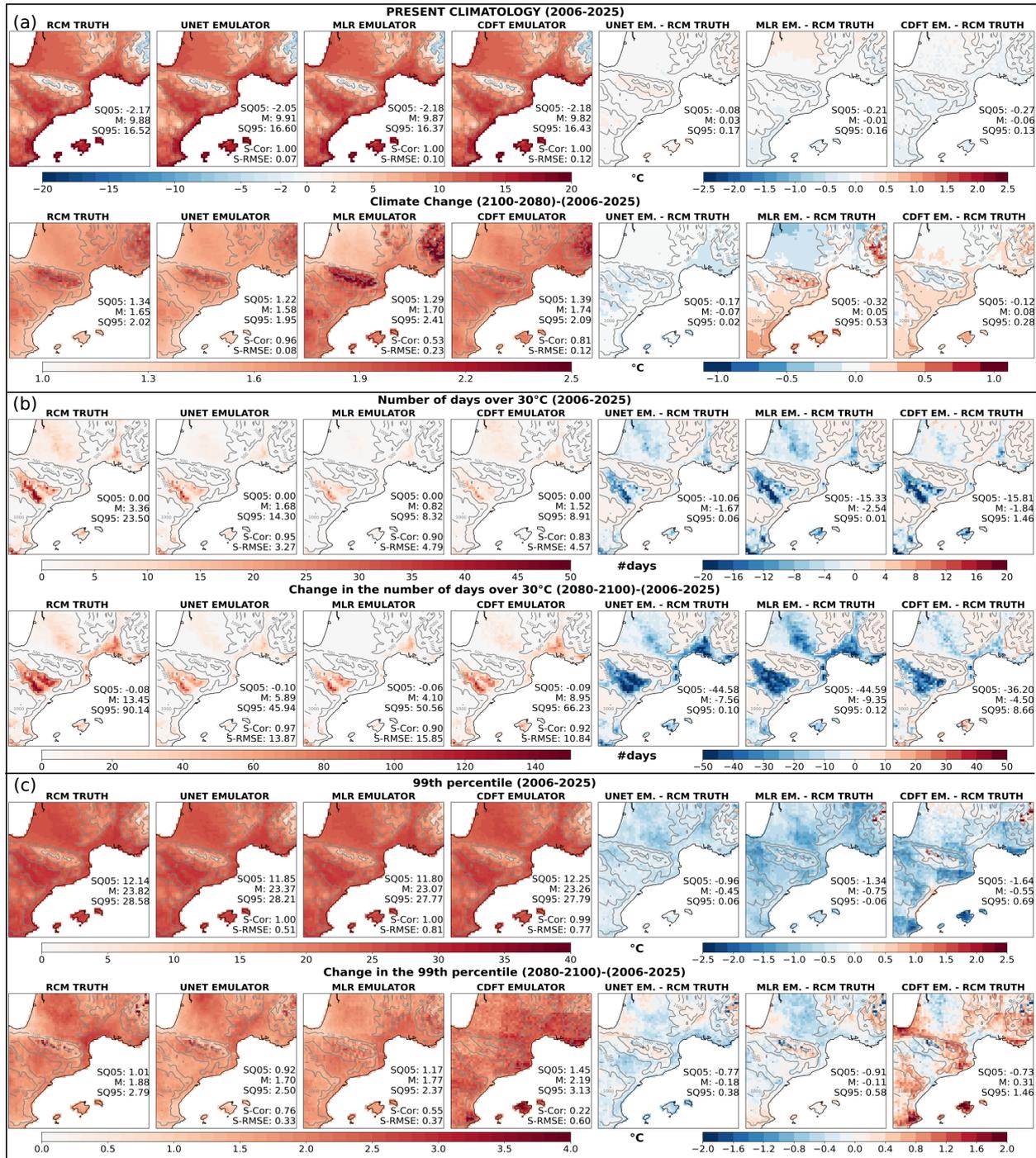


Figure 3.16: (a) Maps of climatologies, (b) Number of days over 30°C and (c) the 99th percentile of daily temperature, for the present-climate period (2006-2025) and for the climate change signal (2080-2100 minus 2006-2025), for the RCM truth, the UNet, the MLR and CDFT emulators in perfect model mode. 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.

3.6.2 Description of the UNet-emulator architecture

We describe here with more details the Figure 3.2, 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 Figure 3.2). 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 Figure 3.2.
- 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.

3.6.3 Supplementary Material: Summary tables

We present here the summary tables of the different tests performed with the UNet-based emulators and for the two steps of the emulator evaluation.

| | RMSE | | | ACC | | | Variance ratio | | | Wasserstein distance | | |
|--|------------|-------------|-------------|------------|------------|--------------|----------------|------------|------------|----------------------|-------------|-------------|
| | SQ 05 | MEAN | SQ 95 | SQ 05 | MEAN | SQ 95 | SQ 05 | MEAN | SQ 95 | SQ 05 | MEAN | SQ 95 |
| MAX | 0.70 (E6) | 0.86 (E6) | 1.26 (E30) | 0.93 (E11) | 0.97 (E11) | 0.98 (E8) | 0.95 (E9) | 0.98 (E9) | 1.00 (E8) | 0.14 (E6) | 0.24 (E6) | 0.36 (E13) |
| MEAN | 0.67 | 0.82 | 1.20 | 0.92 | 0.96 | 0.98 | 0.93 | 0.96 | 0.98 | 0.08 | 0.14 | 0.26 |
| MIN | 0.64 (E19) | 0.8 (E11) | 1.17 (E19) | 0.91 (E13) | 0.96 (E6) | 0.98 (E6) | 0.89 (E6) | 0.92 (E6) | 0.95 (E20) | 0.04 (E9) | 0.11 (E8) | 0.18 (E28) |
| Main Emulator | 0.67 | 0.83 | 1.22 | 0.92 | 0.96 | 0.98 | 0.93 | 0.95 | 0.98 | 0.08 | 0.13 | 0.19 |
| UPRCM | 0.43 | 2.10 | 8.1 | 0.79 | 0.95 | 0.99 | 0.68 | 0.92 | 1.16 | 0.14 | 1.82 | 7.88 |
| Cheap | 0.92 | 1.11 | 1.34 | 0.87 | 0.93 | 0.96 | 0.94 | 0.98 | 1.02 | 0.08 | 0.15 | 0.28 |
| Emul-Hist | 1.16 | 1.35 | 1.70 | 0.90 | 0.95 | 0.97 | 0.88 | 0.93 | 0.96 | 0.74 | 0.87 | 1.01 |
| No_Smoothing | 0.62 | 0.76 | 1.2 | 0.93 | 0.97 | 0.98 | 0.95 | 0.96 | 0.98 | 0.07 | 0.13 | 0.23 |
| TAS_included | 0.36 | 0.55 | 0.98 | 0.96 | 0.98 | 0.99 | 0.96 | 0.98 | 0.99 | 0.05 | 0.08 | 0.15 |
| PRESENT PERIOD 2006-2025 | | | | | | | | | | | | |
| Number of days over 30°C | | | | | | | | | | | | |
| RCM truth values | -2.166 | 9.879 | 16.518 | 0.000 | 3.356 | 23.500 | 12.138 | 23.817 | 28.580 | | | |
| Emul vs RCM truth | SpaCor | SQ 05 | Mean | SQ 95 | SpaCor | SQ 05 | Mean | SQ 95 | SpaCor | SQ 05 | Mean | SQ 95 |
| MAX | 1.0 (E1) | -0.06 (E7) | 0.06 (E16) | 0.31 (E22) | 0.96 (E11) | -5.27 (E22) | -0.57 (E22) | 1.72 (E9) | 1.00 (E3) | -0.61 (E22) | -0.19 (E22) | 0.32 (E9) |
| MEAN | 1.00 | -0.17 | 0 | 0.17 | 0.95 | -8.35 | -1.34 | 0.32 | 1.00 | -0.87 | -0.38 | 0.07 |
| MIN | 1.0 (E1) | -0.30 (E1) | -0.07 (E12) | 0.07 (E4) | 0.93 (E6) | -12.22 (E6) | -1.92 (E6) | 0.03 (E20) | 1.00 (E9) | -1.11 (E2) | -0.6 (E6) | -0.19 (E20) |
| Main Emulator | 1.00 | -0.08 | 0.03 | 0.17 | 0.95 | -10.06 | -1.68 | 0.06 | 1.00 | -0.96 | -0.45 | 0.06 |
| UPRCM | 0.81 | -2.88 | 0.53 | 8.02 | 0.41 | -22.15 | -2.80 | 0.08 | 0.77 | -4.22 | -0.32 | 7.4 |
| Cheap | 1.00 | -0.16 | 0.03 | 0.27 | 0.92 | -7.77 | -1.59 | 0.09 | 1.00 | -0.84 | -0.30 | 0.19 |
| Emul-Hist | 1.00 | -0.39 | -0.28 | -0.09 | 0.83 | -16.93 | -2.79 | 0.02 | 1.00 | -1.66 | -0.98 | -0.49 |
| No_Smoothing | 1.00 | -0.14 | -0.01 | 0.20 | 0.96 | -8.59 | -1.32 | 0.04 | 1.00 | -0.77 | -0.31 | 0.14 |
| TAS_included | 1.00 | -0.12 | -0.03 | 0.06 | 0.97 | -6.24 | -0.98 | 0.07 | 1.00 | -0.65 | -0.25 | 0.07 |
| FUTURE PERIOD 2080-2100 | | | | | | | | | | | | |
| Number of days over 30°C | | | | | | | | | | | | |
| RCM truth values | -0.19 | 11.53 | 17.89 | 0 | 16.80 | 112.29 | 14.04 | 25.7 | 30.55 | | | |
| Emul vs RCM truth | SpaCor | SQ 05 | Mean | SQ 95 | SpaCor | SQ 05 | Mean | SQ 95 | SpaCor | SQ 05 | Mean | SQ 95 |
| MAX | 1.0 (E1) | -0.16 (E7) | 0 (E16) | 0.18 (E22) | 0.98 (E3) | -30.67 (E11) | -5.75 (E11) | 0.36 (E22) | 1.00 (E26) | -0.89 (E22) | -0.35 (E22) | 0.17 (E22) |
| MEAN | 1.00 | -0.27 | -0.08 | 0.07 | 0.97 | -45.22 | -8.06 | 0.04 | 1.00 | -1.15 | -0.57 | -0.06 |
| MIN | 1.0 (E1) | -0.39 (E13) | -0.14 (E12) | -0.03 (E4) | 0.95 (E13) | -61.91 (E6) | -10.63 (E6) | 0 (E30) | 1.00 (E1) | -1.44 (E2) | -0.83 (E6) | -0.24 (E6) |
| Main Emulator | 1.00 | -0.16 | -0.04 | 0.09 | 0.98 | -52.41 | -9.23 | 0.01 | 1.00 | -1.25 | -0.63 | -0.03 |
| UPRCM | 0.80 | -2.87 | 0.5 | 7.79 | 0.36 | -107.14 | -15.35 | 1.66 | 0.78 | -4.42 | -0.34 | 7.29 |
| Cheap | 1.00 | -0.37 | -0.14 | 0.09 | 0.96 | -46.35 | -9.08 | 0.01 | 1.00 | -1.45 | -0.73 | -0.11 |
| Emul-Hist | 1.00 | -1.46 | -1.27 | -1.1 | 0.80 | -100.01 | -16.02 | 0 | 1.00 | -3.74 | -2.48 | -1.67 |
| No_Smoothing | 1.00 | -0.25 | -0.13 | 0.06 | 0.98 | -45.37 | -8.17 | 0 | 1.00 | -1.04 | -0.58 | -0.15 |
| TAS_included | 1.00 | -0.14 | -0.04 | 0.04 | 0.99 | -26.54 | -5.15 | 0.06 | 1.00 | -0.72 | -0.33 | 0.01 |
| CLIMATE CHANGE 2080-2100 vs 2006-2025 | | | | | | | | | | | | |
| Number of days over 30°C | | | | | | | | | | | | |
| RCM truth values | 1.34 | 1.65 | 2.02 | -0.08 | 13.45 | 90.14 | 1.01 | 1.88 | 2.79 | | | |
| Emul vs RCM truth | SpaCor | SQ 05 | Mean | SQ 95 | SpaCor | SQ 05 | Mean | SQ 95 | SpaCor | SQ 05 | Mean | SQ 95 |
| MAX | 0.96 (E26) | -0.16 (E28) | -0.06 (E9) | 0.05 (E1) | 0.98 (E3) | -28.17 (E11) | -4.89 (E11) | 0.31 (E22) | 0.84 (E26) | -0.57 (E11) | -0.06 (E11) | 0.48 (E11) |
| MEAN | 0.95 | -0.19 | -0.08 | 0.01 | 0.97 | -39.67 | -6.7 | 0.1 | 0.80 | -0.73 | -0.19 | 0.31 |
| MIN | 0.95 (E1) | -0.21 (E22) | -0.11 (E6) | -0.02 (E6) | 0.94 (E13) | -52.36 (E6) | -8.71 (E6) | 0.01 (E29) | 0.72 (E1) | -0.84 (E16) | -0.28 (E10) | 0.22 (E16) |
| Main Emulator | 0.96 | -0.18 | -0.07 | 0.02 | 0.97 | -44.58 | -7.56 | 0.1 | 0.76 | -0.77 | -0.18 | 0.38 |
| UPRCM | 0.82 | -0.3 | -0.03 | 0.15 | 0.31 | -87.4 | -12.56 | 1.54 | 0.62 | -0.81 | -0.03 | 0.64 |
| Cheap | 0.94 | -0.3 | -0.17 | 0.07 | 0.96 | -41.43 | -7.47 | 0.11 | 0.73 | -1.03 | -0.43 | 0.21 |
| Emul-Hist | 0.87 | -1.23 | -0.99 | -0.86 | 0.66 | -84.12 | -13.23 | 0.07 | 0.55 | -2.34 | -1.5 | -0.92 |
| No_Smoothing | 0.96 | -0.25 | -0.11 | -0.03 | 0.97 | -38.92 | -6.86 | 0.08 | 0.84 | -0.81 | -0.27 | 0.12 |
| TAS_included | 0.98 | -0.09 | -0.01 | 0.07 | 0.98 | -23.52 | -4.18 | 0.16 | 0.87 | -0.51 | -0.08 | 0.36 |

Table 3.3: Summary of the results obtained in the 1st step of evaluation (UPRCM world). The scores are the one used to summarize the maps on figures 3.6,3.7, for the different versions of the emulator tested in the paper. The first line (RCM truth) is the RCM absolute values (except for the first section). On each sections, the three first lines give the mean and the range of the ensemble of emulators described in Section 3.4.4. For the ensemble minimums and maximums the name (E1 to E30) of the emulator corresponding to the value is given in brackets. “Main Emulator” is the one used for the results section (3.3). “Cheap” emulator is the one including only the variables at 700hpa described in Section 3.4.3. “Emul-Hist” is the emulator trained only on the historical period (section 3.4.2). “NO_smoothing” and “TAS_included” correspond to the emulator trained with no prior smoothing of the data, and the one including TAS among the inputs (Section 3.4.3).

| | RMSE | | ACC | | Variance ratio | | Wasserstein distance | | | | |
|--|-------------|--------------|--------------|--------------|----------------|--------------|----------------------|-------------|--------------|--------------|-------------|
| | SQ 05 | Mean | SQ05 | Mean | SQ 05 | Mean | SQ 05 | Mean | | | |
| MAX | 1.44 (Em30) | 2.27 (Em7) | 2.85 (Em22) | 0.70 (Em8) | 0.76 (Em8) | 0.81 (Em2) | 0.96 (Em8) | 1.01 (Em22) | 0.29 (Em2) | 0.82 (Em7) | 1.23 (Em27) |
| MEAN | 1.40 | 2.24 | 2.80 | 0.69 | 0.76 | 0.81 | 0.93 | 0.99 | 0.21 | 0.75 | 1.11 |
| MIN | 1.38 (Em14) | 2.20 (Em12) | 2.76 (Em4) | 0.69 (Em7) | 0.75 (Em10) | 0.81 (Em1) | 0.90 (Em6) | 0.96 (Em6) | 0.13 (Em12) | 0.66 (Em12) | 1.00 (Em30) |
| Main Emulator | 1.46 | 2.25 | 2.83 | 0.69 | 0.76 | 0.81 | 0.95 | 1.00 | 0.29 | 0.78 | 1.12 |
| GCM | 1.76 | 3.34 | 10.17 | 0.61 | 0.72 | 0.78 | 0.73 | 0.97 | 0.24 | 2.17 | 9.73 |
| Cheap | 1.48 | 2.25 | 2.83 | 0.69 | 0.75 | 0.80 | 0.93 | 0.98 | 0.46 | 0.84 | 1.14 |
| Emul-Hist | 1.54 | 2.16 | 2.75 | 0.68 | 0.75 | 0.80 | 0.89 | 0.95 | 0.16 | 0.28 | 0.55 |
| No.Smoothing | 1.50 | 2.29 | 2.96 | 0.59 | 0.73 | 0.80 | 0.83 | 0.94 | 0.18 | 0.53 | 0.92 |
| TAS_included | 1.81 | 2.48 | 4.13 | 0.65 | 0.73 | 0.79 | 0.83 | 1.03 | 0.16 | 0.84 | 2.93 |
| PRESENT PERIOD 2006-2025 | | | | | | | | | | | |
| CLIMATOLOGY | | | | | | | | | | | |
| Number of days over 30°C | | | | | | | | | | | |
| RCM truth values | -2.17 | 9.90 | 16.52 | 0.00 | 3.36 | 23.50 | 12.14 | 23.82 | 28.58 | | |
| Emul vs RCM truth | SpaCor | SQ 05 | Mean | SQ05 | Mean | SQ05 | SpaCor | SQ 05 | Mean | SQ05 | |
| MAX | 1.00 (Em3) | 0.41 (Em19) | 1.00 (Em7) | 1.41 (Em27) | -1.35 (Em22) | 5.10 (Em22) | 34.22 (Em22) | 1.00 (Em7) | 0.3 (Em22) | 0.96 (Em22) | 1.63 (Em5) |
| MEAN | 1.00 | 0.27 | 0.93 | 1.31 | -2.1 | 3.08 | 23.78 | 1.00 | -0.01 | 0.73 | 1.34 |
| MIN | 1.00 (Em1) | 0.13 (Em27) | 0.83 (Em12) | 1.20 (Em12) | -3.10 (Em21) | 1.64 (Em6) | 13.31 (Em20) | 1.00 (Em27) | -0.35 (Em27) | 0.50 (Em20) | 1.05 (Em20) |
| Main Emulator | 1.00 | 0.43 | 0.96 | 1.29 | -1.61 | 3.23 | 24.06 | 1.00 | 0.26 | 0.83 | 1.43 |
| GCM | 0.70 | -3.16 | 1.19 | 9.99 | -22.53 | -1.49 | 9.95 | 0.58 | -4.63 | 0.77 | 9.38 |
| Cheap | 1.00 | 0.61 | 1.05 | 1.40 | -4.775 | 0.33 | 17.63 | 1.00 | 0.032 | 0.61 | 1.28 |
| Emul-Hist | 1.00 | 0.10 | 0.66 | 0.98 | -10.49 | -2.05 | 0.12 | 1.00 | -0.98 | -0.17 | 0.46 |
| No.Smoothing | 1.00 | -0.05 | 0.68 | 1.20 | -5.26 | 1.95 | 19.75 | 0.98 | -0.97 | 0.52 | 2.39 |
| TAS_included | 0.97 | -1.38 | 0.55 | 3.12 | -5.22 | 3.86 | 33.57 | 0.95 | -1.77 | 0.60 | 2.88 |
| FUTURE PERIOD 2080-2100 | | | | | | | | | | | |
| CLIMATOLOGY | | | | | | | | | | | |
| Number of days over 30°C | | | | | | | | | | | |
| RCM truth values | -0.19 | 11.53 | 17.89 | 0.00 | 16.80 | 112.29 | 14.04 | 25.70 | 30.55 | | |
| Emul vs RCM truth | SpaCor | SQ 05 | Mean | SQ05 | Mean | SQ05 | SpaCor | SQ 05 | Mean | SQ05 | |
| MAX | 1.00 (Em22) | 0.17 (Em11) | 0.73 (Em7) | 1.18 (Em16) | -13.17 (Em22) | 0.86 (Em22) | 32.27 (Em22) | 1.00 (Em4) | -0.33 (Em22) | 0.41 (Em22) | 1.25 (Em5) |
| MEAN | 1.00 | 0.01 | 0.65 | 1.07 | -26.27 | -3.05 | 12.458 | 1.00 | -0.70 | 0.14 | 0.99 |
| MIN | 1.00 (Em1) | -0.15 (Em27) | 0.56 (Em12) | 0.97 (Em30) | -39.84 (Em20) | -6.69 (Em20) | 0.42 (Em20) | 0.99 (Em27) | -1.00 (Em2) | -0.14 (Em20) | 0.74 (Em20) |
| Main Emulator | 1.00 | 0.16 | 0.69 | 1.11 | -25.33 | -2.62 | 12.25 | 1.00 | -0.58 | 0.23 | 1.08 |
| GCM | 0.70 | -3.28 | 1.00 | 9.59 | -109.27 | -12.11 | 21.34 | 0.56 | -5.19 | 0.30 | 9.04 |
| Cheap | 1.00 | 0.3 | 0.70 | 1.03 | -40.66 | -8.31 | 0.18 | 1.00 | -1.14 | -0.29 | 0.69 |
| Emul-Hist | 1.00 | -0.96 | -0.50 | -0.10 | -99.23 | -15.87 | 0.03 | 0.99 | -3.26 | -1.93 | -0.77 |
| No.Smoothing | 0.997 | -0.64 | 0.22 | 0.77 | -86.63 | -9.42 | 23.84 | 0.98 | -2.15 | -0.35 | 1.59 |
| TAS_included | 0.97 | -1.43 | 0.39 | 2.86 | -63.43 | 2.64 | 87.53 | 1.00 | -2.13 | 0.17 | 2.33 |
| CLIMATE CHANGE 2080-2100 vs 2006-2025 | | | | | | | | | | | |
| Annual Mean Delta TAS | | | | | | | | | | | |
| Number of days over 30°C | | | | | | | | | | | |
| RCM truth values | 1.33 | 1.65 | 2.02 | -0.08 | 13.45 | 90.14 | 1.01 | 1.88 | 2.79 | | |
| Emul vs RCM truth | SpaCor | SQ 05 | Mean | SQ05 | Mean | SQ05 | SpaCor | SQ 05 | Mean | SQ05 | |
| MAX | 0.91 (Em28) | -0.39 (Em28) | -0.26 (Em28) | -0.13 (Em11) | -28.15 (Em22) | -4.24 (Em22) | 9.98 (Em22) | 0.76 (Em12) | -1.06 (Em11) | -0.51 (Em11) | 0.22 (Em4) |
| MEAN | 0.89 | -0.44 | -0.28 | -0.155 | -37.72 | -6.13 | 3.56 | 0.666 | -1.19 | -0.59 | 0.14 |
| MIN | 0.87 (Em22) | -0.47 (Em6) | -0.31 (Em6) | -0.18 (Em10) | -49.90 (Em6) | -8.33 (Em20) | 0.27 (Em20) | 0.47 (Em13) | -1.38 (Em13) | -0.66 (Em30) | 0.02 (Em12) |
| Main Emulator | 0.89 | -0.41 | -0.27 | -0.14 | -37.94 | -5.92 | 3.58 | 0.63 | -1.26 | -0.59 | 0.15 |
| GCM | 0.56 | -0.54 | -0.19 | 0.07 | -88.65 | -10.62 | 13.54 | 0.04 | -1.36 | -0.47 | 0.48 |
| Cheap | 0.91 | -0.49 | -0.34 | -0.20 | -47.58 | -8.64 | 0.11 | 0.66 | -1.60 | -0.91 | -0.24 |
| Emul-Hist | 0.73 | -1.44 | -1.16 | -0.99 | -94.60 | -13.83 | 0.18 | 0.36 | -2.59 | -1.77 | -0.96 |
| No.Smoothing | 0.73 | -0.76 | -0.46 | -0.27 | -80.50 | -11.37 | 9.01 | 0.60 | -1.56 | -0.87 | -0.19 |
| TAS_included | 0.68 | -0.44 | -0.17 | 0.10 | -59.36 | -1.22 | 56.57 | 0.29 | -1.31 | -0.43 | 0.43 |

Table 3.4: Summary of the results obtained in the 2nd step of evaluation (GCM world). The scores are the one used to summarize the maps on figures 3.6,3.7, for the different versions of the emulator tested in the paper. The first line (RCM truth) is the RCM absolute values (except for the first section). On each sections, the three first lines give the mean and the range of the ensemble of emulators described in Section 3.4.4. For the ensemble minimums and maximums the name (E1 to E30) of the emulator corresponding to the value is given in brackets. “Main Emulator” is the one used for the results section (3.3). “Cheap” emulator is the one including only the variables at 700hpa described in Section 3.4.3. “Emul-Hist” is the emulator trained only on the historical period (Section 3.4.2). “NO_smoothing” and “TAS_included” correspond to the emulator trained with no prior smoothing of the data, and the one including TAS among the inputs (Section 3.4.3).

Chapter 4

Study of the RCM-Emulator applicability to GCM ensembles

This chapter evaluates the robustness of the emulator introduced in Chapter 3 to various input sources. To simplify the reading, I tried to make it independent, implying that some parts recall and summarise the previous chapter. Moreover, I work with multiple RCM climate simulations and their driving GCM simulations. I have tried to simplify the naming and clearly identify the simulations I use at the beginning of each section. Nevertheless, It might be difficult to follow for a reader non-familiar with those climate models, and I apologize in advance for that.

4.1 Introduction

Evaluating and understanding the uncertainties associated with climate projections is crucial to deliver reliable information about climate change considering all possible futures. There are three main sources of uncertainty. The chaotic nature of the climate system induces a strong natural variability reproduced in climate models and which needs to be quantified. The future evolution of human activities is another source of uncertainty that the scientific community addresses with different scenarios of greenhouse gas emissions along the 21st century (Meinshausen et al, 2011). Finally, each climate model is a possible representation of the Earth System. They are built by different teams in different research centres worldwide with different approaches to solving the same problems. As there is no “best” model, the variety of models is essential, and the robustness of a result can come from the agreement of the majority of these models. Multiple studies are developing methods to make the best out of this large ensemble of GCM simulations (Knutti, 2010; Ribes et al, 2017). Thus, to provide reliable information about climate change, the scientific community cooperates to build large simulation ensembles with multiple Global Climate Models (GCMs) driven by different scenarios. Several *members* from the same GCM/Scenario pair are produced to consider the internal variability.

Regional Climate Models (RCM) are high-resolution limited area models nested in a GCM simulation. They are used to downscale GCM simulations, which are too coarse to reveal the consequences of climate change at local scales. RCMs are an additional source of uncertainty. Ideally, it is necessary to downscale with multiple RCMs each simulation from the GCM large ensembles to consider all possible futures and create reliable information about the local impacts of global climate warming. However, the high cost of RCMs only allows for downscaling some simulations, and the needed exploration of uncertainties is limited. RCMs ensembles are 4-dimensional matrices, and the driving scenarios, the choices of the GCM and the RCM, and the number of members are the four entries. CORDEX, is an international initiative which aims to coordinate the design and filling of such matrices over specific regional domains. Figure 4.1 illustrates the EURO-CORDEX domain and matrix, which results from about ten years of international cooperation.

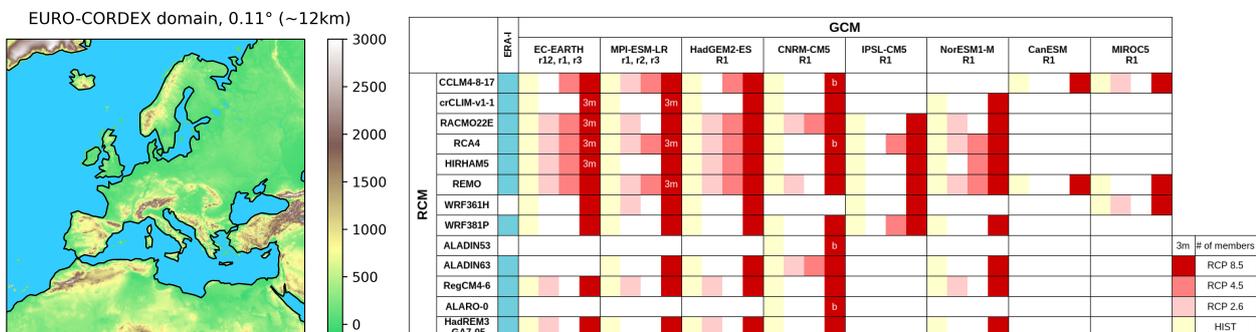


Figure 4.1: EURO-CORDEX domain and matrix illustration.

To tackle this issue, Chapter 3 introduces the RCM-Emulator, a hybrid downscaling approach combining both statistical and dynamical downscaling. The emulator

learns the downscaling function included in the RCM. It is the relationship between low-resolution altitude variables - supposedly well represented by GCMs - and high-resolution surface variables. To ensure perfect coherence between predictors and predictands, the emulator is trained in a perfect model framework, where both inputs and target come from the same RCM simulation. The emulator is designed for a specific RCM and can emulate only this one. Chapter 3 presents the evaluation of the emulator on a single RCM simulation driven by the same GCM as the training simulation but in a different scenario. The present chapter aims to extend this evaluation.

The main objective of the emulator is to downscale a maximum of GCM simulations while respecting and preserving their diversity. It is then important to study its transferability to different sources of inputs. Therefore, different levels of transferability can be considered.

1. Member-transferability: the simplest, it evaluates the emulator's ability to downscale another member from the same GCM/Scenario couple as the one used for training.
2. Scenario-transferability: It is the robustness to different forcing scenarios with the same GCM. In chapter 3, we started to validate this question, and the present chapter will further document it.
3. GCM-transferability: probably the most complex one, it is the ability of the emulator to provide a coherent downscaling of simulations from global models different than the one used to drive the training simulation.

This work is based on ALADIN63, the state-of-the-art regional climate model developed at CNRM. In the EURO-CORDEX and Copernicus C3S projects, 10 ALADIN63 simulations were performed covering different driving GCMs and scenarios. This small ALADIN63 matrix 4.1 allows us to explore levels 2 and 3 of transferability as any of this simulation is a possible test set. Moreover, I think it is legitimate to consider that the second level implies the first one, especially at the beginning of the 21st century, where the scenarios are similar. It is worth recalling here that the emulator is designed to learn the downscaling function of a given RCM, which is why the evaluation focuses on ALADIN63 runs.

This transferability evaluation will be conducted in three steps. First, I evaluate the scenario and GCM-transferability of the emulator for near surface temperature introduced in the last chapter. Secondly, I will extend this analysis by training other emulators of the same RCM (ALADIN63) using different training simulations and check if it provides similar results. Furthermore, I will also investigate the benefits of using simulations from multiple driving GCMs in the training set. Following the methodology and the results from Chapter 3, I will address these questions of transferability in a perfect model framework. This choice is deeper commented in Section 4.2.3. Finally, I will use the emulator in application mode and downscale the GCM simulations used to drive ALADIN63. I will try to assess the quality of these emulated series by comparing them with both the GCM low resolution and the RCM high-resolution temperature.

Table 4.1: The ALADIN63 matrix.

| DRIVING SCENARIO | DRIVING GCMS | | | |
|------------------|--------------------|---------------------|--------------------|---------------------|
| | CNRM-CM5 (CNRM) | MPI-ESM-LR (MPI) | NorESM1-M (NCC) | HadGEM-ES2 (HGM) |
| Historical | x | x | x | x |
| RCP26 | x | | | |
| RCP45 | x | | | |
| RCP85 | x | x | x | x |

This chapter is divided into four sections. The first one presents the simulations used in this chapter and sets the methodology applied to evaluate the transferability of the emulator. After that, sections 4.3 and 4.4 present the results in perfect model evaluation and the application to GCM inputs. The last one discusses different choices made during the construction of the emulator.

4.2 Methodology

This section firstly introduces the available simulations to explore the transferability question. Then I recall the principle of the perfect model framework and justify why I use it for this chapter. Finally, I will expose the different metrics I will use.

4.2.1 The ALADIN63 matrix

In the EURO-CORDEX framework, ten scenario simulations were performed with ALADIN63 with different driving GCMs and scenarios. They are reported in Table 4.1. All GCMs used to drive ALADIN63 are part of the fifth phase of the Coupled Model Intercomparison Project (CMIP5). The first GCM is CNRM-CM5, which is developed at CNRM. Because they are developed in the same institute, ALADIN63 and CNRM-CM5 share very similar components that guarantee some consistency in the downscaling procedure¹. Four simulations from CNRM-CM5 were downscaled by ALADIN63, one historical (1951-2005) and one for each RCP2.6, RCP4.5 and RCP8.5 scenarios (2006-2100) of greenhouse gas emissions. In chapter 1, the emulator is trained on the CNRM-HIST and RCP85, while the RCP4.5 was used to evaluate it. The other driving GCMs are MPI-ESM-LR, developed at the Max Planck Institute (Giorgetta et al, 2013), NorESM1-M, developed at the Norwegian Climate Center (Bentsen et al, 2013) and the HadGEM2-ES, the Met Office climate model (Collins et al, 2011). The historical and RCP8.5 runs were downscaled for these three models with ALADIN63. The CNRM, MPI, NCC and HGM acronyms will respectively refer to the CNRM-CM5, the MPI-ESM-LR, the NorESM1-M and the HadGEM2-ES GCMs.

In Section 4.4 I will use the same GCM simulations to test the emulator in application mode. As they all have different horizontal resolutions, I interpolated all of them on the CNRM grid at 1.4° resolution. All these simulations are available on

¹ALADIN63 relies on a more recent version of the code than CNRM-CM5, so the consistency is not “perfect” but the best we have

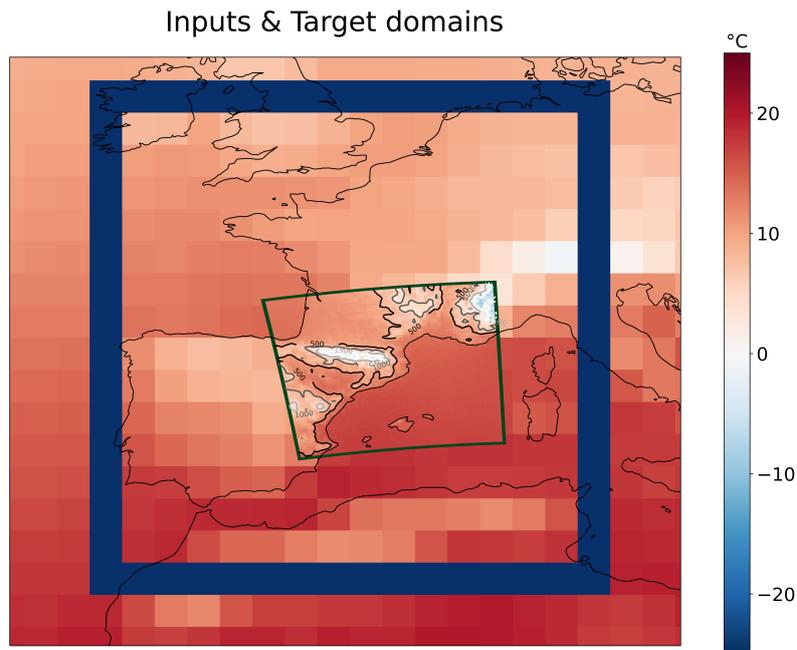


Figure 4.2: Illustration of the inputs domain (blue box including the sides) and the target domain (green box).

the mesocenter ESPRI facility, which is supported by CNRS, UPMC, Labex L-IPSL, CNES and Ecole Polytechnique. Concerning the HGM historical simulation, I could not find some input variables on the ESPRI nor the ESGF (ref) nodes, so I only work on the RCP85 simulation.

4.2.2 Recall on the main emulator and Experiment design

The ALADIN63-emulator from Chapter 3 is built to reproduce the daily near-surface temperature over a 64×64 grid points domain located between South of France and North-East of Spain and illustrated in Figure 4.2. The predictors are five variables at three different pressure levels² plus some surface fields and the aerosols optical depth over a larger input domain of 16×16 GCM (CNRM-CM5) grid points. They are normalised daily according to their spatial means and standard deviations so that only the information about the atmosphere’s spatial structure remains. The temporal information is given through a vector containing for each day the spatial means and standard deviations used for normalising each variable map, plus external forcings such as the greenhouse gas concentration and a seasonal vector. The emulator relies on a neural network architecture called UNet that has been adapted to the problem. More details can be found in the previous chapter.

The emulator is calibrated in the perfect model framework. It implies that, for the training phase, both predictors and predictands come from the same high-resolution simulation. The idea is to ensure a perfect correlation between the predictors and the target surface variable, such that the neural network focuses on the estimation of the downscaling function only. The predictors are then upscaled to a typical GCM resolution, and we apply a moving average window to kill any high-resolution

²i.e. different altitudes

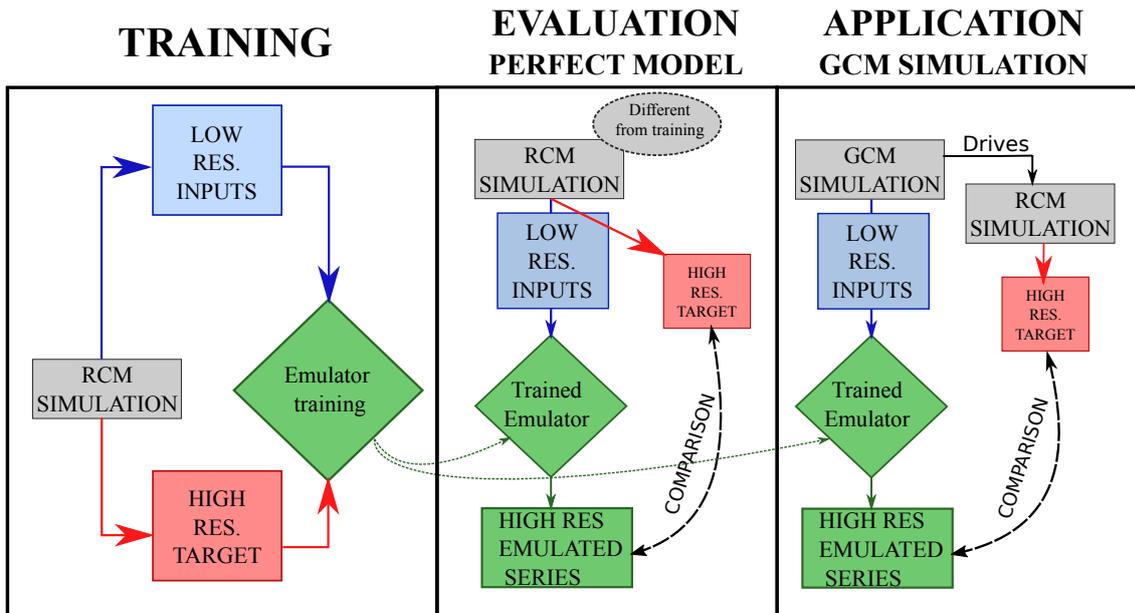


Figure 4.3: Training, evaluation and application of the emulator. Inspired and adapted from Wang et al (2021).

information that could have remained. As explained in chapter 3, the same “filtering” will be applied to GCM data before giving them to the emulator as it allows to reach the effective resolution of the model and to work with more robust data. Figure 4.3 illustrates the training protocol. The perfect model framework also serves for the evaluation with a different simulation than the training set. Thus, the emulated series can be compared with the RCM high-resolution variable we target to emulate. Conversely, when the emulator is used to downscale a GCM simulation, it is impossible to evaluate the emulated series correctly. As the emulator does not reproduce the large scale biases between GCM and RCM, it is normal to have biases between the GCM-downscaled series and the true RCM. Therefore, separating these large scale biases’ local impact and the emulator’s possible error is difficult.

In Chapter 3 the ALADIN63-emulator is trained on CNRM-HIST and CNRM-RCP85 ALADIN63 simulations and evaluated on the CNRM-RCP45, in the perfect model framework. From now on, I will refer to it as the *main emulator*. The first part of this work (Section 4.3.1) analyses the main emulator on all other ALADIN63 simulations in perfect model framework. Following Figure 4.3, predictors are extracted from each of the ALADIN63 simulations (cf. Table 4.1), upscaled and filtered to obtain the *UPRCM* simulations. Those inputs are then given to the main emulator, which recreates high-resolution temperature series that are compared with the corresponding ALADIN63 simulation. The objective is then to assess how the emulator “digests” inputs from different sources. In section 4.3.2, I construct other ALADIN63-emulators using other training sets. The exercise aims to evaluate if it affects the results obtained in chapter 3 and section 4.3.1, and so how the emulator depends on the training set.

4.2.3 Justification of the perfect model framework

For the reasons above, I first address the transferability in the perfect model framework. The RCM-emulator relies on a specific convolutional neural network architecture. Another way to see the current chapter goals is to assess if the neural network and the previous strategy will be robust to slightly different input spaces. Indeed, the driving scenario modifies the atmosphere composition, which impacts all variables differently. Furthermore, two GCMs rely on different assumptions reflecting the choices of the teams building them and impacting the entire structure of the system, the relationship between variables, the vertical dependence etc.. In both cases, the input space from a new simulation to downscale will differ from the training input space, and this chapter aims to study the robustness of the emulator in such cases.

To validate the use of the perfect model framework it is necessary to assess if the difference between two RCM simulations is representative from the one between two GCM simulations. The driving RCP scenario is likely to impact RCM and GCM similarly, so it seems reasonable to study the second level of transferability within this framework. However, it is not intuitive that the intrinsic differences between two GCMs will reflect in the RCM simulations they drive.

The upper panel of Figure 4.4 shows the yearly mean times series (1950-2100, RCP8.5) for the temperature, the geopotential and the humidity at two pressure levels (500 and 850 hPa) over the input domain defined in chapter 1, for the 4 UPscaled RCM (UPRCM) simulations we have. They are an extract of the inputs given to the emulator. This figure helps to illustrate the kind of differences that can exist between the different ALADIN63 simulations. Firstly we can observe that the relationship between variables at a given pressure level differs between two simulations. For example, NCC-UPRCM is generally less humid and colder than MPI-UPRCM at 500hPa but has a similar altitude for the geopotential. Secondly, the vertical dynamic differs, as we can see by comparing the right and left panels. For instance, the MPI-UPRCM is more humid than any other simulation at 500hPa, but it is not true at 850hPa. Finally, we can also remark that the different simulations have different responses to the increase in greenhouse gas concentration imposed by the RCP85 scenario. The HGM-UPRCM (in green) notably simulates more significant changes than the others. Therefore, fundamental differences exist between these 4 UPRCM simulations in the internal relationship of the emulator inputs.

The lower panel shows the yearly mean times series of the same variables but for the driving GCM simulations. The first comforting remark is that the differences observed in the UPRCM simulations are in the same order of magnitude as the ones we can see in the GCM simulations. For a given variable, the vertical structure differs from one model to the other. When comparing right and left columns, it is particularly visible for the MPI-GCM humidity or the CNRM-GCM temperature. This result tends to validate the evaluation of transferability in the perfect model framework. Moreover, we can observe how the RCM transforms the GCM large scale. Regarding almost all six variables, the differences between the four simulations are not the same as in the UPRCM world. The most remarkable example

is probably for the HGM simulations, where ALADIN seems to underestimate the atmospheric warming of the original GCM simulation. Still, regarding the temperature at 850hPa, CNRM-GCM, MPI-GCM, and NCC-GCM are much more similar than their UPRCM versions.

Figure 4.4 confirms that the perfect model framework is a good opportunity to test whether the emulator built in Chapter 3 of this manuscript is robust to different input sources and all the intrinsic differences they carry.

4.2.4 Metrics, Benchmark and results presentation

As in the previous chapter, I use the interpolated low resolution near-surface temperature from the corresponding input simulation as a naive reference. It helps us to locate the emulator's performance with respect to a representation of its driving large scale. On the other hand, I will also evaluate the emulators on their training set and use it as a reference for the best result possible as it is a level 0 of transferability.

As a matter of coherence, I use the same metrics as the ones Chapter 3 over slightly different periods. I compute the maps for the mean and 99th and 1st quantiles over three thirty-year periods: 1951-1980, 2006-2035, and 2071-2100 referred to as past, present and future periods. I replaced in this chapter the number of days over 30°C by the first quantile as it is essential to look at both sides of the distribution. These maps are computed for the RCM truth, the Emulated series and the interpolated UPRCM/GCM. In addition, I add the biases maps to highlight the differences between the RCM and both the emulator and I-UPRCM maps. I also use the four scores from the previous chapter, the RMSE, the variance ratio, the temporal correlation (ACC) and the Wasserstein distance. These scores are computed over the whole simulation, separating the historical (1950-2005) from the projection (2006-2100) part.

These metrics are computed at the grid point scale, implying that I obtain a map for each metric and each evaluated simulation. As in the previous chapter, we use the spatial mean and superquantiles (5th and 95th) to summarise each map. For the raw climatology and quantile maps, I also use the spatial correlation and spatial RMSE to measure the ability of the emulator to reproduce the high resolution spatial structure brought by the RCM. In this chapter, I will evaluate many emulators on multiple metrics. Thanks to these summary metrics, I can synthesise all these results summary plots.

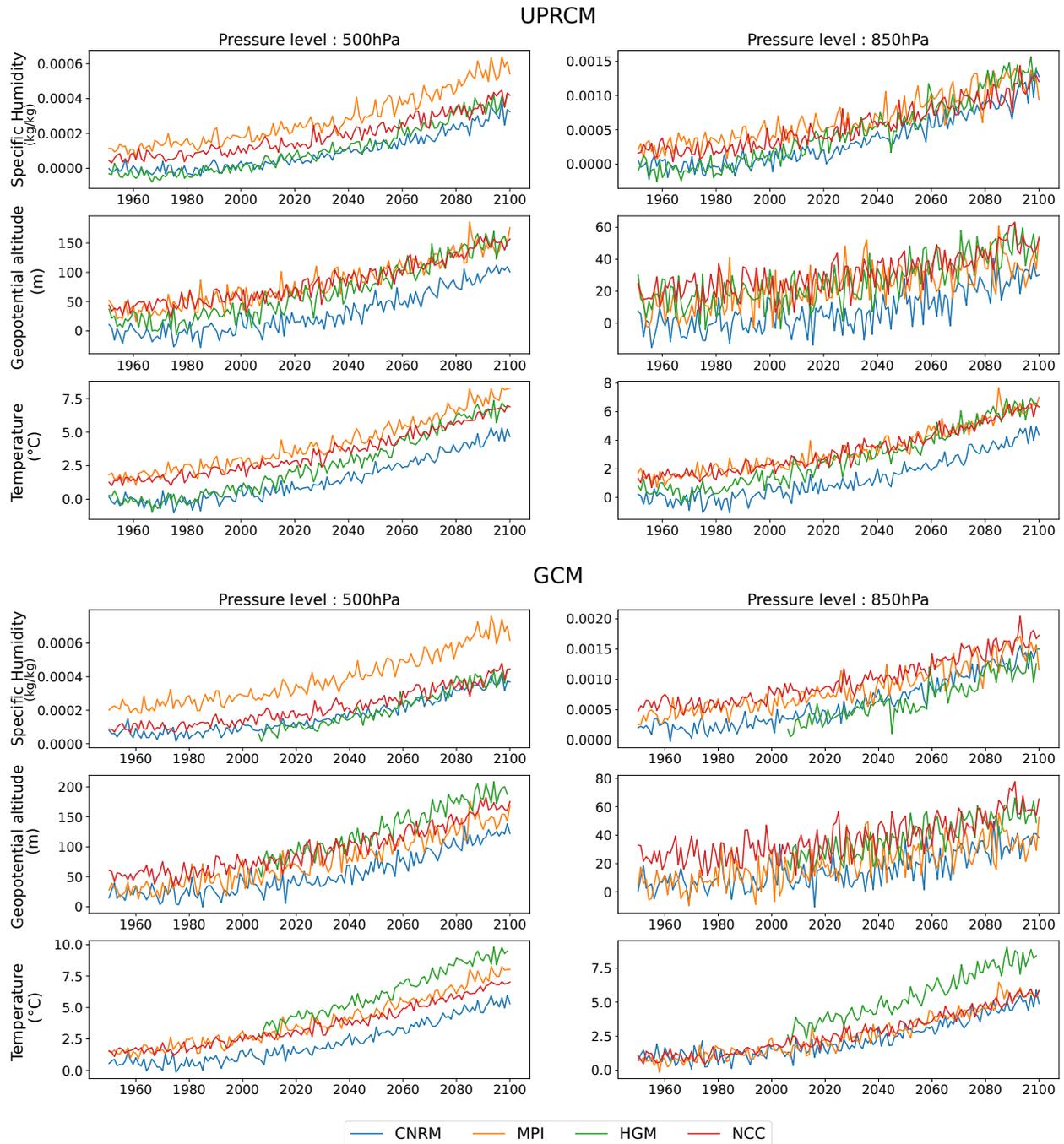


Figure 4.4: Yearly anomalies time series for 3 atmospheric variables (Specific Humidity, Geopotential altitude and Temperature, from up to down) at 2 pressure levels (500 and 850 hPa, from left to right). These series are shown for the 4 UPRCM simulations (upper panel) and their driving GCM simulations (lower panel). We show here the yearly mean over the input domain. The series are plotted as anomalies compared to CNRM-UPRCM climatology over the 1971-2000 period. In the upper panel the simulations are differentiated according to the driving simulation which is plotted in the lower panel.

4.3 Perfect Model evaluation

In this section, I address the reliability of the emulator under different sources of inputs and its capacity to reproduce the daily maps under different climates. We place ourselves here in the perfect model framework to evaluate the emulated series' accuracy with respect to their RCM true series. We divide this section into two parts. Firstly we will study the transferability of the emulator trained in Chap 1, and secondly, we will train 15 other emulators on multiple combinations of the training set.

4.3.1 Main emulator

The UNet-emulator built in chapter 1 was trained using the entire CNRM Historical and RCP85 simulations following the training procedure recalled in Section 4.2.2. It was then evaluated in the perfect model framework on the CNRM RCP45 simulation, showing an excellent ability to downscale a more intermediate RCP scenario than the one used for calibration. I extend this evaluation to all ALADIN63 simulations (see Sec 4.2.1). The results shown here are computed on the novel simulations but also on the CNRM-RCP45 to compare the new results with the ones from the previous chapter. I also downscaled the UPRCM-CNRM HIST and RCP85 used for calibration, which I consider a naive benchmark. Of course, the results on this simulation are expected to be good. As in Chapter 3 we use the interpolated UPRCM temperature as a benchmark. These two benchmarks will help us to analyse and compare the results on the different simulations. In the entire section the RCM simulations are named accordingly to their driving GCMs. The GCM simulations are not used in this section, so the names CNRM, MPI, NCC or HGM only refer to the ALADIN63 simulations.

Yearly mean series reconstruction

Figure 4.5 gives a first insight into the emulator's ability to reproduce the different climates, as it plots the times series of the year means over the whole domain. The emulator produces, as the RCM, daily temperature maps, so these mean series are, in some sense, reconstructed large scale. In a matter of coherence, it is fundamental that the emulator respects and reproduces this kind of chronology and sensitivity aspects well, even if it is not designed for it.

The RCM and emulator series look perfectly similar, with the correct order between the different simulations. The gaps between the 3 CNRM simulations are well reproduced. The RCP85 warms much stronger after 2050, while the RCP26 is slightly cooling at the end of the century. Moreover, the MPI and NCC simulations appear warmer than the CNRM throughout the period, and the emulator respects this well. Finally, the HGM RCM simulation (in brown) is warming the most and exactly as in the emulator plot. These series tend to show that the emulator is robust to different input conditions and correctly follows its driving large scale. Now we can have a closer look into these emulated simulations.

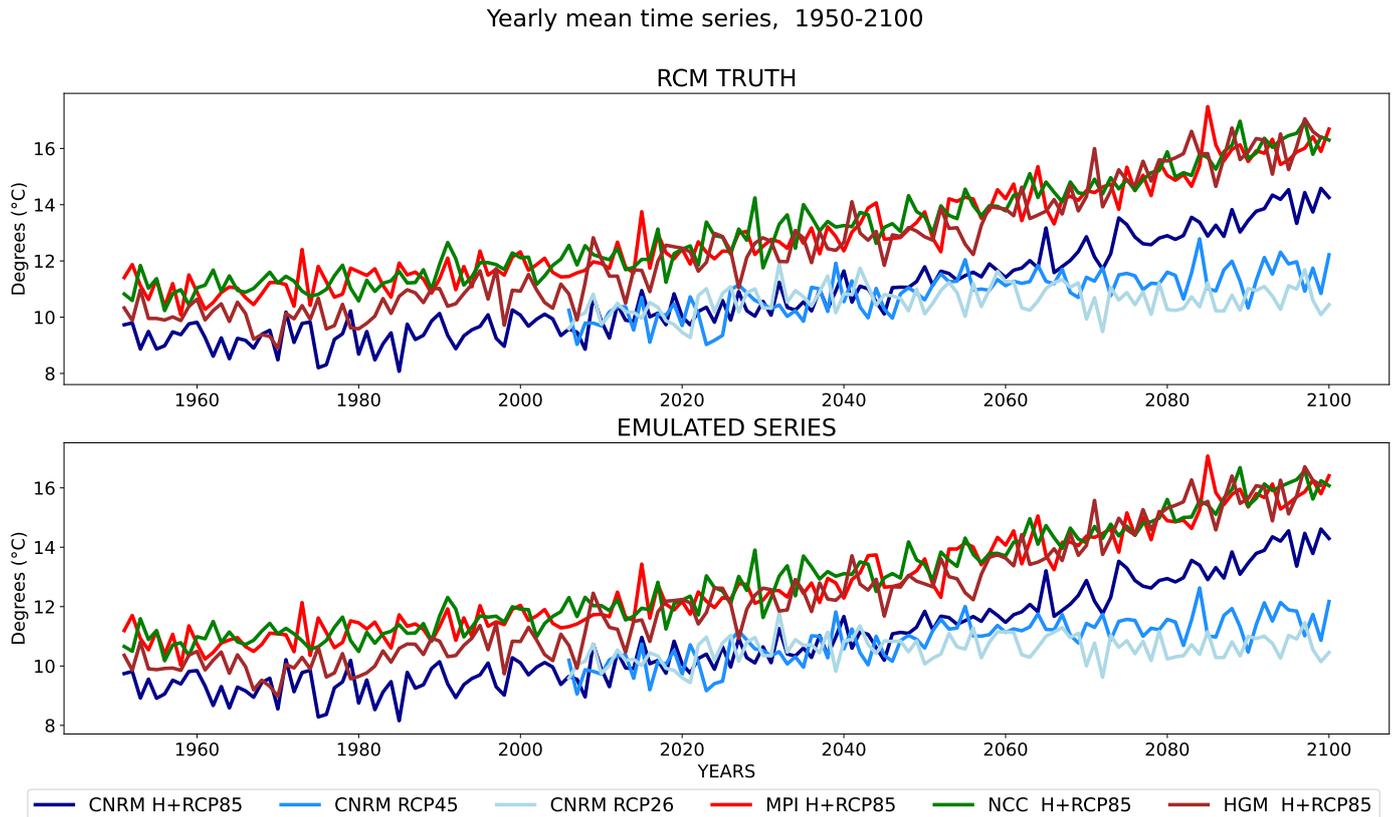


Figure 4.5: Time series of the yearly mean near surface temperature over the target domain for the 6 entire (Historical + Scenario) simulations at our disposal. The upper panel shows the series computed on the RCM true simulations, while the lower panel shows the ones computed from the emulated simulations.

Grid points 30 years statistics

In a second step, we can look at temporal statistics at the grid point scale over 30 years periods. I compute the mean, 99th and 1st percentile maps over the past (1971-2000), present (2006-2035) and future (2071-2100) periods. As explained in Section 4.2.4 we summarise these maps with the spatial mean and spatial 5th and 95th super-quantiles, and I report these numbers in summary bar plots. The biases maps regarding the corresponding RCM truth maps are summarised similarly. Figures 4.6 and 4.7 show the summary plots for the present period and an illustration for one simulation. Figure 4.8 shows the summary plots for the past and future periods. The results are shown for both Scenario and GCM transferability, but we firstly comment the scenario-transferability (i.e. the left part of the plots).

The first conclusions on the yearly series are confirmed at the grid point scale. Concerning the Scenario-transferability, the emulator reproduces accurately the CNRM-RCP26 climatology and quantiles maps. It confirms the results obtained in the last chapter on CNRM-RCP45 and the ability of the emulator to reproduce more intermediate scenarios than the ones used for calibration. For both simulations, the emulator reproduces the spatial structures of the different maps very well, as the spatial correlations are always equal to one and the RMSE around 0.1°C on the present climatology (Figure 4.6). The climatology maps are identical, as the red

and green bars are indistinguishable and the biases very small. Figure 4.7 shows that the emulator reproduces less well the extreme percentile maps on these two simulations. The spatial RMSEs are higher (0.3/0.4°C), and the bias summary plots show an underestimation of the warmest extremes and inversely for the 1st percentile. The maps for the CNRM RCP26 1st percentile (Figure 4.7) help to understand these biases. The emulator captures the spatial structure but has a general warm bias. Remarkably, the worst points do not necessarily correspond to the coldest points, as the bias is more significant in the Rhone valley than on top of the Pyrenees. The analysis is the same for the RCP45 simulation with stronger biases because the first quantile map is colder in this present period than the RCP26.

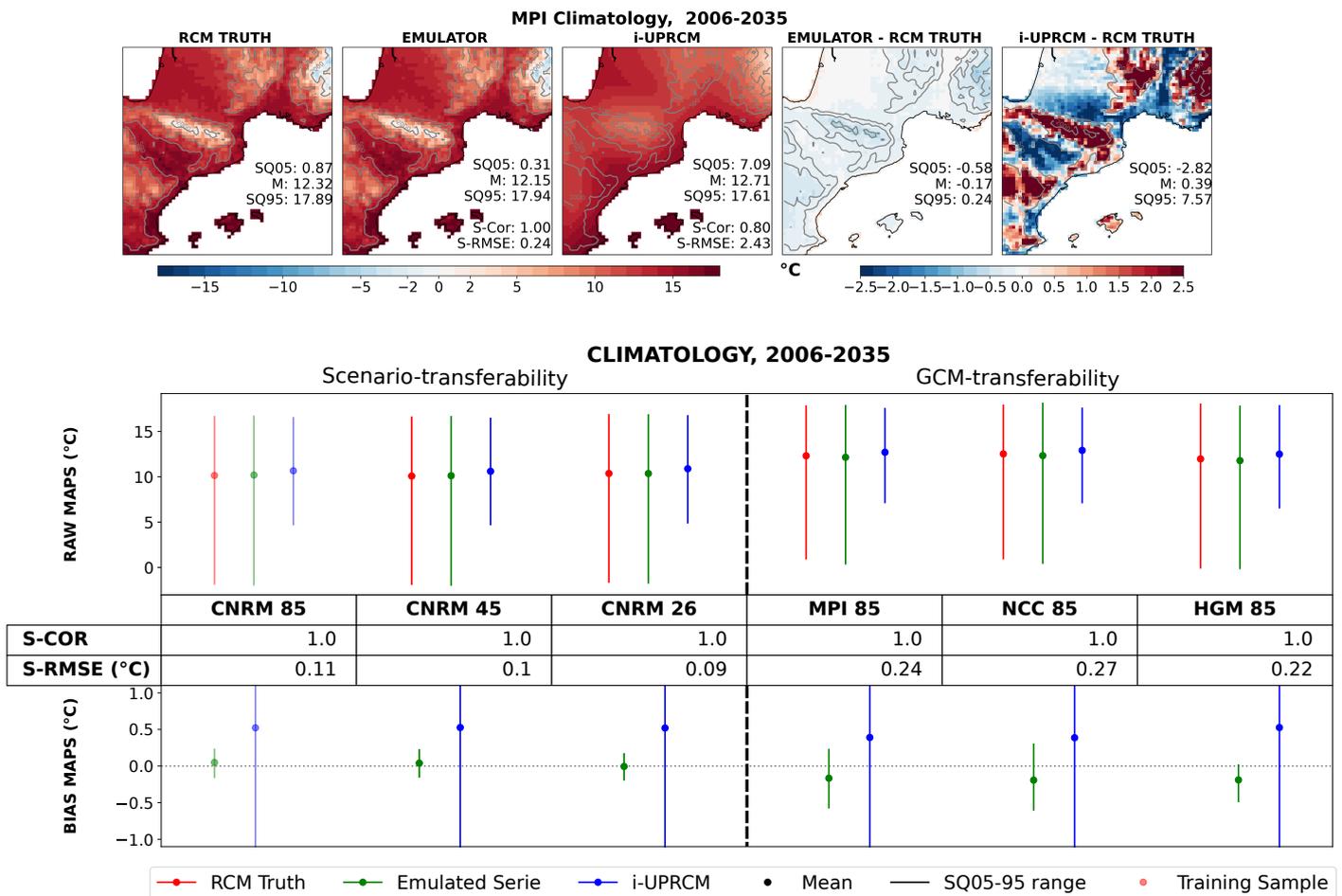


Figure 4.6: Climatology on present climate. The upper panel shows the MPI simulation climatology maps for the RCM truth, the emulator and the interpolated UPRCM temperature. The emulator and i-UPRCM biases with respect to the RCM true map are also shown. These maps are summarised by their mean and super-quantiles. The spatial correlation and RMSE w.r.t the RCM truth are computed for the raw maps. The lower panel regroups these statistics for the 6 simulations. On the upper part for each simulation, the RCM (red), the emulator (green) and the interpolated UPRCM (blue), statistics are represented by an error bar. The lower (resp. upper) boundary of the bar shows the map’s 5th (resp. 95th) super-quantile and the point the mean. The lower plot shows the same statistics for the bias maps. The middle table shows the spatial correlation and RMSE for the emulator maps.

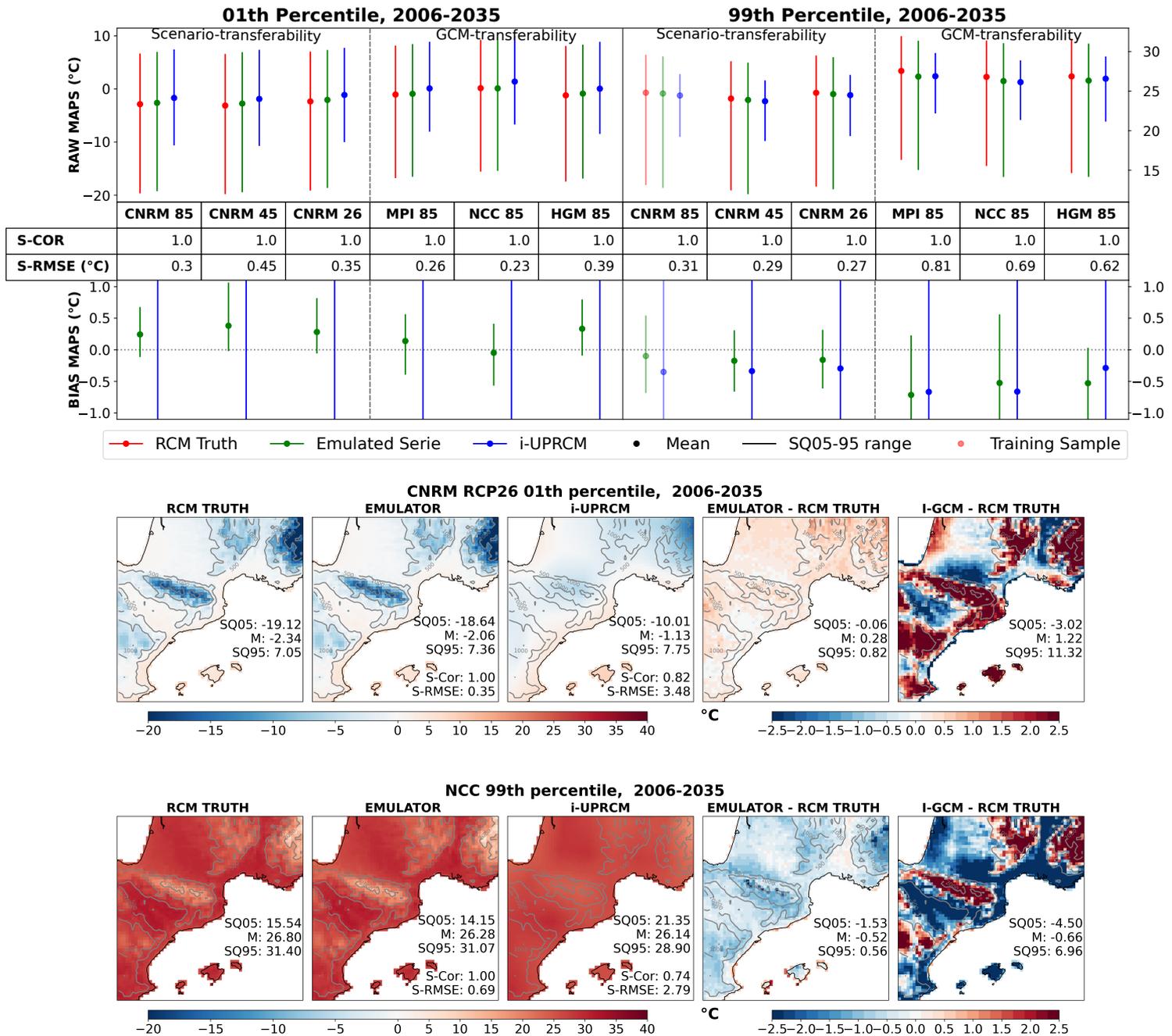


Figure 4.7: Upper pannel : Same as lower panel on Figure 4.6, for 1st and 99th quantiles. Center (resp. Lower) panel : 1st (resp. 99th) quantile maps for the CNRM-RCP26 (resp. NCC) simulation for the RCM truth, the Emulated series and the i-UPRCM, and the biases maps.

Figure 4.8 extend those remarks for future periods³. The spatial structure is always well reproduced with spatial correlation equal to 1. Notably, the green bars always match flawlessly with the red ones. In future periods, the effect of the different scenarios is well visible in the first percentile, for example, and the emulator respects those different evolutions. We can conclude that even if the emulator has difficulties reproducing the extremes, it generally fits well with the original RCM

³The historical is common to RCP85, RCP45 and RCP26 and has been used in the training set.

series validating the scenario-transferability.

Moreover, one can observe that the results of reproducing the CNRM Hist and RCP85 are similar to those for the other CNRM RCPs. This brings confidence to the neural network training since it shows no over-fitting on the training set.

Regarding the transferability to other driving GCMs, the emulator reproduces the different maps correctly even if the results are less good than in the CNRM simulations. Figure 4.6 shows a cold bias (about -0.2 °C) in the 3 emulated series which reflects in the RMSE. The spatial structure is well captured, as we can observe on the MPI maps given for illustration. The cold bias is similar over the entire domain with no spatial structure. Regarding Figure 4.7, the emulator has a clearer bias on the 99th percentile for the MPI and NCC simulations. The NCC maps (Figure 4.7) show a general bias which seems stronger on the reliefs. Conversely, the emulator is quite accurate on the 1st quantile.

A possible explanation is that the emulator has more difficulties in a warmer climate. The MPI and NCC are constantly warmer than CNRM (cf Fig 4.5) and the emulator provides slightly colder (about -0.2 °C) simulations. The HGM simulation has a similar average temperature to the CNRM in the past period. Figure 4.8 shows that its emulated series has very good climatology and quantile maps. However, due to a stronger response to climate change, HGM gets warmer than CNRM at the end of the simulation. We can then observe that the HGM emulated climatology map is slightly colder than the RCM one. This cold bias might come from the highest temperature, which the emulator never saw in its training sample and has more difficulties reinventing. In Figure 4.8, the summary plots for the 99th percentile maps support this hypothesis, the emulator maps on MPI Hist+85, NCC Hist+85, and HGM 85 are too cold with RMSEs over 1°C . Notably, the emulator performs poorly on the representation of the MPI RCP85 99th percentile map with a significant cold bias. On the other hand, the emulator reproduces the colder extremes on these simulations very well. These results highlight the role of the emulator, which is not designed to extrapolate too far from its calibration set. Similarly, the emulator is less good at reproducing the coldest extremes in the HGM HIST and CNRM RCP26, the two coldest simulations.

Nevertheless, the emulator still provides very accurate maps. It recreates almost perfectly the temperature range over the maps as the red and green bars on all summary plots are always very close. Indeed the temperature difference between CNRM and MPI or NCC is about two degrees, while the emulator bias is only about 0.2 degrees. On the illustrating maps on Figures 4.6 or 4.7, the emulator maps are indistinguishable from the RCM ones. Moreover, the order between the simulations is well respected. In Figure 4.8, the HGM, MPI, and NCC simulations reach similar temperatures at the end of the century, while HGM HIST is colder than the others, and the emulator reproduces this aspect. To conclude on this part, the emulator seems robust to the GCM transferability as it produces realistic and coherent maps. Nevertheless, it seems complicated to extrapolate too far from its training set, notably to reproduce the same range of warmest extremes in a warmer climate. This is a problem that a better training set could probably overcome.

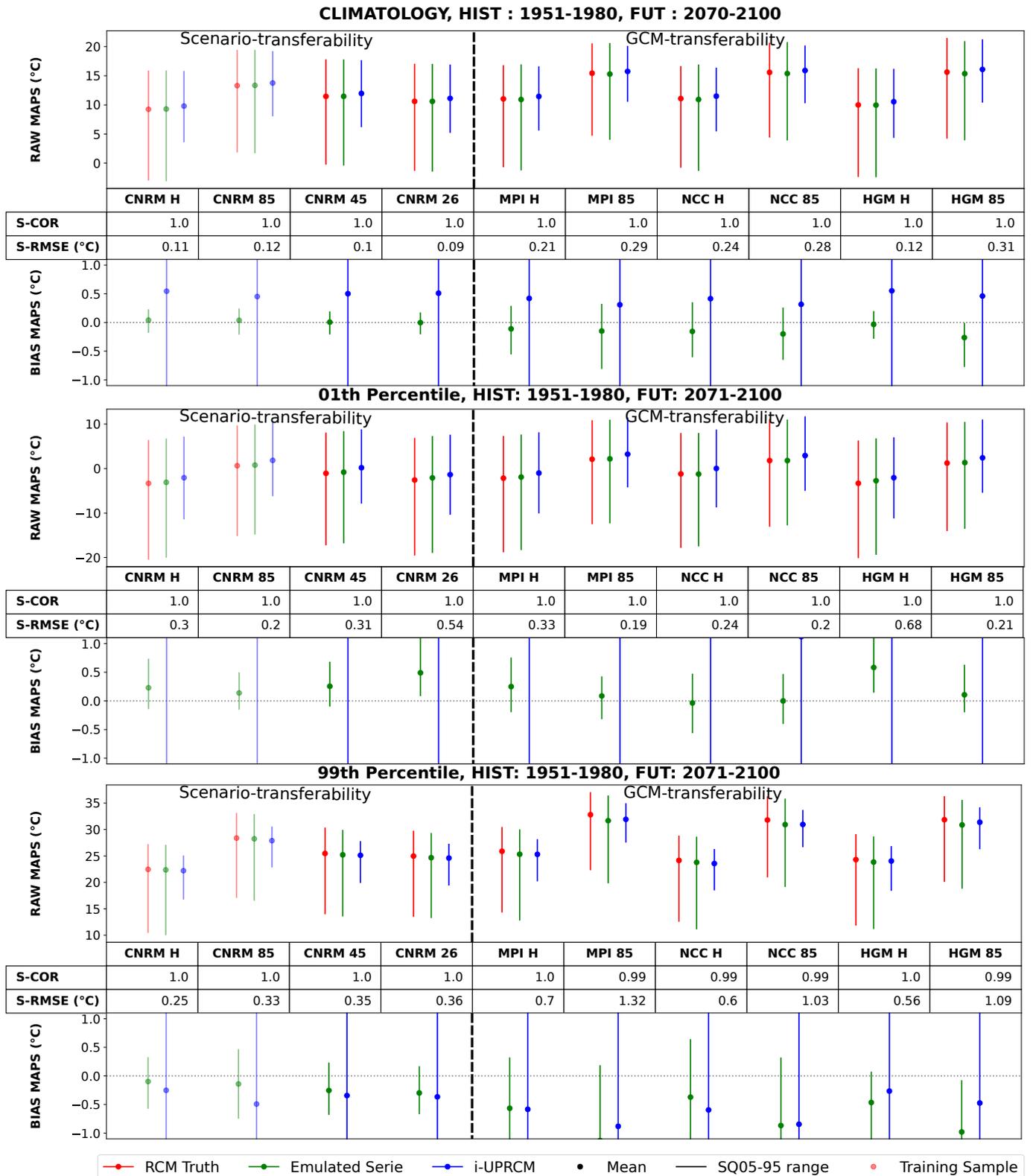


Figure 4.8: Summary plots for Climatology, 1st and 99th percentiles on past (1970-2000, "H" columns) and future (2071-2100, "85", "45", "26" columns) period.

Grid points climate change analysis

Figure 4.9 shows the summary plots for the climate change responses maps for the three statistics studied previously. The emulator reproduces well the mean climate change for all six simulations. The biases observed in the climatology maps for MPI and NCC simulations are constant over past and future periods, so it does not impact the delta. The warming in the HGM simulation is slightly underestimated (RMSE about 0.2°C), but the emulated simulation warms way more ($\geq 1^{\circ}\text{C}$ more) than any other, just as the RCM does. For all six simulations, the emulated maps have the right spatial structure of the warming, with the right spatial range.

It is interesting to notice in Figure 4.9 that, for example, the structure of the change for the first percentile is different between the HGM and the MPI simulations and the emulator correctly captures this difference. Indeed, one can observe that MPI warms more in the plains from South-West of France than HGM. On the other hand, HGM has a stronger warming just South of the Pyrenees. Furthermore, the spatial correlation between these two RCM maps is equal to 0.79 and accurately reproduced on the emulator maps. The results are less good concerning the 99th percentile with larger biases and less good spatial correlations (about 0.85 and even 0.76 for the MPI). Still, the emulator simulations have always the proper spatial range of warming, with, for instance, a more considerable spatial magnitude in the NCC map than in the HGM one. It is also noteworthy that the differences between the blue (i-UPRCM) and red (RCM) bars are not always the same, demonstrating that the downscaling transformation differs from one driving large scale to another.

Grid point daily analysis

As a third step and to have a closer look at these emulated series, it is necessary to evaluate them daily. The four scores used in Chapter 1 and recalled in Section 4.2.4 are computed over the entire ten simulations (Historical + RCPs) on the daily time series in each grid point. The results are summarised in Figure 4.10, through summary plots where each map's mean and superquantiles are reported. These scores reflect the emulator's ability to reproduce the daily time series in all simulations as they compare the entire time series at each point.

Regarding the two CNRM simulations (RCP45 and 26), the emulator reproduces almost identically the daily time series all over the target domain. For instance, even the worst points have a Wasserstein distance less than 0.2 and a variance ratio higher than 90%. As expected from the climatology and quantile maps analyse, the scores on the MPI, HCM and NCC simulations are less "perfect", which is probably somehow expected. The first remark is that the scores spatial means on these simulations are completely comparable with CNRM ones. The average RMSE is lower than 1°C , the variance ratio and correlation are almost equal to one and the mean Wasserstein is less than 0.2 with a lower bound similar to the CNRM simulations scores. However, some points are less well reproduced, as shown by the upper bound of the error bars on the Wasserstein distance plot and illustrated on the MPI scores maps. The MPI's Wasserstein map indicates that the emulator performs well everywhere but has more difficulties on the reliefs. It is not shown

here, but the analysis is the same for NCC and HGM Wasserstein distance.

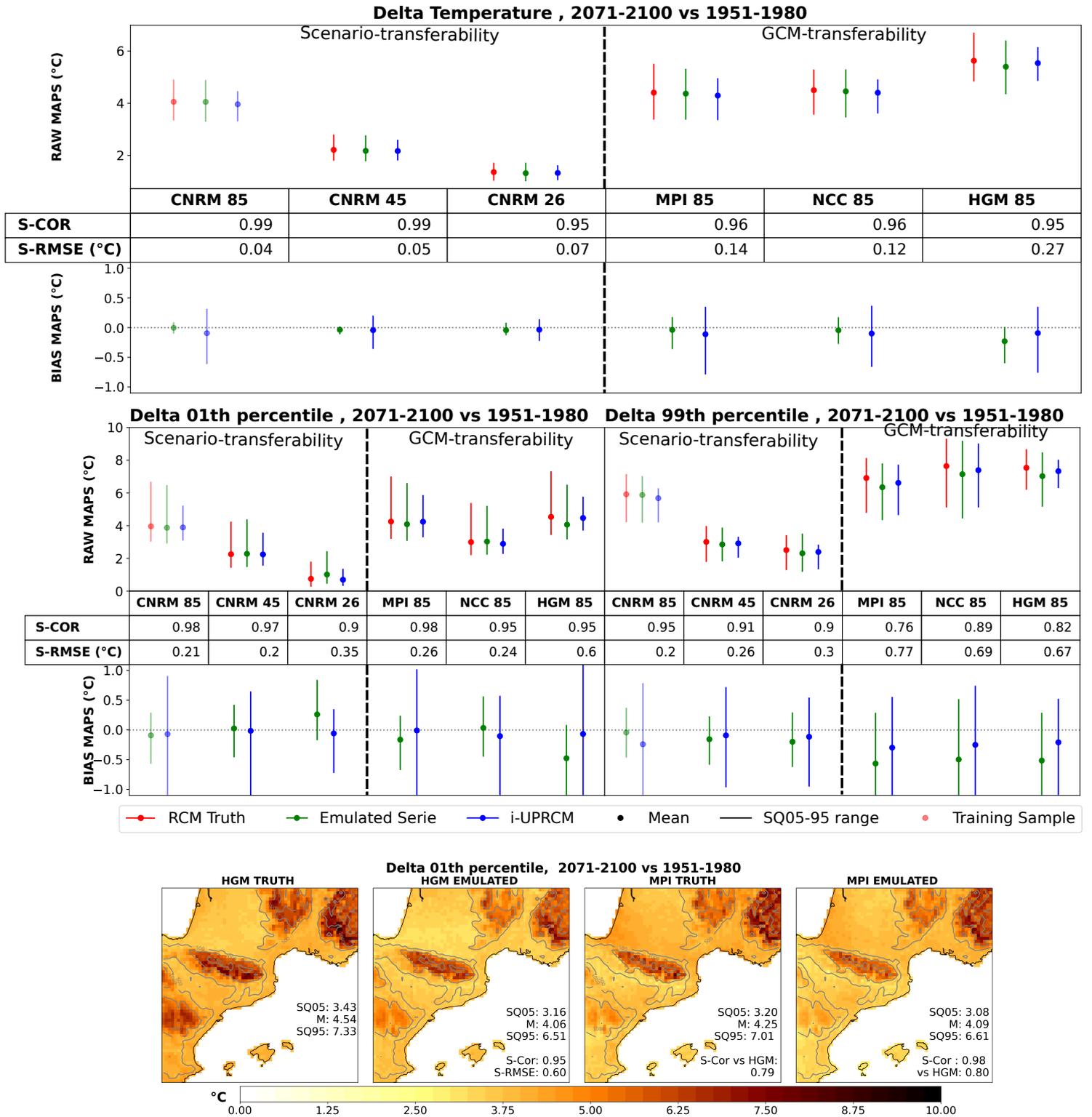


Figure 4.9: Summary plots for climatology, 1st and 99th percentiles deltas between past (1970-2000) and future (2071-2100) period.

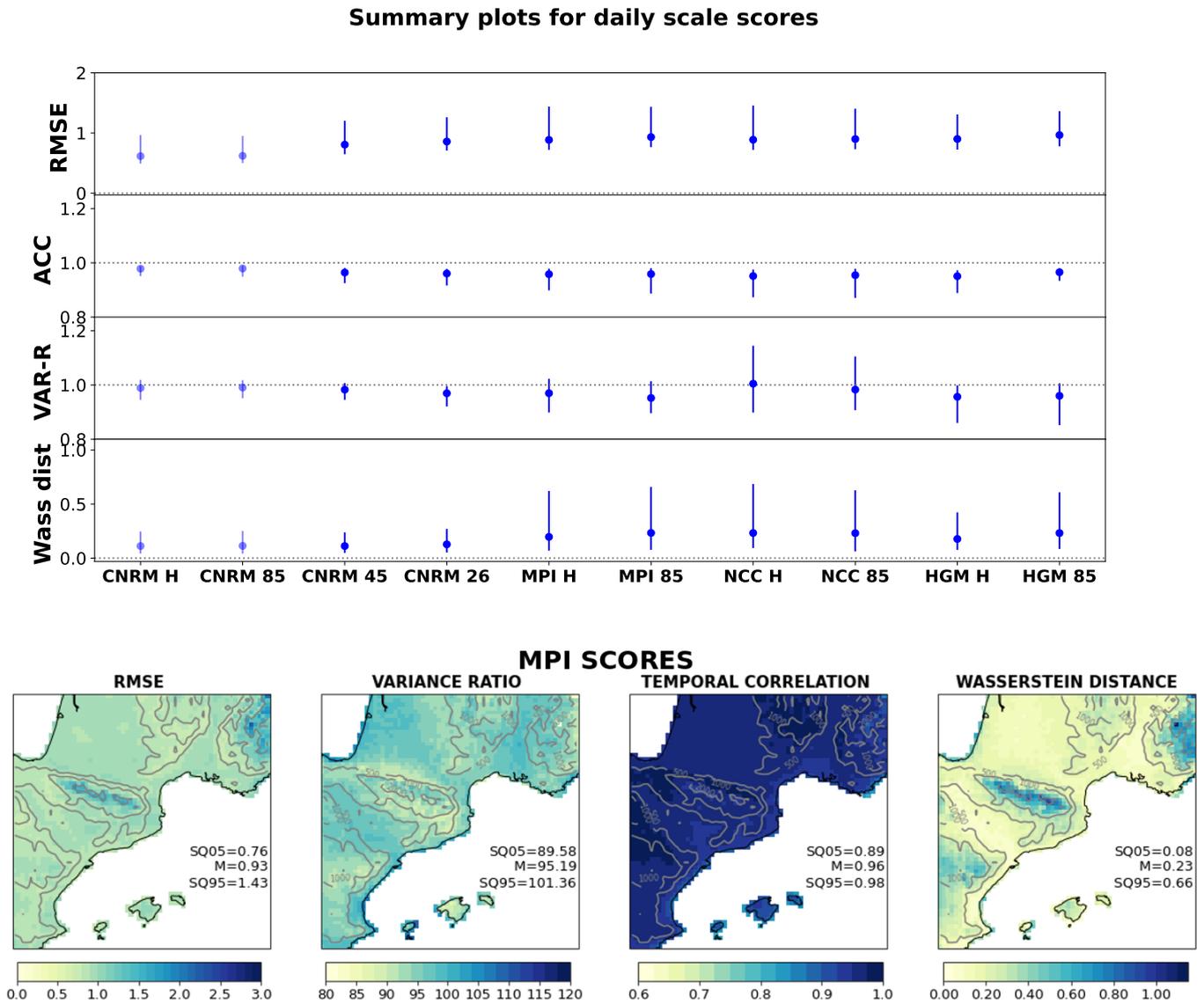


Figure 4.10: Upper : Summary plots for the daily scores: RMSE, the deseasonalised temporal correlation (ACC), the ratio of variance (VAR-R) and the Wasserstein distance, for all 10 simulations, on their entire length. Lower: Illustration of the scores maps on the MPI RCP85 simulation.

Another way to look at the transferability of the emulator is to analyse its ability to reproduce the transformation of the daily probability functions across models and time. Figure 4.11 plots the spatio-temporal probability density function of the daily temperature over all Pyrennees points higher than 1000m. It compares the true RCM (full line) and the emulated (dotted line) pdfs over a historical (1971-2000) and future (2071-2100) periods. It also plots the difference between the past and future pdfs to better visualise the transformation due to global GHG emissions. On each plot, two simulations are represented on each side of the x-axis. This figure shows that the emulator reproduces almost perfectly the different pdfs over both periods and all driving GCMs. It also matches exactly the climate change transformation. This figure also illustrates the greater Wasserstein distance (Fig. 4.10) for MPI simulation on the Pyrenees. We can see that the emulator has more difficulties in reproducing the right tail of the future MPI distribution. However, it

is observable here that even over these bad points, the pdf of the emulator looks credible regarding the RCM one. Finally, it is remarkable that the driving GCM has a non-negligible impact on the pdfs and that the emulated series reproduce this aspect. The CNRM/MPI comparison illustrates well this aspect, with the apparition of a dipole in the MPI future pdf, which does not exist in the CNRM simulation.

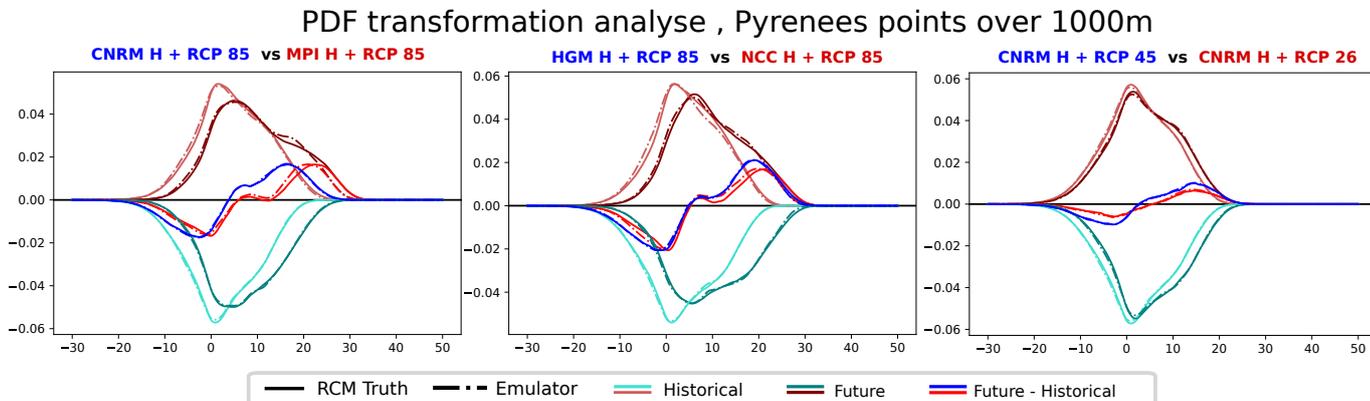


Figure 4.11: Comparison of the pdf transformation from the emulator and the RCM over the high Pyrenees points for all ALADIN63 simulations.

Conclusion on the main emulator transferability

This section has been dedicated to evaluate the emulator on various simulations according to multiple aspects. It was lengthy, but I believe it is necessary to have better confidence in the tool introduced in Chapter 3. The strong take-home message here is that the emulator is robust to other inputs as it provides simulations close to original RCM ones. We have shown here that the emulator shows as good performances on intermediate scenarios as on the training set for a given driving emulator, confirming its truthfulness for the member and scenario level of transferability.

The results are still very accurate concerning the GCM transferability, showing the robustness of our methodology to eventual differences in the representation of the atmospheric conditions. Notably, the emulator showed excellent accuracy in reconstructing the complex spatial structure on the climatologic and quantile maps across time and on the climate change maps. It shows, nevertheless, some limitations in reproducing some extremes, especially in a globally warmer simulation. The training set is probably an important factor here. We have seen that the simulations drove by CNRM, used for training the main emulator, are generally colder than the three other GCMs and it warms less. The emulator had to extrapolate on those and showed its limits. The next section will be helpful to answer this question. Some of these limitations might also come from memory effects, notably in the soil (snow or humidity), not taken into account or reproduced by the emulator. For instance, it has been shown in various studies that a deficit in soil moisture leads to stronger summer hot extremes (Seneviratne et al, 2010; Hauser et al, 2017). As the emulator downscales each day independently, there are no chances it takes this effect into account. It could explain the less good performances in representing the highest percentile.

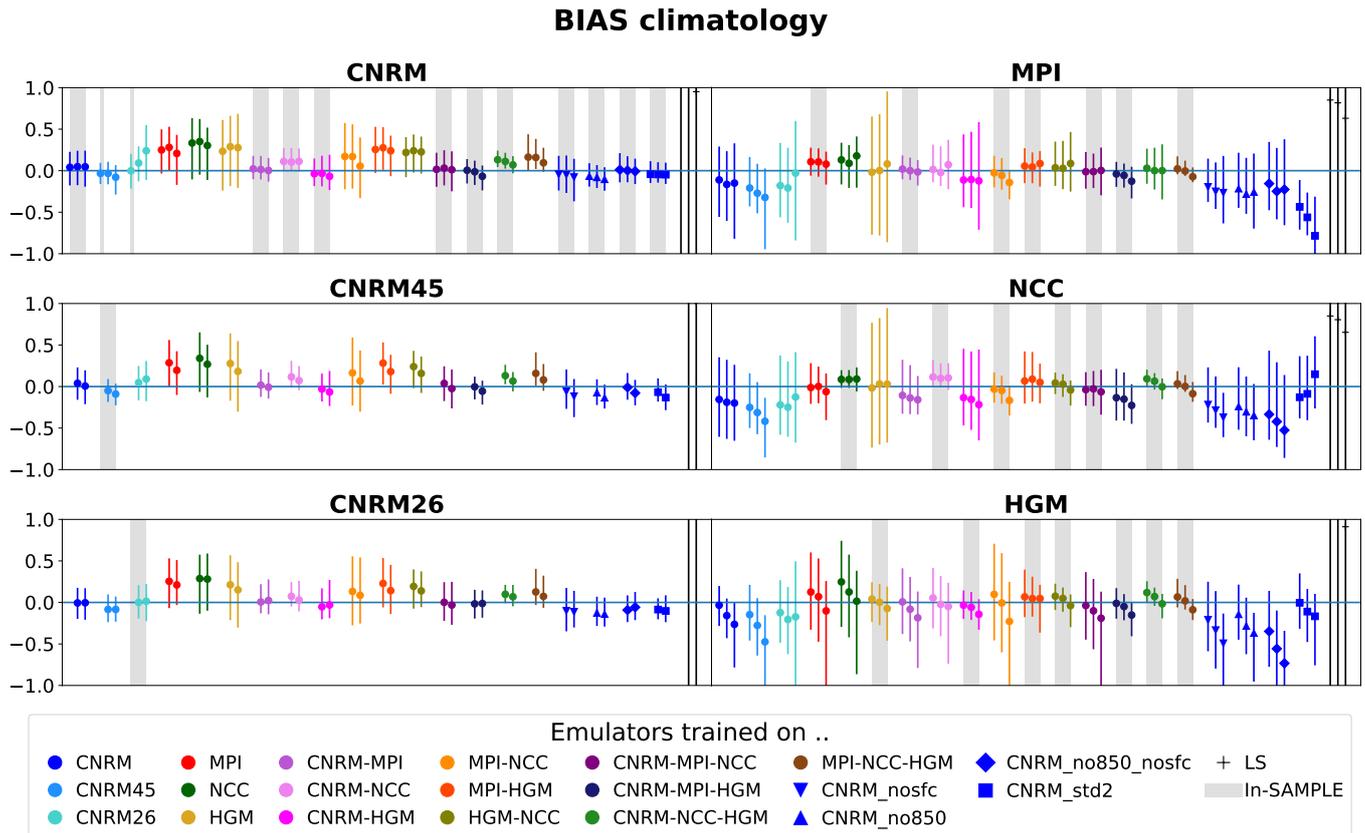


Figure 4.12: Summary plots for the mean biases maps, in the cross validation experiment. The panels corresponds to the evaluation simulation. Each color corresponds to the different emulators: it indicates the training simulations. The marker shape corresponds to the version of the emulator: the 'o' is the main version while the others are sensitivity test described in Sec. 4.5.1 and 4.5.2. The black 'plus' at the extreme right is the Large scale near surface temperature (I-UPRCM). The grey columns indicates if the evaluation simulation is part of the emulator training set.

4.3.2 Cross-Validation on other training sets

In the previous section, we evaluated the emulator trained on CNRM HISTORICAL and RCP85 simulations on its capacity to reproduce all other ALADIN63 simulations. It is now interesting to see if these results are comparable when we use a different training set. In this section, I train 15 more ALADIN63-emulators using different combinations of the available simulations. First, on top of the main emulator, one emulator is built using each MPI, NCC and HGM 150 years of simulation (HIST+RCP85). Secondly, all possible combinations of two or three of these four simulations are used to calibrate one emulator. I also added emulators trained on CNRM Historical plus RCP45 and RCP26, which will allow us to assess the scenario transferability through warmer scenarios. All these emulators are then evaluated in the perfect model framework following the same protocol as in the previous section on all ALADIN63 simulations, including the calibration set. Indeed, we have seen in the previous section that the results obtained when emulating the training set are quite representative to what could be obtained on simulation driven by the same GCM and same or close scenario. In this section the use of CNRM, MPI, NCC and HGM acronyms refer to the corresponding

ALADIN63 simulation on the entire 150 years combining HISTORICAL and RCP85 scenario.

In this section, we will look at the biases of the emulators versus the RCM true series for the mean climatology and the 99th and 1st quantiles, the variance ratio and Wasserstein distance. The results are reported using summary plots in Figures 4.12, 4.13, 4.14. Each figure presents the results for a given metric. Each figure is composed of 6 panels, where each panel corresponds to the evaluation simulation. One colour represents one emulator, and the shaded rectangles identify the emulator trained on the concerned evaluation simulation. For each emulator, on each panel, I plotted consecutively the past (except for CNRM RCP45 and RCP26), present and future periods. These figures contain a lot of results so we will discuss them step by step.

One simulation based emulators

First, we can focus on the emulators trained on CNRM45 (Emul-45) and CNRM26 (Emul-26). They globally show similar results as the main emulator. Regarding the climatology bias maps in Figure 4.12 the three CNRM emulators offer identical performances to reproduce the three CNRM simulations in the past and present climates. It confirms that, for perfect input conditions, an emulator train using a given driving GCM and warming scenario can perfectly reproduce the RCM temperature simulation driven by the same GCM and at the same or more intermediate level of warming. It also confirms that our training methodology prevents over-fitting, and the training set could reasonably be considered as an evaluation set. It is observable, however, that the Emul-26 performs less well in the future period of both RCP45 and 85 simulations. The same remark can be made to a smaller extent for the Emul-45 on the future RCP85. It appears more evident on the 99th quantile summary plot (Figure 4.13), where Emul-26 has considerable local biases. This result is expected as the emulator is not supposed to extrapolate and this test is similar to training only on the historical part (Sec 4.2 Chap1). Regarding all metrics, Emul-45 performs less well than the two others and even on its calibration set. It notably underestimates the local temporal variance leading to biases on the quantile maps. The explanation have not properly been explored here and a hidden bug is not to exclude. It might also be due to the randomness of the training mentioned in Sec.X of Chapter 3, as the scores are consistent with the range of emulators built in this same section. Considering these observations, it is worth noting that these three emulators have similar performances on the three other simulations. For example, they have similar errors on the Wasserstein distance plots in Figure 4.14.

Secondly, we can look at the three other emulators trained on single driving GCM: Emul-MPI, Emul-NCC and Emul-HGM are trained using the upscaled ALADIN63 simulations driven by the MPI, NCC and HGM GCMs. They are represented in red, green and yellow on the summary plots. First, each performs well on its own training set, with the same defaults on the extremes as the main emulator. Each bias map shows very low local biases (visible from the size of the error bar), and quantiles are slightly under or over-estimated. The Emul-HGM has less good performance, with

a more extensive spatial range on all metrics. Looking now at their results on the other simulations, the first remarkable point is the similarity between the Emul-MPI and Emul-NCC. They have comparable scores on each other simulations and seem to produce similar maps on the other simulations as it is particularly remarkable on the 99th quantile summary plot (Fig 4.13). Emul-HGM shows poorer results on MPI and NCC simulations with much stronger local biases on the climatology and 99th quantiles maps and overestimating the temporal variance at some points (Fig 4.14). Finally, it is worth noting that these three emulators behave similarly on each of the three CNRM simulations. For instance, on the three simulations, they all have a slight warm bias on the climatology map and capture well the average 99th quantile but with significant local biases. These biases are coherent with the cold bias of the main emulator on these three simulations in the previous section. The conclusions on these three emulators are very close to the ones in Section 4.3. All these emulators are robust to other input sources, but they might produce some biases, mainly when the simulation to downscale represents a too different climate. Finally, they all reproduce adequately the climate change maps, which are not shown here but it is remarkable that eventual biases are consistent through time.

It seems reasonable to conclude here that for a given RCM, the emulator will present similar performances whatever the simulation used for training. Emul-MPI and Emul-NCC behave analogous than the Main emulator. They have excellent results regarding the entire distribution of data as long as the input space to downscale isn't too far from the training one. The more it deviates, the more complicated it is reproduce the series and notably the extremes. Emul-HGM has globally less good performances. It highlights that using a single simulation for training might be risky. It would be necessary to explore further why this specific emulator has different results. A possible explanation is that it is the GCM simulation that ALADIN63 transforms the most. In section 4.2.3, we showed for instance that HGM-GCM was warming way more than HGM-UPRCM across the century. This might have created some atmospheric conditions different from the three other ALADIN63 simulation.

Multiple simulations based emulators

It is now natural to question what happens when two or more simulations driven by different GCMs are used for calibration. As we have four different driving GCMs, we used a maximum of three historical plus RCP85 simulations for training to keep one out of the training sample for the evaluation. Using different simulation from different driving GCM helps to enlarge the input space in the training phase. It is closer to the reality of a CORDEX matrix where the RCM are used to downscale multiple GCMs. In order to build the best emulator one can legitimately imagine that it is better to use all available simulations to train it. This experiment allow to test this aspect. On the summary plots, the shaded area shows the emulators including the evaluation simulation.

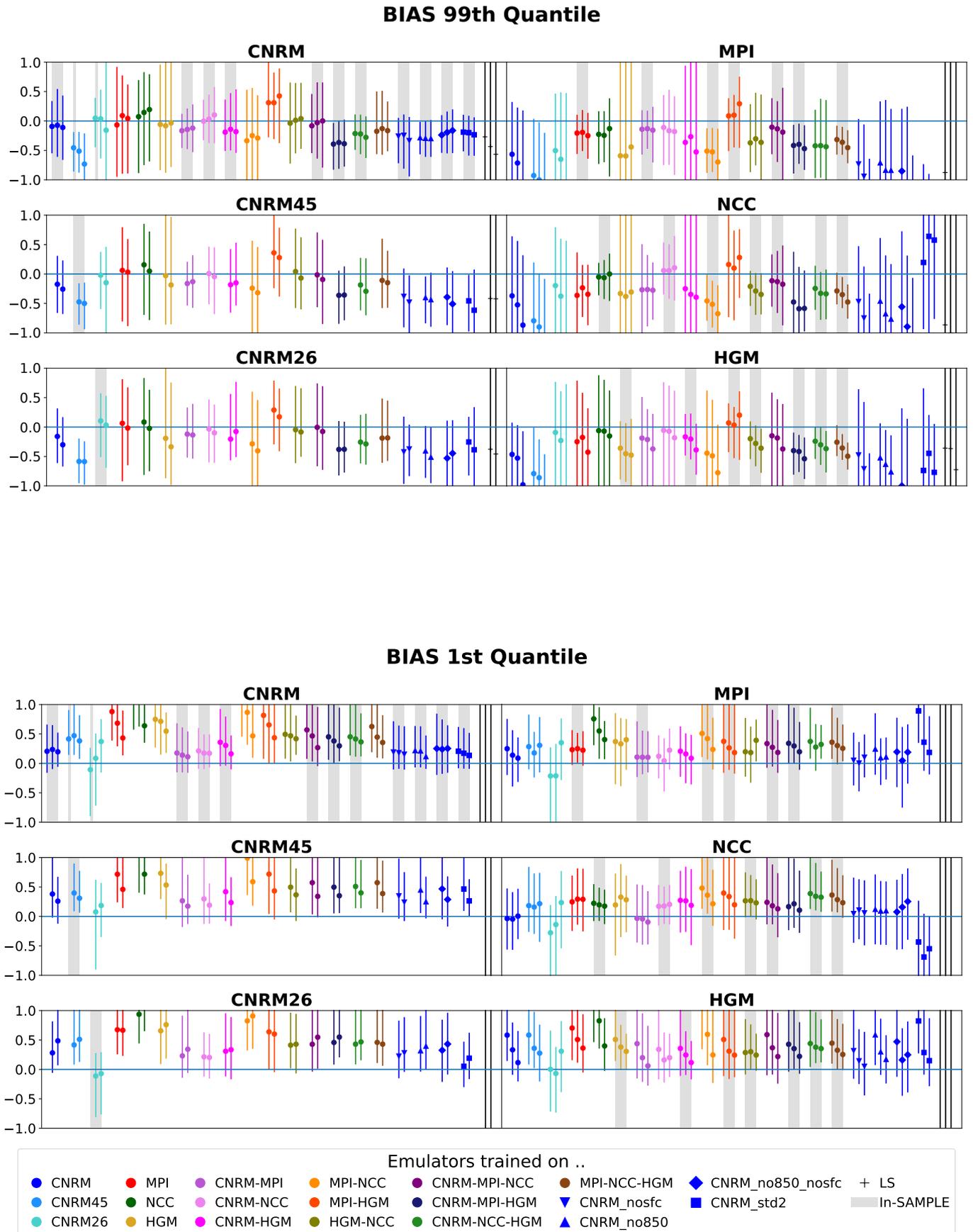


Figure 4.13: Same as Fig 4.12 for the 99th and 1st quantile biases maps.

The first comment is that even when the driving GCM has been used for training, the results are similar or better when the emulator is trained on more simulations. It is evident on the Wasserstein and variance plots (Fig. 4.14) that on the grey areas, the range of the 5% most extremes points is getting smaller when going to the right of the colourful bars. The conclusions are more challenging on the actual evaluation set, i.e. the white areas on the plots. But when looking only at the CNRM evaluation simulations, the more we use simulations for training, the better the reproduction of the original RCM series. It is also the case for the HGM series, even if it is the most difficult to reproduce and the results are less good. Regarding the MPI simulation, the presence of the NCC simulation is of primary importance to improve the results, and the inverse is also true. However, it still comes that having more than one simulation for the training improves the outcomes. For example, the Emul-CNRM-HGM is better at reproducing MPI and NCC than Emul-HGM. On the other hand, when two simulations are similar, it brings nothing to combine them for training. Indeed Emul-MPI-NCC does not show significantly better results than Emul-NCC or Emul-MPI.

Cross experiment conclusions

There are three important conclusions from this cross-validation experiment.

- **The scenario transferability works better from more extremes to intermediate scenarios than conversely.**

The main emulator is able to reproduce the CNRM RCP45 and 26 very well. The results are even comparable to the ones obtained on the calibration set. However, Emul-RCP45 and especially Emul-RCP26 showed difficulties to reproduce the end of the RCP85 simulation. It implies that training the emulator using a single GCM and the most extremes scenarios allows to downscale at least all simulations of the same GCM driven by similar or less warming scenarios.

- **All conclusion on the main emulator are still valid when using another simulation for training.**

It seems that, in most cases, the conclusion made regarding the Main in Chapter 3 and in section 4.3.1 are deployable to an other emulator trained on different simulations.

- **Having more than a single driving GCM for the simulations used for calibration helps for the GCM-transferability.**

We have seen that increasing the number of simulation for training the emulator enlarge the input space and generally improve the emulator robustness. At worst it did not improve the results when they were already good. On the other hand it helped to correct the bad poor results of the Emul-HGM.

On top of bringing more confidence in the emulator methodology, this section gives insight on how to design a matrix of RCM simulations to train the best emulator.

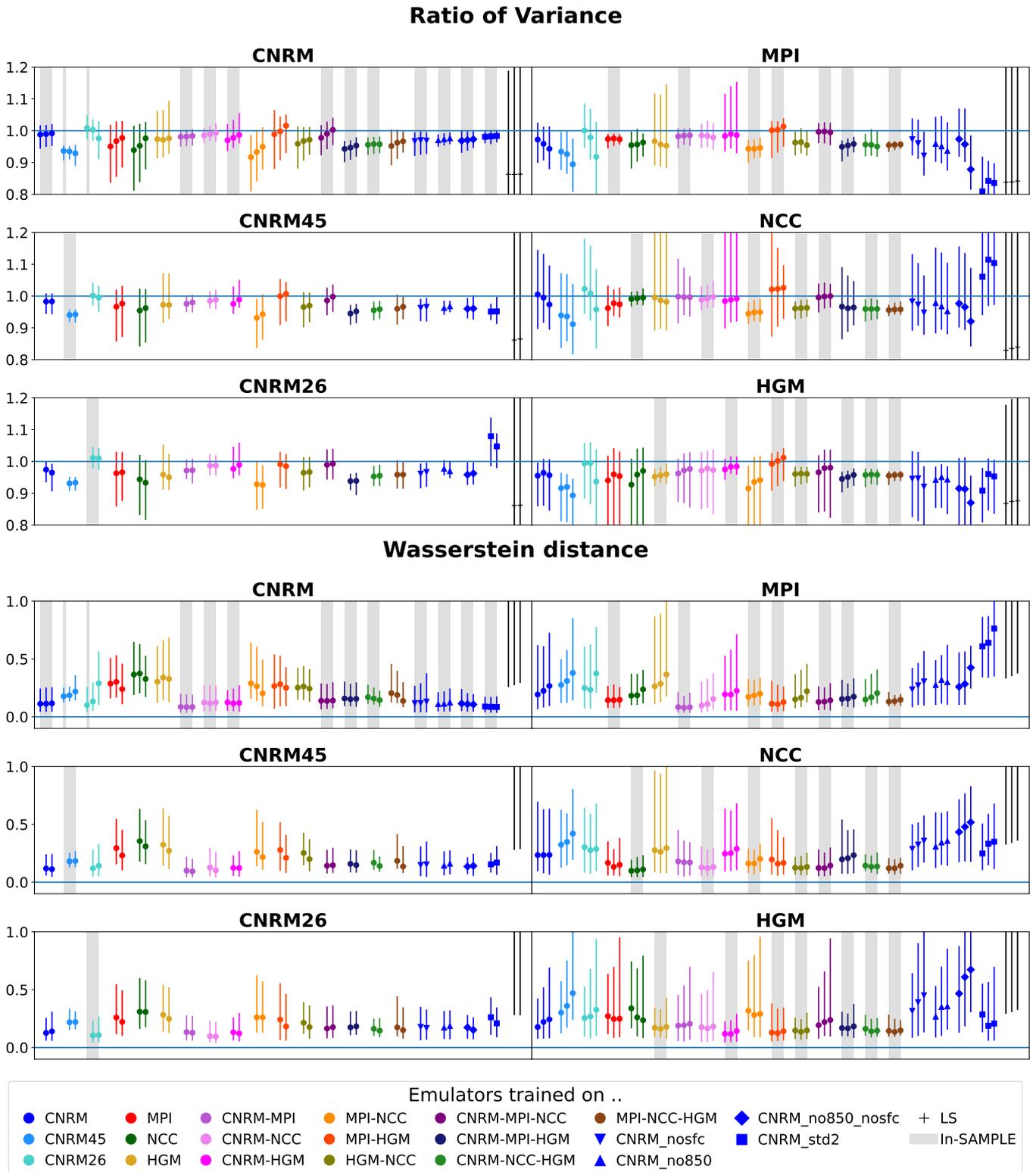


Figure 4.14: Same as Fig 4.12 for the Variance ratio and Wasserstein distance maps.

4.4 Application on GCM data

In chapter 3, I tried to evaluate the main emulator when downscaling a GCM simulation. I downscaled the CNRM RCP45 simulation with the emulator and compared the result with the ALADIN63 simulation driven by it. The emulator was producing a coherent simulation, with a precise reproduction of the high resolution spatial structure and daily variability. However, it had a warm bias, which we found consistent with a large scale bias between the GCM and UPRCM inputs. Indeed the emulator came out to logically follow its driving large scale without reproducing the RCM large scale transformation. As there are daily but also statistical differences between the GCM and the RCM large scales, I concluded in Chapter 3 the RCM true series is not the correct reference to evaluate the GCM-emulated series. It is then truly complicated to assess the emulator in this exercise, whereas it sounds necessary to have complete trust in this downscaling approach as downscaling GCM simulation is the final aim of RCM-emulators. I propose looking at the downscaling of the GCM simulations used to drive the ALADIN63 simulations. I will try then to evaluate the emulated simulations by comparing them with the RCM truth and keeping an eye on the GCM near-surface temperature interpolated on the target domain. I remind the reader that I am missing the historical part of the HGM GCM simulation.

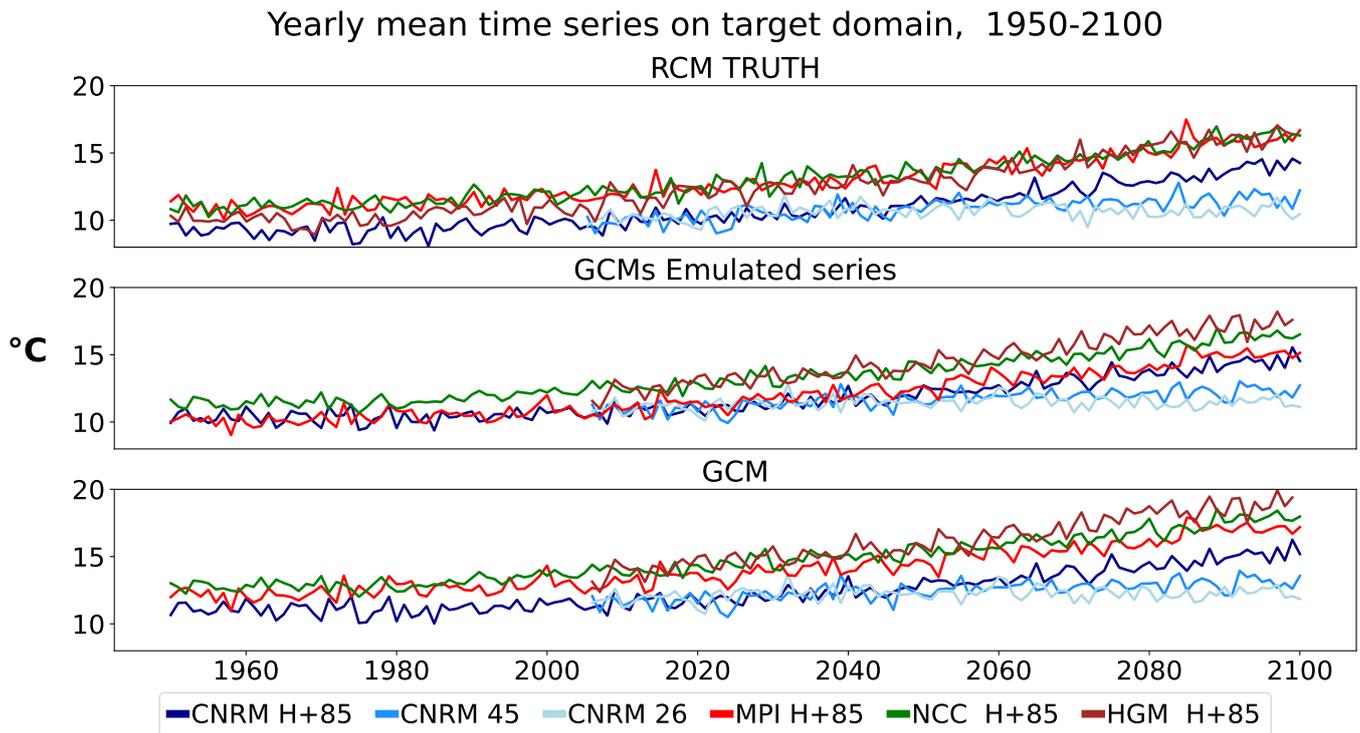


Figure 4.15: Yearly time series over the whole domain for the 6 simulations, for the RCM true series, the GCM-driving main emulator series and the GCM series.

Figure 4.15 shows the yearly mean time series over the target domain for the RCM, the GCM-Emulated and the interpolated GCM. The RCM and emulator series are similar, but two significant differences arise. Firstly, the emulated HGM series is much warmer than the HGM RCM simulation all along the century, especially at

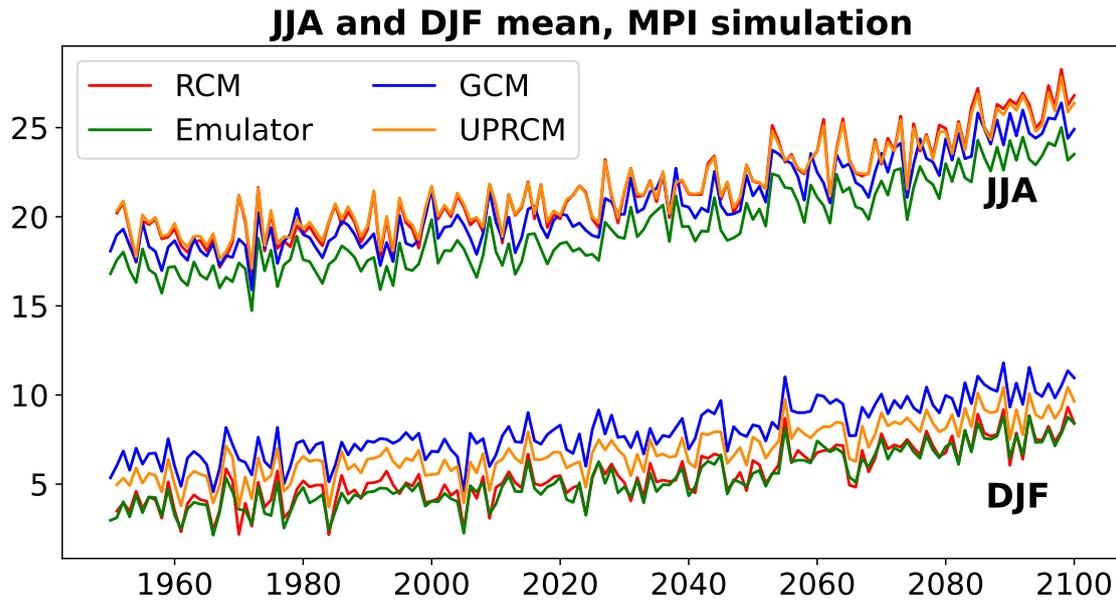


Figure 4.16: Summer (JJA) and Winter (DJF) mean time series for the RCM (red), the Emulated series from the GCM (green), the GCM (blue) and the UPRCM (yellow) over the target domain.

the end since the emulator produces more intense warming. The emulated series follows the GCM and seems, in this aspect, more coherent than the RCM series. The HGM-GCM simulates a warmer climate at the end of the century than the three other GCMs over our target region, and the ALADIN63 simulations did not reproduce this.

Conversely, the emulated MPI simulation may be colder than it should be as it is closer to the emulated CNRM than the emulated NCC, while it is the opposite in both GCM and RCM series. It is particularly visible on the historical part. It might be explained by a cold bias of the emulated MPI in summer. Figure 4.16 shows the winter and summer time series means for the MPI simulation. It comes that the emulated MPI is much colder than the RCM series in summer. It can, at least partially, be explained by the large scale as the GCM is also colder than the RCM. Moreover, in summer, RCM and UPRCM are very close, with the UPRCM being sometimes slightly colder while the emulated series is much colder than the GCM. It is, of course, very complex to interpret. The low resolution near-surface temperature is not a predictor given to the emulator, and the UPRCM mean is naturally very similar to the RCM mean. Thus, many good reasons could explain the bigger differences between the GCM and emulator. For example, the GCM-MPI has a poorer resolution than the CNRM grid (where we interpolate all inputs), implying a poorer representation of the relief. Nevertheless, this cold behaviour of the emulator in summer for the MPI simulation could be suspect and necessitates more investigation to be understood. Here is probably the biggest challenge for the emulator evaluation as it is complicated to evaluate without a truth series as in perfect model evaluation.

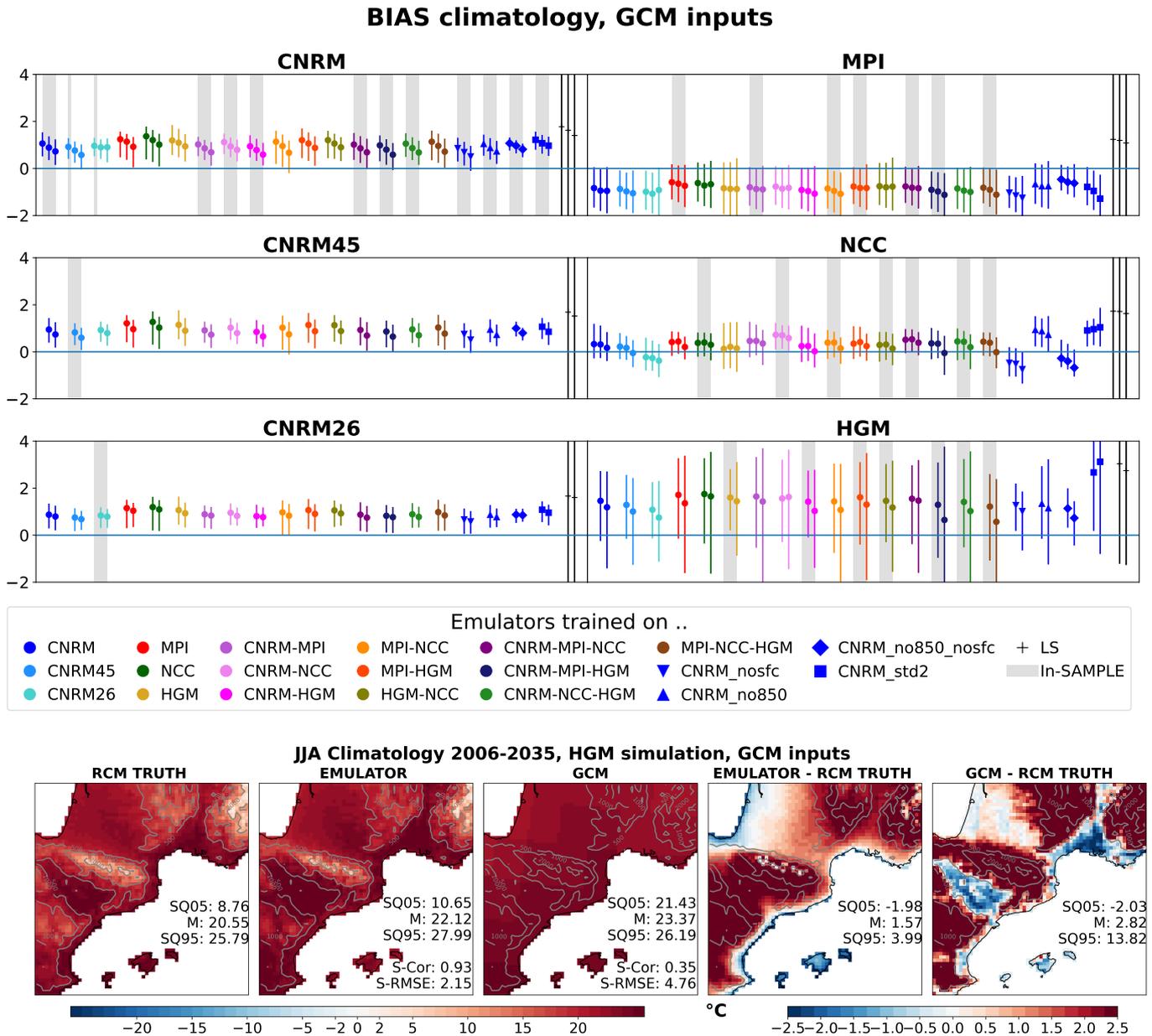


Figure 4.17: Upper panel: Summary plots for the bias in mean climatology maps, past (1970-2000), present (2006-2035) and future (2071-2100) periods. Lower: Climatology maps for the HGM simulations with the RCM simulation, the GCM-emulated simulation CNRM emulator and the GCM simulation. It also shows the biases.

I used all emulators built in Section 4.3.2 on these 4 GCM simulations to evaluate the training set's impact when downscaling GCM simulations. Figure 4.17 shows the bias summary plots in the climatology maps. The reference is the corresponding RCM temperature. Interestingly, all these emulators have similar biases, as shown by the spatial means and superquantiles. Surprisingly, training on the corresponding ALADIN63 simulation does not impact the performances. I show only the results for the climatological bias here, but the conclusion is precisely the same on all other metrics used in the previous sections. Globally we can see that the emulated series have larger mean and local biases with greater superquantiles than in the perfect model. The most substantial differences appear in the HGM

simulation, especially in the summer, as shown in Figure 4.17. The emulated climatology is warmer over land and relief while there is a cold bias over the coasts. This cold bias could come from missing sea surface temperature information. On the other hand the warm bias seems consistent with the large scale. However, we can see here again that the climatology map from the emulated series is coherent with a good representation of the high-resolution spatial structure brought by the RCM.

These similar results on all emulators show that the downscaling function learnt by the emulators is globally similar on all training sets, and the emulator’s production always stays consistent with its driving large scale. It confirms the robustness of the emulator, and if some issues need to be fixed, notably regarding some biases, the general result here is that the downscaling of GCM simulations with the emulator works and provides the adapted high resolution information of climate according to the given large scale.

4.5 Discussion

4.5.1 On the standardization method

In Chapter 3 of this manuscript, I presented my choice for the standardization method used to prepare the inputs for the neural network. I chose to separate the temporal and spatial information into two sets of inputs. The first set regroups all 2D variables (reported in Table 1chaptab1) normalized according to their daily spatial mean and standard deviation. It results for every day in a collection of maps, all centred in 0, highlighting the spatial anomalies or structure of the given day. The second set contains somehow temporal information. It is a 1D array including the collection of daily means and standard deviations used to normalize the 2D variables, the yearly greenhouse gas concentration used to drive the RCM simulation and a vector including a cosine and a sinus representing the day of the year. This set of 1D variables is also normalized according to its climatology over 30 years. To summarize, for each day, the set of 1D variables indicates the position of the day in comparison to others in terms of global climate (through the GHG concentration), its place in the year and somehow an indication of the climate variability mode over the input domain through the set of spatial mean and standard deviation. The 2D maps bring information on the daily spatial structure completing the information on the climate mode of the day.

I made this choice because of the robustness to new inputs. Indeed, it is well established that neural networks (ref) and especially convolutional neural networks are very sensitive to input space changes. As shown in section 4.2.3, the relationship between different variables and altitude levels can differ between 2 GCMs. Figure 4.4 shows these differences at the input domain scale. They are probably bigger when we look at the grid point scale, especially because two GCMs have different horizontal resolutions. My intuition here was that this normalization would help us to prevent this effect.

A probably more intuitive way to normalize the inputs is to do it at the grid point

Table 4.2

| NAME | DESCRIPTION with respect to the main emulator |
|------------------|--|
| CNRM_no850 | No variables at 850 hectopascal |
| CNRM_nosfc | No surface variables |
| CNRM_nosfc_no850 | No variables at 850 hectopascal & No surface variables |

scale. I constructed an emulator called CNRM_std2 to test our standardization strategy and my hypothesis. This emulator is, in every aspect, trained in the same way as the main emulator as it takes the same set of inputs. It is trained on the same simulation (CNRM HIST+RCP85) and uses the same neural network. The only difference is the normalization procedure since each daily 2D map is directly normalized by the mean and standard deviation maps computed over a 30 years reference period of the corresponding variable. Following this, the set of 1D inputs contains only the external forcings (GHG, Solar and ozone constraints) and the cosinus sinus vector for the day of the year. The results are reported in all plots from the cross-validation exercise (Fig. 4.12,4.13,4.14). The emulator CNRM_std2 results are very similar to our main emulator on the CNRM simulations, with sometimes slightly better results regarding the climatology plots, for example (Fig, 4.12). However, regarding the three other simulations driven by different GCMs, the main emulator performs better than the novel one. It is obvious for all metrics on the MPI simulation, which seems to be the most different from the CNRM one in Figure 4.4. To conclude here, the CNRM_std2 has, in this perfect model evaluation, stronger biases and a less good representation of the variances and pdfs on the simulation driven by different GCMs, which tends to confirm the hypothesis that the two steps normalization helps for the robustness of the emulator.

4.5.2 Some tests on inputs choices

The large set of simulations to evaluate the emulators is a good opportunity to make small tests on some input choices from chapter 1. For instance, I showed in Chapter 1 that including the surface temperature in the inputs set improves the emulator's performance in the perfect model evaluation but seemed to degrade the downscaling of GCM inputs. This is logical as the UPRCM near-surface temperature might still contain high-resolution information that the GCM near-surface temperature cannot include. However, I decided to keep the surface wind and sea pressure level, and this choice could be questionable. In the same spirit, I wanted to test whether removing the variable at 850hpa would change something, as it is still close to the surface, especially over the relief. To test these two points, I built three emulators described in table 4.2. As in Sec.4.5.1, the results are reported in the figures from the cross-validation and GCM sections (sections 4.3.2 and 4.4). It seems that excluding these variables slightly degrade the results, especially regarding the 99th quantiles (Fig. 4.13) or the variance ratio for the MPI, NCC and HGM simulations. Regarding the results on the GCM simulation, removing these variables could have improved the results even if it is difficult to evaluate. The impact is small, but it seems to reduce the spatial range of differences in the climatology maps around a global bias. It is not shown here, but the same comment can be made on the maps of variance ratio and Wasserstein distance, especially when both levels (surface and 850hPa)

are removed. This result should be kept in mind for future development and use of the emulator, but it has not been deeper investigated in this work.

4.6 Conclusion

In this chapter, I tried to address the overriding question of the emulator's transferability to other input sources. The aim of the emulator is to downscale large ensembles of simulations. In the previous chapter, I introduced the emulator's concept and methodology. I showed that it can reproduce a simulation close from the training one (same driving GCM, intermediate scenario of GHG). However, ensuring that the emulator works well for downscaling any GCM and any scenario is essential. ALADIN63 is an ideal case study for the emulator since there are three simulations driven by the same GCM and three different scenarios of GHG concentration, and four simulations driven by four different GCMs following both the historical and the RCP85 scenarios.

As in the previous chapter, I address this question in the perfect model framework where the inputs come from upscaling and smoothing the RCM simulations as described in Sec.4.2.3 in Chapter 3. This question of transferability remains a challenging exercise in the perfect model world, as the driving GCM strongly impacts the relationship between the input variables. Furthermore, these differences between the different UPRCM simulations are comparable to those observed in GCM simulations. Therefore, the perfect model framework is a good opportunity to evaluate the emulator's robustness under other inputs.

Firstly, I evaluate the emulator built in the last chapter on all available simulations. The results are globally excellent. The emulator produces the high resolution structure and variability brought by the RCM. Furthermore, it recreates the differences between all the simulations regarding temperature levels and warming strength at the entire domain, regional and grid point scales, even on the 1st and 99th quantile change maps. For instance, I show that each simulation simulates different density functions over the Pyrenees and transforms the regional climate differently across time. The emulated series replicate all these observed differences. This result gives us solid confidence in the tool, even if it probably could be improved notably in reproducing the most intense extremes. Indeed, I notably showed that the emulator underestimates the warmest extreme in a simulation globally warmer than the training set and it reflects in the mean climatology.

In a second step, I propose a cross-validation experiment to evaluate the impact of the training set on the emulator's performance. The ALADIN63 matrix is also an excellent opportunity to build multiple emulators trained on different combinations of the available simulations. This experiment strengthens the robustness of our methodology as all emulators show akin results. They all correctly recreate each simulation, with some biases or defaults but all in the same order of magnitude. There is nevertheless a little damper as the emulators trained on the HGM ALADIN63 simulations are significantly less good. The other conclusion from this experiment is the importance of having the greatest spread in the training set, including the most extreme scenarios and, when possible, various driving GCMs.

The aim is to cover the most extensive range of climates to minimize the risk of extrapolation in application mode. Moreover, it also prevents using a too specific simulation, as was the case here for the HGM simulation.

Finally, I tried to validate the emulator in application mode using GCM inputs. The emulator was able to downscale these GCMs simulations and provide consistent results. It was particularly interesting to see that the emulator recreates the strong sensitivity of the HGM simulation better than the RCM. However, the large scales biases between GCM and RCM make the evaluation very hard since there is no truth to confront the emulated series. In particular, the emulator seemed too cold in summer in the MPI simulation. Still, it is difficult to assess if it is coherent with the driving large scale or if there is an issue in the emulator construction. A potential limitation of the emulator which could lead to such biases is the absence of soil variables, such as soil moisture or snow accumulation, which can carry a strong memory effect not represented in the atmospheric variables. These variables are less well represented in the GCM, so it is impossible to use them directly as predictors. Therefore, I think this is a possible improvement method that should be investigated.

This chapter gives a lot of confidence in the emulator for the daily near-surface temperature to apply it to actual GCM data and create big ensembles of high resolutions temperature series. Of course, the emulator still needs improvements, notably regarding the extremes, but its current version provides already a robust basis for the downscaling of new GCM simulations. Moreover, the results of this transferability study could also be used as guidelines for future RCM simulations in order to give the best chance to complete the matrix with RCM-Emulators. In the conclusions of the manuscript (Chapter 6) I will give more details on this perspective.

Chapter 5

Precipitation downscaling with the RCM-Emulator : The role of the loss function

This chapter presents the use of the emulator for the downscaling of the daily amount of precipitation. It has been written as an article to be submitted. Therefore, some parts of the introduction and the methodology sections might be redundant with the previous chapters (notably Chapter 3).

5.1 Introduction

Precipitations are the principal source of usable fresh water on Earth. It is a key variable to study and predict because of its fundamental role in the Earth's system balance, in the ecosystem's life, and above all, human survival and activities rely directly on it. On the other hand, it is a potential source of catastrophic events. Heavy rainfalls are responsible for floods devastating cities and cultures. Severe droughts can cause colossal damage to ecosystems and agricultural activities and limit access to drinkable water. In the current context of global climate warming, the eventual changes in precipitations climatology and extremes are crucial to study for adaptation policies.

The study of rainfall is very complex. Precipitation is a non-continuous variable neither in time nor in space. It is necessary to look at precipitation series through different time and spatial scales to completely understand their nature. Their occurrence and intensity characterise them. For example, the monthly mean precipitation amount indicates seasonal cycle or large-scale variability but could underline very different situations. Moderate and regular rain or a dry spell followed by heavy precipitation will result in the same monthly mean but will impact the local ecosystems and hydrological cycle very differently. Precipitations can also be very local as they might strongly depend on the local topography, the distance to the sea or a lake, etc., and on small scale physical processes such as convective instability or cold pool (Caillaud et al, 2021). It is therefore necessary to study them at refined resolutions.

Global Climate Models (GCMs) represent the global Earth system. They are today the primary tool to study the impact of human greenhouses gases emissions on the climate. They divide the globe into 3d grids. Each cell represents a specific location with its characteristics (altitude, kind of surface, etc.). A set of physical equations is resolved at regular steps in each cell and between cells to represent the different processes driving atmospheric circulation. The horizontal size of the cell is the resolution of the model. Finer resolution implies more grid points and higher computational costs. GCMs have a resolution between 50 and 300km, according to the model. It is too coarse to accurately represent the precipitation in all its complexity, especially when looking at the most extreme events. For these reasons, multiple studies have shown the added value of Regional Climate Models (RCMs) for representing precipitations, especially extreme events (Torma et al, 2015; Giorgi et al, 2016; Fantini et al, 2018).

RCMs are similar in all aspects to GCMs, except that they represent only a limited area domain allowing a higher resolution (1 to 25 km). An RCM simulation receives at regular time interval information from a GCM simulation at the borders of its domain. The global circulation is then simulated by the GCM and passed to the RCM, which integrates it and produces a higher resolution simulation. The RCM is then used to downscale the GCM coarse resolution and we talk about dynamical downscaling (by contrast to the statistical downscaling introduced below). We say that the GCM drives or forces the RCM. The main limitation of RCMs is their high computational costs which do not allow for downscaling large ensembles of

GCM simulations, although it is essential to produce reliable information. Various studies (Déqué et al, 2012; Evin et al, 2021) have pointed out at least four sources of uncertainty associated with RCM simulations: the choice for the RCM and GCM with the defaults they carry, the uncertainty around the evolution of human activities and especially the emissions of GHG and finally, the internal variability of the climate. In order to deliver a reliable message about future climate, it is necessary to run numerous long simulations from different RCMs, driven by different GCMs and themselves forced by multiple economic scenarios. Multiple simulations with the same Scenario, GCM, RCM triplet are then necessary to differentiate the impacts of climate change and the internal variability.

In this study, I propose testing whether the RCM-emulator introduced in Chapter 3 is suitable for emulating daily precipitation in the RCM ALADIN63 (12km resolution). The concept of the RCM-emulator is to learn, with machine learning tools, the relationship between low-resolution large-scale variables describing the atmosphere circulation of a given day and a local high-resolution surface variable, such as precipitation daily amount. This downscaling function is learnt inside existing RCM simulations. The aim is to tackle the cost limitation of RCM by mimicking its downscaling function for a specific variable at a low computational cost and then by applying it to any GCM simulations. RCM-Emulators belong to the hybrid downscaling methods as they involve both statistical and dynamical downscaling. Statistical downscaling aims to estimate or reproduce the same large-scale/local-scale relationship with statistical methods. The fundamental difference is that statistical downscaling uses historical observation data to learn the downscaling function. By using historical and future RCM simulations in the training set, the RCM-emulator is more robust to the possible non-stationarity of this relationship. Moreover, it can also be applied over regions with no long series of observational data.

Various studies have proposed statistical downscaling methods to downscale GCM or RCM simulations. Maraun et al (2010) and more recently Gutiérrez et al (2019) present an overview of available statistical methods for downscaling precipitations. Some very recent studies (Baño-Medina et al, 2020, 2021; Vandal et al, 2019; Wang et al, 2021) have successfully used convolutional neural networks for this purpose. The RCM-emulator used in Chapter 3 and this study relies on a convolutional neural network architecture called UNet (Ronneberger et al, 2015). It has shown an excellent ability to emulate the temperature, notably in reproducing the complex spatial structure and daily variability brought by the RCM.

The second objective of this study is to look at the impact of the loss function on emulator prediction. If precipitations are complex to study, they are also challenging to model. Their intermittent nature implies a substantial variability in time and space and high non-linearity. Baño-Medina et al (2020) undertake this issue with a well-designed probabilistic neural network. It outputs the three parameters of a Bernoulli-Gamma distribution for each day and grid point. Wang et al (2021) use a GAN neural network architecture, which produces for each day a possible realisation. Here, I will compare two standard loss functions for regression problems with a third designed explicitly for the precipitation

downscaling task. It relies on an asymmetric regularisation parameter which penalises stronger the network when it under-evaluates higher precipitation intensities.

As in the rest of this manuscript, I train and evaluate the emulators using a perfect model framework. It implies that both predictors and predictands (precipitation here) come from the same RCM simulation. The target domain here is centred over the Alps and is four times bigger than in the preceding chapters. It will also help to address the emulator’s transferability to another domain and the needed computational time for a larger target domain. This chapter contains four main sections. In Section 5.2, I present the framework of the study: the data, the target domain and the associated predictors. I will also recall the neural network architecture and the concept of perfect model training and present the three losses functions used to calibrate three emulators and the evaluation tools I will use. Section 5.3 presents the detailed evaluation of the emulators in a perfect model framework, while Section 5.4 shows the results of the asymmetric emulator application on GCM simulations. Finally, in a discussion part (Section 5.5), I come back to the construction of the asymmetric loss function and the computational performances of the emulator.

5.2 Methodology

In this section, I set the framework used to build and evaluate the RCM emulator for precipitation. I firstly present briefly the simulations used in this work. Then I present the emulator base: the chosen target domain and predictors and the neural

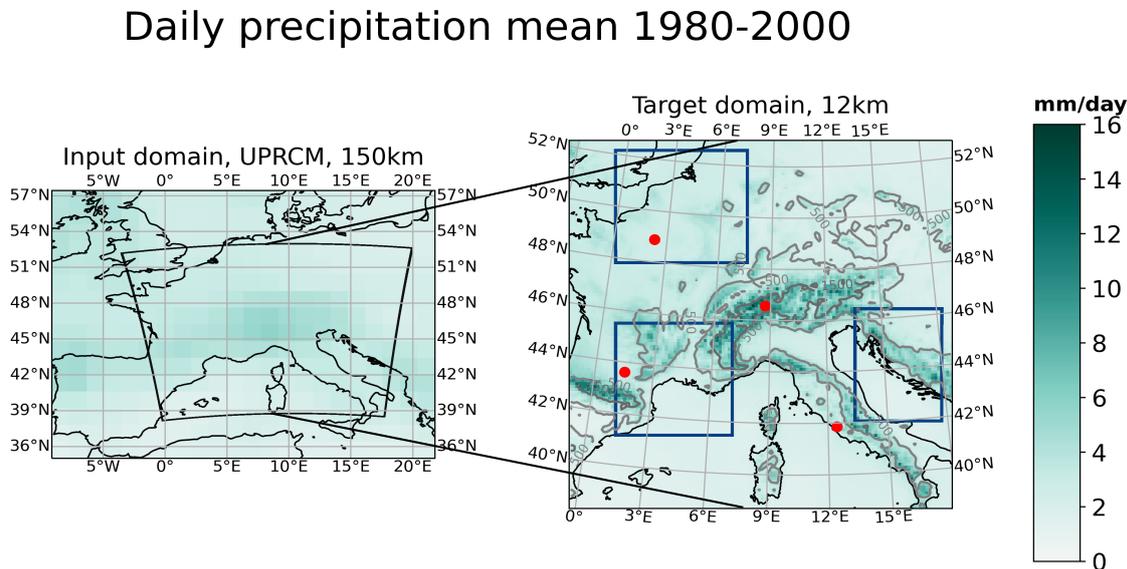


Figure 5.1: Illustration of the input (left) and target (right) domains through the climatology of the daily rainfall over the 1980-2000 period. The black line on the left panel shows the target domain while the input domain is the entire map. On the target domain: the red points are the 4 illustrating points on Figure 5.7 and 5.12. From North to South, there is Paris, a high point (2247 meters) in the Swiss Alps, Toulouse and Roma. The three blue boxes are the three regions used for the SAL evaluation in section 5.3.2: The north region, centred over Belgium, the Cevennes region (south-east France) and the Dinaric Alps.

Table 5.1: The ALADIN63 matrix.

| DRIVING SCENARIO | DRIVING GCMS | | | |
|------------------|--------------------|---------------------|--------------------|---------------------|
| | CNRM-CM5 (CNRM) | MPI-ESM-LR (MPI) | NorESM1-M (NCC) | HadGEM-ES2 (HGM) |
| Historical | x | x | x | x |
| RCP26 | x | | | |
| RCP45 | x | | | |
| RCP85 | x | x | x | x |

network architecture. I will present the three loss functions used to train the three emulators for the inter-comparison. I will also shortly recall the perfect model framework approach used to train and evaluate the emulators. Finally, I will detail all the metrics used to evaluate the emulator under different aspects.

5.2.1 Datas : the ALADIN63's matrix

The emulator proposed in this study relies on the regional climate model ALADIN63 (Nabat et al, 2020). A total of ten simulations have been published with this RCM over the whole Europe in the EURO-CORDEX framework (Vautard et al, 2020). They downscale four different GCMs and four different scenarios (cf Table 5.1). The CNRM-CM5 global climate model is developed in the same institute as ALADIN63, so they belong to the same family of models. CNRM-CM5 drove 4 ALADIN63 simulations, the historical (1951-2005) and three RCP scenarios (2.6, 4.5 and 8.5, on the period 2006-2100). MPI-ESM-LR, NorESM1-M and HadGEM2-ES are the three other GCMs used to drive ALADIN63 following the historical and RCP8.5 scenarios of greenhouses gases emissions. From now CNRM-CM5 will be referred to as CNRM, MPI-ESM-LR as MPI, NorESM1-M as NCC and HadGEM2-ES as HGM.

5.2.2 RCM-Emulator conception : Predictors, architecture and target.

In this chapter, I focus on the challenging emulation of daily precipitation from ALADIN63 at 11° horizontal resolution (about 12km). In order to also test the emulator's transferability to another domain, I chose a more extensive domain centred over the Alps. The target domain is visible on the left side of Figure 5.1. It is four times bigger than the previous domain as it counts 128×128 grid points. It includes the entire Alps and goes from Sardinia until the north of France and from the Pyrenees until Croatia. This domain is interesting because it has different areas with different precipitation regimes. For example, the Cevennes (South-East of France) region is known for its very extreme events in autumn, similarly to other coastal areas of the Mediterranean region. The reliefs receive more precipitation than plane regions, and they are regions where the higher resolution of RCMs brings a lot of added values, especially for the extremes. The flat regions of the north of the domain receive a lot of precipitation throughout the year but have less strong daily extremes than the southern regions. The emulator is trained to

Table 5.2: List of predictors

| 2D Variables | | | | | |
|--|-----------------------------------|--------------------------------|-------|----------------------|-----------|
| Field | Altitude Levels | Variables Notation | Units | Temporal Aggregation | Dimension |
| 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] |

reproduce land and overseas precipitations, even if I will sometimes evaluate it only on land. Finally, the bigger domain will allow characterising the computational time needed for the emulator’s calibration.

The emulator I use in this chapter for the downscaling of precipitation is built exactly like the one for the temperature in the two last chapters. It can be seen as a classical machine learning problem

$$Y_t = F(\tilde{X}_t, \tilde{Z}_t)$$

where $(\tilde{X}_t, \tilde{Z}_t)$ are the low resolution predictors, Y_t the high resolution target variable (here daily amount of precipitation) at day t and F the downscaling function that we aim to estimate with the neural network. The list of predictors is the same as in Chapter 3, with both sets of 1D and 2D inputs recalled in Table 5.2. The standardisation procedure is also the same. For each day, I normalise each 2D input spatially. The daily spatial mean and standard deviation are then given to the emulator through the set of 1D inputs, which also contains the external forcings (yearly greenhouse gases concentration, solar and ozone constraints) and the seasonal indicator (sinus, cosinus vector). More details can be found in Chapter 3. The input domain is adapted to match the new target domain. It is a 22×16 grid point on the CNRM-CM5 grid (1.4°) centred over the target domain, (the whole map on Figure 5.1).

The neural network architecture is also the same as in the temperature emulator. It relies on the well-known UNet architecture (Ronneberger et al, 2015). The only differences with Chapter 3 are due to the size of the input and target domains. As shown on Figure 5.2, the first layer of the network reshapes the 2D inputs from $[16, 22, 32]$ to $[16, 16, 64]$ in order to obtain squared images before the encoding path. On the other side, the expanding path is extended to reach the target domain size.

5.2.3 Loss function for the neural network training

In this chapter, I propose a deeper look on the impact of the loss function on the emulator’s performance. The loss function is an essential part of the neural network training. In the training phase, the network sees examples of inputs and target variable pairs. For each day of the training set, it makes a prediction and compares it with the truth. The loss function evaluates the network prediction against the expected outcome. The network parameters are then updated according to the loss function results. This operation is repeated until the cost (i.e. the average loss results) stabilises. The best combination of parameters has the lowest cost over a validation set, different from the training set. This is then a minimisation problem to find the best estimate \hat{F} such that :

$$\hat{F} = \arg \min_{\theta \in \Theta} L(\mathcal{D}, \theta) \quad (5.1)$$

Where Θ is the ensemble of possible parameters, \mathcal{D} the validation set and L the loss function.

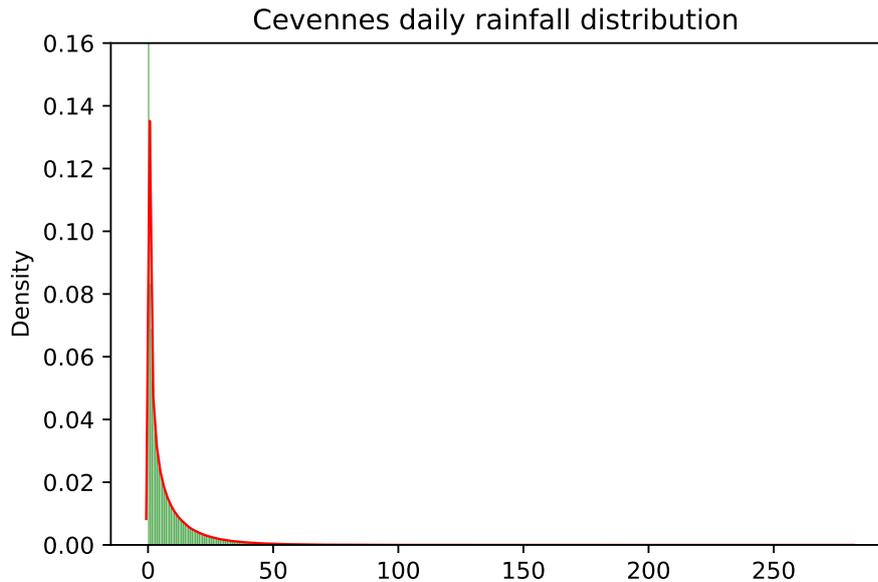


Figure 5.3: Illustration of daily precipitation distribution, in the Cevennes box (cf Fig 5.1) all points and days are pooled.

which penalizes stronger the emulator when it underestimates the true value while it was a raining day.

$$L(y, \hat{y}) = \frac{1}{N \times T} \sum_{t=0}^T \sum_{i \in \mathcal{D}} |y_{i,t} - \hat{y}_{i,t}| + \gamma_{i,t}^2 \times \max(0, y_{i,t} - \hat{y}_{i,t}) \quad (5.4)$$

With $\gamma_{i,t} = F_i(y_{i,t})$ and F_i the cumulative distribution function of a random variable Y_i following a gamma distribution

$$Y_i \sim \Gamma(\alpha_i, \beta_i) \sim \Gamma_i$$

where α_i and β_i are fitted on the training set.

The MAE and MSE losses are the most commonly used loss functions for regression problems. The MAE loss sums the absolute distance between an observation and its prediction. It gives the same weight to each observation. Knowing that daily rainfalls are strongly left-skewed, with a vast number of observations with a small amount of precipitation, the EMUL-MAE should be able to suit these days well. However, the rare cases with large precipitations should be less well reproduced.

The MSE loss function gives more weight to the significant errors than the small ones. The MSE generally shows the best results in regression problems. It is equivalent to the maximum likelihood estimation in a Gaussian setting. It leads theoretically to the best estimate for normally distributed data knowing the inputs. In the case of precipitations, it is not likely to be the case because of their highly intermittent nature. So the MSE loss function might not be well adapted. Emul-MSE is the same emulator as the one of Chapter 3.

The choice of the asymmetric loss function is more widely discussed at the end of this chapter. The intuition comes from the results of both EMUL-MAE and

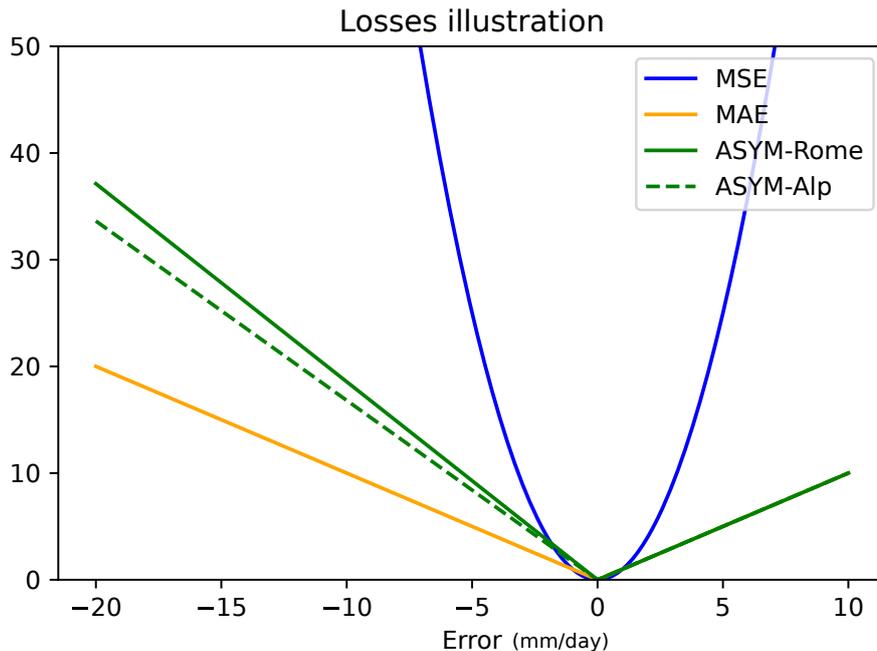


Figure 5.4: Illustration of the three loss functions according to the error ($y - \hat{y}$). For the ASYM loss, as it depends on the true prediction and the location, I illustrate it with $y = 20$ and 2 locations: Roma and the Alps point already mentioned (Fig 5.1).

EMUL-MSE presented in section 5.3. The idea is to add a penalty when the emulator underestimates consequent precipitations. The more extreme this observation, the higher the penalty should be. The $\gamma_{i,t}$ parameter determines how extreme is a given observation and defines the weight accordingly. At each grid point, I estimate the parameters of a gamma distribution, which I fit on the rainy days (over 1mm) of the training set (using the scipy python package, Virtanen et al (2020)). The Gamma distribution has been widely used to described precipitation data (Katz, 1977; Vrac and Naveau, 2007) but other distribution could be considered. In order to make this parameter estimation more robust, I fit them yearly and then average these parameters over the years. It gives a map of the shape and scale parameters. The $\gamma_{i,t}$ parameter is then the evaluation of $y_{i,t}$ by the Cumulative Distribution Function (CDF) associated to the gamma distribution Γ_i fitted at the point i .

5.2.4 Evaluation Metrics

In order to evaluate and compare the performances of the emulators I will evaluate them with respect to the daily precipitation series from the corresponding RCM simulation (cf Fig.5.5). I will use various metrics to compare the targeted (Y) and the predicted (\hat{Y}) series to have the most complete evaluation possible and understand the strengths and weaknesses of each emulators. The different metrics are detailed in the sections below.

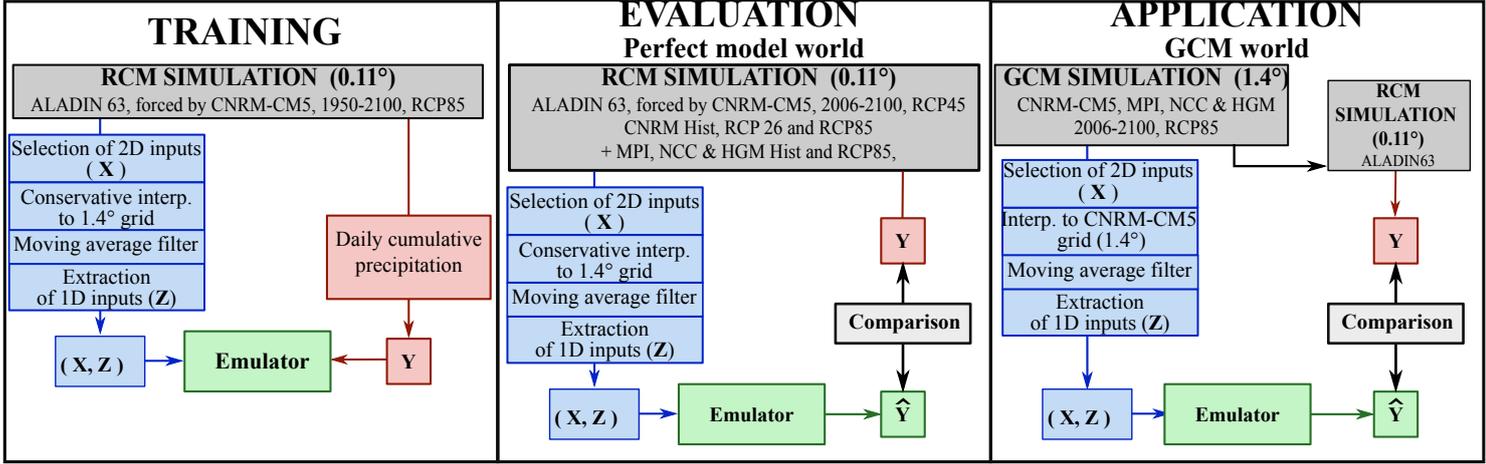


Figure 5.5: Scheme of the training (left), perfect model evaluation (middle) and GCM world application (right) protocols.

Time series comparison

First of all I will evaluate in each grid point if the emulated time series matches the original RCM series through two metrics already used in this manuscript:

- **Temporal Anomalies Correlation.** This is the Pearson correlation coefficient after removing the seasonal cycle:

$$ACC(Y, \hat{Y}) = \rho(Y_a, \hat{Y}_a), \quad (5.5)$$

with ρ the Pearson correlation coefficient and Y_a and \hat{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:

$$RoV(Y, \hat{Y}) = \frac{Var(\hat{Y})}{Var(Y)} * 100 \quad (5.6)$$

Both metrics are computed at each grid point. According to what was done in the previous chapters of this manuscript, each map is summarised with its spatial mean and 5th and 95th super-quantiles. 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.

Climatological scale metrics

It is necessary to evaluate the emulators at the climatological scale. I will look at three statistics over at least 20 years: the daily precipitation mean, the 99th quantile and the percentage of dry days. These three metrics are prints of the variable distribution from the mean and extreme sides. I will compare the emulators' maps (\hat{Y}) with the RCM truth maps (Y) and compute the relative biases (eq. 5.7) from the emulated series maps against the RCM truth. The biases maps are then in percentage. When the biases are too strong, notably because of comparing very

small values, I use the simple bias $(\hat{Y} - Y)$, expressed in mm/days. Again, the statistics are computed point-wise, and I summarise each map with the spatial mean and super-quantiles.

$$\text{Relative bias} = \frac{(\hat{Y} - Y)}{Y} \times 100 \quad (5.7)$$

These three statistics will be looked at in present climate but also in climate change context. Each statistic will be computed in a future period and the climate change statistic is the relative difference with respect to the past period. Then the simple bias is computed between RCM and emulator climate change statistics.

PDF normalisation

Since the pdf for the rainfall are very heavy-tailed, it is difficult to compare them. I propose here to have a deeper look into the distributions thanks to the ASoP method introduced in Klingaman et al (2017) and used in multiple studies as Berthou et al (2020) or Vergara-Temprado et al (2020). It consists in computing the precipitation frequency following some well-chosen bins b_n defined in Eq 5.8. The bins are such that they contain a similar number of events for bins over 1mm and as long as the number of events is sufficient.

$$b_n = e \left(\ln(0.005) + \left[n \frac{(\ln(120) - \ln(0.005))^2}{59} \right]^{\frac{1}{2}} \right) \text{ with } n \in \llbracket 0, 100 \rrbracket \quad (5.8)$$

Then we can look at each bin's contribution C_n to the mean by multiplying each frequency by the corresponding bin's mean as described in Eq. 5.9. Both frequency and contribution are interesting in comparing the emulated series with the true RCM.

$$C_n = f_n m_n \quad (5.9)$$

where f_n and m_n are the frequency and the mean of bin b_n

I use the skill score proposed in Berthou et al (2020) to evaluate the difference between the emulators and the RCM truth contributions curves. The fractional contributions are the actual contributions divided by the total mean precipitation of the series. They give information on the shape of the distribution independently from the mean. The Fractional Contribution Skill Score (FCSS) sums the absolute difference in each bin between the fractional contributions of an emulator and the targeted true series. The area under the FC curve is equal to 1, so the FCSS is equal to 0 when the two distributions are identical and to 2 when there is no overlap between them. It measures the differences between the two distribution shapes independently from the series mean. This score is illustrated on Figure 5.12 and further commented in the results section 5.3.1.

$$FCSS(Emul, RCM) = \sum_{n \in \llbracket 0, 100 \rrbracket} |FC_n^{Emul} - FC_n^{RCM}| \quad (5.10)$$

where $FC_n = \frac{C_n}{\sum_n C_n} = \frac{C_n}{mean}$

SAL score

In order to further evaluate the performances of the Emulator, I use an object-oriented score introduced in Wernli et al (2008). The SAL score aims to evaluate the spatial structure of precipitation objects from a predicted map versus a reference. It compares two maps of precipitation at a given time step. It accounts for the objects' structure (S-component) and location (L-component) plus the total amplitude of precipitation (A-component). In perfect model evaluation, the Emulator should be able to reproduce the precipitation events accurately. This score indicates if the Emulator recreates objects like the RCM and if they have the same characteristics.

The first step is to identify the precipitation objects. To do so, I used the *pysteps* (Pulkkinen et al, 2019) python library, which integrates a SAL implementation. Each day, the objects are all the groups of 5 or more consecutive points with more precipitation than a minimum threshold equal to $R^* = \frac{1}{15}R^{95}$, R^{95} being the 95th quantile on the map. Multiple objects can be detected every day. Then, the three components are computed aiming to differentiate objectively different precipitation objects. Figure 5.6, from Wernli et al (2008), illustrates the behavior of the SAL score between observed (O) and forecasted (F) objects on some idealized and basic cases.

The three SAL components are defined as follow:

- **Amplitude component** : It compares the normalized total amount of precipitation over the domain between the Emulator and the RCM truth.

$$A = 2 \frac{D(R_{Emul}) - D(R_{RCM})}{D(R_{Emul}) + D(R_{RCM})} \quad (5.11)$$

$$\text{Where } D(R) = \frac{1}{N} \sum_{i \in \mathcal{D}} R_i$$

With \mathcal{D} the ensemble of all grid points (even with no rain), N the number of grid points, and R_i is the precipitation amount in the i th grid point. Then $D(R)$ is the average precipitation per grid point over the whole map. A takes values between -2 and 2, and 0 is the perfect reproduction.

- **Location component** : It is composed of two parts $L = L_1 + L_2$. L_1 measures the distance between the two maps' centre-of-mass of the total precipitation fields.

$$L_1 = \frac{|x(R_{Emul}) - x(R_{RCM})|}{d} \quad (5.12)$$

$x(R)$ is the centre-of-mass of the precipitation field, considering all precipitation points and not only the identified objects. d is the maximum distance on

the map. The maximum value for L_1 is 1 since it is normalized by d . The second component aims to distinguish the individual objects. L_2 computes the normalized difference between the map average of the distances of the objects to the global centre-of-mass.

$$L_2 = 2 \frac{|r(R_{Emul}) - r(R_{RCM})|}{d} \quad (5.13)$$

Where $r(R) = \frac{\sum_{k \in \mathcal{O}} R_k |x - x_k|}{\sum_{k \in \mathcal{O}} R_k}$

With \mathcal{O} the ensemble of identified objects, R_k the total amount of rain in object k , x_k the centre-of-mass of the object k and x the total centre-of-mass of the map. The maximum value for r is $d/2$, so L_2 ranges between 0 and 2. The case where L_2 is equal to 0 is if both maps contain at most a single object or if, on both maps, all objects have the same centre of mass. The L-component is then between 0 and 2.

- **Structure component :** On each object, we can compute a scaled volume :

$$V_k = \frac{R_k}{R_k^{max}} \forall k \in \mathcal{O} \quad (5.14)$$

which is the sum of all object's points divided by the object's maximum point. Then we compute V the weighted mean of all scaled volume over the maps:

$$V(R) = \frac{\sum_{k \in \mathcal{O}} R_k V_k}{\sum_{k \in \mathcal{O}} R_k}. \quad (5.15)$$

And finally the S-component is the normalized difference in V :

$$S = 2 \frac{V(R_{Emul}) - V(R_{RCM})}{V(R_{Emul}) + V(R_{RCM})}. \quad (5.16)$$

Wernli et al (2008) illustrates with several examples that this part of the score captures information on both size and shape of each object. It notably differentiates peaked objects, with intense precipitation but affecting few grid points, than an other one, larger but with comparable amplitude (i.e low intensity but affecting more points). It corresponds to examples (c) and (e) on Figure 5.6 where object F would have smaller maximum. The structure component scales between -2 and +2.

The results are then represented through a diagram where each day is represented by a point with the S and A components on the x and y axis respectively, and the L component given by the color of the point. SAL diagram are visible in Figure 5.14 and commented in Section 5.3.2. Following the recommendation of Wernli et al (2009), I will apply this score on sub-domains of a maximum of 500km by side represented with blue squares on Figure 5.1.

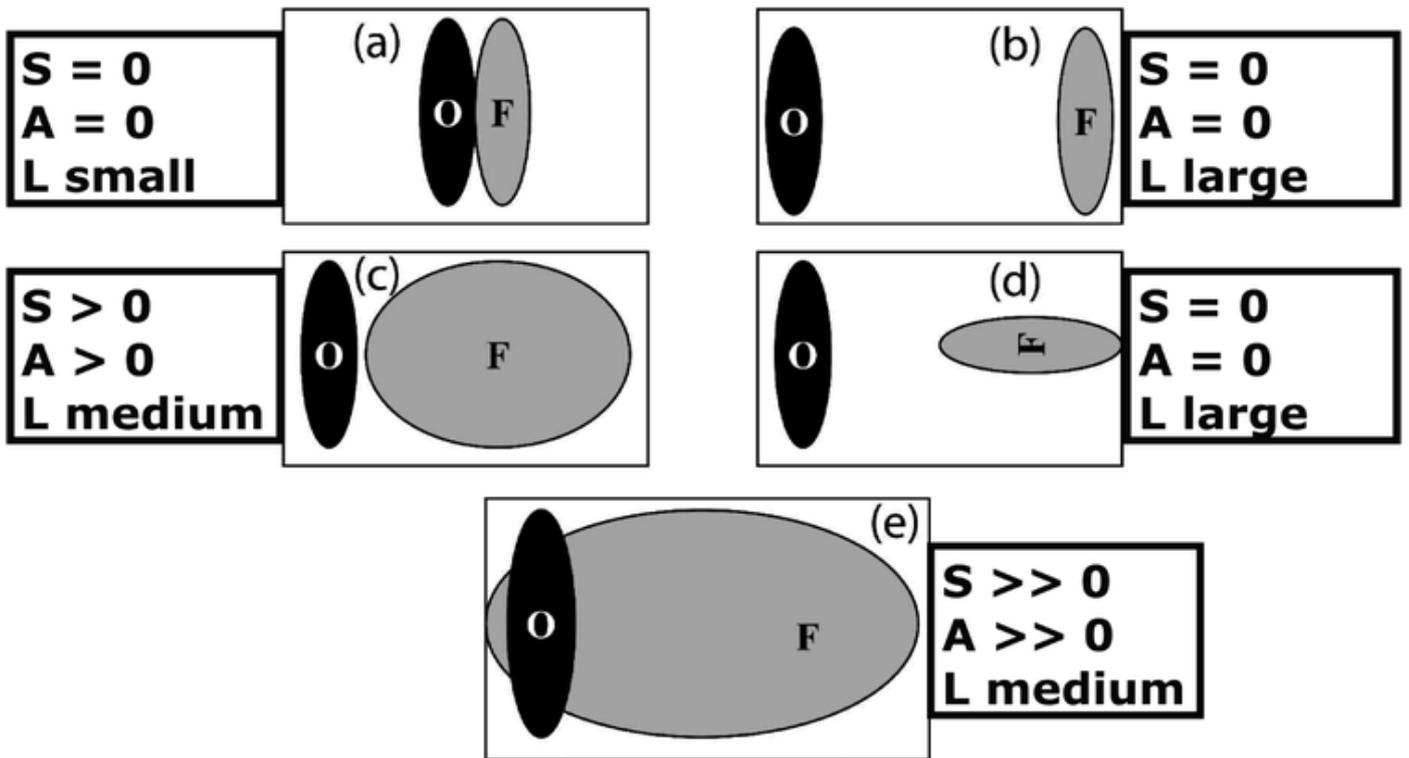


Figure 5.6: Illustration of SAL scores with idealized examples, extracted from Wernli et al (2008).

5.3 Perfect model Evaluation

This section is divided in two part. In a first evaluation step I evaluate and compare the three emulators in perfect model framework using the CNRM-ALADIN RCP45 simulation, from 2006 to 2100, as hidden truth. After a first impression on the emulators' abilities through some examples, I will extend the analysis with climatological and daily scores. This section also helps to understand the impact of the loss function on the trained emulator. In a second step, I limit the evaluation to the Emul-ASYM and comment the SAL results which will help us to objectively determine if the emulator is able to create precipitation objects. In a second part I extend the analysis to all available ALADIN63 simulations (cf. Table 5.1) and study the emulator ability to reproduce their climate change projections.

5.3.1 Comparison of the three emulators

First look into the emulators' prediction

Before evaluating the emulators' performances with metrics, it seemed worthwhile to look into the raw series they produce. Figure 5.7 shows the times series at four grid points for the year 2022 in the evaluation simulation for the RCM truth and the three emulators. The four grid points show very different series. The Alps point series shows the strongest variability and intensities, with many days over 50 mm and almost no dry spell. The Paris series has minimal variability with numerous small precipitation days and low extremes compared with the other points. The

Toulouse and Roma series are the two most similar, with some dry periods and heavy precipitation days, especially in spring and autumn in Toulouse. Roma is pretty dry during spring and summer 2022 in this simulation and has a very strong rainfall event in fall.

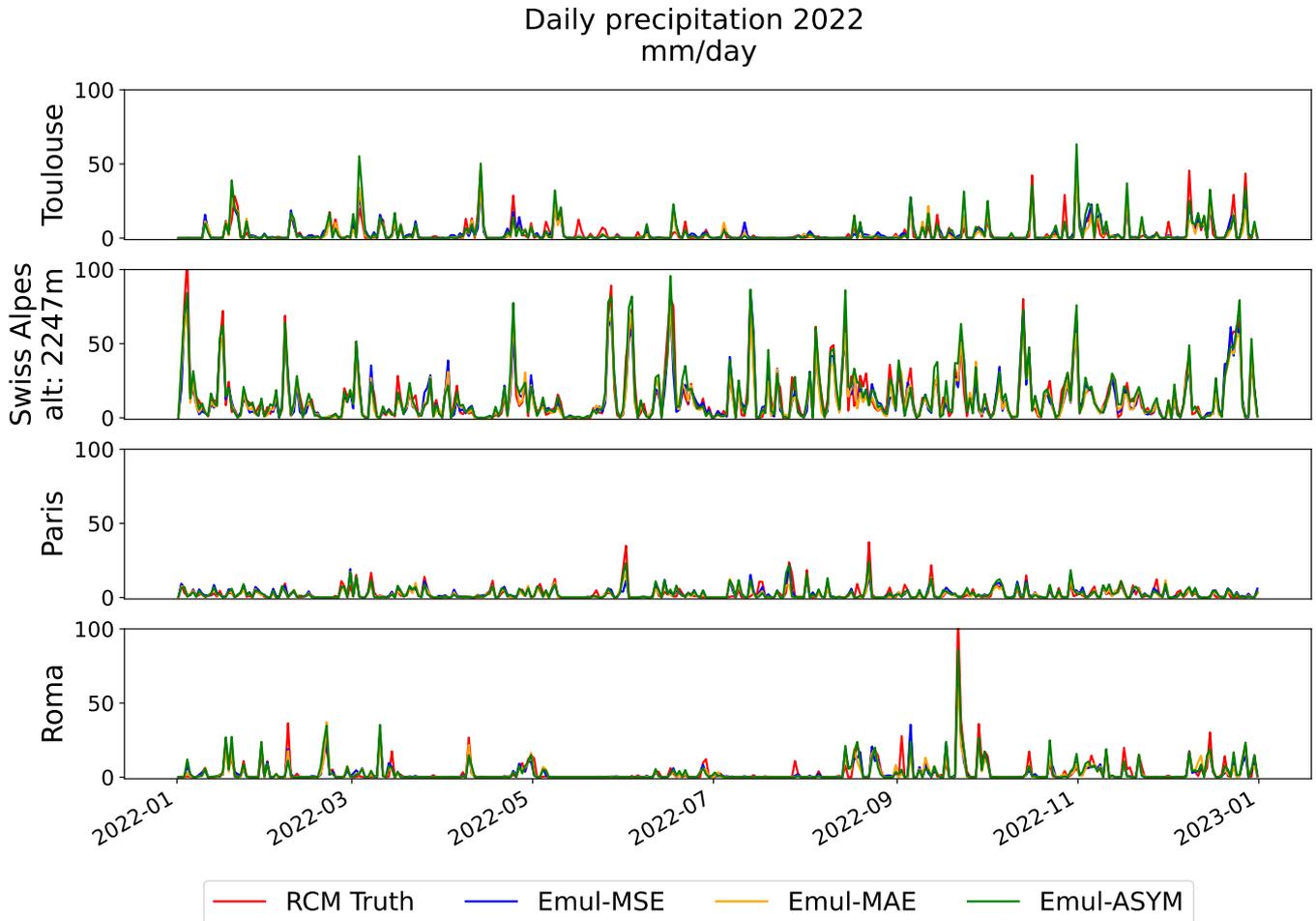


Figure 5.7: Daily precipitation time series for four grid points. The RCM truth (in red) and the three emulators are plotted on each panel.

The emulators series are very encouraging. They reproduce the four original series accurately, respecting each point's characteristics. They look like precipitation series as they appear to be able to produce days with no precipitation and days with heavy rainfall. All emulators capture the extreme autumn rainfall in Roma and the dry spell between May and June. The very high variability over the Alps point also appears well reproduced by the three emulators. On all four points, the three emulators seem to miss some extremes simulated by the RCM, as it occurs several times that the red line comes higher than the others. It also seems to be the case for the green curve (Emul-ASYM), but it is more difficult to see as it covers the other lines. However, it does not seem that Emul-MSE or Emul-MAE ever make stronger extremes than the RCM. At this point, it is impossible to decide if an emulator performs better than the others.

Figure 5.8 shows the precipitation field over the target for three days randomly

picked along the simulation. It shows the RCM truth, the three emulators and the UPscaled precipitation field (UPRCM). The UPRCM helps to have an insight into the input resolution and shows how the RCM and the emulators refine it, even if precipitation is not part of the predictors. Several exciting points appear in this figure. First of all, the emulators' prediction on each panel is very coherent with the RCM. The precipitations are always well located with coherent intensity. It seems, however, that the emulators are producing too smooth objects. On the RCM maps, there are some very sharp and precise structures that the emulators fail to reproduce with the same precision. For example, on the lower panel, there is a hole with no rain over the southwest of France, which is missed by all emulators, even if Emul-MAE and Emul-ASYM make less intense precipitation over this area. The middle panel RCM map also shows very sharp structures that appear smoother in the emulators' maps. Nevertheless, the extreme points are well located for the three days.

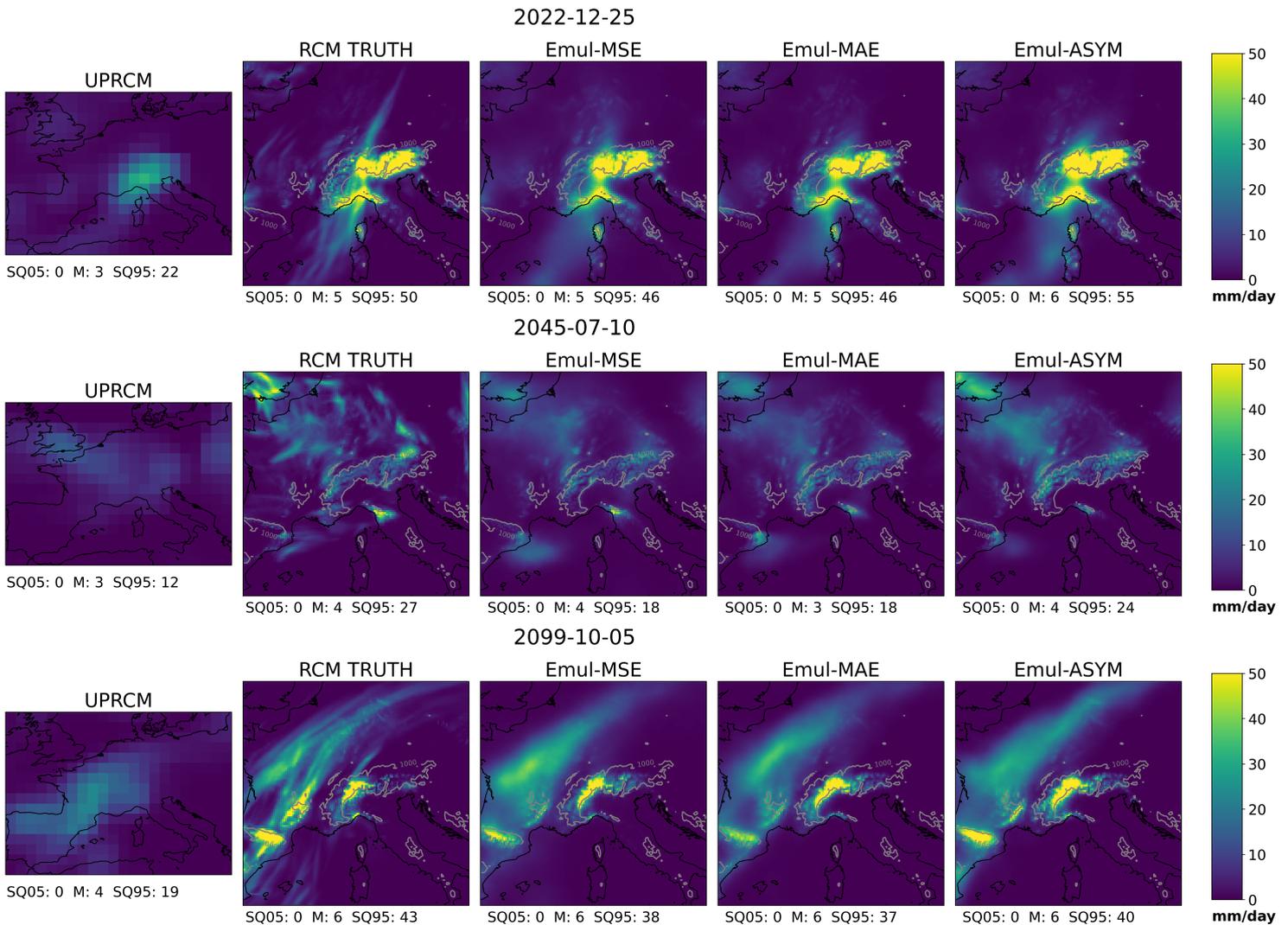


Figure 5.8: 3 randomly chosen days illustrating the precipitation field of ALADIN63 at the Upscaled resolution (UPRCM), its native resolution (RCM truth). The three right-most plots show the precipitation field for each of the three emulators.

In terms of intensities, the three emulators have mostly the correct spatial mean. Emul-ASYM reproduces better the spatial extremes as it has closer 95th superquantiles than Emul-MSE and Emul-MAE, which are both under-estimating the spatial extremes on these three days. Emul-ASYM is overestimating the spatial SQ95 on the first panel, as it creates a more significant local extreme over the Alps than in the RCM map. It is, however, remarkable that this extreme is not inconsistent with the UPRCM map. Indeed it is interesting to notice the differences between the RCM and the UPRCM maps, which attest to the resolution's impact. The RCM is able to create sharp and well defined objects, with locally strong intensities. Regarding this aspect, the emulators seem to have an adequate capacity to refine the low-resolution maps and always recreate consistent high-resolution maps. Nevertheless, it seems that the objects created by the emulator are less *precise* than on the original RCM maps.

Daily scale analysis

In a second step, and to extend the first observations from the previous section, we can look at some scores over the time series. Firstly, the upper panel on figure 5.9 shows the Pearson correlation coefficients calculated between the RCM and the emulators' series in each grid point. The three emulators appear to have similar performances regarding this aspect, with a reasonable deseasonalised correlation with the true series over the whole domain. The best correlations are over the reliefs with Pearson coefficients larger 0.9. The lowest correlation appears over the driest area (cf Fig. 5.11), like the south of the Pyrenees or the North-East corner of the domain, but the correlations are still around 0.75.

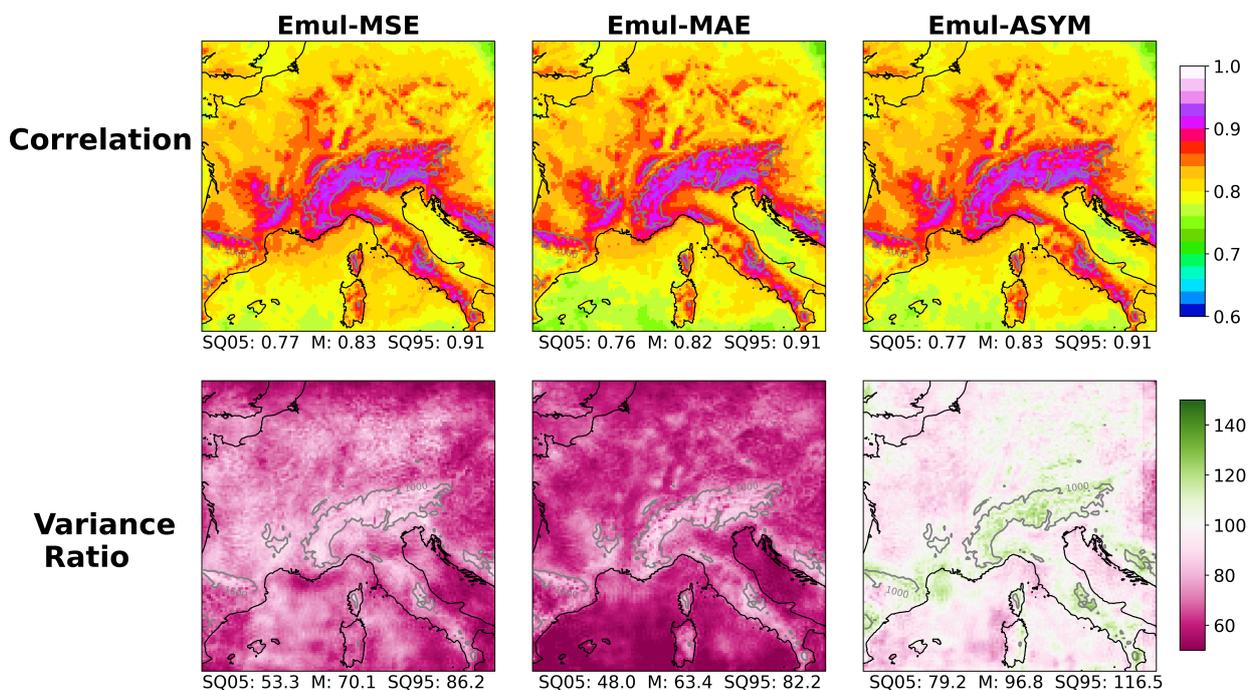


Figure 5.9: Temporal Anomalies Correlation (up) and Ratio of variance (bottom) computed on the entire evaluation simulation (2006-2100) for the three emulators.

The lower panel on Figure 5.9 shows the variance ratio for the three emulators against the RCM truth. Emul-ASYM manages to reproduce in each point the RCM variance much better than the two others. Its variance ratio ranges from 80 to 120 percent, with a big part of the map being very light, showing about 100% variance reproduction. It slightly overestimates the relief's variance and slightly underestimates it over the regions with low rain average (cf Fig. 5.11). On the other hand, both Emul-MSE and Emul-MAE vastly underestimate the variance over the whole domain, even if Emul-MSE is slightly better.

It seems that the three emulators understand the large scale the same way, as they can recreate the chronology of the original RCM series very accurately. They can identify where and when the precipitations occur at the grid point scale, as shown by the good correlation maps. However, the loss choice seems to substantially impact the reproduction of the events' intensity as the emulators have different variance ratio maps. Let us see if this is confirmed when we look at aggregated statistics.

Climatological scale analysis

In this section, we look at some aggregated statistics to evaluate if the series produced by the emulator are statistically similar to the RCM one and how they differ. Figure 5.11 shows three climatological metrics over 20 years in the present period for the RCP4.5 simulation which is not in the training set. The upper panel shows the average daily precipitation over 2006-2025, the middle one is the 99th quantile, and the lower one shows the proportion of dry days. This figure illustrates the impact of each loss function on the emulator.

The Emul-MSE mean is very similar to the RCM map. The spatial mean and superquantiles are the same. The bias map shows that it slightly underestimates the RCM values, but at maximum by 15% and over regions with low precipitations. However, it presents much poorer results on the other part of the distribution: it largely underestimates the 99th quantile (-15% on average) and the number of dry days (-10% on average). It is due to the nature of the mean squared error loss, mainly concentrating around the mean. The Emul-MAE is, meanwhile, very accurate for the representation of dry days, very slightly overestimating them. However, it fails to reproduce the mean and the 99th quantile maps, broadly underestimating them. The MAE loss gives the same weight to all errors. Since the number of dry days is the most represented (between 35 and 85% of the days are between 0 and 1 mm) they weigh much more in the emulator training, so it mainly focuses on them.

The Emul-ASYM aims to correct the EMUL-MAE by giving more weight to the rainy days, proportionally to the amount of rain. It has similar performances to Emul-MAE over the dry days' map, which is expected since both emulators have the same loss function on this part of the distribution. However, the Emul-ASYM mean and 99th quantile maps are also very accurate. It shows in both cases less than 15% bias over the worst points and almost no bias on average over the maps. Regarding both climatologic maps, it seems to slightly overestimate the precipitation where it is raining the most and under-estimates at the driest points. Nevertheless, these errors are shallow, and the Emul-ASYM is clearly the best option if we aggregate the performances for the three metrics.

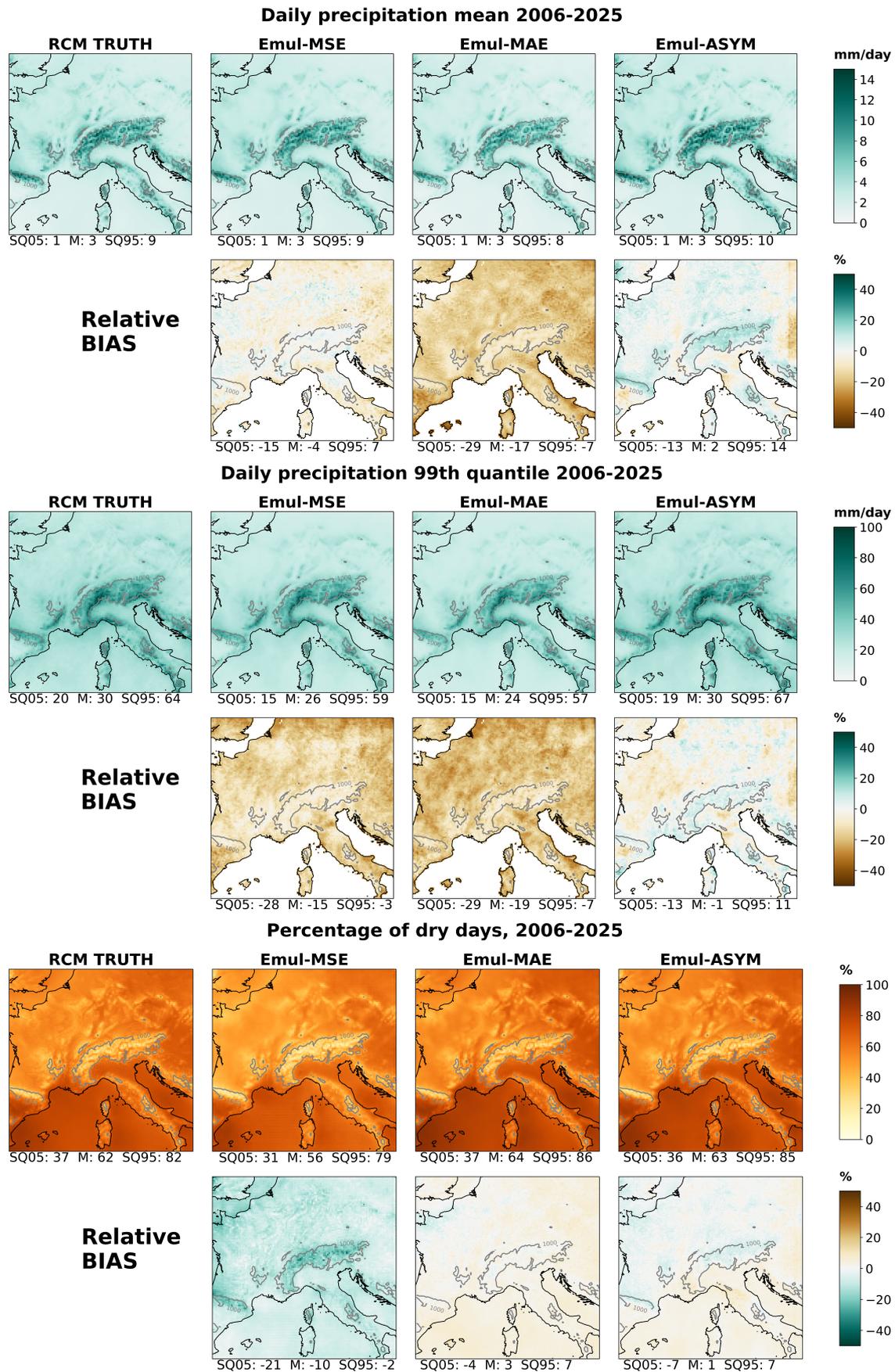


Figure 5.11: (Upper) the mean map of daily precipitation accumulations over the 2006-2025 period, (middle) the 99th quantile map over the same period and (lower) the percentage of dry days. These three statistics are shown for the RCM and the three emulators. For each emulator and each metric, the relative bias maps are shown. The spatial mean and 95th and 5th superquantiles are given for each map.

On all maps in Figure 5.11, it is striking to see how well the emulators reproduce the complex spatial structures. Emul-MAE and Emul-MSE have strong biases that are uniform over the domain. All three statistics present locally different patterns, and the emulators reproduce that. For instance, on the 99th quantile maps, there is a strong pattern in the Cevennes, just south of the Massif Central (France), which is much less intense in the daily mean map. It is the same for the emulators' maps. The spatial structure over Italy is also very complex; there is a thin line over the reliefs with more rainy days and higher extremes, which is also perfectly reproduced by the emulators. Similar examples exist for the entire domain.

In order to extend this result, we can look at the entire distribution using the ASoP method described in section 5.2.4. In Figure 5.12, the pdf analysis is detailed for the four grid points previously used: Toulouse, a high point in the Swiss Alps, Paris and Roma. The first column shows the events frequencies for each bin defined in section 5.2.4. Most days fall in bins under 0.1mm as the red curve comes from high on the left part of the plots. The Emul-ASYM and the Emul-MAE reproduce this part well, while the Emul-MSE underestimates the very low precipitations ($\leq 0.1mm/day$) and overestimates the ones between 0.01 and 10mm. It is less pronounced for the Alps point, where the event distribution is more uniform across the bins with more intensities 10mm and 100mm than the other three points. Emul-ASYM reproduces the frequency of these stronger events better than the two other emulators.

The second column shows the actual contributions to the mean, which are the frequencies multiplied by the bins' mean. The first remark is that Emul-ASYM slightly overestimates the contribution of the precipitations around 10mm, which probably led to the wet bias on the mean map of figure 5.11. Emul-MAE produces insufficient rainfall over $\sim 8mm$ as the right part of the distribution is shifted to the left. The same remark applies to the Emul-MSE to a minor extent, which has a better reproduction of the mean, confirming what we saw in figure 5.11. Nevertheless, the Emul-ASYM matches better the right tail of the curve.

The last column illustrates the fractional contributions skill score by plotting the difference between the emulators and the RCM distributions. The fractional contributions are the actual contributions normalized by the mean of the series, allowing us to compare only the shape of the distribution across the bins. It helps to see that the Emul-MSE and Emul-MAE distribution are generally left-shifted, with too many small precipitations and not enough big ones. The Emul-ASYM curve generally looks better even if it tends to be right-shifted as it produces too many heavy precipitations. The regularization term in the Emul-ASYM loss appears to play its role pretty well as the distribution of precipitation is closer to the real one but might sometimes be too strong.

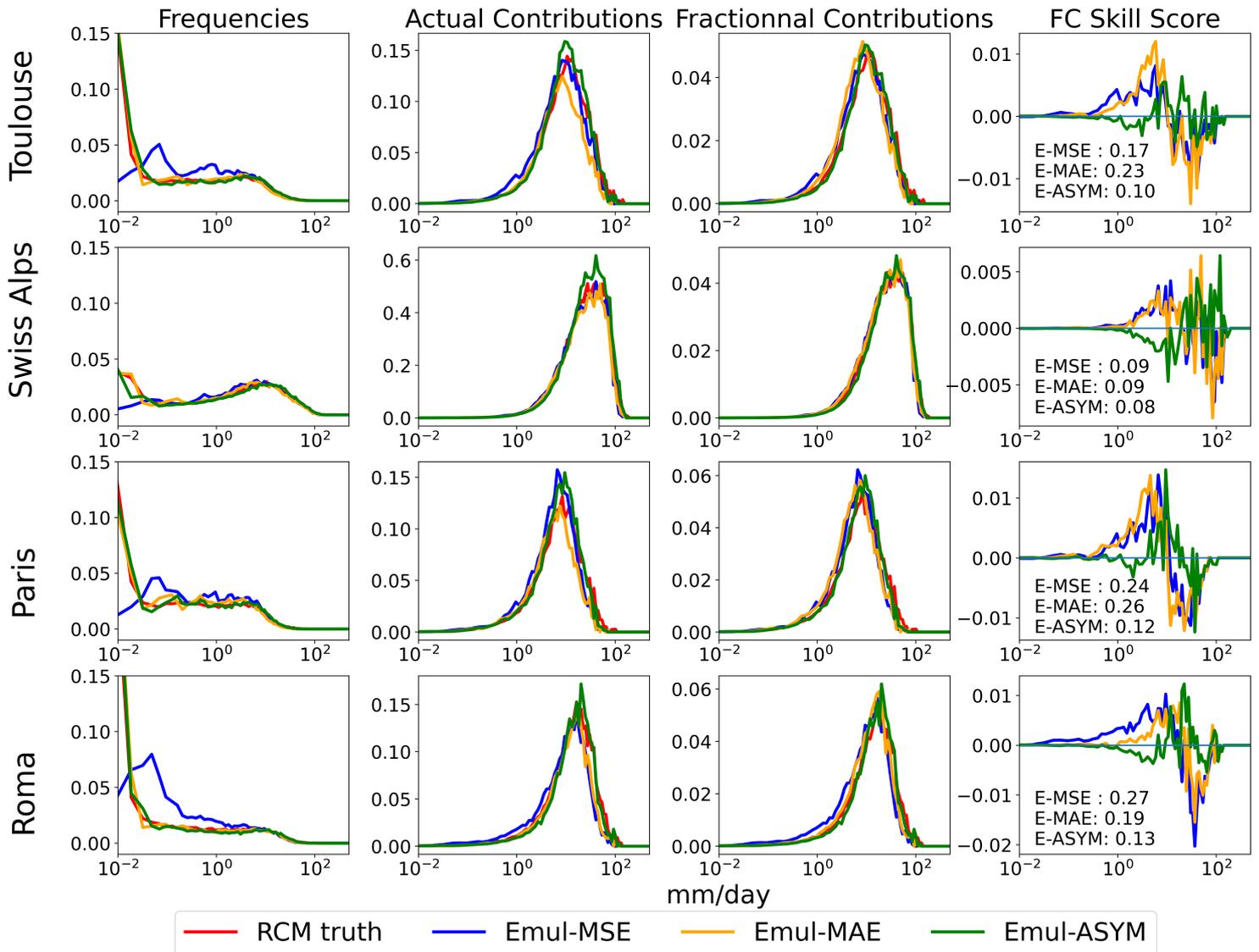


Figure 5.12: Illustration of the probability density function analysis following the ASoP method (Klingaman et al, 2017) on four example grid points. Each line is a point and each column is a different step of the method. The first column shows the frequency of events in each bins, the second and the third the actual and the fractional contribution and the last column illustrates the skill score. The number in the last column plots are the scores for each emulator at the corresponding point.

The skill score measures the area between the emulators' fractional contribution and the RCM one, and we can see that the Emul-ASYM outperforms the others over these 4 points. It is interesting to notice that Emul-MSE and Emul-MAE perform better over the Alps point, where the precipitations are more uniformly distributed across the bins. Finally, Figure 5.13 shows that the Emul-ASYM skill score is better over the whole domain. It generalizes the distribution analysis and confirms that the specifically designed loss function is more adapted than the two others to reproduce the highly skewed distribution of precipitation.

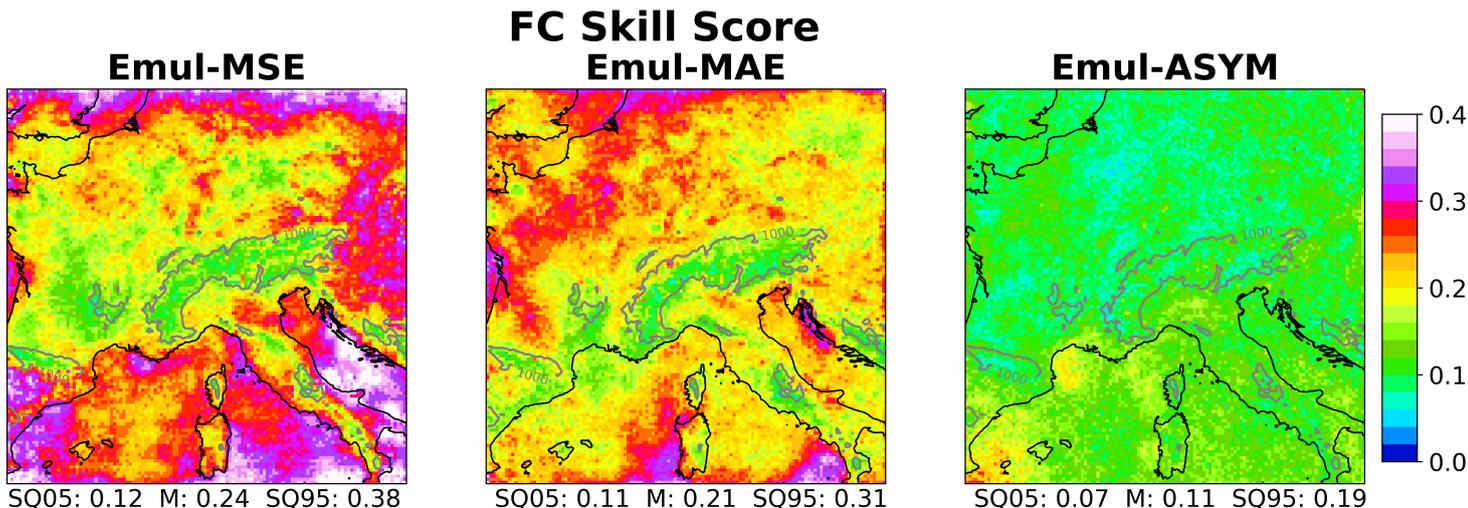


Figure 5.13: Fractional Contribution Skill Score maps for the three emulators.

Conclusion on the comparison

Until here, we have analysed the role the loss function can play in the calibration of the emulator. Table 5.3 summarise the results obtained on the three emulator. They all demonstrated an excellent capacity to reproduce the daily precipitation time series with a good temporal correlation with the original RCM series. Nevertheless the loss function impacts strongly the intensity of the events. The MSE loss function penalizes strongly the large error which centers the prediction around the mean because of the chaotic nature of precipitations. Thus, if the mean daily precipitations is well represented the extremes are underestimated. On the other hand the MAE reproduces well the low precipitations but underestimate the intensity of larger events. Finally, the Emul-ASYM, thanks to a regularization term added in the loss function, managed to reproduce better the entire rainfall distribution at each grid point of the domain with notably a better reproduction of the extremes. Therefore, the loss function plays here as a cursor to set the event intensities, while the chronology of the series is captured from the predictors. From now on, we will consider only the EMUL-ASYM.

Table 5.3: Summary of the emulators' comparison results

| Emulators | Temporal correlation | Average precipitation | Dry days | Heavy precipitation | Variance | PDF Shape |
|-----------|----------------------|-----------------------|----------|---------------------|----------|-----------|
| MSE | ++ | ++ | - | - | - | - |
| MAE | ++ | - | ++ | - | - | - |
| ASYM | ++ | ++ | ++ | ++ | + | + |

5.3.2 Deeper evaluation of Emul-ASYM

Object oriented analysis

The illustration Figure 5.8 that the precipitation objects created by the emulator are smoother than in the RCM. The SAL method presented in section 5.2.4 is an objected-oriented evaluation approach which compares on two maps the object

similarities.

Following the recommendation of Wernli et al (2009), I limited the evaluation to three subdomains of about 500km by the side. The blue boxes represent them on figure 5.1. The first subdomain focuses on the Cevennes regions. This part of South France is well known for its extreme autumn precipitation events. These events are the object of multiple studies (Ribes et al, 2019; Caillaud et al, 2021) because of their strong socio-economic impacts. It is then important to assess whether the emulator is able or not to reproduce such events. The second domain is another hotspot for Mediterranean extreme precipitation events (Ivušić et al, 2021) located in Croatia, over the Dinaric Alps and the North of the Adriatic Sea. The last subdomain is centred around Belgium, including the South-East of England, the North-East of France and West of Germany. This region presents a different climatology with extreme events of smaller intensities occurring more in winter.

Figure 5.14 presents the SAL scores' results. For each region, there are five SAL diagrams. The left most diagram represents the results for all raining days. Then going to the right we consider only days where the spatial 99th percentile of the RCM truth series is above an increasing threshold. The threshold and the number of considered days are indicated on each diagram. Thus, from left to right we consider only more and more extreme events. The first general comment is that over all these diagrams, the emulator reproduces accurately the extreme majority of the events. Indeed the red boxes regroup 90% of the days and they are always centred around 0 with most points are deep blue, showing good Location score. On the first column representing all raining days, the emulator underestimates the global amount of precipitation over the domain, with the red box being pulled down left. It can be explained by the fact that the emulator misses some small precipitation objects. The SAL metrics are one-sided: they evaluate how the predicted map matches the reference one. Knowing the chaotic nature of rainfall, it is perfectly fine that the emulator misses or add some small events. Besides, we fix the threshold according to the RCM true series. It is then logical that events, especially small ones, are missed or underestimated by the emulator. It is not shown here, but when we fix the threshold according to the emulated series, then the emulator overestimates the amplitude of some small RCM events and the red box is pushed up-right.

On the right of the figure, when we look at days with heavier precipitation, the amplitude gets centred around zero or slightly positive on the right-most column of the two Mediterranean regions. In addition, the emulator tends to produce larger objects with a positive S-component. However, the centre of the object is most of the time well located. It tends to generalize that the emulator produces smoother objects than the RCM, especially on significant intensities events.

There is a correlation between the amplitude and the structure metrics. It can attest that the emulator always creates objects consistent with the RCM. They are either smaller or bigger in terms of both shape and amplitude. On all maps, we can see some days with lousy location and structure scores but the correct amplitude. They are typical of days where the emulator produced too smooth objects and did not peak like the RCM. The emulator produces one large object with medium

intensity, while the RCM produces multiple peaked objects with high intensities. It implies bad locations and structure scores but good amplitude.

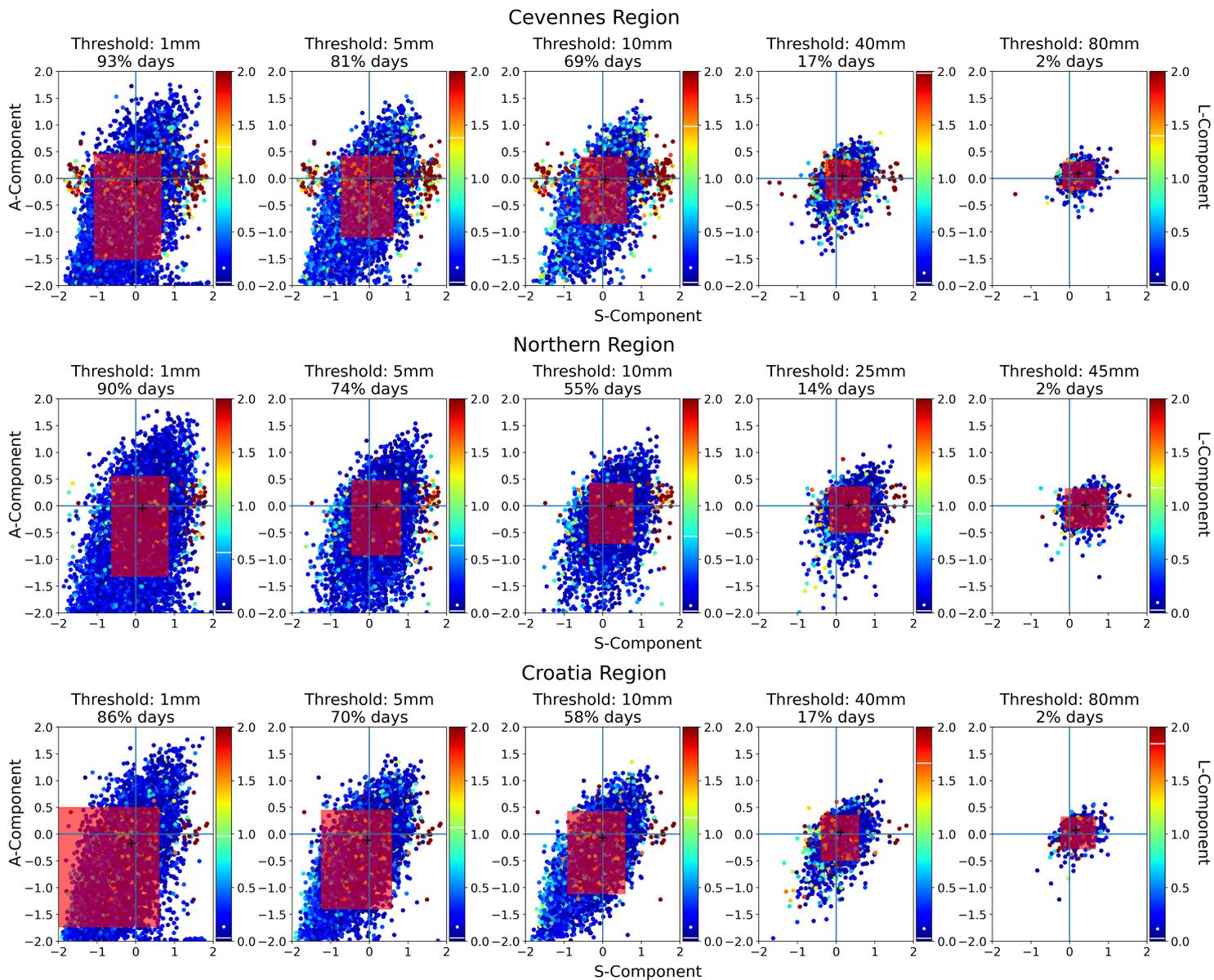


Figure 5.14: SAL diagram for the three regions: Cevennes (up), North of the domain centred around Belgium (middle), and a region over Croatia and the North of the Adriatic sea. From left to right, the panel show the SAL results for days with maximum events intensities above an increasing threshold. Each point on the diagram represents a day with the A-Component on the y-axis, the S on the x-axis and the color give the Location score. The red box includes 90% of the points, and the black cross indicates the A and S median. The 5th, 50th and 95th quantiles are given in white on the colormap for the Location component.

Generally speaking, the emulator manages to reproduce the precipitation objects simulated by the RCM, even if they do not always have the perfect characteristics. The emulator captures most of the extreme events with the most suitable characteristics. The emulator seems nevertheless to produce smoother objects. A further

analysis, with an application to a hydrological impact study, should be conducted to determine whether it is a fundamental limitation and how we could maybe adapt the emulator.

ALADIN63 matrix extension

In order to give more robustness to the good performances of the Emul-ASYM, we can extend the evaluation to all ALADIN63 simulations available for our target domain. Figure 5.15 summarizes climatological maps as the ones shown on Figure 5.11. The three panels (from left to right) correspond to the three statistics we looked at in Section 5.3.1: the mean amount of daily precipitation, the 99th quantile and the percentage of dry days over the 2006-2025 period. On each panel, the upper part shows the summary statistics for the raw maps of the RCM and the emulator, and the lower part summarises the relative bias maps of the emulator with respect to the RCM truth. On each panel, the columns correspond to a simulation. Each bar shows the spatial mean of the map, the upper bound shows the 95th super-quantile and the lower bound shows the 05th super-quantile. The first column shows the results for the CNRM RCP85 simulation, which has been used to train the emulator. The results on this simulation are given here as an indicator and cannot be taken alone to evaluate the emulator's performances.

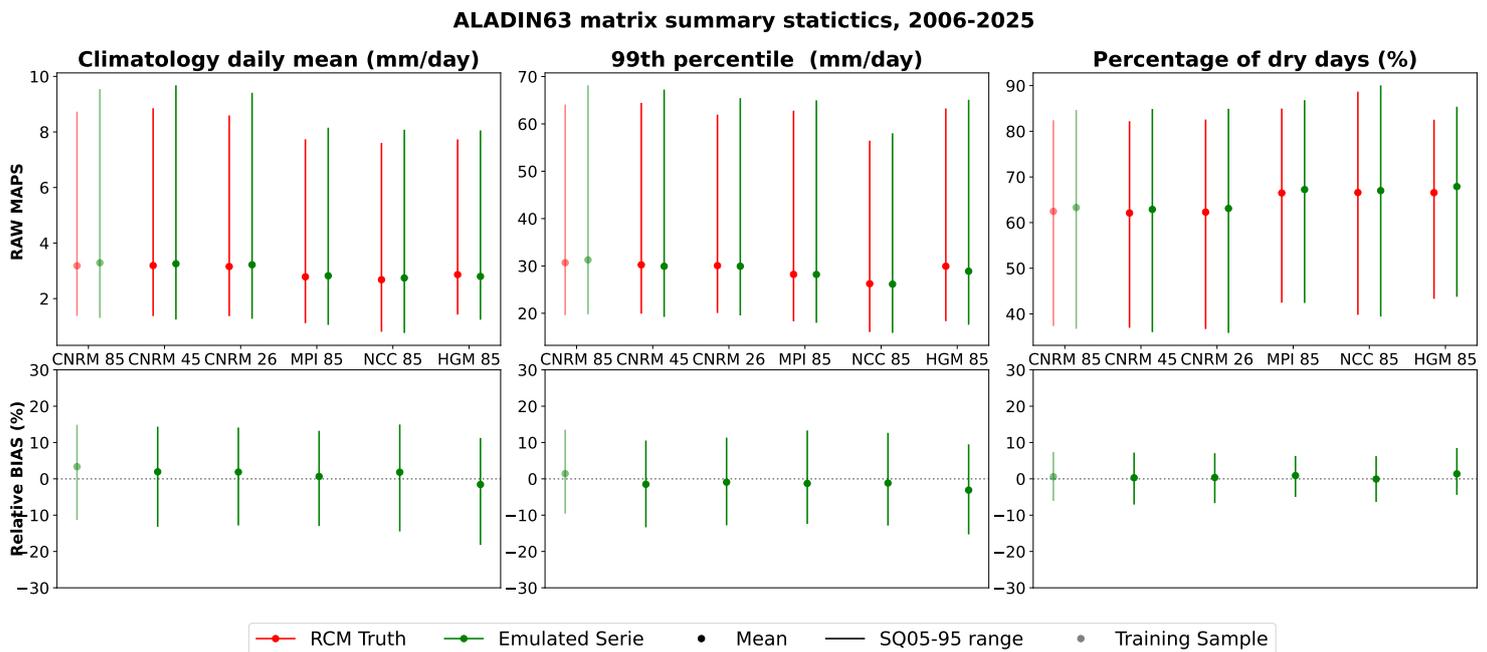


Figure 5.15: Summary plots of the three climatological statistics regrouping the results on all ALADIN63 simulations. On each error bar, the lower (resp. upper) bound is the spatial 5th (resp. 95th) superquantile and the spatial mean is represented by the dot. The upper panels show the raw maps summary statistics for the RCM (in red) and the Emul-ASYM (in green), and the lower panels show them for the relative bias maps.

The results are very encouraging as the performances of the emulator are very similar across simulations, even if those are different in several aspects. For instance, the 3 CNRM simulations have higher daily means than the three others

since the spatial mean and superquantiles are higher. It is less evident on the 99th quantile maps, where only the NCC simulation produces “eye-visible” less intense extremes. The emulator’s bars reproduce these aspects. As observed previously, the emulator overestimates the average daily precipitation on the wettest points and underestimates it over the driest points whatever the statistics, which stays valid for all simulations. The biases on land points are similar to the ones observed for the CNRM RCP45 simulation, showing that the emulator reproduces each simulation with the same accuracy. In all these simulations, the emulator reproduces the three parts of the distribution well over the whole domain.

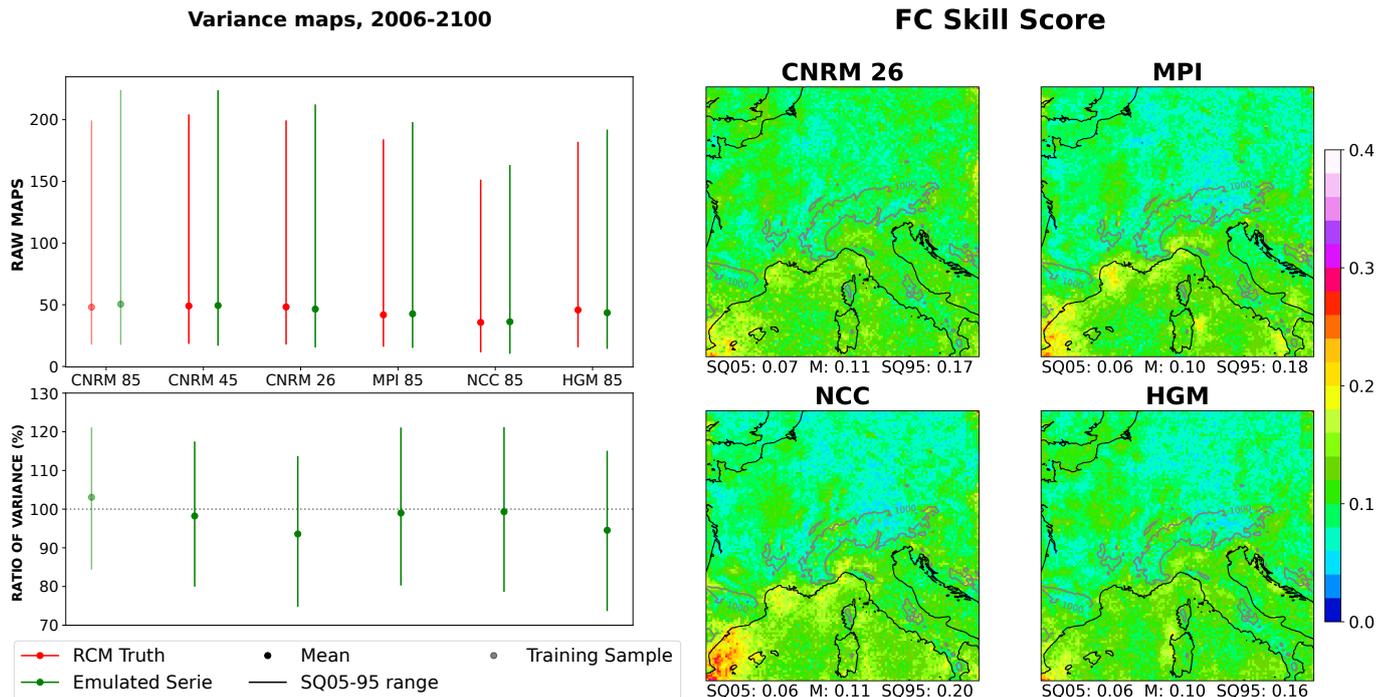


Figure 5.16: The left panel show the summary plots for the temporal variance, the upper panel summarizes the raw variance maps, and the lower one shows the ratio of variances. The right panel shows the Fractional Contribution Skill Score (FCSS) for the 2006-2100 period for the four simulations that have not been shown yet (CNRM 26, MPI, NCC, HGM).

The analysis is the same regarding the variance maps summarized in Figure 5.16. The temporal variance differs according to the simulation. For example, the RCM simulation driven by NCC has a smaller variance than the CNRM simulations or the HGM. The emulator reproduces in each case the variance maps quite accurately. However, in every simulation, it strengthens the variance where it is the strongest. The variance ratio summary plot confirms that the analysis made for the CNRM RCP45 in section 5.3.1 extends to all other simulations. The emulator can reproduce the daily time series with globally acceptable variance at every grid point. In the worse cases, it misestimates the variance by about 20%. Figure 5.16 also shows the FC skill score maps for the four missing evaluation simulations (the RCP45 simulation in in 5.13). Here again, we can observe that the emulators reproduce the shape of the precipitation distribution correctly at each grid point in all simulations. It is impressive to see how similar are these four maps. The emulator has similar

performances across all simulations at the grid point scale.

Climate change reproduction

In order to finalise the evaluation of the emulator in the perfect model framework, we can look at the climate change maps. To do so, we will look at the three statistics used in the previous sections: the mean daily precipitation, the 99th quantile and the percentage of dry days. In each simulation, I compute the relative changes in a future period (2070-2100) versus a past period (1950-1980). The changes in precipitation are likely to be different according to the seasons over western Europe so we will look at the seasonal climate change here. The different studies about changes in precipitation amount over the region project a decrease in summer precipitations, notably around the Mediterranean sea, and an increase of winter precipitation on the North. Besides, a possible increase in extreme precipitation, especially over northern Europe, is expected. The results for the four seasons and the three statistics on all simulations are summarised through summary plots in Figure 5.17 while the results for the MPI and HGM simulations are illustrated in Figure 5.18.

The first remark is that on all plots summarising the raw maps, the green bar sticks very well to the red one, implying that the emulator correctly reproduces the maps and the intensity of the local changes. It is particularly notable on the summer plot, where the differences between the projections are the strongest. The MPI and NCC simulations show a substantial decrease in the mean daily precipitation over the entire map, associated with a global increase in the percentage of dry days. On the other hand, the HGM simulation projects an increase in average daily rainfall over some regions in summer. The emulator reproduces each simulation specificity with mainly the right intensity. Figure 5.18 shows summer and winter changes for the MPI and HGM simulations. It illustrates well that the emulator correctly captures the big spatial pattern. Still, in summer, we can observe that the emulator precisely places the regions where the HGM simulation produces an increase in average rainfall. This increase matches an increase of the 99th quantile in the same regions, and the emulator produces the same relationship. Similar analysis can exist on the winter maps, concluding that the emulator reproduces the ALADIN63 simulation with excellent accuracy.

Nevertheless, the emulator's maps are more continuous than the RCM maps, especially for the 99th quantile maps, which are a lot patchy. It results in significant local biases between the emulator and the RCM maps. It partly explains the large biases on the bias maps summary plots in Figure 5.17. Generally, the emulator tends to overestimate some changes as we can see that the green bar is often longer than the red one. The number given on top of the bias maps summary plots shows the percentage of sign agreement between RCM and emulator over the grid points. It shows that the emulator identifies well the changes as these numbers are very high (always above 75%, very often above 90%). Moreover, on the bias maps of Figure 5.18, the hatching shows the points where RCM and Emulator disagree on the signs. It is visible that they mostly correspond to points with minor changes.

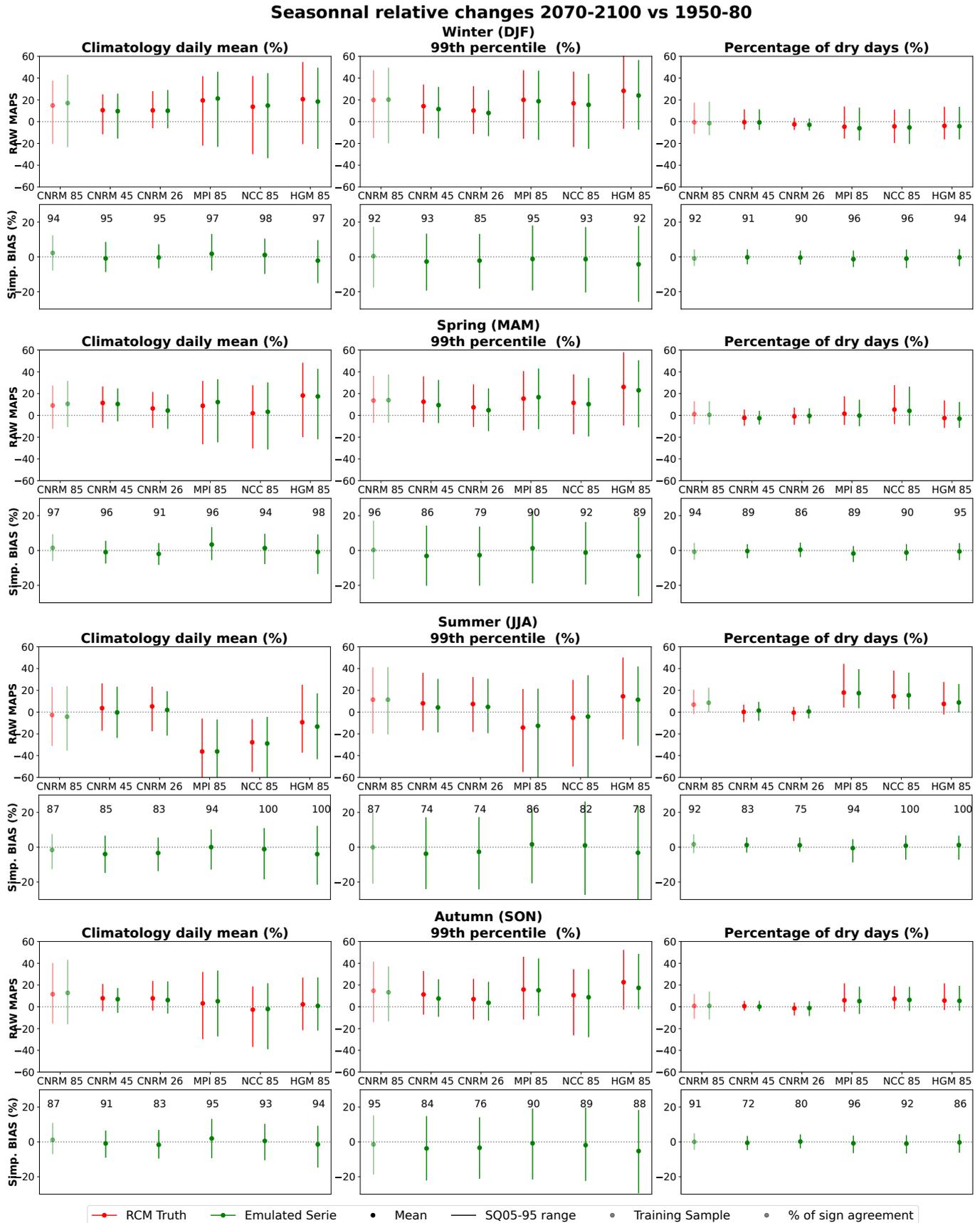


Figure 5.17: Same as Fig 5.15 for the seasonal climate change (2070-2100 vs 1950-80) summary plots for the three statistics of interest: the daily precipitation mean, the 99th quantile and the percentage of dry days. The changes are the relative difference between the future period and the past one. The biases are simple bias between the Emulator and RCM relative change maps. On each bias summary plot the number indicates the % of points where RCM and Emulator agree on the sign.

To conclude, the emulator can reproduce high-resolution climate change maps with the same strong spatial pattern and intensities. Another relevant remark, not shown here, is that Emul-MSE and Emul-MAE have the same ability as Emul-ASYM to reproduce the climate change maps. It means that each emulator keeps the same biases along the simulation, and the changes are mainly driven by the large scale, which the emulators understand well.

Conclusions on Emul-ASYM

Through sections 5.3.1 and 5.3.2 we have analysed the ability of the emulator trained with the asymmetric loss function to reproduce the precipitation field simulated by the RCM. The conclusion on the emulator performances are summarised here:

- The emulator is able to produce realistic precipitation time series well correlated to the RCM ones and with the right variability.
- The grid-point regularization term in the asymmetric loss function helps to respect and reproduce the entire complex distribution of precipitation everywhere on the target domain.
- The emulator tend to underestimate the precipitation in generally dry regions and inversely in the wettest parts of the domain.
- The emulator creates coherent objects of precipitation, with generally the right characteristics even if they tend to be too smooth (i.e. less sharp and precise than the RCM objects).
- Those conclusion on a single evaluation series stand also for the other evaluation simulations that we have at our disposal, respecting both scenario and GCM transferability defined in Chapter 4.
- Finally the climate change maps obtained from the emulated series are almost identical to the RCM ones. It gives a lot of confidence to use the emulator in climate change context.

The emulator present therefore satisfactory results in perfect model evaluation and, even is there is space for improvements, we will study its behavior in GCM application.

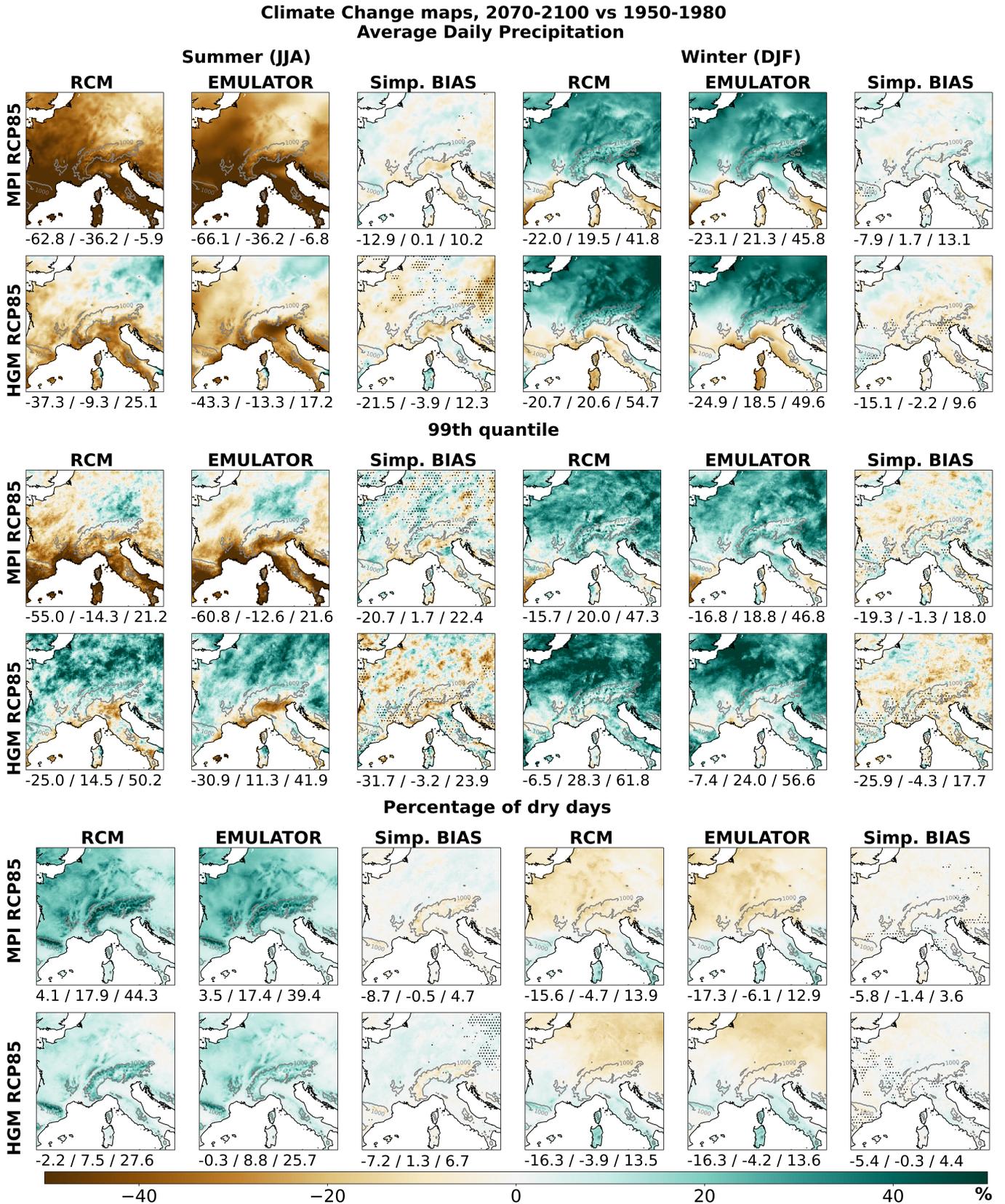


Figure 5.18: Relative changes (in %) between 2070-2100 and 1950-1980 for the MPI and HGM driven simulations regarding (up) the mean map of daily precipitation accumulations, (middle) the 99th quantile map and (low) the percentage of dry days. These three statistics are shown for the RCM and the emulator, plus the simple bias map between the two. For each map, the spatial mean and 95th and 5th superquantiles are given. The hatching indicates the point where RCM and Emulator disagree on the sign.

5.4 GCM data application

In this section, I apply the emulator directly to GCM data as in its application mode. In order to have a reference to analyse those results, I will downscale with the emulator the GCMs used to drive ALADIN63: CNRM-CM5, MPI-ESM-LR, HadGEM2-ES and NorESM1 (cf. Table 5.1). I will focus only on the RCP85 simulations from 2006 to 2100. This section aims to evaluate the emulator’s behaviour when applied to GCM inputs. However, as shown in Chapters ?? and 4, the RCM simulations cannot be used as the exact reference to evaluate the emulator. The large-scale differences between an RCM simulation and its driving GCM imply low day-to-day correlation and long-term statistical differences. The difficult task here is to assess if the emulator produces a series consistent with the GCM large scale and the high resolution added value from the RCM.

5.4.1 Illustration of the daily GCM/RCM differences

Figure 5.19 shows five consecutive autumn days of the precipitation field in the CNRM RCP85 simulation. For each day, we see the RCM truth simulation and the emulated maps in perfect model and application mode. In perfect model mode, the emulator inputs come the UPscaled RCM predictors, and in the application mode, it takes GCM predictors. I also plot the UPRCM and GCM precipitation maps of the corresponding days. It is important to remember here that the low-resolution precipitation field is not a predictor. The UPRCM precipitation is simply the RCM map interpolated on the GCM grid, and I use it to compare with the GCM precipitation map.

It is striking that over these six days, the GCM emulated maps never look the same as the RCM and the UPRCM emulated maps. As shown in the previous section, the UPRCM’s emulator reproduces the RCM map very well. It recreates most objects in the right place and with the right intensity. We have seen that the objects are more spatially continuous than in the RCM maps. Looking at the maps produced by the emulator when it downscales GCM predictors, we can see that they are very different. The rainfall occurrence, location or intensity do not match the RCM map but is perfectly coherent with the low-resolution map from the GCM. For instance, on the first day, the RCM produces heavy precipitation on both sides of the Adriatic sea, while there is nothing on the GCM emulated nor in the GCM maps. Then a strong precipitation event arrives from south and stays blocked on the Alps at day 3. The GCM seems produces the same event but one day after the RCM and it stays south of the Alps. Remarkably, the GCM emulated maps are always perfectly coherent with the precipitation field produced by the GCM, which gives some confidence that the emulator creates realistic maps. Indeed, the emulator seems robust to GCM predictors, as it “understands” and follows the large-scale atmospheric conditions.

The sentiment after this little illustration, is that the emulator seems to do the job. It recreates high-resolution fields which look like the RCM ones, or at least the emulator in perfect model mode. Indeed, the downscaled objects from the GCM look similar to the UPRCM emulated maps. They seem smoother than what creates the

RCM, as we saw in section 5.3. Nevertheless, similarly to the perfect model mode, the emulator creates objects with realistic intensities or characteristics. These six days also illustrate the difficulty of evaluating the emulator in application mode without proper reference. Indeed these day-to-day mismatches make it impossible to differentiate an eventual emulator's bad behaviour and large-scale induced divergence.

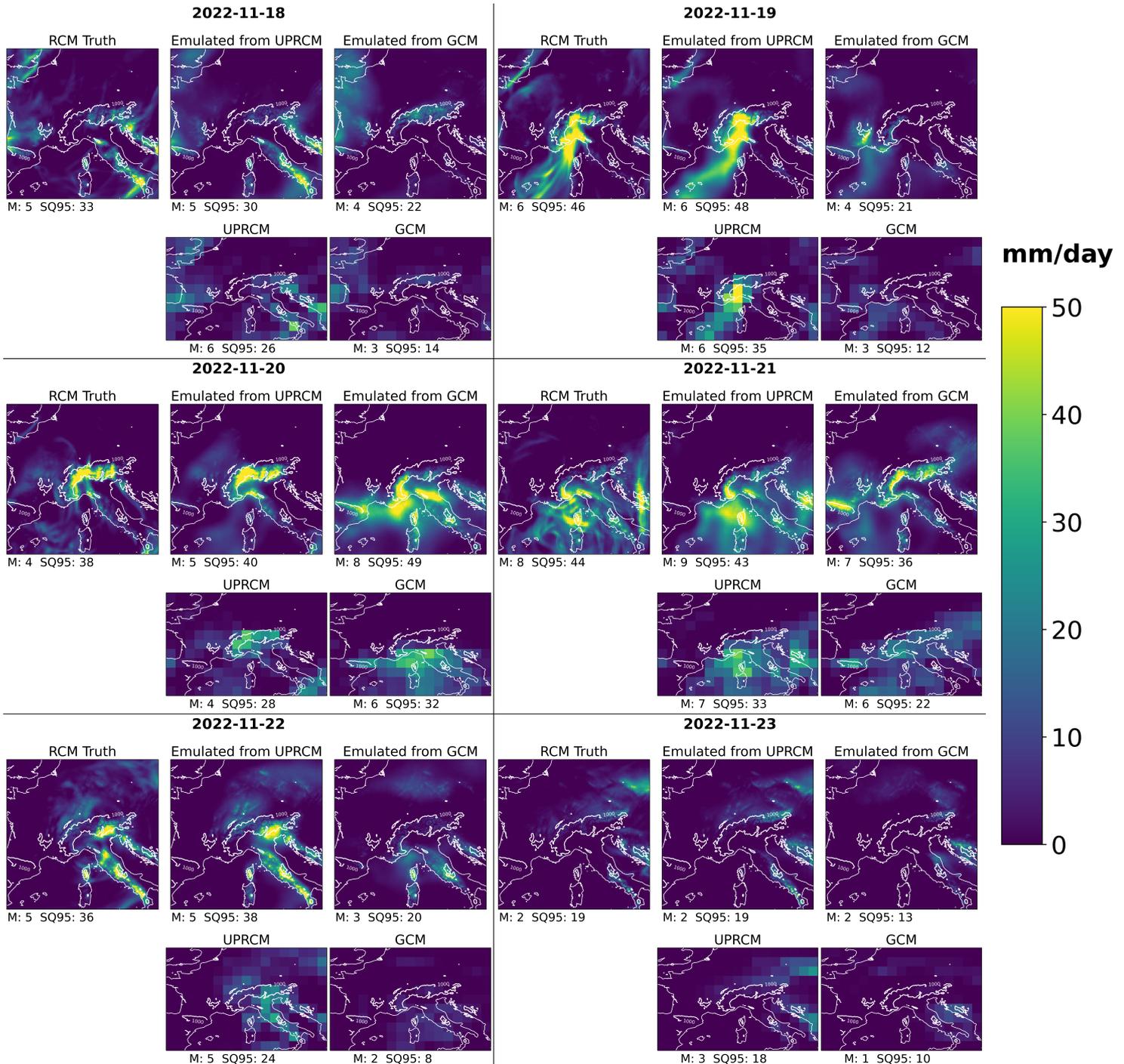


Figure 5.19: Illustration of 6 consecutive days rainfall maps for the RCM, the emulator downscaling the UPRCM, the emulator downscaling the GCM, and both the UPRCM and GCM precipitation fields.

5.4.2 Present climate analysis

Since studying time series reproduction does not make sense, we can look at climatological statistics. As in the perfect model evaluation, I compute the annual average daily rainfall, the 99th quantile and the percentage of dry days in the present climate (2006-2025) in the four simulations. The results are shown as maps in Figure 5.20 because there are too big local biases which cannot be analysed on summary plots. For the same reason, we look at simple biases as relative bias can get very high for small biases on small rain amount.

The first general comment on all these maps is that they all seem realistic, which testify to the emulator's robustness. Indeed, because of the well-known limits of neural networks regarding the input space, it risked making completely unrealistic series. The spatial extremes and mean on the raw maps are in the same order as for the RCM maps. The second reassuring remark is that most biases differ from one model to the other. The emulated CNRM is wetter in the Alps than its corresponding RCM simulation, while it is the opposite for the HGM simulations. It is furthermore notable that the three statistics agree on these two examples. The emulated CNRM 99th quantile is higher over the northern Alps than in the RCM simulation combined with fewer dry days, leading to a stronger average of daily precipitation. The complete opposite analysis can be done for the HGM simulations over the southern Alps. This coherence also gives some confidence in the downscaling of the emulator.

On the other hand, some biases might be more questioning. For instance, all emulated simulations undervalue the 99th quantile over the Cevennes in South France. This specific region is well known for its extreme events, and it is an added value from the RCM to reproduce them better. The emulator produces substantial extreme in this region, but less intense than over the Alps, while there are comparable on the RCM maps. Likewise, the emulated maps show a dry bias on all models around the Adriatic sea and a wet bias on the North of the Pyrenees. Though, it is difficult to assess without a dedicated study if this is indeed a limitation of the emulator.

5.4.3 Climate change analysis

In order to complete the study of the emulator performances on downscaling GCM simulations, it is necessary to look at its ability to create consistent high-resolution climate change maps, as it is the primary purpose of the emulator. To do so, I decided to look at the average precipitation changes in summer for the four simulations used in this section. The aim here is really to illustrate what the emulator produces. Figure 5.21 shows for each model the summer changes for the RCM simulation, the GCM-emulated simulation, the bias between emulator and RCM projections, and the bias map between UPRCM and raw GCM. The changes are shown as the relative differences to the past period, while the biases are the simple difference between the emulator (resp. GCM) and the RCM (resp. UPRCM) maps.

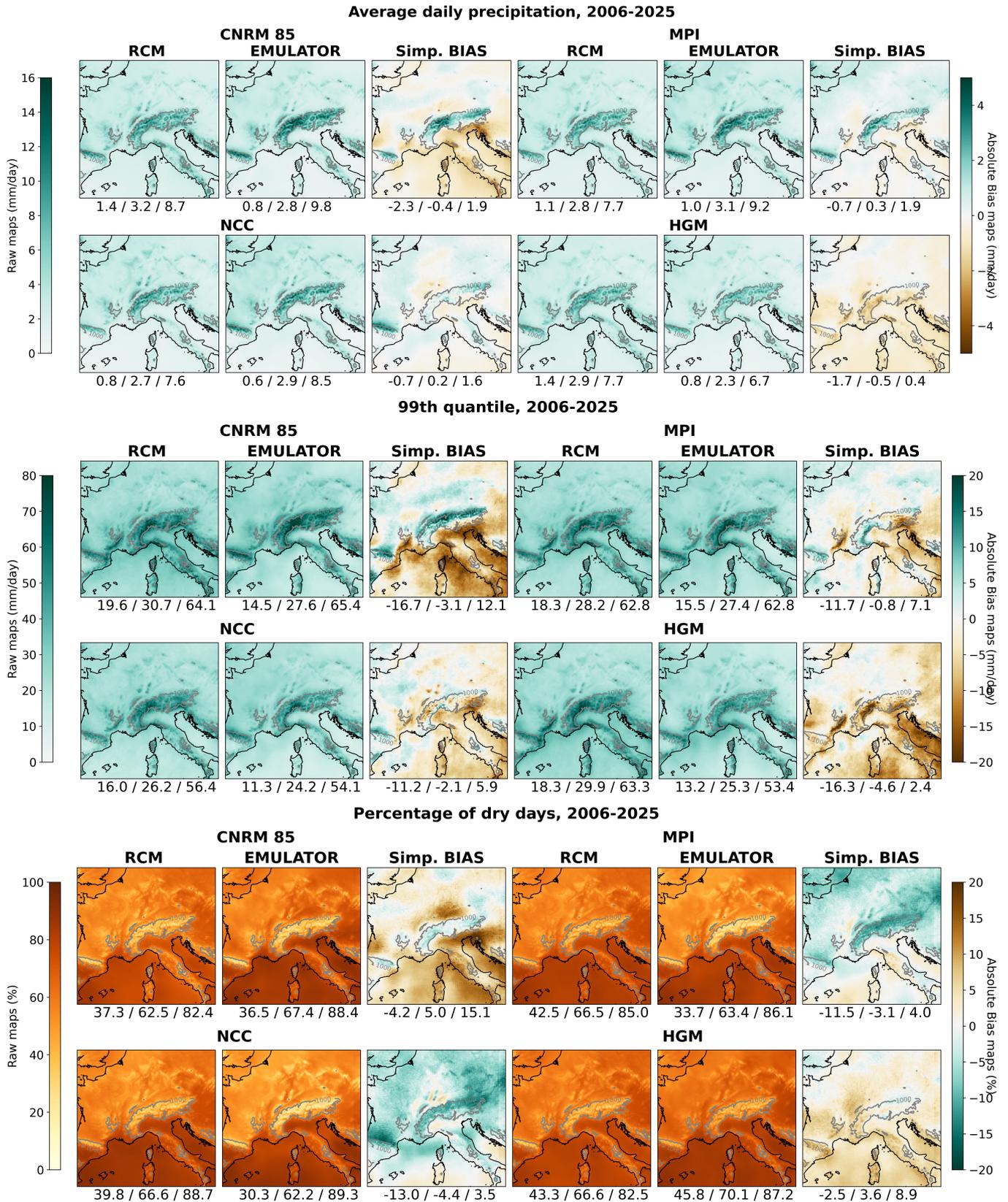


Figure 5.20: Present (2006-2025) climate statistics of 4 simulations (CNRM RCP85, MPI, NCC and HGM) for (Upper) the mean map of daily precipitation accumulations, (middle) the 99th quantile map and (lower) the percentage of dry days. For each simulation, we see the RCM, the emulated one and the simple bias between them. The spatial mean and 95th and 5th superquantiles are given for each map.

The MPI simulations are the ones which agree the most. Both emulator and RCM are drying over the entire domain, with more intensity over the Mediterranean sea and the Atlantic Ocean coasts. The emulator is drying more over the Mediterranean than the RCM. Both Emulator and GCM bias maps are consistent. No strong bias over the land, and more pronounced on the south of the domain.

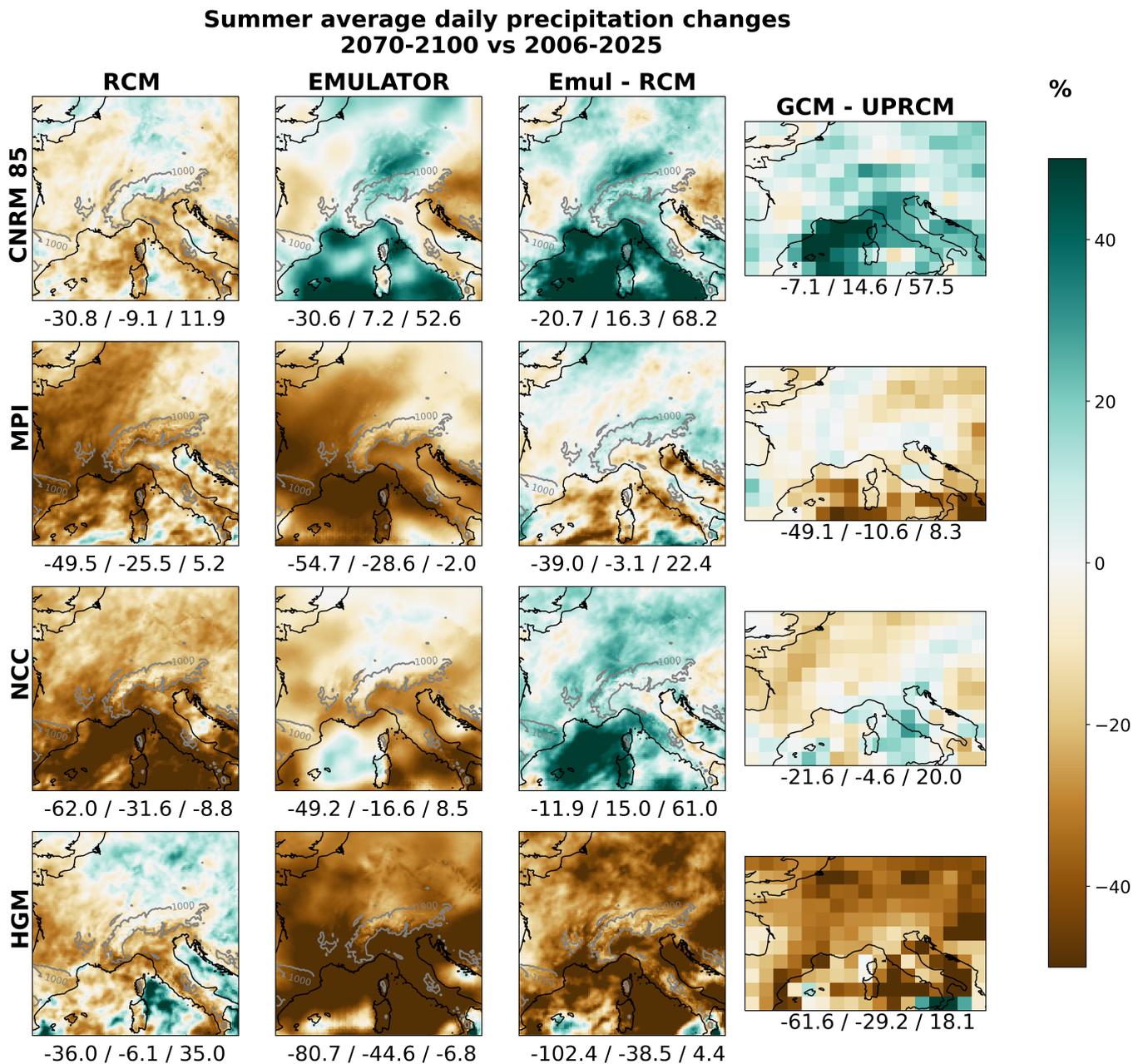


Figure 5.21: Summer relative changes of average daily precipitation between future (2070-2100) and present (2006-2025) period for the 4 GCM simulations downscaled with the emulator: CNRM, MPI, NCC and HGM under RCP85 scenario. From left to right, the column shows: the RCM map, the emulated map (from GCM), the difference between emulator and RCM, and the difference between GCM and UPRCM. The spatial mean and 95th and 5th superquantiles are given for each map.

The HGM emulator is drying much more than the RCM, which completely matches the GCM, as both bias maps are brown, with some extreme differences from the RCM. Surprisingly the CNRM GCM simulation is wetter around the Mediterranean sea in the future. The emulator reproduces it well, as the bias maps are very close. The RCM produces an entirely different change, much more consistent with the other simulations.

Finally, the emulated NCC simulation is quite similar to the RCM one. It is drying over most of the domain, more intensively in the south, such as the RCM. The emulator changes are, however, less pronounced than in the RCM simulation, as shown by the bias map. It is getting wetter in the middle of the Mediterranean Sea, which looks weird, but there is low precipitation on the sea so that it might be simply noise. However, the emulator’s bias map does not match the one from the GCM, which is surprising. The GCM occurs to be globally drier than the UPRCM, while we have the opposite from the emulator. Again, it is difficult to conclude the soundness of the emulator downscaling. Nevertheless, once again, the emulator produces realistic maps without evidence invalidating its application to GCM simulations.

5.5 Discussion

5.5.1 On the asymmetric loss

In this section, I wanted to reflect on the construction of the Emul-ASYM loss. The idea came from the behaviour of Emul-MAE. It predicted the low precipitation very well but largely underestimated the medium and large rainfall. It is mostly due to the unbalancing of the data. Figure 5.11 shows that over the domain, at least 35% of days are dry (less than 1mm), with some grid points reaching more than 80% of dry days. Emul-MAE reproduces better the most raining points as shown by the variance ratio (Fig. 5.9) and the FCSS (Fig. 5.13) maps. The aim was then to give more weight to the underrepresented raining days. I chose to handle this issue by penalising the neural network stronger when it underestimates a precipitation amount. It felt like the right answer to the Emul-MAE default. It is the asymmetric part done by the $\max(0, y_{i,t} - \hat{y}_{i,t})$ parameter in equation 5.4. However, it needed to depend on the intensity of the rain. The more intense the event is, the rarer it is, so the stronger the emulator needs to be penalised if it underestimates it, especially because these events constitute an essential part of RCM’s added value. It also depends on each grid point because of the strong spatial variability of extremes visible on the 99th quantile maps (Fig. 5.11). It is the role of the $\gamma_{i,t}$ parameter.

(Eq. 5.17) is the first asymmetric loss function I tried with good results. It has fixed values of γ according to a certain threshold based on the local quantile function. In each grid point i we compute the p^{th} quantile, and for every target value $y_{i,t}$ the associated $\gamma_{i,t}$ parameter can take two values if $y_{i,t}$ is higher or lower than the threshold. There are 3 parameters to define for this loss function and figure 5.22 shows different attempts. We can see that the value of γ^{UP} strongly impacts the results. Too low is not enough to correct the Emul-MAE, while too

high leads to a considerable overestimation of the precipitations (expected by construction). The γ^{LOW} influences mainly the results on the low precipitations days, where the MAE is already accurate. So 0 or a low value is preferable. The quantile choice also has a strong impact. It is not shown here, but by analysing the pdf curves of these tries, we could see an artificial bump after the given quantile.

$$L(y, \hat{y}) = \frac{1}{N \times T} \sum_{t=0}^T \sum_{i \in \mathcal{D}} |y_{i,t} - \hat{y}_{i,t}| + \gamma_{i,t} \times \max(0, y_{i,t} - \hat{y}_{i,t})$$

$$\text{With: } \gamma_{i,t} = \begin{cases} \gamma^{UP} & \text{if } y_{i,t} \geq Q_i(p) \\ \gamma^{LOW} & \text{if } y_{i,t} < Q_i(p) \end{cases} \quad p \in [0, 1] \quad (5.17)$$

$\gamma^{UP}, \gamma^{LOW}$ and p are fixed parameters to be defined and Q_i is the quantile function in point i .

For this last reason, I searched something continuous which would give similar results. The Gamma distribution has been shown to fit precipitation data correctly. The idea was then to fit in each grid point a gamma distribution and use the associated cumulative function to set the γ parameter according to the target value. Once again, I made multiple tests. The orange bar on Figure 5.22 shows the results when the gamma is fitted on all days. The star red bar shows the results for the loss function as used in this chapter, with the gamma fitted only on days over 1mm rainfall, and the brown bar shows the results without taking the square of the γ . Fitting the Gamma distribution on all days seems to lead to an undervaluation of the number of dry days, increasing the mean value. Taking the square of the γ parameter seems to correct an intensification of the medium rainfall.

I also added on Figure 5.22, two others emulators based on the same loss function as the Emul-ASYM used in the rest of this chapter. The training of neural networks implies a part of randomness (see 3), which can lead to slightly different results. These two other versions help to put into perspective the results. Indeed, if Emul-ASYM overestimated the mean daily rainfall, a twin emulator shows different results and underestimates the 99th quantile map. It does not change the conclusion that Emul-ASYM produces accurate results in reproducing the entire local rainfall distribution. We can see that the three red bars perform better over the three statistics than the Emul-MAE and Emul-MSE and than all other tries. Nevertheless, it needs to be considered when analysing the best loss.

The discussion on the asymmetric loss function felt important, as I do not claim that the choice made here is the best. There could be a better and more robust way re-weight the training sample. We have seen that Emul-ASYM tends to overestimate the medium-range precipitation, probably because the associated γ parameter is too strong. The problem was mainly over the relief where the Emul-MAE was doing a better job due to more balanced data. Maybe the regularisation parameter γ should also consider the MAE performance. The point here is to present what I have reached as it constitutes a basis for more development.

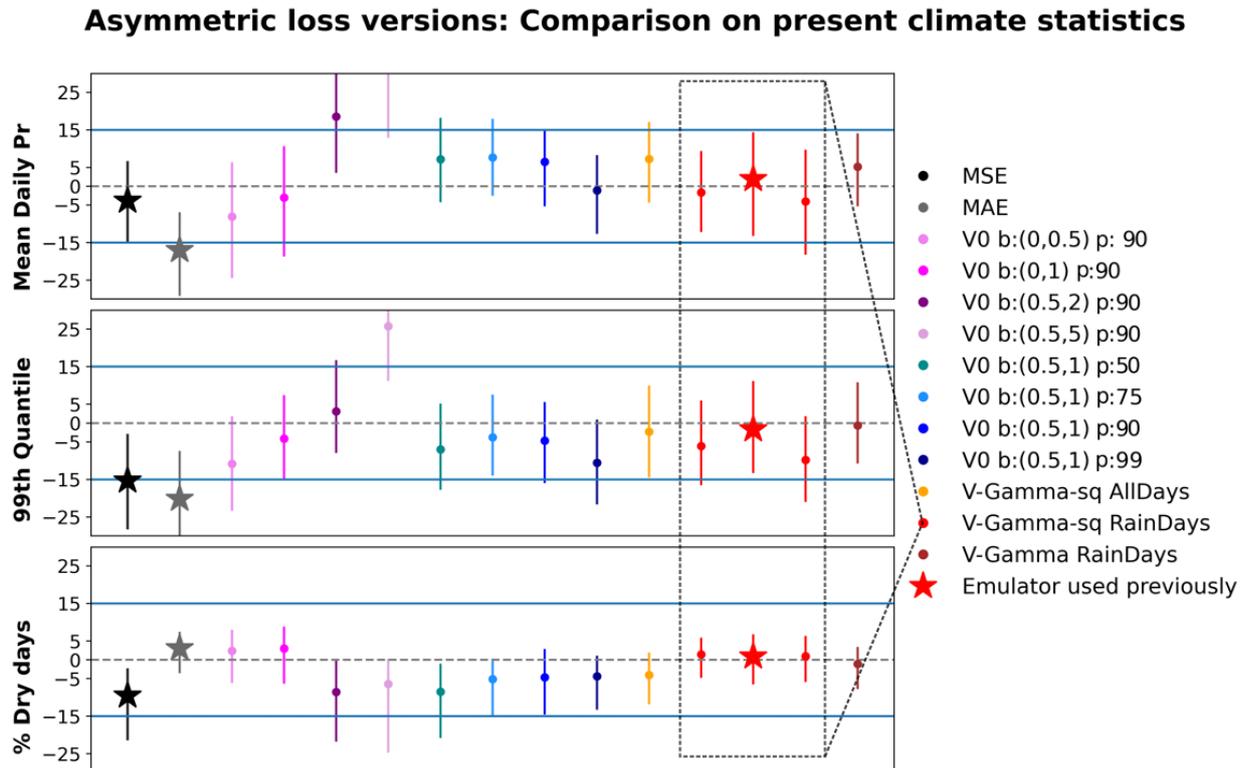


Figure 5.22: Summary plots of the three climatological statistics for different emulators trained with different versions of the asymmetric loss. The “V0” versions correspond to the equation 5.17 with the γ parameter noted “b” here and p the threshold quantile. See sec 5.5.1.

5.5.2 On the computational efficiency

The target domain chosen for this chapter is four times larger than in the previous chapters. We saw that the RCM-emulator showed good abilities in this new domain, and even if it is not the same variable, it seemed to follow the large-scale predictors perfectly. This more extensive domain is an excellent opportunity to make a short evaluation of computational efficiency. In the previous chapters, the target domain is a square with 64 grid points side, and the input domain is a square of 16×16 low-resolution grid points. In this chapter, the target domain size is 128×128 , while the input domain includes 22 grid points in longitude and 16 in latitude. Both neural networks include about 25 million parameters to fit. Indeed the layers with the most parameters are at the bottom of the architecture.

Regarding available memory, it was impossible to train the precipitation emulator on the new domain with the same GPU as in Chapter 1. So I trained both emulators on a bigger GPU to compare the computational times. The GPU used here is an Nvidia Tesla V100 PCIe 16GB. Both emulators necessitate about 60 epochs to be trained. Actually, only 45 are necessary, as the neural network does not improve on the last 15 epochs. 1 epoch on the new domain takes about 130 seconds, which implies a total training time of 2 hours and 10min.

On the other hand, the emulator trained on the smaller domain needs 65 seconds

per epoch. The training of the emulator for the more extensive domain is two times longer, while the domain is four times bigger. The computational resources are then not linearly proportional to the target domain size. Moreover, the entire ALADIN63 domain is of size 450×450 grid points, less than four times bigger than the target domain considered in this chapter. Even when we use 4 emulators to cover the entire ALADIN63 domain, it will take less than 10 hours to train. It is not comparable with the computational time of ALADIN63, which needs several weeks to perform one simulation. Besides, the training of the emulator can probably be more optimised as I am not an expert in GPU resource management. Finally it is important to note here that once the emulator trained the producing time¹ takes only few seconds even on the bigger domain.

5.6 Conclusion

This chapter aims to propose a credible solution to the high computational costs of Regional Climate Models to build large ensembles of high-resolution precipitation projections. It relies on the RCM-emulator introduced in Chapter 3 for the case of temperature downscaling. RCM-emulators belong to the family of hybrid downscaling methods. They use RCM simulations to estimate the downscaling relationship between low-resolution and large-scale variables and a high-resolution surface variable. This chapter aims to see if it is suitable for the complex variable of precipitation, including the extreme parts of its distribution, and, if not, how it can be adapted.

The RCM-Emulator used here is exactly the same as in the rest of this manuscript. It takes the same set of predictors: various two-dimensional variables at different pressure levels and external forcing, such as greenhouse gas concentrations. The predictors are normalised exactly the same way. The neural network architecture is also the same, and the emulator is trained in the same perfect model framework where both predictors and targets come from the same RCM simulations. The predictors are then upscaled to low resolution so that they look like a GCM simulation output. The perfect model framework ensures a perfect match between input and output and grants the focus of the neural network on the downscaling function. The only difference with chapter 3 is the target domain which is more extensive and centred over the Alps here. It implied an adaptation of the input domain and a little extension of the neural network. Of course, the target variable is also different as we want to emulate the precipitation field from ALADIN63.

The same neural network is referred to as Emul-MSE in this chapter. In perfect model evaluation, it showed an excellent capacity to follow the large-scale atmospheric circulation and to recreate the precipitation times series with reasonable accuracy. However, the climatological analysis revealed a problem in the intensity. The emulator concentrates its prediction around the climatological mean, with too many medium-low precipitations and a clear underestimation of the number of dry

¹This do take into account the preparation of the data. The longest part is to actually get the data from online servers. Then the cropping, interpolation and smoothing can be done in few hours.

days and the 99th quantile. It is due to the complexity of the precipitation variable, which is non-continuous and highly unbalanced. In this chapter, I proposed to adjust the loss function to tackle this issue.

I proposed an asymmetric loss function designed to penalise the emulator stronger in the training phase when it underestimates the true precipitation amount; the more extreme the rain, the stronger the penalty. It showed that the loss function is an excellent means of constraining neural networks to focus on specific aspects as it corrected the emulator prediction to fit better the precipitation intensity. This adapted emulator showed the same good properties as the original emulator in terms of the temporal match with the original RCM series in perfect model evaluation. However, it reproduces the temporal variance and the entire distribution better. At the climatological scale, it reproduces almost perfectly the maps of the daily average precipitation, the 99th quantile or the percentage of dry days, with notably the reproduction of the high-resolution spatial structure. It has some local biases but they never exceed 15%. Moreover, the precipitation objects recreated by the emulator are also coherent with the ones created by the RCM. They are well located and have the right intensity, but they tend to be more spatially continuous than on the RCM simulation. Nevertheless, this does not affect the statistical properties of the series, even at the grid point scale.

I extended the evaluation of the emulator to multiple RCM simulations. The perfect model evaluation grants the perfect match between the predictor and the expected output. So the emulator was expected to reproduce all differences between the RCM simulations. The abilities of the emulator extended perfectly to all simulations. It showed the same good skills and default for all simulations. The emulator tended to overestimate the precipitations where they are the strongest, which stayed the case, in the same proportions for all tested simulations. In the climate change context, the emulator has also shown an excellent ability to reproduce the complex spatial patterns of change for each simulation and the three statistics we watched. Furthermore, the loss function does not impact the climate change maps from the emulator. It does not impact how the neural network follows the large scale but corrects the intensity. If the emulator underestimates the extreme precipitation, it does it similarly throughout the simulation. The event's occurrence drives the change, and we saw that the loss function does not impact the temporal correlation.

The application of the emulator to GCM simulations is a real-world exercise. Nevertheless, it is very challenging to evaluate. Indeed, the emulator is constructed in a perfect model framework to ensure a perfect match between predictors and target variables. It focuses only on the downscaling function and can be applied to any GCM simulation. On the other hand, it implies that if there are any large-scale biases between an RCM simulation and its driving GCM, the emulator will reproduce it. Such biases have been demonstrated in chapters 3 and 4 and in various studies (Boé et al, 2020; Bartók et al, 2017; Taranu et al, 2022). There is then no high-resolution reference to evaluate the emulator when it downscales a GCM simulation. I applied the emulator on four RCP85 simulations from the four GCMs used to drive ALADIN63. The results are very consistent with the perfect model evaluation. The emulator seems to follow the GCM large

scale very accurately. It produces coherent and realistic climatological maps with locally strong biases with the RCM. Looking at climate change maps, they also might differ from the RCM ones. However, they stay coherent with the GCM/RCM biases. In my opinion, the emulator fits the job for downscaling GCM simulations. However, to have better trust in it, the emulated GCM simulations should be looked at deeper and maybe with different approaches. In particular, setting a GCM/RCM pair without large scale discrepancies may be the opportunity for a proper evaluation of the emulator when downscaling a GCM simulation.

In terms of computational efficiency, the more extensive domain implied to use a GPU with more available memory to train the emulator. We could see that it takes about two hours to train the emulator while the production takes less than a minute. It is not comparable to the time necessary to produce a new ALADIN63 simulation. Moreover, the emulator was easily transferable to a new variable and a new domain. The loss function is a good way to constrain and force a neural network to specific events and better fit the entire distribution of a variable as complex as precipitation. I think the RCM-emulator is a promising way to create large ensembles of high-resolution precipitation simulations by downscaling all existing GCM simulations.

Chapter 6

Conclusion & Perspectives

This final chapter aims to summarise the main conclusions of this work, to present the big questions raised and introduce some perspectives on improvements, applications or developments. This PhD targeted to propose a solution to enlarge the size of regional climate model (RCM) simulation ensembles. These models' increasing resolution and complexity imply high computational costs that strongly limit the number of simulations and, thus, the study of the associated uncertainties. The reliability of the regional climate change messages extracted from the actual RCMs ensembles is then insufficient. The correct exploration of all possible futures at local scales is necessary to impact studies and organise adaptation policies. To address this issue, I propose in this manuscript a hybrid downscaling approach as it relies on the two distinct families of downscaling: dynamical and statistical. While the former relies on RCMs, the latter aims to learn the statistical relationship between what I called all along this report large and local scales. The large scales correspond to some low-resolution fields in altitude or averaged surface fields over a vast domain that global climate models (GCMs) should represent correctly. On the other hand, the local scale are characterizing the whether that is actually experimented by humans. They range from one to hundred kilometres. They depend on the local environment and necessitate high-resolution climate models to be represented.

Therefore, I proposed here to build an RCM-Emulator that relies on the same idea as statistical downscaling. However, instead of targeting the *real* one, the emulator aims to learn the large scale/local scale relationship simulated by a given RCM. The advantage of this approach is that the relationship can be estimated in non-stationary climates and regions with no observations. Of course, the emulator does not aim to correct the relationship simulated by the RCM; it learns to reproduce it with its defaults. By constructing one emulator for all existing RCMs, it would be possible to downscale, for some variables, all existing GCMs simulations with all RCMs, resulting in large RCMs/GCMs matrices. Studying those ensembles would be an important step towards reliable local information on climate change.

In what follows, I propose to come back on the choices I made to build the RCM emulator, the reasons behind them, and the essential results and questions that this work has raised. This final chapter comports six sections. The first one recalls the critical choices in defining the RCM emulator framework. Sections 6.2 to 6.5 answer

the three other questions about the efficiency, the trustability and the applicability of the emulator that I have set at the end of the introductory chapter. In those four sections, I also present some perspectives of improvement or development directly linked to the issues that stay open regarding those specific aspects. Finally, the last section presents some more general perspectives.

6.1 RCM-Emulator concept

This work's first part consisted of defining the appropriate framework to build the emulator. The aim was first to identify and isolate the downscaling function of the RCM and then find the most appropriate tool and dataset to learn it. I decided to describe the RCM dynamical downscaling as a combination of two functions. The RCM receives large scale inputs from the GCM that it firstly transforms, creating its own large scale, and in a second time it downscales this *new* large scale to create high-resolution fields. There is no reason to assume that the large scale transformation is the same for any GCM, and results from Chapter 4 tends to confirm it. Therefore, we need to focus only on the downscaling function to have a universal emulator. This choice implies some important assumptions on the causes of the large scales transformation that I will discuss later. I proposed to use a perfect model framework with both large scale and local scale coming from the same RCM simulation. The idea is to create, from the RCM simulation, a low-resolution large scale looking like a GCM output and learn the transfer function to the high-resolution surface field. This strategy ensures a perfect coherence between the large scale low-resolution predictors and the high-resolution local scale predictands.

To estimate this relationship, I used a fully convolutional neural network architecture called UNet that I adapted to fit our regression task. The efficiency of CNNs in dealing with large images made them the perfect candidate to build the emulator. Indeed both inputs and outputs of the emulator are gridded objects that can be considered as images. Moreover, their ability to deal with large dimensional objects allows us to make a minimal prior selection of needed inputs and leave the network finding the right combinations for each output grid point. Various studies have already used convolutional neural networks in statistical downscaling (Vandal et al, 2019; Baño-Medina et al, 2020) or hybrid frameworks (Serifi et al, 2021) with success.

I defined a common set of inputs that I kept for the whole study. It mainly includes classical atmospheric variables (temperature, geopotential height, humidity and wind) at three pressure levels plus some surface variables that should be accurate in the GCM and the aerosols as they play an important role at the regional scale (Nabat et al, 2020). I also included external forcings such as the concentration of greenhouse gases and a vector representing the seasonal cycle. The flexibility of the neural network architecture allows separating the predictors' spatial structure information and the temporal part so that the neural network considers them equally before reconstructing the high-resolution field. The choice of the input set could still be discussed and maybe optimised. In particular, the predictors come from the RCM simulation during the emulator calibration. It is crucial to ensure

that the predictors do not include some RCM high resolution signal. Otherwise, the downscaling function we try to learn is truncated and thus cannot be well reproduced in the application world. It is why the low-resolution target variable is not included in the predictors set (cf chapter 3). It could be that such local information remains in the inputs set I used and further analysis to detect it is necessary. We have also seen that a RCM-emulator trained with a minimal set of predictor can provide reasonable results. Regarding an optimisation of the input set, a sensitivity analysis to identify the influence of each predictor is a necessary study that I did not have time to explore in this work.

Thus once this is set, this work aimed to assess if the RCM emulator was a suitable option to fill up RCM matrices. To do so, we need to give some guarantees regarding three main aspects: the efficiency, trustability and applicability. In what follow, I detail those three aspects.

6.2 Efficiency

Efficiency is the first reason behind this work and the statistical RCM emulator. Indeed the objective is to bypass the extremely high cost of regional climate models. In this manuscript, we have seen that the neural network proposed could be trained on GPU within only a few hours, even on a large domain. The production of a new high-resolution series given a low-resolution simulation takes less than a minute. The longest part is finally to get the input fields from the simulation to downscale and interpolate them to the emulator input grid. This would also take some hours and could easily be automatised (at least the interpolation part). Regarding this aspect, the efficiency of the emulator is satisfactory.

Nevertheless, efficiency can be considered under other facets not raised in this work. An important point is the easiness of use and application. A very practical point, once trained, the emulator can be stored in a `.h5`¹ file of less than 2GB. So it is easy to transport and can be used on GPU and CPU. A second point is the easiness of application to a slightly different problem. We have seen in the manuscript that the emulator was used on different domains and variables. The code available on a public GitHub page has not been changed. Indeed, I created a python function that designs the neural network architecture according to the size of inputs and target. The choice of the target variable does not impact the code unless specific adaptations, such as the loss function in the precipitation case, for example. Moreover, during the time of my PhD, two intern students in other research centres deployed our methodology to the MAR RCM over Antarctica and for different variables. Emma Amblard² emulated the surface temperature and precipitation, while Marijn Van der Meer³, worked with a very different variable, the Surface Mass Balance. Both internships got excellent results showing the emulator methodology and concept's good transferability and ease of use.

¹Hierarchical Data Format 5

²Internship at the IPSL, supervised by Cecile Agosta, Patrick Gallinari, Christoph Kittel and myself

³Internship at the TU-DELFT, supervised by Stef Lhermite

Nonetheless, there is still space for improvement. First, regarding the UNet architecture I built, there is probably a way to improve computational efficiency. A costing part of the architecture is the bottom of the “U”. I chose to go down to a 1-dimensional encoding vector in order to put on the same level the temporal information, which is 1D, and the spatial information, originally 3-D, before the reconstruction. It implies many layers and parameters to fit to reach the target dimension. One could modify this neural network architecture, making the encoded array larger and improving the emulator’s computational cost. Besides, it would allow getting the best out of the UNet architecture. Indeed, some studies have developed a dilated UNet allowing multiple input sizes thanks to its fully convolutional aspect. Until now, once trained, the emulator focuses on a specific domain and can be applied only to this domain. One could imagine a RCM-emulator for the entire RCM domain and able to consider only a subdomain. The orography and longitude/latitude coordinates of the target domain would be added to the input list, as well as the coordinate map of the input domain. Each training sample would represent a piece of the domain. In application mode, one could imagine a very interactive tool located on a server close to the data centres. The user could choose its target domain of interest, the adapted input domain, and easily produce high-resolution emulated simulations. A more complete tool, could allow the user to choose the RCM to downscale, the training set, the input list and the target variable to build a new emulator. Such a tool implies a lot of development (and human resources), but I believe that it is realistic.

Finally, simpler or more efficient architectures could also be used. I believe that this work, and the other pioneer works of CNN application to downscaling, open the door to constructing different emulators. The diversity of large emulated ensembles will benefit the production of reliable regional climate information.

6.3 Trustability

The second key point was to assess the trustability of the RCM emulator. The objective is to have a tool reproducing the downscaling function from a given RCM to apply it to other GCM simulations. It is then of primary importance to assess if the emulator well estimates the function. Because it is trained in the perfect model framework, the emulator *lives* in the RCM world, it is then easy to evaluate if this large scale/local scale relationship is correctly learned. I firstly used an RCM simulation similar to the training one (i.e. driven by the same GCM and the closest, more intermediate scenario). I recreated some low resolution from it to evaluate the emulator in the perfect model framework and to have a perfect reference to compare the emulated series. We saw that the emulator perfectly captures the atmospheric circulation in both precipitation and temperature cases. The emulator is always perfectly coherent with its driving large scale (i.e. the simulation it downscales) and notably shows excellent temporal correlations with the reference series. Moreover, we saw that it also captures the high-resolution spatial structure brought by the RCM. The reliefs, coastal areas or islands have specific climates that the emulator well reproduced. In this work, I took time to deeply evaluate the emulator under

various aspects, similarly as it is done for RCMs. It was essential to assess if the production of the emulator is good enough to be used in impact studies. The main conclusion regarding this aspect is that the emulator produces realistic maps and series that are almost indistinguishable from the RCM ones. We also have seen that the statistical properties from the emulated series were almost identical from the original RCM series, with notably a complete reproduction of the probability density function. We also looked at indices of extremes, climatology mean and quantiles, and the changes in future with good reproduction of the emulator. We also validated the neural network algorithm by comparing to standard statistical downscaling approaches used as emulator. The UNet-emulator showed globally better performances even if those strategies also had reasonable results.

Nevertheless, there is still some room for improvement. We notably have seen that the emulator encounters difficulties reproducing some temperature extremes and the precipitation objects tend to be too smooth. For the former, we identified that an adapted loss function could help to reproduce better the extremes of precipitation. A similar strategy could easily be used to constrain the emulator to focus more on temperature extremes. A possible explanation could also be the lack of memory in the downscaling, as the days are considered separately, and the chronology comes only from the predictors. For example, the presence of snow impacts the near-surface temperature. However, the emulator cannot see it as it results from the previous days, and the emulator does not consider it. Similarly, a deficit in soil moisture can reinforce the high temperature extremes, and the emulator can not consider this aspect either. Therefore, it might be a good idea to include some temporal link between emulator predictions. This could be done using Recurrent Neural Network and Long Short Term memory networks.

The too smooth representation of rainfall objects is explainable by the determinism of the neural network used here. Indeed the probability that rainfall occurs in a grid point or next to it is often the same. So the network, by minimising the error, puts rain in both grid points. Nevertheless, we saw that the emulated series' statistical distribution is similar to the original RCM one, with a good reproduction of dry days and heavy rainfalls. A deeper analysis, including an impact study, might be necessary to see if it is a real issue. If it is, I can see at least one option to tackle this issue. Generative Adversarial Networks are neural networks that produce one possible outcome instead of producing what minimises the error. Precipitation nowcasting studies based on GAN showed their ability to produce realistic rainfall images with precise and sharp objects. They are probably a good option for our case. As a brief reminder, GANs are composed of two fighting networks: the Generator generates images, and the Discriminator learns to differentiate the generated images from the true ones. In our case, the Generator would be the adapted UNet. The Discriminator can be considered simply as a loss function, telling the Generator if the image is realistic. I started to develop this aspect, but the results were too preliminary (and not good enough) to be included here.

6.4 Applicability step 1: different sources of input

Until now, we have an efficient emulator that correctly captures the relationship it is designed to learn. The next step to assess if the emulator is the right tool to enlarge RCM matrices is to ensure that the relationship learnt holds in a slightly different world. Neural networks might be unstable to a slight modification of the input space. I have tried to prevent such an effect in this work with two strategies. In the network architecture, the pooling layer and the Relu activation function help to ensure the stability of the emulator. On the other hand, by artificially dissociating the spatial information from the temporal information and standardising the input, I have tried to control the discrepancies between two simulations. Indeed, the aim was somehow to reduce the input space by leaving only the spatial information at the grid point scale and extracting the temporal one at the domain scale. In chapter 4 I have shown that the perfect model framework is well adapted to test the robustness of the RCM-emulator to different input sources. Therefore, I took all ALADIN63 simulations and extracted some large scales predictors that I downscaled with the emulator. In both temperature and precipitation cases, we saw that the emulator maps and local time series were always close to the RCM ones. The statistical evaluation regarding the mean or some specific quantiles maps and the entire pdfs showed similar performance on all simulations. Moreover, the differences between simulations in climate change were also reproduced in the emulated series. This result shows that the relationship learnt in a given simulation holds in other simulations from the same RCM. Besides, a short experiment presented in chapter 4 showed that the normalisation strategy is crucial to secure this result.

Nevertheless, the study of the transferability to other input sources also highlighted some emulator limitations. Indeed, we saw that the tool is not appropriate to extrapolate too far from its training set, notably regarding a warmer or colder climate. To further explore the relationship's robustness, I trained various emulators using combinations of all available simulations. It helped to draw the following conclusions. Firstly regarding the driving scenario, we concluded that training in the coldest and warmest scenarios allows downscaling more intermediate scenarios, but the inverse does not hold. Secondly, we saw that some RCM simulations are constantly warmer than others. Thus, training in a *cold* simulation does not allow reproducing the warmer simulation's warmest extremes, leading to a slight cold bias when looking at climatological means. Conversely, training in a globally warmer simulation leads to warm bias in a colder simulation. Finally, we could also observe that some simulations seem less good than others to be considered as training sets.

All these results provide some indications of how to build the best training set for the emulator. The great principle could be “the most diversified, the better”, which seems logical. At fixed computational costs, the best way to build a training set for the emulator is to downscale with an RCM the most different GCMs and the most extreme scenarios possible. Moreover, having more than one driving GCM for the training set prevents using only a too-specific simulation for calibrating the emulator and thus ensures better transferability. Finally, I have also shown that

more extended simulations are better. Indeed, I did not present the result in this manuscript. However, a short experiment done for the EUCP project seemed to show that 20 years of historical and 20 years of far future are the strict minima to have a reasonable emulator. And longer is better as it explores more climate and better the internal variability.

The strong conclusions here are that a RCM-emulator trained in a given Scenario-GCM-RCM triplet will be able to downscale all other members from the same Scenario-GCM pair. Secondly, there is quite high confidence that it can also downscale simulation from the same GCM and more intermediate scenarios. Finally, there is a good confidence that it produces accurate series when downscaling simulations from an other GCM as long as it is not too far from the training one. Therefore, using the most diversified training set is probably a good solution to prevent such a situation.

6.5 Applicability step 2: GCM world

Finally, the last point to validate the actual applicability of the RCM-Emulator proposed here is its good behaviour when applied to GCM simulations. It is the *real life* of the emulator, as all previous steps are perfect model validation. The final aim of the RCM-emulator is to downscale GCM simulations never downscaled by the RCM. This is definitively the most challenging point of this work. Indeed, over the last years, various studies have raised the inconsistencies that can occur between RCM and their driving GCM (Laprise et al, 2008; Bartók et al, 2017; Boé et al, 2020; Taranu et al, 2022). We also have been confronted with this problem when evaluating the downscaling performed by the emulator. Indeed, as we focused only on the downscaling function, the emulator does not reproduce GCM/RCM large scale biases. Therefore, the RCM simulation driven by a given GCM simulation is not an exact reference when the emulator is applied to this GCM simulation. The emulated series needs to be coherent with its driving large scale but should also include all the added value brought by the RCM. In both temperature and precipitation cases, the emulator seemed to follow the chronology of the GCM simulation it downscaled. Moreover, we saw that the biases between the emulator and RCM are generally coherent with the RCM/GCM ones. The emulated maps contained the complex spatial structure resulting from the RCM resolution at daily or climatological time scales and regarding both mean states or extremes. This result shows that the emulator includes the RCM added value on the GCM large scale quite well. We have seen in precipitation some counter-example that might imply an incomplete reproduction of the RCM added value. However, it is complicated to assess whether it is an emulator error.

Consequently, it is necessary to find a proper manner to evaluate the RCM-Emulator when it is applied to a GCM simulation. Several options could be considered. Firstly, using a nudged simulation for the evaluation as this technique forces the RCM large scales to respect the GCM ones. Then the RCM series should be a better reference under the hypothesis that the nudging does not modify the downscaling function. It is an easy test that is necessary to do soon. A second option could be to build a perfect RCM/GCM pair. Taranu et al (2022) show that

setting up a nearly-perfect GCM-RCM with minimal large scale biases is feasible, and it could become a proper test for the emulator. The last option (in my opinion, at least) could be to apply the emulator to some reanalysis and compare its result to the observed data and the RCM downscaling. If the emulator reproduces the entire RCM downscaling function without large scale biases, the emulated series should be closer to the observations than the RCM simulation.

A fundamental question here is the soundness of those RCM/GCM biases. In this work, we somehow assumed that the large scale biases are not legitimate as the framework I set implies not learning them. This strong assumption seems to be confirmed by some recent studies (Taranu et al, 2022). In this case, an actual result from this work is that we no longer aim to complete an RCM/GCM matrix but to construct another one. Indeed we assume that the existing matrix includes inconsistent biases that we get rid of with the emulator. The emulated matrix is somehow the one we should have in the perfect world where RCMs only downscale the GCM information without transforming it. Of course, it relies on two strong assumptions that are not verified yet: the emulator captures the entire downscaling function in the RCM perfectly, and the transformation function of the RCM is for the wrong reasons. On the other hand, if the last hypothesis does not hold, then the framework I proposed to train the emulator breaks down, and it is necessary to learn also this large scale transformation.

6.6 More general perspectives

I believe this work opens the door to multiple studies to evaluate better, improve or apply the emulator. In the last sections, I presented some perspectives to answer different problems raised during my work. Nevertheless, I still see some ideas for further development and application.

The first point necessary to assess is the multivariate consistency of the emulator. In other words, the emulator must reproduce the exact relationship between temperature and precipitation. Most impact studies necessitate temperature and precipitation, and a real relationship must exist between them. In perfect model mode, we have seen that the emulated series follows almost perfectly the RCM one. If the relationship is realistic in the RCM, it should also be in the emulated simulation. Nevertheless, I did not watch this specifically, and it should be done. Besides, it could also be a way to verify the accuracy of downscaling a GCM simulation. A multivariate emulator is perfectly imaginable if the relationship is not well reproduced. A constraint on the joint loss could force the network to respect a good consistency between precipitation and temperature.

A second point concerns higher resolution in space and time. Indeed, in the introduction, I mentioned the existence of Convection Permitting Models with a horizontal resolution of less than 3km. Those models have notably shown an excellent ability to reproduce hourly precipitation and are necessary to study extreme and devastating precipitation changes. The high resolution of these models implies very high computational costs. RCM emulator could then be necessary

to extend the number of simulations. I have already done some preliminary tests on a daily scale with such models. It provided good results, but the emulator could not capture the finest spatial structure that the CPM creates. Today, I can see two possible reasons behind this: the resolution jumps (from 150km to 2.5km) are too big as I used the same input grid as the one in this work, or I did not have a long enough simulation (20 years in total). This transfer to higher spatial resolution necessitates further developments but is accessible. On the other hand, hourly prediction is another challenge. I can see two possibilities to do it. Firstly, following the same principle as for the daily prediction. The emulator considers each hour independently, with hourly inputs and maybe a cosinus/sinus vector to represent the day's hour (similarly to the seasonal one I implemented here). The big disadvantage of this option is that it implies a good quality hourly input which the GCM might not well represent. Moreover, it implies heavy input files and more preprocessing. The second option is to use lower temporal resolution inputs (daily or 6 hours, for example) and also down-scale the temporal resolution using recurrent approaches like in nowcasting exercises.

Finally, the last perspective, probably the most evident, concerns the application of RCM emulators. If the tool is trusted, it could be used to downscale huge ensembles of GCM simulation. It could completely change the way performing local studies of climate change. An entirely statistical approach could be used to extract robust and reliable messages on the local implications of global climate warming. The internal variability could be entirely explored, and the most extreme possible events could be studied. Moreover, I believe that a diversity of emulators based on different statistical approaches would serve the purpose of exploring the possibilities and extracting robust information.

Bibliography

- Alexandru A, De Elia R, Laprise R, Separovic L, Biner S (2009) Sensitivity study of regional climate model simulations to large-scale nudging parameters. *Monthly Weather Review* 137(5):1666–1686, DOI 10.1175/2008MWR2620.1
- Ayzel G, Scheffer T, Heistermann M (2020) RainNet v1.0: A convolutional neural network for radar-based precipitation nowcasting. *Geoscientific Model Development* 13(6):2631–2644, DOI 10.5194/GMD-13-2631-2020
- Babaousmail H, Hou R, Gnitou GT, Ayugi B (2021) Novel statistical downscaling emulator for precipitation projections using deep convolutional autoencoder over northern africa. *Journal of Atmospheric and Solar-Terrestrial Physics* 218:105,614, DOI <https://doi.org/10.1016/j.jastp.2021.105614>, URL <https://www.sciencedirect.com/science/article/pii/S1364682621000754>
- Balogh B, Saint-Martin D, Ribes A (2022) How to Calibrate a Dynamical System With Neural Network Based Physics? *Geophysical Research Letters* 49(8):e2022GL097,872, DOI 10.1029/2022GL097872, URL <https://onlinelibrary-wiley-com.insu.bib.cnrs.fr/doi/full/10.1029/2022GL097872> <https://onlinelibrary-wiley-com.insu.bib.cnrs.fr/doi/abs/10.1029/2022GL097872> <https://agupubs-onlinelibrary-wiley-com.insu.bib.cnrs.fr/doi/10.1029/2022GL097872>
- Ban N, Caillaud C, Coppola E, Pichelli E, Sobolowski S, Adinolfi M, Ahrens B, Alias A, Anders I, Bastin S, Belušić D, Berthou S, Brisson E, Cardoso RM, Chan SC, Christensen OB, Fernández J, Fita L, Frisius T, Gašparac G, Giorgi F, Goergen K, Haugen JE, Hodnebrog O, Kartsios S, Katragkou E, Kendon EJ, Keuler K, Lavin-Gullon A, Lenderink G, Leutwyler D, Lorenz T, Maraun D, Mercogliano P, Milovac J, Panitz HJ, Raffa M, Remedio AR, Schär C, Soares PM, Srnec L, Steensen BM, Stocchi P, Tölle MH, Truhetz H, Vergara-Temprado J, de Vries H, Warrach-Sagi K, Wulfmeyer V, Zander MJ (2021) The first multi-model ensemble of regional climate simulations at kilometer-scale resolution, part I: evaluation of precipitation. *Climate Dynamics* 57(1-2):275–302, DOI 10.1007/s00382-021-05708-w
- Baño-Medina J, Manzananas R, Gutierrez JM, Gutiérrez JM (2020) Configuration and intercomparison of deep learning neural models for statistical downscaling. *Geoscientific Model Development* 13(4):2109–2124, DOI 10.5194/gmd-13-2109-2020, URL <https://gmd.copernicus.org/articles/13/2109/2020/>
- Baño-Medina J, Manzananas R, Gutiérrez JM (2021) On the suitability of deep convolutional neural networks for continental-wide downscaling of climate change

- projections. *Climate Dynamics* (0123456789), DOI 10.1007/s00382-021-05847-0, URL <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. *Climate Dynamics* 49(7-8):2665–2683, DOI 10.1007/s00382-016-3471-2
- Bentsen M, Bethke I, Debernard JB, Iversen T, Kirkevåg A, Seland O, Drange H, Roelandt C, Seierstad IA, Hoose C, Kristjánsson JE (2013) The Norwegian Earth System Model, NorESM1-M – Part 1: Description and basic evaluation of the physical climate. *Geoscientific Model Development* 6(3):687–720, DOI 10.5194/gmd-6-687-2013
- 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. *Journal of Climate* 28(2):401–421, DOI 10.1175/JCLI-D-14-00316.1, URL <http://dx.doi.org/10.1175/JCLI-D-14->
- Berthou S, Kendon EJ, Chan SC, Ban N, Leutwyler D, Schär C, Fosser G (2020) Pan-European climate at convection-permitting scale: a model intercomparison study. *Climate Dynamics* 55(1-2):35–59, DOI 10.1007/s00382-018-4114-6
- Bettolli ML, Solman SA, da Rocha RP, Llopart M, Gutierrez JM, Fernández J, Olmo ME, Lavin-Gullon A, Chou SC, Rodrigues DC, Coppola E, Balmaceda Huarte R, Barreiro M, Blázquez J, Doyle M, Feijoó M, Huth R, Machado L, Cuadra SV (2021) The CORDEX Flagship Pilot Study in south-eastern South America: a comparative study of statistical and dynamical downscaling models in simulating daily extreme precipitation events. *Climate Dynamics* 56(5-6):1589–1608, DOI 10.1007/S00382-020-05549-Z/FIGURES/7, URL <https://link.springer.com/article/10.1007/s00382-020-05549-z>
- Bhojanapalli S, Wilber K, Veit A, Rawat AS, Kim S, Menon A, Kumar S (2021) On the Reproducibility of Neural Network Predictions. arXiv preprint arXiv:210203349 pp 1–19, URL <http://arxiv.org/abs/2102.03349>
- Boé J (2021) The physiological effect of CO₂ on the hydrological cycle in summer over Europe and land-atmosphere interactions. *Climatic Change* 167(1-2):1–20, DOI 10.1007/S10584-021-03173-2/FIGURES/10, URL <https://link.springer.com/article/10.1007/s10584-021-03173-2>
- 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. *Climate Dynamics* DOI 10.1007/s00382-020-05153-1
- Brajard J, Carrassi A, Bocquet M, Bertino L (2021) Combining data assimilation and machine learning to infer unresolved scale parametrization. *Philosophical Transactions of the Royal Society A* 379(2194), DOI 10.1098/RSTA.2020.0086, URL <https://royalsocietypublishing.org/doi/10.1098/rsta.2020.0086>

- Caillaud C, Somot S, Alias A, Bernard-Bouissières I, Fumière Q, Laurantin O, Seity Y, Ducrocq V (2021) Modelling Mediterranean heavy precipitation events at climate scale: an object-oriented evaluation of the CNRM-AROME convection-permitting regional climate model. *Climate Dynamics* 2021 56:5 56(5):1717–1752, DOI 10.1007/S00382-020-05558-Y, URL <https://link.springer.com/article/10.1007/s00382-020-05558-y>
- Cannon AJ (2008) Probabilistic Multisite Precipitation Downscaling by an Expanded Bernoulli–Gamma Density Network. *Journal of Hydrometeorology* 9(6):1284–1300, DOI 10.1175/2008JHM960.1
- Careto JAM, Soares PMM, Cardoso RM, Herrera S, Gutiérrez JM (2022) Added value of EURO-CORDEX high-resolution downscaling over the Iberian Peninsula revisited - Part 2: Max and min temperature. *Geoscientific Model Development* 15(6):2653–2671, DOI 10.5194/GMD-15-2653-2022
- Castro CL (2005) Dynamical downscaling: Assessment of value retained and added using the Regional Atmospheric Modeling System (RAMS). *Journal of Geophysical Research* 110(D5):D05,108, DOI 10.1029/2004JD004721, URL <http://doi.wiley.com/10.1029/2004JD004721>
- Chadwick R, Coppola E, Giorgi F (2011) An artificial neural network technique for downscaling GCM outputs to RCM spatial scale. *Nonlinear Processes in Geophysics* 18(6):1013–1028, DOI 10.5194/NPG-18-1013-2011, URL www.nonlin-processes-geophys.net/18/1013/2011/
- Christensen OB, Kjellström E (2022) Filling the matrix: an ANOVA-based method to emulate regional climate model simulations for equally-weighted properties of ensembles of opportunity. *Climate Dynamics* 58(9-10):2371–2385, DOI 10.1007/S00382-021-06010-5/FIGURES/11, URL <https://link.springer.com/article/10.1007/s00382-021-06010-5>
- Cinquini L, Crichton D, Mattmann C, Harney J, Shipman G, Wang F, Ananthakrishnan R, Miller N, Denvil S, Morgan M, Pobre Z, Bell GM, Doutriaux C, Drach R, Williams D, Kershaw P, Pascoe S, Gonzalez E, Fiore S, Schweitzer R (2014) The Earth System Grid Federation: An open infrastructure for access to distributed geospatial data. *Future Generation Computer Systems* 36:400–417, DOI 10.1016/J.FUTURE.2013.07.002
- Colette A, Vautard R, Vrac M (2012) Regional climate downscaling with prior statistical correction of the global climate forcing. *Geophysical Research Letters* 39(13):1–5, DOI 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, Series A: Dynamic Meteorology and Oceanography* 62(5):591–604, DOI 10.1111/j.1600-0870.2010.00467.x
- Collins WJ, Bellouin N, Doutriaux-Boucher M, Gedney N, Halloran P, Hinton T, Hughes J, Jones CD, Joshi M, Liddicoat S, Martin G, O’Connor F, Rae J, Senior

- C, Sitch S, Totterdell I, Wiltshire A, Woodward S (2011) Development and evaluation of an Earth-System model - HadGEM2. *Geoscientific Model Development* 4(4):1051–1075, DOI 10.5194/gmd-4-1051-2011
- Coppola E, Nogherotto R, Ciarlo' JM, Giorgi F, van Meijgaard E, Kadygrov N, Iles C, Corre L, Sandstad M, Somot S, Nabat P, Vautard R, Levvasseur G, Schwingshackl C, Sillmann J, Kjellström E, Nikulin G, Aalbers E, Lenderink G, Christensen OB, Boberg F, Sørland SL, Demory ME, Bülow K, Teichmann C, Warrach-Sagi K, Wulfmeyer V (2021) Assessment of the European Climate Projections as Simulated by the Large EURO-CORDEX Regional and Global Climate Model Ensemble. *Journal of Geophysical Research: Atmospheres* 126(4):e2019JD032,356, DOI 10.1029/2019JD032356, URL <https://onlinelibrary.wiley.com/doi/full/10.1029/2019JD032356>, <https://onlinelibrary.wiley.com/doi/abs/10.1029/2019JD032356>, <https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2019JD032356>
- Davies HC, Turner RE (1977) Updating prediction models by dynamical relaxation: an examination of the technique. *Quarterly Journal of the Royal Meteorological Society* 103(436):225–245, DOI 10.1002/qj.49710343602, URL <https://onlinelibrary.wiley.com/doi/10.1002/qj.49710343602>
- 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, DOI 10.1002/2014JD022236
- Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L (2009) ImageNet: A Large-Scale Hierarchical Image Database. URL <http://www.image-net.org>.
- Denis B, Laprise R, Caya D (2003) Sensitivity of a regional climate model to the resolution of the lateral boundary conditions. *Climate Dynamics* 20(2):107–126, DOI 10.1007/S00382-002-0264-6, URL <https://link.springer.com/article/10.1007/s00382-002-0264-6>
- 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. *Climatic Change* 81(SUPPL. 1):53–70, DOI 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. *Climate Dynamics* 38(5-6):951–964, DOI 10.1007/s00382-011-1053-x
- Dewitte S, Cornelis JP, Müller R, Munteanu A (2021) Artificial Intelligence Revolutionises Weather Forecast, Climate Monitoring and Decadal Prediction. *Remote Sensing* 2021, Vol 13, Page 3209 13(16):3209, DOI 10.3390/RS13163209, URL <https://www.mdpi.com/2072-4292/13/16/3209/htm>, <https://www.mdpi.com/2072-4292/13/16/3209>
- Dickinson RE, Errico RM, Giorgi F, Bates GT (1989) A regional climate model for the western United States. *Climatic Change* 15(3):383–422, DOI 10.1007/BF00240465

- Dixon KW, Lanzante JR, Jo Nath M, Hayhoe K, Stoner A, Radhakrishnan A, Balaji V, Gaitán CF, Dixon Keith Dixon KW (2016) Evaluating the stationarity assumption in statistically downscaled climate projections: is past performance an indicator of future results? *Climatic Change* 135:395–408, DOI 10.1007/s10584-016-1598-0
- Dubrovský M, Buchtele J, Žalud Z (2004) High-Frequency and Low-Frequency Variability in Stochastic Daily Weather Generator and Its Effect on Agricultural and Hydrologic Modelling. *Climatic Change* 2004 63:1 63(1):145–179, DOI 10.1023/B:CLIM.0000018504.99914.60, URL <https://link.springer.com/article/10.1023/B:CLIM.0000018504.99914.60>
- Erlandsen HB, Parding KM, Benestad R, Mezghani A, Pontoppidan M (2020) A hybrid downscaling approach for future temperature and precipitation change. *Journal of Applied Meteorology and Climatology* 59(11):1–46, DOI 10.1175/jamc-d-20-0013.1, URL <https://doi.org/10.1175/JAMC-D-20-0013.1>
- Evin G, Favre ACC, Hingray B (2019) Stochastic generators of multi-site daily temperature: comparison of performances in various applications. *Theoretical and Applied Climatology* 135(3-4):811–824, DOI 10.1007/s00704-018-2404-x, URL <http://link.springer.com/10.1007/s00704-018-2404-x>
- 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 Dynamics Discussions* (March):1–40, DOI 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. *Geoscientific Model Development* 9(5):1937–1958, DOI 10.5194/gmd-9-1937-2016, URL <https://gmd.copernicus.org/articles/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 System Dynamics* 9(1):313–338, DOI 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. *Climate Dynamics* 51(3):877–900, DOI 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. *Climate Dynamics* 52(1-2):1139–1156, DOI 10.1007/s00382-018-4181-8, URL <https://doi.org/10.1007/s00382-018-4181-8>

- François B, Vrac M, Cannon AJ, Robin Y, Allard D (2020) Multivariate bias corrections of climate simulations: Which benefits for which losses? *Earth System Dynamics* 11(2):537–562, DOI 10.5194/ESD-11-537-2020
- François B, Thao S, Vrac M (2021) Adjusting spatial dependence of climate model outputs with cycle-consistent adversarial networks. *Climate Dynamics* 57(11-12):3323–3353, DOI 10.1007/S00382-021-05869-8
- Fumière Q, Déqué M, Nuissier O, Somot S, Alias A, Caillaud C, Laurantin O, Seity Y (2020) Extreme rainfall in Mediterranean France during the fall: added value of the CNRM-AROME Convection-Permitting Regional Climate Model. *Climate Dynamics* 55(1-2):77–91, DOI 10.1007/s00382-019-04898-8
- Gardner M, Dorling S (1998) Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric Environment* 32(14):2627–2636, DOI [https://doi.org/10.1016/S1352-2310\(97\)00447-0](https://doi.org/10.1016/S1352-2310(97)00447-0), URL <https://www.sciencedirect.com/science/article/pii/S1352231097004470>
- Giffard-Roisin S, Yang M, Charpiat G, Kumler Bonfanti C, Kégl B, Monteleoni C (2020) Tropical Cyclone Track Forecasting Using Fused Deep Learning From Aligned Reanalysis Data. *Frontiers in Big Data* 3:1, DOI 10.3389/FDATA.2020.00001/BIBTEX
- Giorgetta MA, Jungclaus J, Reick CH, Legutke S, Bader J, Böttinger M, Brovkin V, Crueger T, Esch M, Fieg K, Glushak K, Gayler V, Haak H, Hollweg HD, Ilyina T, Kinne S, Kornbluh L, Matei D, Mauritsen T, Mikolajewicz U, Mueller W, Notz D, Pithan F, Raddatz T, Rast S, Redler R, Roeckner E, Schmidt H, Schnur R, Segschneider J, Six KD, Stockhause M, Timmreck C, Wegner J, Widmann H, Wieners KH, Claussen M, Marotzke J, Stevens B (2013) Climate and carbon cycle changes from 1850 to 2100 in MPI-ESM simulations for the Coupled Model Intercomparison Project phase 5. *Journal of Advances in Modeling Earth Systems* 5(3):572–597, DOI 10.1002/jame.20038, URL <http://doi.wiley.com/10.1002/jame.20038>
- Giorgi F (2019) Thirty Years of Regional Climate Modeling: Where Are We and Where Are We Going next? *Journal of Geophysical Research: Atmospheres* 124(11):5696–5723, DOI 10.1029/2018JD030094
- Giorgi F, Bates GT (1989) The climatological skill of a regional model over complex terrain. *Monthly Weather Review* 117(11):2325–2347, DOI 10.1175/1520-0493(1989)117<2325:TCSOAR>2.0.CO;2
- Giorgi F, Gutowski WJ (2015) Regional Dynamical Downscaling and the CORDEX Initiative. *Annual Review of Environment and Resources* 40(1):467–490, DOI 10.1146/annurev-environ-102014-021217, URL <https://www.annualreviews.org/doi/10.1146/annurev-environ-102014-021217>
- 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. *Nature Geoscience* 9(8):584–589, DOI 10.1038/ngeo2761

- Glorot X, Bordes A, Bengio Y (2011) Deep Sparse Rectifier Neural Networks. *JMLR Workshop and Conference Proceedings*, pp 315–323, URL <https://proceedings.mlr.press/v15/glorot11a.html>
- Goodfellow I, Bengio Y, Courville A (2016) *Deep Learning*
- Gudmundsson L, Bremnes JB, Haugen JE, Engen-Skaugen T (2012) Hydrology and Earth System Sciences Technical Note: Downscaling RCM precipitation to the station scale using statistical transformations—a comparison of methods. *Hydrol Earth Syst Sci* 16:3383–3390, DOI 10.5194/hess-16-3383-2012, URL www.hydro-earth-syst-sci.net/16/3383/2012/
- Guidotti R, Monreale A, Ruggieri S, Turini F, Giannotti F, Pedreschi D (2018) A survey of methods for explaining black box models. *ACM Computing Surveys* 51(5), DOI 10.1145/3236009
- Gutiérrez C, Somot S, Nabat P, Mallet M, Corre L, Van Meijgaard E, Perpiñán O, Gaertner M (2020) Future evolution of surface solar radiation and photovoltaic potential in Europe: investigating the role of aerosols. *Environmental Research Letters* 15(3):034,035, DOI 10.1088/1748-9326/AB6666, URL <https://iopscience.iop.org/article/10.1088/1748-9326/AB6666>, <https://iopscience.iop.org/article/10.1088/1748-9326/AB6666/meta>
- Gutiérrez JM, San-Martín D, Brands S, Manzanas R, Herrera S (2013) Reassessing Statistical Downscaling Techniques for Their Robust Application under Climate Change Conditions. *Journal of Climate* 26(1):171–188, DOI 10.1175/JCLID-11-00687.1, URL <https://journals.ametsoc.org/view/journals/clim/26/1/jclid-11-00687.1.xml>
- 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, Rätty 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. *International Journal of Climatology* 39(9):3750–3785, DOI 10.1002/joc.5462
- Gutmann E, Pruitt T, Clark MP, Brekke L, Arnold JR, Raff DA, Rasmussen RM (2014) An intercomparison of statistical downscaling methods used for water resource assessments in the United States. *Water Resources Research* 50(9):7167–7186, DOI 10.1002/2014WR015559, URL <https://onlinelibrary.wiley.com/doi/full/10.1002/2014WR015559>, <https://onlinelibrary.wiley.com/doi/abs/10.1002/2014WR015559>, <https://agupubs.onlinelibrary.wiley.com/doi/10.1002/2014WR015559>
- Hauser M, Orth R, Seneviratne SI (2017) Investigating soil moisture-climate interactions with prescribed soil moisture experiments: An assessment with the Community Earth System Model (version 1.2). *Geoscientific Model Development* 10(4):1665–1677, DOI 10.5194/GMD-10-1665-2017

- Hawkins E, Sutton R (2009) The potential to narrow uncertainty in regional climate predictions. *Bulletin of the American Meteorological Society* 90(8):1095–1107, DOI 10.1175/2009BAMS2607.1
- Hertig E, Seubert S, Paxian A, Vogt G, Paeth H, Jacobeit J (2013) Changes of total versus extreme precipitation and dry periods until the end of the twenty-first century: Statistical assessments for the Mediterranean area. *Theoretical and Applied Climatology* 111(1-2):1–20, DOI 10.1007/S00704-012-0639-5/TABLES/4, URL <https://link.springer.com/article/10.1007/s00704-012-0639-5>
- Hertig E, Maraun D, Bartholy J, Pongracz R, Vrac M, Mares I, Gutiérrez JM, Wibig J, Casanueva A, Soares PM (2019) Comparison of statistical downscaling methods with respect to extreme events over Europe: Validation results from the perfect predictor experiment of the COST Action VALUE. *International Journal of Climatology* 39(9):3846–3867, DOI 10.1002/JOC.5469, URL <https://onlinelibrary.wiley.com/doi/full/10.1002/joc.5469>
<https://onlinelibrary.wiley.com/doi/abs/10.1002/joc.5469>
<https://rmets.onlinelibrary.wiley.com/doi/10.1002/joc.5469>
- Hoeser T, Kuenzer C (2020) Object detection and image segmentation with deep learning on Earth observation data: A review-part I: Evolution and recent trends. *Remote Sensing* 12(10), DOI 10.3390/RS12101667
- Höhlein K, Kern M, Hewson T, Westermann R (2020) A comparative study of convolutional neural network models for wind field downscaling. *Meteorological Applications* 27(6):e1961, DOI 10.1002/MET.1961, URL <https://onlinelibrary.wiley.com/doi/full/10.1002/met.1961>
<https://onlinelibrary.wiley.com/doi/abs/10.1002/met.1961>
<https://rmets.onlinelibrary.wiley.com/doi/10.1002/met.1961>
- 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*:505–507, DOI 10.1109/CVPRW.2019.00073
- Huth R (2002) Statistical Downscaling of Daily Temperature in Central Europe. *Journal of Climate* 15(13):1731–1742, DOI 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. *Theoretical and Applied Climatology* 120(3-4):533–553, DOI 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):448–456
- Ivušić S, Güttler I, Somot S, Guérémy JF, Horvath K, Alias A (2021) Modelling extreme precipitation over the Dinaric Alps: An evaluation of the CNRM-ALADIN regional climate model. *Quarterly Journal of the Royal Meteorological Society* 147(741):4425–4453, DOI 10.1002/qj.4187

- 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. *Regional Environmental Change* 14(2):563–578, DOI 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. *Canadian Water Resources Journal* 40(4):392–408, DOI 10.1080/07011784.2015.1089191, URL <http://dx.doi.org/10.1080/07011784.2015.1089191>
- Katz RW (1977) Precipitation as a Chain-Dependent Process. *JApMe* 16(7):671–676, DOI 10.1175/1520-0450(1977)016, URL <https://ui.adsabs.harvard.edu/abs/1977JApMe..16..671K/abstract>
- Keller DE, Fischer AM, Liniger MA, Appenzeller C, Knutti R (2017) Testing a weather generator for downscaling climate change projections over Switzerland. *International Journal of Climatology* 37(2):928–942, DOI 10.1002/JOC.4750, URL <https://onlinelibrary.wiley.com/doi/full/10.1002/joc.4750>
<https://onlinelibrary.wiley.com/doi/abs/10.1002/joc.4750>
<https://rmets.onlinelibrary.wiley.com/doi/10.1002/joc.4750>
- Kilsby CG, Jones PD, Burton A, Ford AC, Fowler HJ, Harpham C, James P, Smith A, Wilby RL (2007) A daily weather generator for use in climate change studies. *Environmental Modelling & Software* 22(12):1705–1719, DOI 10.1016/J.ENVSOFT.2007.02.005
- Kim S, Kim H, Lee J, Yoon S, Kahou SE, Kashinath K, Prabhat (2019) Deep-hurricane-tracker: Tracking and forecasting extreme climate events. *Proceedings - 2019 IEEE Winter Conference on Applications of Computer Vision, WACV 2019* pp 1761–1769, DOI 10.1109/WACV.2019.00192
- Kingma DP, Lei Ba J (2015) ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION. *arXiv*
- Klaver R, Haarsma R, Vidale PL, Hazeleger W (2020) Effective resolution in high resolution global atmospheric models for climate studies. *Atmospheric Science Letters* 21(4):1–8, DOI 10.1002/asl.952, URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/asl.952>
- Klingaman NP, Martin GM, Moise A (2017) ASoP (v1.0): A set of methods for analyzing scales of precipitation in general circulation models. *Geoscientific Model Development* 10(1):57–83, DOI 10.5194/gmd-10-57-2017
- Knutti R (2010) The end of model democracy? An editorial comment. *Climatic Change* 102:395–404, DOI 10.1007/s10584-010-9800-2

- Knutti R, Sedláček J (2013) Robustness and uncertainties in the new CMIP5 climate model projections. *Nature Climate Change* 3(4):369–373, DOI 10.1038/nclimate1716
- Kotlarski S, Lüthi D, Schär C (2015) The elevation dependency of 21st century European climate change: An RCM ensemble perspective. *International Journal of Climatology* 35(13):3902–3920, DOI 10.1002/joc.4254
- Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems* 25, URL <http://code.google.com/p/cuda-convnet/>
- Lanzante JR, Dixon KW, Nath MJ, Whitlock CE, Adams-Smith D (2018) Some Pitfalls in Statistical Downscaling of Future Climate. *Bulletin of the American Meteorological Society* 99(4):791–803, DOI 10.1175/BAMS-D-17-0046.1, URL <https://journals.ametsoc.org/view/journals/bams/99/4/bams-d-17-0046.1.xml>
- 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, DOI 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. *Natural Hazards and Earth System Science* 12(3):651–670, DOI 10.5194/nhess-12-651-2012
- LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11):2278–2323, DOI 10.1109/5.726791
- Lecun Y, Bengio Y, Hinton G (2015) Deep learning. *Nature* 521(7553):436–444, DOI 10.1038/nature14539
- Leduc M, Mailhot A, Frigon A, Martel JL, Ludwig R, Brietzke GB, Giguère M, Brissette F, Turcotte R, Braun M, Scinocca J (2019) The ClimEx project: A 50-member ensemble of climate change projections at 12-km resolution over Europe and northeastern North America with the Canadian Regional Climate Model (CRCM5). *Journal of Applied Meteorology and Climatology* 58(4):663–693, DOI 10.1175/JAMC-D-18-0021.1
- Lehner F, Deser C, Maher N, Marotzke J, Fischer EM, Brunner L, Knutti R, Hawkins E (2020) Partitioning climate projection uncertainty with multiple large ensembles and CMIP5/6. *Earth System Dynamics* 11(2):491–508, DOI 10.5194/esd-11-491-2020
- 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, p 11389, DOI 10.5194/egusphere-egu2020-11389

- Long J, Shelhamer E, Darrell T (2015) Fully Convolutional Networks for Semantic Segmentation
- Lorenz EN (1963) Deterministic Nonperiodic Flow. *Journal of the Atmospheric Sciences* 20(2):130–141, DOI 10.1175/1520-0469(1963)020<0130:dnfj.2.0.co;2
- Lorenz EN (1969) Atmospheric Predictability as Revealed by Naturally Occurring Analogues. *Journal of Atmospheric Sciences* 26(4):636–646
- Lorenz P, Jacob D (2005) Influence of regional scale information on the global circulation: A two-way nesting climate simulation. *Geophysical Research Letters* 32(18), DOI <https://doi.org/10.1029/2005GL023351>, URL <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2005GL023351>, <https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2005GL023351>
- Lu C, Huang D, Shen Y, Xin F (2022) Data-driven Global Subseasonal Forecast Model (GSFM 1 v1.0) for intraseasonal oscillation components 2. *GMD [PREPRINT]* DOI 10.5194/gmd-2022-146, URL <https://doi.org/10.5194/gmd-2022-146>
- Lucas-Picher P, Caya D, Elía R, Laprise R (2008) Investigation of regional climate models' internal variability with a ten-member ensemble of 10-year simulations over a large domain. *Climate Dynamics* 31(7-8):927–940, DOI 10.1007/S00382-008-0384-8/FIGURES/8, URL <https://link.springer.com/article/10.1007/s00382-008-0384-8>
- Lucas-Picher P, Cattiaux J, Bougie A, Laprise R (2016) How does large-scale nudging in a regional climate model contribute to improving the simulation of weather regimes and seasonal extremes over North America? *Climate Dynamics* 46(3-4):929–948, DOI 10.1007/s00382-015-2623-0
- Lucas-Picher P, Argüeso D, Brisson E, Trambly Y, Berg P, Lemonsu A, Kotlarski S, Caillaud C (2021) Convection-permitting modeling with regional climate models: Latest developments and next steps. DOI 10.1002/wcc.731
- Maher N, Milinski S, Ludwig R (2021) Large ensemble climate model simulations: Introduction, overview, and future prospects for utilising multiple types of large ensemble. *Earth System Dynamics* 12(2):401–418, DOI 10.5194/esd-12-401-2021
- 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. *Climate Services* 9:44–56, DOI 10.1016/j.cliser.2017.06.004, URL <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, DOI 10.1017/9781107588783, URL <https://www.cambridge.org/core/product/identifier/9781107588783/type/book>
- Maraun D, Wetterhall F, Ireson AM, Chandler RE, Kendon EJ, Widmann M, Brienen S, Rust HW, Sauter T, Themel M, Venema VK, Chun KP, Goodess CM, Jones RG, Onof C, Vrac M, Thiele-Eich I (2010) Precipitation downscaling under

- climate change: Recent developments to bridge the gap between dynamical models and the end user. *Reviews of Geophysics* 48(3), DOI 10.1029/2009RG000314
- Maraun D, Widmann M, Gutiérrez JM, Kotlarski S, Chandler RE, Hertig E, Wibig J, Huth R, Wilcke RA (2015) VALUE: A framework to validate downscaling approaches for climate change studies. *Earth's Future* 3(1):1–14, DOI 10.1002/2014EF000259, URL <https://onlinelibrary.wiley.com/doi/full/10.1002/2014EF000259>, <https://onlinelibrary.wiley.com/doi/abs/10.1002/2014EF000259>, <https://agupubs.onlinelibrary.wiley.com/doi/10.1002/2014EF000259>
- Maraun D, Huth R, Gutiérrez JM, Martín DS, Dubrovsky M, Fischer A, Hertig E, Soares PM, Bartholy J, Pongrácz R, Widmann M, Casado MJ, Ramos P, Bedia J (2019a) The VALUE perfect predictor experiment: Evaluation of temporal variability. *International Journal of Climatology* 39(9):3786–3818, DOI 10.1002/JOC.5222, URL <https://onlinelibrary.wiley.com/doi/full/10.1002/joc.5222>, <https://onlinelibrary.wiley.com/doi/abs/10.1002/joc.5222>, <https://rmets.onlinelibrary.wiley.com/doi/10.1002/joc.5222>
- Maraun D, Widmann M, Gutiérrez JM (2019b) Statistical downscaling skill under present climate conditions: A synthesis of the VALUE perfect predictor experiment. *International Journal of Climatology* 39(9):3692–3703, DOI 10.1002/JOC.5877
- 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-Sept*:0–4, DOI 10.23919/EUSIPCO.2019.8902690
- Martin ZK, Barnes EA, Maloney E (2022) Using Simple, Explainable Neural Networks to Predict the Madden-Julian Oscillation. *Journal of Advances in Modeling Earth Systems* 14(5):e2021MS002,774, DOI 10.1029/2021MS002774, URL <https://onlinelibrary.wiley.com/doi/full/10.1029/2021MS002774>, <https://onlinelibrary.wiley.com/doi/abs/10.1029/2021MS002774>, <https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2021MS002774>
- Masson-Delmotte V, Zhai P, Chen Y, Goldfarb L, Gomis MI, Matthews JBR, Berger S, Huang M, Yelekçi O, Yu R, Zhou B, Lonnoy E, Maycock TK, Waterfield T, Leitzell K, Caud N (2021) Working Group I Contribution to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change Edited by. DOI 10.1017/9781009157896, URL www.ipcc.ch
- Maurer EP, Hidalgo HG, Das T, Dettinger MD, Cayan DR (2010) The utility of daily large-scale climate data in the assessment of climate change impacts on daily streamflow in California. *Hydrology and Earth System Sciences* 14(6):1125–1138, DOI 10.5194/HESS-14-1125-2010
- McCulloch WS, Pitts W (1943) A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics* 1943 5:4 5(4):115–133, DOI 10.1007/BF02478259, URL <https://link.springer.com/article/10.1007/BF02478259>

- Medeiros B, Stevens B (2011) Revealing differences in GCM representations of low clouds. *Climate Dynamics* 36(1):385–399, DOI 10.1007/S00382-009-0694-5/FIGURES/9, URL <https://link-springer-com.insu.bib.cnrs.fr/article/10.1007/s00382-009-0694-5>
- 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. *Bulletin of the American Meteorological Society* 88(9):1383–1394, DOI 10.1175/BAMS-88-9-1383
- Meinshausen M, Smith SJ, Calvin K, Daniel JS, Kainuma ML, Lamarque J, Matsumoto K, Montzka SA, Raper SC, Riahi K, Thomson A, Velders GJ, van Vuuren DP (2011) The RCP greenhouse gas concentrations and their extensions from 1765 to 2300. *Climatic Change* 109(1):213–241, DOI 10.1007/s10584-011-0156-z
- Michelangeli PA, Vrac M, Loukos H (2009) Probabilistic downscaling approaches: Application to wind cumulative distribution functions. *Geophysical Research Letters* 36(11):2–7, DOI 10.1029/2009GL038401
- Miguez-Macho G, Stenchikov GL, Robock A (2004) Spectral nudging to eliminate the effects of domain position and geometry in regional climate model simulations. *Journal of Geophysical Research: Atmospheres* 109(D13):n/a–n/a, DOI 10.1029/2003JD004495, URL <http://doi.wiley.com/10.1029/2003JD004495>
- 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, DOI 10.1038/nature08823
- Mounier A, Raynaud L, Rottner L, Plu M, Arbogast P, Kreitz M, Mignan L, Touzé B (2022) Detection of Bow Echoes in Kilometer-Scale Forecasts Using a Convolutional Neural Network. *Artificial Intelligence for the Earth Systems* 1(2), DOI 10.1175/AIES-D-21-0010.1, URL <https://journals.ametsoc.org/view/journals/aies/1/2/AIES-D-21-0010.1.xml>
- Nabat P, Somot S, Mallet M, Sanchez-Lorenzo A, Wild M (2014) Contribution of anthropogenic sulfate aerosols to the changing Euro-Mediterranean climate since 1980. *Geophysical Research Letters* 41(15):5605–5611, DOI 10.1002/2014GL060798, URL <http://doi.wiley.com/10.1002/2014GL060798>
- 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. *Atmospheric Chemistry and Physics* 20(14):8315–8349, DOI 10.5194/acp-20-8315-2020, URL <https://doi.org/10.5194/acp-20-8315-2020>
- Novikoff AB (1962) On Convergence Proofs on Perceptrons. In: *Proceedings of the Symposium on the Mathematical Theory of Automata*, Polytechnic Institute of Brooklyn, New York, NY, USA, vol 12, pp 615–622

- Overpeck JT, Meehl GA, Bony S, Easterling DR (2011) Climate data challenges in the 21st century. *Science* 331(6018):700–702, DOI 10.1126/SCIENCE.1197869
- Prein AF, Langhans W, Fosser G, Ferrone A, Ban N, Goergen K, Keller M, Tölle M, Gutjahr O, Feser F, Brisson E, Kollet S, Schmidli J, Van Lipzig NP, Leung R (2015) A review on regional convection-permitting climate modeling: Demonstrations, prospects, and challenges. DOI 10.1002/2014RG000475
- Prein AF, Gobiet A, Truhetz H, Keuler K, Goergen K, Teichmann C, Fox Maule 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? *Climate Dynamics* 46(1-2):383–412, DOI 10.1007/s00382-015-2589-y
- Pryor SC, Barthelmie RJ (2014) Hybrid downscaling of wind climates over the eastern USA. *Environmental Research Letters* 9(2), DOI 10.1088/1748-9326/9/2/024013
- Pulkkinen S, Nerini D, Pérez Hortal AA, Velasco-Forero C, Seed A, Germann U, Foresti L (2019) Pysteps: An open-source Python library for probabilistic precipitation nowcasting (v1.0). *Geoscientific Model Development* 12(10):4185–4219, DOI 10.5194/gmd-12-4185-2019
- Racah E, Beckham C, Maharaj T, Kahou SE, Pal C (2017) ExtremeWeather: A large-scale climate dataset for semi-supervised detection, localization, and understanding of extreme weather events. In: *Neural Information Processing System*, URL <https://github.com/eracah/hur-detect>.
- Raina R, Madhavan A, Ng AY (2009) Large-scale Deep Unsupervised Learning using Graphics Processors
- Rasp S, Thuerey N (2021) Data-Driven Medium-Range Weather Prediction With a Resnet Pretrained on Climate Simulations: A New Model for WeatherBench. *Journal of Advances in Modeling Earth Systems* 13(2):e2020MS002,405, DOI 10.1029/2020MS002405, URL <https://onlinelibrary.wiley.com/doi/full/10.1029/2020MS002405>, <https://onlinelibrary.wiley.com/doi/abs/10.1029/2020MS002405>, <https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2020MS002405>
- Rasp S, Dueben PD, Scher S, Weyn JA, Mouatadid S, Thuerey N (2020) WeatherBench: A Benchmark Data Set for Data-Driven Weather Forecasting. *Journal of Advances in Modeling Earth Systems* 12(11):e2020MS002,203, DOI 10.1029/2020MS002203, URL <https://onlinelibrary.wiley.com/doi/full/10.1029/2020MS002203>, <https://onlinelibrary.wiley.com/doi/abs/10.1029/2020MS002203>, <https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2020MS002203>
- Ravuri S, Lenc K, Willson M, Kangin D, Lam R, Mirowski P, Fitzsimons M, Athanassiadou M, Kashem S, Madge S, Prudden R, Mandhane A, Clark A, Brock A, Simonyan K, Hadsell R, Robinson N, Clancy E, Arribas A, Mohamed S (2021a) Skilful precipitation nowcasting using deep generative models of radar. *Nature* 597(7878):672–677, DOI 10.1038/s41586-021-03854-z

- Ravuri S, Lenc K, Willson M, Kangin D, Lam R, Mirowski P, Fitzsimons M, Athanassiadou M, Kashem S, Madge S, Prudden R, Mandhane A, Clark A, Brock A, Simonyan K, Hadsell R, Robinson N, Clancy E, Arribas A, Mohamed S (2021b) Skilful precipitation nowcasting using deep generative models of radar. *Nature* 2021 597:7878–7883, DOI 10.1038/s41586-021-03854-z, URL <https://www.nature.com/articles/s41586-021-03854-z>
- 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, DOI 10.1038/s41586-019-0912-1, URL <https://doi.org/10.1038/s41586-019-0912-1> <http://dx.doi.org/10.1038/s41586-019-0912-1>
- Ribes A, Zwiers FW, Azaïs JM, Naveau P (2017) A new statistical approach to climate change detection and attribution. *Climate Dynamics* 48(1-2):367–386, DOI 10.1007/S00382-016-3079-6/FIGURES/5, URL <https://link.springer.com/article/10.1007/s00382-016-3079-6>
- Ribes A, Thao S, Vautard R, Dubuisson B, Somot S, Colin J, Planton S, Soubeyrou JM (2019) Observed increase in extreme daily rainfall in the French Mediterranean. *Climate Dynamics* 52(1-2):1095–1114, DOI 10.1007/S00382-018-4179-2/FIGURES/11, URL <https://link.springer.com/article/10.1007/s00382-018-4179-2>
- Ribes A, Boé J, Qasmi S, Dubuisson B, Douville H, Terray L (2022) An updated assessment of past and future warming over France based on a regional observational constraint. *Earth System Dynamics Discussions* 2022:1–29, DOI 10.5194/esd-2022-7, URL <https://esd.copernicus.org/preprints/esd-2022-7/>
- Richardson CW (1981) Stochastic simulation of daily precipitation, temperature, and solar radiation. *Water Resources Research* 17(1):182–190, DOI 10.1029/WR017i001p00182, URL <https://onlinelibrary.wiley.com/doi/full/10.1029/WR017i001p00182>
<https://onlinelibrary.wiley.com/doi/abs/10.1029/WR017i001p00182>
<https://agupubs.onlinelibrary.wiley.com/doi/10.1029/WR017i001p00182>
- Robin Y, Vrac M, Naveau P, Yiou P (2019) Multivariate stochastic bias corrections with optimal transport. *Hydrology and Earth System Sciences* 23(2):773–786, DOI 10.5194/HESS-23-773-2019
- Ronneberger O, Fischer P, Brox T (2015) U-net: Convolutional networks for biomedical image segmentation. In: *Lecture Notes in Computer Science (including sub-series Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol 9351, pp 234–241, DOI 10.1007/978-3-319-24574-4_28
- Rosenblatt F (1958) The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review* 65(6):386–408, DOI 10.1037/H0042519
- Rumelhart DE, Hinton GE (1986) Learning Representations by Back-Propagating Errors. *Cognitive Modeling* (2):3–6, DOI 10.7551/mitpress/1888.003.0013

- Rummukainen M (2010) State-of-the-art with regional climate models. John Wiley & Sons, Ltd 1, DOI 10.1002/wcc.008
- Rummukainen M, Rockel B, Bärring L, Christensen JH, Reckermann M (2015) Twenty-First-Century Challenges in Regional Climate Modeling. *Bulletin of the American Meteorological Society* 96(8):ES135–ES138, DOI 10.1175/BAMS-D-14-00214.1, URL <https://journals.ametsoc.org/view/journals/bams/96/8/bams-d-14-00214.1.xml>
- San-Martín D, Manzanas R, Brands S, Herrera S, Gutiérrez J (2017) Reassessing model uncertainty for regional projections of precipitation with an ensemble of statistical downscaling methods. *Journal of Climate* 30(1):203–223, DOI 10.1175/JCLI-D-16-0366.1, URL <http://www.aemet>.
- 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. *Climate Dynamics* 51(3):1005–1021, DOI 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. *Climate Dynamics* 33(5):723–736, DOI 10.1007/s00382-008-0502-7
- Scher S, Messori G (2021) Ensemble Methods for Neural Network-Based Weather Forecasts. *Journal of Advances in Modeling Earth Systems* 13(2), DOI 10.1029/2020MS002331, URL <https://onlinelibrary.wiley.com/doi/full/10.1029/2020MS002331>
<https://onlinelibrary.wiley.com/doi/abs/10.1029/2020MS002331>
<https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2020MS002331>
- Schmith T (2008) Stationarity of regression relationships: Application to empirical downscaling. *Journal of Climate* 21(17):4529–4537, DOI 10.1175/2008JCLI1910.1
- Schultz MG, Betancourt C, Gong B, Kleinert F, Langguth M, Leufen LH, Mozafari A, Stadtler S (2021) Can deep learning beat numerical weather prediction? *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 379(2194), DOI 10.1098/RSTA.2020.0097, URL <https://doi.org/10.1098/rsta.2020.0097>
- Schwingshackl C, Davin EL, Hirschi M, Sørland SL, Wartenburger R, Seneviratne SI (2019) Regional climate model projections underestimate future warming due to missing plant physiological CO₂ response. *Environmental Research Letters* 14(11), DOI 10.1088/1748-9326/ab4949
- Sein DV, Mikolajewicz U, Gröger M, Fast I, Cabos W, Pinto JG, Hagemann S, Semmler T, Izquierdo A, Jacob D (2015) Regionally coupled atmosphere-ocean-sea ice-marine biogeochemistry model ROM: 1. Description and validation. *Journal of Advances in Modeling Earth Systems* 7(1):268–304, DOI 10.1002/2014MS000357, URL <http://doi.wiley.com/10.1002/2014MS000357>
- Seneviratne SI, Corti T, Davin EL, Hirschi M, Jaeger EB, Lehner I, Orlowsky B, Teuling AJ (2010) Investigating soil moisture–climate interactions in a

- changing climate: A review. *Earth-Science Reviews* 99(3-4):125–161, DOI 10.1016/J.EARSCIREV.2010.02.004
- Separovic L, de Elía R, Laprise R (2012) Impact of spectral nudging and domain size in studies of RCM response to parameter modification. *Climate Dynamics* 38(7-8):1325–1343, DOI 10.1007/s00382-011-1072-7
- Serifi A, Günther T, Ban N (2021) Spatio-Temporal Downscaling of Climate Data Using Convolutional and Error-Predicting Neural Networks. *Frontiers in Climate* 3:26, DOI 10.3389/FCLIM.2021.656479/BIBTEX
- Sevault F, Somot S, Alias A, Dubois C, Lebeaupin-Brossier C, Nabat P, Adloff F, Déqué M, Decharme B (2014) A fully coupled Mediterranean regional climate system model: design and evaluation of the ocean component for the 1980–2012 period. *Tellus A: Dynamic Meteorology and Oceanography* 66(1):23,967, DOI 10.3402/tellusa.v66.23967, URL <https://a.tellusjournals.se/article/10.3402/tellusa.v66.23967/>
- Sha Y, Gagne DJ, West G, Stull R (2020a) Deep-Learning-Based Gridded Downscaling of Surface Meteorological Variables in Complex Terrain. Part I: Daily Maximum and Minimum 2-m Temperature. *Journal of Applied Meteorology and Climatology* 59(12):2057–2073, DOI 10.1175/JAMC-D-20-0057.1, URL <https://journals.ametsoc.org/view/journals/apme/59/12/jamc-d-20-0057.1.xml>
- Sha Y, Gagne DJ, West G, Stull R (2020b) Deep-Learning-Based Gridded Downscaling of Surface Meteorological Variables in Complex Terrain. Part II: Daily Precipitation. *Journal of Applied Meteorology and Climatology* 59(12):2075–2092, DOI 10.1175/JAMC-D-20-0058.1, URL <https://journals.ametsoc.org/view/journals/apme/59/12/jamc-d-20-0058.1.xml>
- Simonyan K, Zisserman A (2015) VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION. URL <http://www.robots.ox.ac.uk/>
- Sitz LE, Di Sante F, Farneti R, Fuentes-Franco R, Coppola E, Mariotti L, Reale M, Sannino G, Barreiro M, Nogherotto R, Giuliani G, Graffino G, Solidoro C, Cossarini G, Giorgi F (2017) Description and evaluation of the Earth System Regional Climate Model (Reg CM-ES). *Journal of Advances in Modeling Earth Systems* 9(4):1863–1886, DOI 10.1002/2017MS000933, URL <http://doi.wiley.com/10.1002/2017MS000933>
- Soares PM, Maraun D, Brands S, Jury MW, Gutiérrez JM, San-Martín D, Hertig E, Huth R, Belušić Vozila A, Cardoso RM, Kotlarski S, Drobinski P, Obermann-Hellhund A (2019) Process-based evaluation of the VALUE perfect predictor experiment of statistical downscaling methods. *International Journal of Climatology* 39(9):3868–3893, DOI 10.1002/JOC.5911, URL <https://onlinelibrary.wiley.com/doi/full/10.1002/joc.5911>
<https://onlinelibrary.wiley.com/doi/abs/10.1002/joc.5911>
<https://rmets.onlinelibrary.wiley.com/doi/10.1002/joc.5911>
- Solomon S, Qin D, Manning M, Marquis M, Averyt K, Tignor M, LeRoy Miller H, Zhenlin C (2007) *Climate Change 2007*

- Somot S, Sevault F, Déqué M, Crépon M (2008) 21st century climate change scenario for the Mediterranean using a coupled atmosphere-ocean regional climate model. *Global and Planetary Change* 63(2-3):112–126, DOI 10.1016/j.gloplacha.2007.10.003
- Sønderby CK, Espenholt L, Heek J, Dehghani M, Oliver A, Salimans T, Agrawal S, Hickey J, Kalchbrenner N (2020) MetNet: A Neural Weather Model for Precipitation Forecasting. arXiv DOI 10.48550/arxiv.2003.12140, URL <https://arxiv.org/abs/2003.12140>
- Sørland SL, Schär C, Lüthi D, Kjellström E (2018) Bias patterns and climate change signals in GCM-RCM model chains. *Environmental Research Letters* 13(7), DOI 10.1088/1748-9326/aacc77
- Štěpánek P, Zahradníček P, Farda A, Skalák P, Trnka M, Meitner J, Rajdl K (2016) Projection of drought-inducing climate conditions in the Czech Republic according to Euro-CORDEX models. *Climate Research* 70(2-3):179–193, DOI 10.3354/CR01424, URL <https://www.int-res.com/abstracts/cr/v70/n2-3/p179-193/>
- Taranu IS, Delire C, Somot S, Alias A, Julien Boé (2022) Mechanisms behind large-scale inconsistencies between regional and global climate model-based projections over Europe. *Clim Dyn* [preprint] DOI 10.21203/RS.3.RS-1915811/V1, URL <https://www.researchsquare.com> <https://www.researchsquare.com/article/rs-1915811/v1>
- Taylor KE, Stouffer RJ, Meehl GA (2012) An overview of CMIP5 and the experiment design. *Bulletin of the American Meteorological Society* 93(4):485–498, DOI 10.1175/BAMS-D-11-00094.1
- Thiemeßl MJ, Gobiet A, Heinrich G (2012) Empirical-statistical downscaling and error correction of regional climate models and its impact on the climate change signal. *Climatic Change* 112(2):449–468, DOI 10.1007/S10584-011-0224-4/FIGURES/10, URL <https://link.springer.com/article/10.1007/s10584-011-0224-4>
- Tignor M (2013) CLIMATE CHANGE 2013 The Physical Science Basis WORKING GROUP I CONTRIBUTION TO THE FIFTH ASSESSMENT REPORT OF THE INTERGOVERNMENTAL PANEL ON CLIMATE CHANGE WG I IN T ERGOVERNMENTAL PANEL ON climate change. Tech. rep.
- Timbal B, Fernandez E, Li Z (2009) Generalization of a statistical downscaling model to provide local climate change projections for Australia. *Environmental Modelling & Software* 24(3):341–358, DOI 10.1016/J.ENVSOFT.2008.07.007
- Torma C, Giorgi F, Coppola E (2015) Added value of regional climate modeling over areas characterized by complex terrain-precipitation over the Alps. *Journal of Geophysical Research* 120(9):3957–3972, DOI 10.1002/2014JD022781, URL <http://doi.wiley.com/10.1002/2014JD022781>

- von Trentini F, Leduc M, Ludwig R (2019) Assessing natural variability in RCM signals: comparison of a multi model EURO-CORDEX ensemble with a 50-member single model large ensemble. *Climate Dynamics* 53(3-4):1963–1979, DOI 10.1007/s00382-019-04755-8
- Turco M, Quintana-Seguí P, Llasat MC, Herrera S, Gutiérrez JM (2011) Testing MOS precipitation downscaling for ENSEMBLES regional climate models over Spain. *Journal of Geophysical Research Atmospheres* 116(18):1–14, DOI 10.1029/2011JD016166
- Vaithinada Ayar P, Vrac M, Bastin S, Carreau J, Déqué M, Gallardo C (2015) Intercomparison of statistical and dynamical downscaling models under the EURO- and MED-CORDEX initiative framework: present climate evaluations. *Climate Dynamics* 2015 46:3 46(3):1301–1329, DOI 10.1007/S00382-015-2647-5, URL <https://link.springer.com/article/10.1007/s00382-015-2647-5>
- 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 F1296:1663–1672*, DOI 10.1145/3097983.3098004
- Vandal T, Kodra E, Ganguly AR (2019) Intercomparison of machine learning methods for statistical downscaling: the case of daily and extreme precipitation. *Theoretical and Applied Climatology* 137(1-2):557–570, DOI 10.1007/s00704-018-2613-3, URL <https://arxiv.org/abs/1702.04018>
- Vautard R, Kadygrov N, Iles C, Boberg F, Buonomo E, Bülow K, Coppola E, Corre L, Meijgaard E, Nogherotto R, Sandstad M, Schwingshackl C, Somot S, Aalbers E, Christensen OB, Ciarlò JM, Demory M, Giorgi F, Jacob D, Jones RG, Keuler K, Kjellström E, Lenderink G, Levavasseur G, Nikulin G, Sillmann J, Solidoro C, Sørland SL, Steger C, Teichmann C, Warrach-Sagi K, Wulfmeyer V (2020) Evaluation of the large EURO-CORDEX regional climate model ensemble. *Journal of Geophysical Research: Atmospheres* pp 0–3, DOI 10.1029/2019jd032344
- Vergara-Temprado J, Ban N, Panosetti D, Schlemmer L, Schär C (2020) Climate models permit convection at much coarser resolutions than previously considered. *Journal of Climate* 33(5):1915–1933, DOI 10.1175/JCLI-D-19-0286.1
- Vigaud N, Vrac M, Caballero Y (2013) Probabilistic downscaling of GCM scenarios over southern India. *International Journal of Climatology* 33(5):1248–1263, DOI 10.1002/joc.3509
- Villani C (2009) *Optimal Transport*, vol 338. DOI 10.1007/978-3-540-71050-9
- Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, Burovski E, Peterson P, Weckesser W, Bright J, van der Walt SJ, Brett M, Wilson J, Millman KJ, Mayorov N, Nelson AR, Jones E, Kern R, Larson E, Carey CJ, Polat I, Feng Y, Moore EW, VanderPlas J, Laxalde D, Perktold J, Cimrman R, Henriksen I, Quintero EA, Harris CR, Archibald AM, Ribeiro AH, Pedregosa F, van Mulbregt P, Vijaykumar A, Bardelli AP, Rothberg A, Hilboll A, Kloeckner

- A, Scopatz A, Lee A, Rokem A, Woods CN, Fulton C, Masson C, Häggström C, Fitzgerald C, Nicholson DA, Hagen DR, Pasechnik DV, Olivetti E, Martin E, Wieser E, Silva F, Lenders F, Wilhelm F, Young G, Price GA, Ingold GL, Allen GE, Lee GR, Audren H, Probst I, Dietrich JP, Silterra J, Webber JT, Slavič J, Nothman J, Buchner J, Kulick J, Schönberger JL, de Miranda Cardoso JV, Reimer J, Harrington J, Rodríguez JLC, Nunez-Iglesias J, Kuczynski J, Tritz K, Thoma M, Newville M, Kümmerer M, Bolingbroke M, Tartre M, Pak M, Smith NJ, Nowaczyk N, Shebanov N, Pavlyk O, Brodtkorb PA, Lee P, McGibbon RT, Feldbauer R, Lewis S, Tygier S, Sievert S, Vigna S, Peterson S, More S, Pudlik T, Oshima T, Pingel TJ, Robitaille TP, Spura T, Jones TR, Cera T, Leslie T, Zito T, Krauss T, Upadhyay U, Halchenko YO, Vázquez-Baeza Y (2020) SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods* 17(3):261–272, DOI 10.1038/s41592-019-0686-2
- Voldoire A, Sanchez-Gomez E, Salas y Méliá 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. *Climate Dynamics* 40(9-10):2091–2121, DOI 10.1007/s00382-011-1259-y
- Volosciuk C, Maraun D, Vrac M, Widmann M (2017) A combined statistical bias correction and stochastic downscaling method for precipitation. *Hydrology and Earth System Sciences* 21(3):1693–1719, DOI 10.5194/HESS-21-1693-2017
- Von Storch H, Langenberg H, Feser F (2000) A spectral nudging technique for dynamical downscaling purposes. *Monthly Weather Review* 128(10):3664–3673, DOI 10.1175/1520-0493(2000)128<3664:ASNTFD>2.0.CO;2
- Vrac M (2018) Multivariate bias adjustment of high-dimensional climate simulations: The Rank Resampling for Distributions and Dependences (R2D2) bias correction. *Hydrology and Earth System Sciences* 22(6):3175–3196, DOI 10.5194/HESS-22-3175-2018
- Vrac M, Naveau P (2007) Stochastic downscaling of precipitation: From dry events to heavy rainfalls. *Water Resources Research* 43(7):7402, DOI 10.1029/2006WR005308, URL <https://onlinelibrary.wiley.com/doi/full/10.1029/2006WR005308>
<https://onlinelibrary.wiley.com/doi/abs/10.1029/2006WR005308>
<https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2006WR005308>
- Vrac M, Marbaix P, Paillard D, Naveau P (2007a) Non-linear statistical downscaling of present and LGM precipitation and temperatures over Europe. *Climate of the Past* 3(4):669–682, DOI 10.5194/CP-3-669-2007
- Vrac M, Stein ML, Hayhoe K, Liang XZ (2007b) A general method for validating statistical downscaling methods under future climate change. *Geophysical Research Letters* 34(18), DOI 10.1029/2007GL030295
- Vrac M, Drobinski P, Merlo A, Herrmann M, Lavaysse C, Li L, Somot S (2012) Dynamical and statistical downscaling of the French Mediterranean climate: Un-

- certainty assessment. *Natural Hazards and Earth System Science* 12(9):2769–2784, DOI 10.5194/nhess-12-2769-2012
- Waldron KM, Paegle J, Horel JD (1996) Sensitivity of a spectrally filtered and nudged limited-area model to outer model options. *Monthly Weather Review* 124(3):529–547, DOI 10.1175/1520-0493(1996)124<0529:SOASFA>2.0.CO;2
- Walton DB, Sun F, Hall A, Capps S (2015) A hybrid dynamical-statistical downscaling technique. Part I: Development and validation of the technique. *Journal of Climate* 28(12):4597–4617, DOI 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. *Geoscientific Model Development* 14(10):6355–6372, DOI 10.5194/gmd-14-6355-2021
- Wernli H, Paulat M, Hagen M, Frei C (2008) SAL - A novel quality measure for the verification of quantitative precipitation forecasts. *Monthly Weather Review* 136(11):4470–4487, DOI 10.1175/2008MWR2415.1
- Wernli H, Hofmann C, Zimmer M (2009) Spatial forecast verification methods inter-comparison project: Application of the SAL technique. *Weather and Forecasting* 24(6):1472–1484, DOI 10.1175/2009WAF2222271.1
- Wilby RL, Hassan H, Hanaki K (1998) Statistical downscaling of hydrometeorological variables using general circulation model output. *Journal of Hydrology* 205(1-2):1–19, DOI 10.1016/S0022-1694(97)00130-3
- Williams PD, Alexander MJ, Barnes EA, Butler AH, Davies HC, Garfinkel CI, Kushnir Y, Lane TP, Lundquist JK, Martius O, Maue RN, Peltier WR, Sato K, Scaife AA, Zhang C (2017) A Census of Atmospheric Variability From Seconds to Decades. DOI 10.1002/2017GL075483
- Yao W, Zeng Z, Lian C, Tang H (2018) Pixel-wise regression using U-Net and its application on pansharpening. *Neurocomputing* 312:364–371, DOI 10.1016/j.neucom.2018.05.103, URL <https://doi.org/10.1016/j.neucom.2018.05.103>