



École Polytechnique Fédérale de Lausanne

Regional Surface Mass Balance Emulator based on Deep Learning:
local-scale SMB predictions over the Antarctic Peninsula

by Marijn van der Meer

Master Thesis

Approved by the Examining Committee:

Prof. Dr. sc. EPFL Martin Jäggi
Thesis Advisor

Prof. Dr. sc. TU Delft Stef Lhermitte
External Expert

Sophie de Roda Husman
Thesis Supervisor

EPFL IC IINFCOM MLO
INJ 341 (Bâtiment INJ)
Station 14
CH-1015 Lausanne

Wednesday 27th July, 2022

Nevertheless, she persisted.

Diese Arbeit ist meiner Grossmutter Brigitte Dorigo Baumann gewidmet. Danke liebes Grosi, für
all deine Unterstützung, ich habe dich sehr lieb.

Thank you also to my parents, Jan Roelof van der Meer and Barbara Baumann, who have always
been my biggest inspiration to join the scientific world through their work and dedication.

Acknowledgments

I would like to express my deep gratitude to Prof. Stef Lhermitte and Prof. Martin Jäggi, my thesis supervisors, for their patient guidance, enthusiastic encouragement, and valuable critiques of this research work. I would also like to thank Sophie de Roda Husman for her helpful advice and assistance on my thesis.

I would like to express my special thanks to Christoph Kittel for providing me with the MAR(ACCESS1.3) RCM dataset and for his availability to answer my questions. Likewise, I would like to thank Doury et al.'s team for providing me with information about their RCM-emulator.

I would like to thank my proofreaders in order of occurrence but not of importance: Melissa Corboz, Jan Roelof van der Meer, Jantina van der Meer, and Maxime Fellrath.

Last but not least, I would like to thank my dear friends and family, whose love and support kept me motivated and confident.

Finally, I would like to thank Prof. Payer from the EPFL for creating the practical latex package to write this EPFL thesis. The latest version of this package is available [here](#).

Lausanne, Wednesday 27th July, 2022

Marijn van der Meer

Abstract

Surface mass balance (SMB), a measure of processes leading to the addition or removal of ice, is crucial in controlling the stability and dynamics of the Antarctic ice sheet. It is imperative to have precise estimations of the local SMB to predict Antarctic ice sheet variations and sea-level rise. Currently, however, it is challenging to model SMB correctly because of its high variability across space and time and its complex interactivity with atmospheric variables. In this thesis, we developed a machine learning framework that learns to create local SMB predictions from global-scale climate model (GCM) outputs, which are readily available. We focused on the Antarctic Peninsula, the fastest warming region of the Antarctic ice sheet, which loses ice mass at significant rates. For this, detailed SMB predictions were available from the MAR(ACCESS1.3) regional climate model dataset (RCM). We built the RCM-emulator using a U-Net with convolutional block attention mechanisms, depth-wise separable convolutions, and a normalized RMSE loss. To train the emulator, we proposed two frameworks. In the first, called the perfect model framework, the RCM-emulator received upscaled RCM features at GCM resolution (\hat{F}_U) instead of low-resolution inputs from the GCM. In this case, the model learns only the RCM downscaling function and is undisturbed by potential RCM-GCM inconsistencies and biases. \hat{F}_U was evaluated both on upscaled RCM and GCM features. In the second framework, we trained the emulator with large-scale features directly sourced from the GCM (\hat{F}_G) to evaluate whether it could make accurate predictions despite RCM-GCM inconsistencies. Our results show that \hat{F}_U can almost perfectly reproduce the high-resolution SMB truth from upscaled RCM features. However, its predictions from GCM features lead to consistent underestimation of true SMB. In contrast, \hat{F}_G can reproduce the true SMB values successfully from GCM, and despite large-scale bias between the GCM and RCM, except for very dry regions. This suggests that the RCM-emulator can work directly with GCM data rather than with a perfect model framework because it can learn the proper GCM to RCM downscaling. The emulator also presents a large computational gain compared to running an RCM simulation, with training under ten minutes and predictions in a few seconds.

We conclude that machine learning emulators can be applied to produce fast and fine-scaled reproductions of RCM simulations from GCM data. Therefore, they can be an interesting tool for providing low-cost local-scale information on the ice sheet evolution and dynamics in a changing climate.

Plain Language Summary

The Antarctic ice sheet contains the majority of ice on Earth and would be the most significant contributor to sea-level rise if it were to melt completely. A critical variable in studying the stability and evolution of ice on Antarctica, such as glaciers and ice shelves, is surface mass balance (SMB). SMB is the balance between processes that add and remove ice on top of an ice sheet, glacier, or ice shelf. For example, those processes can include snowfall, surface melt, or erosion by winds. To predict changes in the Antarctic ice sheet, climate scientists need highly detailed information about SMB.

High-resolution maps of SMB are usually only produced by regional climate models (RCM). But these are costly and time-consuming to create because they require calculations on very powerful computers. The goal of this project was to accomplish this faster and with similar accuracy by using machine learning to predict SMB changes.

We trained a machine-learning algorithm i.e., a computer program that recognizes patterns in data, to learn the relationship between a group of low-resolution images of climate variables and a high-resolution image of regional SMB. The low-resolution data consisted of a bundle of atmospheric variables influencing SMB, such as precipitation, temperature, and surface pressure over the Antarctic Peninsula, an interesting region of Antarctica. Those coarse-grain atmospheric variables are easier and cheaper to obtain from worldwide simulations made by climate scientists.

Our machine-learning model can recreate regional images of SMB almost identical to ground truth observations by learning the relationship between low and high-resolution inputs. The model only struggles to predict proper SMB values for very dry areas (where the SMB is small) because it tends to emphasize points with larger changes in SMB. The predictive model is very fast, with training under ten minutes and output in a few seconds. What remains to be studied is whether the model would need new training for every new (unseen) region in Antarctica or if it could be deployed across areas.

We conclude that it is possible to make fast and detailed reproductions of SMB at regional scales using machine learning. Therefore, machine learning can be an interesting and cheap tool for gathering local-scale information about how ice sheets vary with climate change.

Contents

Acknowledgments	1
Abstract	2
Plain Language Summary	3
1 Introduction	6
2 Model and methods	11
2.1 Data	11
2.1.1 Climate model	11
2.1.2 Target and input domain	12
2.1.3 Features	13
2.2 Model	14
2.2.1 Architecture	14
2.2.2 Perfect model framework	16
2.3 Training	17
2.4 Evaluation	18
2.4.1 Evaluation metrics	19
3 Results	21
3.1 Prediction maps	21
3.2 Time-series of predictions	22
3.3 Evaluation metrics	23
3.4 Bias between UPRCM and GCM variables	23
3.5 Computational efficiency	24
4 Discussion	28
4.1 Extending an emulator of near-surface temperature to SMB	28
4.2 Necessity of the perfect model framework	29
4.3 Usability of the RCM-emulator	30
5 Conclusion	32

Bibliography **34**

A Data processing **40**

 A.1 Pre-processing of the GCM 40

 A.2 Pre-processing of the RCM 41

 A.3 Architecture selection 41

Chapter 1

Introduction

The Antarctic ice sheet¹ (AIS) is the largest class of glaciers and contains the majority of ice on Earth, with a volume of 30 million cubic kilometers [41]. It extends over 14 million square kilometers, roughly the combined areas of the US and Mexico [3]. This amount of ice constantly changes through melting processes and the addition of new mass from snow and ice.

Since the early 1900s, numerous glaciers worldwide have been melting at high rates due to human activity [59]. If the AIS were to melt entirely, sea-levels would rise by approximately 60 m [11, 24, 34], making the AIS the most significant potential contributor to sea-level rise. Indeed, the AIS has been steadily losing ice mass since 2002 [36, 51], mainly because of calving² and melting at the water-ice interface underneath ice shelves³ (basal melting) [24, 36, 44]. In the case of grounded ice, this water flows into the oceans and directly affects sea-level rise [23]. Antarctic ice can also indirectly contribute to sea-level rise through, for example, the collapse of ice shelves. Ice shelves are floating ice formed by the extension of grounded ice at the coastline, and they restrict the direct flow of the glaciers into the ocean. Because they float on the water, they do not contribute directly to sea-level rise, but if they break, the ice flow of grounded ice into the water can speed up and lead to sea-level rise [18, 42].

A primary variable used to observe the AIS is surface mass balance (SMB). SMB is the net balance between inputs and outputs of mass i.e., ice, snow, and water, on top of the ice sheet (Figure 1.1) [27]. It is influenced by the addition of mass through precipitation and snow-transporting winds and ablation via wind erosion and sublimation [24]. At any point, SMB close to zero indicates an arid climate with little change in the ice mass, while a positive SMB means that mass was added, through precipitation, for example. Inversely, a negative SMB indicates that a glacier is losing mass on the surface. SMB is highly important in monitoring the ice sheet's stability and

¹Ice sheet, otherwise referred to as a continental glacier, is a mass of glacial ice that covers land and is greater than 50'000 km². The only present-day ice sheets are in Antarctica and Greenland [22].

²Calving is the process of chunks of ice breaking off from the edge of a glacier [32].

³Ice shelves are floating ice that is connected to a continental landmass [42].

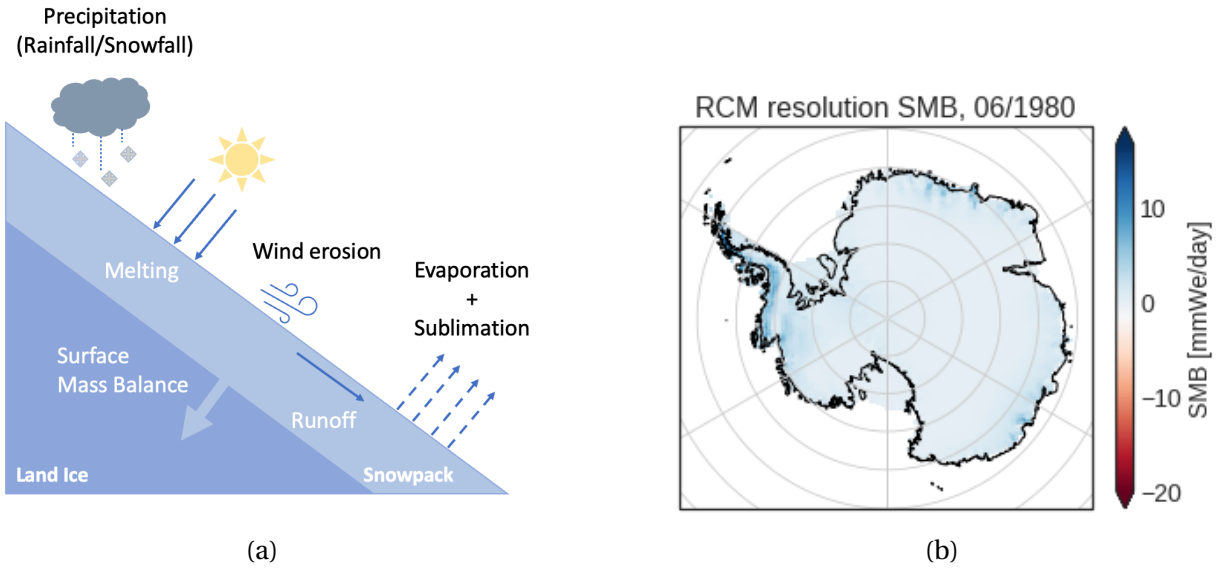


Figure 1.1: (a): Illustration of processes influencing the surface mass balance (SMB) on the Antarctic ice sheet. (b) Example of daily SMB values aggregated over a random month (06/1980) over Antarctica at regional climate model resolution.

evolution. Changes in surface mass of the AIS impact the global mass balance⁴ and therefore, the ice dynamics and sea-level rise [36]. Thus, it is imperative to correctly estimate the SMB of the AIS [23]. Currently, however, it is challenging to model SMB accurately.

Although SMB can be monitored with on-site tools (e.g., geodetic mass balance stakes), the vastness and climate of Antarctica make these expeditions challenging [27]. A more reasonable alternative is to extrapolate SMB in high-resolution⁵ climate model predictions [36]. To understand and predict the behavior of different climate phenomena, climatologists have developed complex climate models, which are simulations of climate variables over time and space in different parts of the world. However, at our current technological level, the complexity of these climate models is a compromise between computational costs, resolution of pixels, and the domain covered [9]. Depending on the spatial resolution, two types of climate models are typically defined: global (GCM) and regional (RCM). GCMs are simulations that cover the whole planet. They are relatively cheap in terms of computational costs because their domain size (i.e., the geographical region covered) is wide, while their spatial resolution is low (50-300 km) [9]. Unfortunately, for a fine-scale representation of the AIS, such as its edges or peripheral ice, the resolution of a GCM is too coarse [24, 50]. In addition, GCMs do not correctly incorporate critical polar physical processes, such as snow melt, albedo feedback, etc [24, 28]. On the other hand, RCMs are a dynamic downscaling of GCMs and are directly created from a GCM. They have a higher spatial resolution (1-50 km) but a smaller domain size. Polar-oriented RCMs, such as the

⁴Mass balance is the balance of overall mass, not only at the surface, gained by snow deposition, and the loss of mass by melting and calving [23].

⁵Note here that by low-resolution, we mean large-scale coarse images (e.g. 50-300 km); by high resolution, we mean local-scale fine images (e.g. 1-50 km).

Modèle Atmosphérique Régional (MAR), tackle the problem of low spatial resolution of GCMs over the AIS and give a significantly more robust evaluation of mass and energy fluxes at the AIS surface [13, 24]. Unfortunately, due to the RCMs' higher spatial resolution, they come at a high computational cost and time (several weeks on supercomputers) [9]. To save time and expenses, we studied the potential of emulating SMB predictions over the AIS from a polar-oriented RCM simulation using a machine learning framework.

To our knowledge, no machine learning model emulates regional-scale SMB over Antarctica yet. Using machine learning emulators as surrogate models to substitute computationally expensive and complex RCMs is still a new and recent approach. Nevertheless, the cryosphere community has already started harnessing the power of deep learning in recovering cryospheric features and representing cryospheric dynamics at regional scales [30]. For example, Hu et al. deployed a deep multi-layer perceptron model to correct surface melt simulations over the Larsen Ice Shelf in Antarctica from the RACMO2 RCM [21]. Similarly, for French Alpine glaciers, Bolibar et al. simulated and reconstructed the annual glacier-wide surface mass balance (SMB) series based on a deep artificial neural network [7].

Other more traditional methods without machine learning also recently obtained fine-scaled SMB simulations. For example, Gallée et al. combined RCMs and statistical methods by using a cascade of atmospheric models from large-scale to local-scale (1-20 km) to simulate a high-resolution map of SMB over Antarctica [14]. Similarly, Agosta et al. developed SMHiL, a downscaling of Antarctic SMB components at a high resolution (15 km) from large-scale atmospheric forcings [2]. Finally, over Greenland, Sellevold et al. used an elevation class method [49] and Geyer et al. statistical downscaling [15] as an SMB downscaling alternative to RCMs.

In this study, we followed the example of a novel machine learning temperature emulator proposed by Doury et al. [9]. The authors emulated the near-surface temperature (TAS) of a high-resolution RCM (CNRM-ALADIN63) over Southwestern Europe using a traditional U-Net. Their model study showed a substantial computational gain in computing an RCM and could almost perfectly reproduce the original near-surface temperature series.

The U-Net is a fully convolutional neural network (CNN) architecture developed by Ronneberger et al. [45]. It shows high performance in classification and segmentation tasks. Still, U-Net models are also used for image processing tasks, such as super-resolution, because they are excellent at gathering pixels belonging to the same object [9, 20]. Since the RCM-emulator ought to recognize distinct meteorological entities present in the low-resolution inputs to create the corresponding high-resolution target, U-Nets are highly relevant to use on images of atmospheric variables [9].

We expected to face several challenges when extending the temperature-emulator from [9] to emulating SMB over Antarctica. First, developing an SMB emulator is challenging because SMB varies highly across multiple scales of space and time. Moreover, SMB is impacted by complex interactivity between the atmosphere and the ice sheet surface, large-scale atmosphere circulations, and ice sheet topography [27]. To address the difficulty of modeling SMB, we equipped the U-Net used to emulate near-surface temperature with attention mechanisms and a normalized

RMSE loss designed to handle different ranges in SMB values across our physical domain.

Attention in a neural network helps it focus on specific input parts by emphasizing relevant activations during training [5, 39, 60]. We were inspired by the SmaAt-UNet by Trebing et al. [55], designed for a precipitation nowcasting task⁶. The SmaAt-UNet uses depthwise-separable convolutions (DSC) and attention implemented at the skip connections of the encoder⁷. Using DSC instead of traditional convolutions reduces the number of parameters of the model and builds a smaller and more efficient network. This is particularly interesting for this study because we aim for computational gain compared to running an RCM.

The traditional U-Net uses skip connections to combine features going via the decoder⁸ of the model with spatial information from the encoder. Yet, because feature representation in the first layers of the encoder can be lacking, this can add redundant low-level feature extractions in the upsampling path. Therefore, adding attention to the skip connections suppresses activations in irrelevant regions and decreases the number of redundant features [5].

In this study, we trained two SMB emulators. We trained the first emulator (\hat{F}_U) following the perfect model framework inspired by [9] where we used upscaled RCM features as low-resolution inputs (UPRCM) instead of inputs from the GCM. The perfect model framework allowed us to evaluate how the U-Net performs when it has to learn only the downscaling function of the RCM. For this, the RCM-emulator needed perfect spatial and temporal consistency between the coarse inputs and fine-scaled maps it recreated. This alignment cannot be guaranteed between global and regional climate models because of biases and temporal and geographical inconsistencies, and thus we implemented the perfect model framework [9, 47, 48]. Then, in a second step, we trained another emulator (\hat{F}_G) with coarse features directly from the GCM. In this training set, we wanted to see whether the model could learn the underlying dynamics, despite inconsistencies and biases, between GCM and RCM.

The core contributions of this thesis are the following:

- The extension of an RCM-emulator built for emulating local-scale temperature over Europe to reproducing SMB values over Antarctica is possible to extend .
- The predictions from our local-scale SMB emulator are computationally faster than performing an RCM simulation.
- Our local-scale SMB emulator can be trained directly on low-resolution inputs from a GCM and can learn underlying large-scale inconsistencies and biases.

⁶Nowcasting is the prediction of the present, the very near future, and the very recent past state of a weather indicator [58].

⁷The encoder is the first half in the U-Net architecture where the inputs are downsampled to be encoded into feature representations at multiple different levels [19].

⁸The decoder is the second half of the U-Net architecture where features learned by the encoder are concatenated with features going through the upsampling path [19].

Table 1.1

Notation	Description	Dimensions
\mathcal{D}	Input Domain	$\llbracket 1, I \rrbracket \times \llbracket 1, J \rrbracket$
\mathcal{E}	Target Domain	$\llbracket 1, K \rrbracket \times \llbracket 1, L \rrbracket$
(i, j)	Spatial indexes over input grid	\mathcal{D}
(k, l)	Spatial indexes over target grid	\mathcal{E}
X	Input: 2D variables over \mathcal{D}	$T \times \llbracket 1, I \rrbracket \times \llbracket 1, J \rrbracket \times C_1$
Z	Input: 1D variables over \mathcal{D}	$T \times C_2$
Y	Target: SMB over \mathcal{E}	$T \times \llbracket 1, K \rrbracket \times \llbracket 1, L \rrbracket$
t	Monthly temporal index	T
x	2D variables index	
z	1D variables index	
F	Downscaling function of the RCM	
\hat{F}	Emulator: estimation of F	
RCM	Regional Climate Model	
GCM	Global Climate Model	
UPRCM	RCM upscaled to GCM resolution	
$\hat{F}_U(\text{UPRCM})$	Emulator trained on UPRCM data, prediction on UPRCM	$\llbracket 1, K \rrbracket \times \llbracket 1, L \rrbracket$
$\hat{F}_U(\text{GCM})$	Emulator trained on UPRCM data, prediction on GCM	$\llbracket 1, K \rrbracket \times \llbracket 1, L \rrbracket$
$\hat{F}_G(\text{GCM})$	Emulator trained on GCM data, prediction on GCM	$\llbracket 1, K \rrbracket \times \llbracket 1, L \rrbracket$
AIS	Antarctic ice sheet	
SMB	Surface mass balance	
DSC	Depth-wise separable convolutions	
CBAM	Convolutional block attention mechanisms	
MAR	Modèle Atmosphérique Régional	

Table 1.1. Notations used in this thesis.

Chapter 2

Model and methods

In this study, we built an RCM-emulator \hat{F} that uses a neural network architecture to estimate the downscaling function F in

$$Y = F(X) \quad X \subset \mathcal{D}, Y \subset \mathcal{E} \quad (2.1)$$

where X are low-resolution variables from a global climate simulation (GCM) over an input domain \mathcal{D} and Y is a high-resolution surface variable of a regional climate model (RCM) over a target domain \mathcal{E} .

Note that all notations used in this study are described in Table 1.1.

2.1 Data

2.1.1 Climate model

The RCM variable emulated by our model is the daily SMB values from MAR(ACCESS1.3), a regional downscaling by MAR of the ACCESS1.3 GCM (CMIP5 [6, 54]). MAR is known for accurately modeling physical processes in polar regions such as SMB, air-snow interactions, and atmospheric circulation over ice sheets [31]. According to studies comparing climate models conducted in [1, 24], ACCESS1.3 and MAR(ACCESS1.3) were the models that most accurately represented the present Antarctic climate compared to ERA-Interim. For this reason, we selected MAR(ACCESS1.3) and its corresponding GCM as the climate simulations for our RCM-emulator.

The time frame of MAR(ACCESS1.3) covers historical values from 1980-2006 and future climate simulation RCP8.5 from 2006-2100 [35]. In terms of resolution, the RCM grid is in south polar stereographic coordinates of 35×35 km while the GCM resolution is of 1.25° latitude by 1.875° longitude (approximately 68×206 km) [6, 8]. So that both regional and global climate models are in the same projection system, we projected the GCM to south polar stereographic coordinates (c.f. section A.1 for more details).

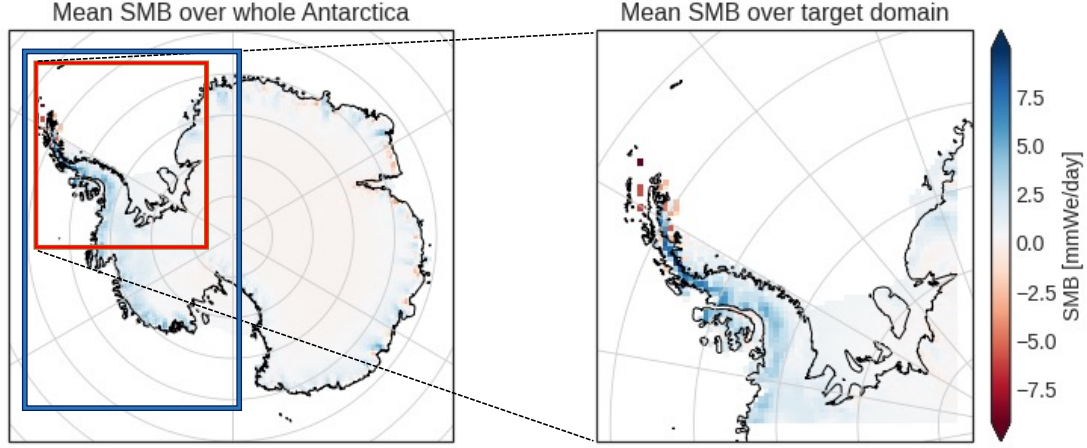


Figure 2.1: Regions chosen as target domain \mathcal{E} (in red) and input domain \mathcal{D} (in blue) for the RCM-emulator. Illustrated underneath are the daily mean surface mass balance (SMB) over Antarctica (left) and the Antarctic Peninsula (right) from 1980 to 2100.

2.1.2 Target and input domain

The target domain \mathcal{E} chosen for the RCM-emulator is a grid box of 64×64 pixels at 35 km resolution that contains the Antarctic Peninsula (AP) in West Antarctica (Figure 2.1). The AP is the most northern part of Antarctica, extending over approximately 5 million square kilometers and mainly covered by ice. The region is very mountainous, with its highest peaks rising to about 3'000 m and major ice shelves on the AP include the Larsen and Ronne ice shelves. In terms of temperature, the Peninsula has the mildest climate in Antarctica. Its temperatures are warmest in January, averaging 2°C , and coldest in June with -15 to -20°C [4]. Interestingly, precipitation varies significantly within the AP. The Peninsula's tip has the highest precipitation levels, with 35-50 cm per year. On the west and northeast coast, occasional precipitation leaves levels at 35 cm. On the other hand, along the east coast and the interior of Antarctica, the climate is drier, with 10-15 cm [10, 56].

Because of the specific patterns of climate variables such as temperature and precipitation, the AP has a high annual and geographical variability in SMB values. This shows when looking at mean daily SMB values (Figure 2.1). As expected from precipitation patterns, the Peninsula's tip and west coast have higher SMB values than the rest. Mean SMB values over the AP range from approximately -10 to 10 mmWe/day¹ while the extremes observed in this region from 1980 to 2100 are a minimum of -59 mmWe/day and a maximum of 30 mmWe/day. This variation in

¹ millimetre water equivalent per day

SMB values made this region very interesting to emulate because we wanted to see how the RCM-emulator adapted to different annual patterns of SMB over our target domain.

The input domain \mathcal{D} of the RCM-emulator covers approximately 17 million square kilometers and is a 48×25 pixels grid box (at 68×206 km resolution) defined around the target domain. Because it is favorable to give our machine learning model a square input, it was resized to 32×32 by bilinear interpolation.

2.1.3 Features

The RCM-emulator receives input features (X, Z) that consist of two-dimensional variable X , and one-dimensional Z (Table 2.1). X is a spatial encoding of eight different atmospheric variables $x \in C_1$ measured daily at near surface level over \mathcal{D} . After a monthly mean aggregation of the data, the frequency of the variables becomes monthly. Each atmospheric variable at time-step $t \in T$ is normalized according to its spatial mean $\bar{X}_{t,x}$ and standard deviation $\sigma(X_{t,x})$:

$$\tilde{X}_{t,i,j,x} = \frac{X_{t,i,j,x} - \bar{X}_{t,x}}{\sigma(X_{t,x})} \quad \forall (i, j) \in \mathcal{D}, t \in T, x \in C_1 \quad (2.2)$$

In terms of feature selection for X , we followed the same procedure as the authors of the temperature-emulator and used all available variables [9]. This gives the following equation for X :

$$X = \left[\tilde{X}_{t \in T, x \in C_1} \right] \subset T \times \mathcal{D} \times C_1 \quad (2.3)$$

Z is a one-dimensional temporal encoding that includes the time series of spatial means $\bar{X}_{t,x}$ and standard deviations $\sigma(X_{t,x})$ for each $x \in C_1$ and $t \in T$. It also includes a cosine and sine vector to encode the information about the month of the year.

$$\cos\left(\frac{2\pi t}{12}\right); \sin\left(\frac{2\pi t}{12}\right) \quad \forall t \in T \quad (2.4)$$

Overall this gives the following equation for Z :

$$Z = \left[\bar{X}_{t \in T, x \in C_1}, \sigma(X)_{t \in T, x \in C_1}, \cos, \sin \right] \subset T \times C_2 \quad (2.5)$$

Each of the spatial means $\bar{X}_{t,x}$ and standard deviations $\sigma(X_{t,x})$ in Z are normalised according to a reference period (ref = 1980-2000) [9]:

$$\tilde{Z}_{t,x} = \frac{Z_{t,x} - \bar{Z}_{\text{ref},x}}{\sigma(Z_{\text{ref},x})} \quad t \in T, x \in C_1 \quad (2.6)$$

where $\bar{Z}_{\text{ref},x}$ and $\sigma(Z_{\text{ref},x})$ are respectively the temporal mean and standard deviation of the spatial means or standard deviations of $X_{\text{ref},x} \subset T_{\text{ref}} \times C_1$ for variable x . Because the X variables

Table 2.1

Variable Name	Variable Notation	Units	Dimensions
2D variables			
Northward Wind	NW	$[\text{ms}^{-1}]$	\mathcal{D}
Eastward Wind	EW	$[\text{ms}^{-1}]$	\mathcal{D}
Shortwave Downward Radiation	SWD	$[\text{Wm}^{-2}]$	\mathcal{D}
Longwave Downward Radiation	LWD	$[\text{Wm}^{-2}]$	\mathcal{D}
Specific Humidity	QQP	$[\text{g/Kg}]$	\mathcal{D}
Temperature	TT	$[\text{°C}]$	\mathcal{D}
Precipitation	PR	$[\text{mmWe/day}]$	\mathcal{D}
Pressure	SP	$[\text{hPa}]$	\mathcal{D}
1D variables			
Spatial mean of 2D variables	\bar{X}_x		$[C_1]$
Spatial std of 2D variables	$\sigma(X_x)$		$[C_1]$
Seasonal Indicators	cos, sin		$[2]$

Table 2.1. Two and one-dimensional input features given to the RCM-emulator at time-step $t \in T$. Each feature is measured daily at near surface level over domain \mathcal{D} . The frequency of variables is monthly after a monthly mean aggregation.

are normalized at each time step by their spatial mean, they no longer carry any temporal encoding. Adding Z to the RCM-emulator allows it to have access to this information.

2.2 Model

2.2.1 Architecture

Our emulator’s architecture extends the traditional U-Net [46] used to emulate near-surface temperature in [9] with mechanisms from the SmaAt-UNet model [55] (Figure 2.2). The SmaAt-UNet equips the U-Net architecture with convolutional block attention mechanisms (CBAM) and depthwise-separable convolutions (DSC) instead of regular convolutional operations.

The network is U-shaped, divided into a down-sampling (encoder) section that forms the left side and an up-sampling (decoder) on the right. The encoder consists of double convolutions (blue arrow in Figure 2.2) followed by max-pooling (red arrow). This process halves the image size and doubles the number of channels at each layer of the U-Net. Instead of traditional convolutional

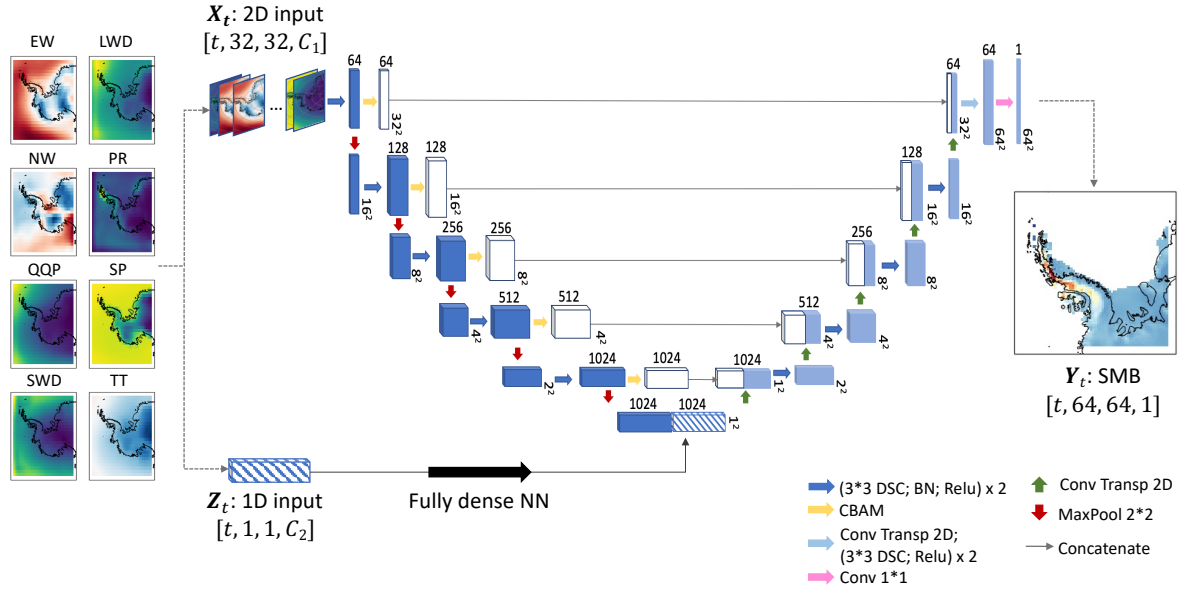


Figure 2.2: Illustration of an observation at time-step t . Left: 2D input variables X_t on the input domain with its corresponding 1D variable Z_t . Right: target surface mass balance (SMB) Y_t on the target domain. Middle: scheme of the U-Net architecture used for this paper.

operations, the emulator uses DSC, designed to reduce the number of parameters and make the model faster [55].

At the bottom of the U-Net, encoded spatial information from X_t and temporal information from Z_t are concatenated. The corresponding 1D input Z_t of X_t first goes through a fully dense neural network to reach the same number of channels as the output of the last layer of the encoder. Then, it is concatenated with the encoder output at the bottom of the "U". This constrains the U-Net to give equal importance to the spatial and temporal inputs before starting the decoding path and generating the high-resolution SMB image [9].

The decoder is built out of three parts. First, a 2D transposed convolution operation (green arrow) doubles the image size. Then, the resulting feature maps are concatenated with the previous encoder's attention maps (white block) via skip-connections (grey arrow). Lastly, a double depthwise-separable halves the number of channels (blue arrow). Skip-connections between layers (grey arrow) make it possible to skip large sections if required and create a smoother loss surface [29]. An additional up-sampling layer (light blue arrow) is added to the decoder to reach the target size (64×64), and a 1×1 convolution (pink arrow) changes the number of channels from 64 to 1. The model outputs a single image representing the SMB values predicted by the network at time-step t .

Convolutional block attention modules (yellow arrow) are placed after each double convolution in the encoder and used to detect important features over the channels and spatial regions of

the inputs [55]. In CBAMs, the attention mechanism is first applied over the channels of the image and then the spatial dimension. Note that the input to the next layer of the encoder is not the attention feature map (white block) but the convoluted and downsampled image of the previous layer (dark blue block). This way, the original image features are preserved throughout the encoder layers. The attention blocks are fed through skip-connections to the corresponding decoder layer to be concatenated. Contrary to the DSC convolutions used in the encoder and decoder, CBAMs use regular convolutions.

2.2.2 Perfect model framework

We were inspired by the *perfect model framework* from [9] where both the low-resolution inputs and high-resolution target SMB come from the same RCM simulation when training the emulator. In this setting, the model aims to learn only the downscaling function F of the RCM (Equation 2.1). For this, it needs perfect consistency i.e., high temporal and spatial correlation, between the global-scale inputs and the regional-scale target. Otherwise, the emulator will try to learn a relationship between them that does not exist or is imprecise [9]. Because of large-scale biases and inconsistencies between GCM and RCM variables, a perfect consistency cannot be guaranteed [47, 48]. The perfect model framework provides it.

To test the effect of this perfect model training framework on our SMB-emulator, we created "GCM-like" features from the RCM (UPRCM). First, we used conservative interpolation to upscale RCM features to GCM resolution (68×206 km). Then, the upscaled RCM features were smoothed with a 3×3 moving average filter. This filter conserves the GCM grid while smoothing out the information of each pixel. This further ensured the removal of local-scale information that might remain after the upscaling [9, 25].

To assess the presence of biases and inconsistencies between UPRCM and GCM features, we used two correlation statistics:

- Temporal correlation: for each atmospheric variable $x \in C_1$ and point $p = (i, j)$ in the input domain \mathcal{D} , we calculated the Pearson correlation coefficient between the GCM and UPRCM time series:

$$\rho(G_p^x, U_p^x) = \frac{\text{cov}(G_p^x, U_p^x)}{\sigma(G_p^x)\sigma(U_p^x)} \quad \forall p \in \mathcal{D}, x \in C_1 \quad (2.7)$$

where $G_p^x = \text{GCM}[1 : T, i, j, x]$, $U_p^x = \text{UPRCM}[1 : T, i, j, x]$, $\text{cov}(\cdot)$ is the covariance and $\sigma(\cdot)$ is the standard deviation.

- Spatial correlation: for each $x \in C_1$ and time-step $t \in T$, we calculated the spatial correlation between GCM and UPRCM images:

$$\text{sc}(G_t^x, U_t^x) = \frac{\text{cov}(G_t^x, U_t^x)}{\sigma(G_t^x)\sigma(U_t^x)} \quad \forall t \in T, x \in C_1 \quad (2.8)$$

where $G_t^x = \text{GCM}[t, 1 : I, 1 : J, x]$ and $U_t^x = \text{UPRCM}[t, 1 : I, 1 : J, x]$.

2.3 Training

As aforementioned in subsection 2.1.3, each single observation given to our emulator is composed of features X_t and Z_t for a month $t \in T$. X_t is an array in $I \times J \times C_1$, where the two first dimensions are the spatial size of the input domain, and the number of channels C_1 is the different atmospheric variables chosen as predictors. $Z_t \subset C_2$ is its corresponding temporal encoding (Figure 2.2).

We created two emulators by following two scenarios. We trained the first emulator (\hat{F}_U) following the perfect model framework (subsection 2.2.2) where we used UPRCM features as low-resolution inputs. Then, in a second step, we trained another emulator (\hat{F}_G) with coarse features directly from the GCM. The perfect model framework allowed us to evaluate how the U-Net performs when it has to learn only the downscaling function of the RCM. In the second training setting, we wanted to see whether the model could learn the underlying dynamics, despite inconsistencies and biases, between GCM and RCM.

In the temperature-emulator from [9], the authors propose to view the problem as a regression and use MSE as a loss function. However, our emulator performed poorly using an MSE loss, and it did not seem appropriate to our setting. As aforementioned, the scale of SMB values varies significantly across the target domain of the Antarctic Peninsula. For example, dry inland points have small annual variations of SMB, with maximum values reaching under 2 mmWe/day, while a point on the west coast of the Peninsula can reach low extremes of -20 mmWe/day. In this unbalanced setting, MSE penalizes the ill-fitting of low SMB regions less than points with high values of SMB. For this reason, we chose to use a normalized RMSE (NRMSE) loss. Normalizing the RMSE facilitates comparing datasets with different magnitude scales, as in our case. For each observation i.e., each time step t , given to the emulator, we calculated the NRMSE loss between the predicted SMB maps \hat{Y}^t and the true Y^t over all positions $p \in \mathcal{E}$:

$$\begin{aligned} \text{NRMSE} \left(Y^t, \hat{Y}^t \right) &= \frac{\text{RMSE}}{Y_{\max} - Y_{\min}} \\ &= \frac{\sqrt{\frac{1}{P} \sum_p (\hat{y}_p^t - y_p^t)^2}}{Y_{\max} - Y_{\min}} \quad \forall t \in T \end{aligned}$$

where \hat{y}_p^t is the SMB value predicted at location p and time t , P the number of points in \mathcal{E} and Y_{\max}, Y_{\min} are respectively the maximum and minimum value of SMB over T and \mathcal{E} .

We trained both emulators using a batch size of 100 and over 50 epochs. We used early stopping and a learning-rate scheduler that reduced the learning rate on loss plateaus ($\text{LR}_0 = 0.005$).

The RCM-emulator architecture was implemented in PyTorch 1.11 and is available on our GitHub. Training was done on Google Colab's GPU (NVIDIA Tesla K80).

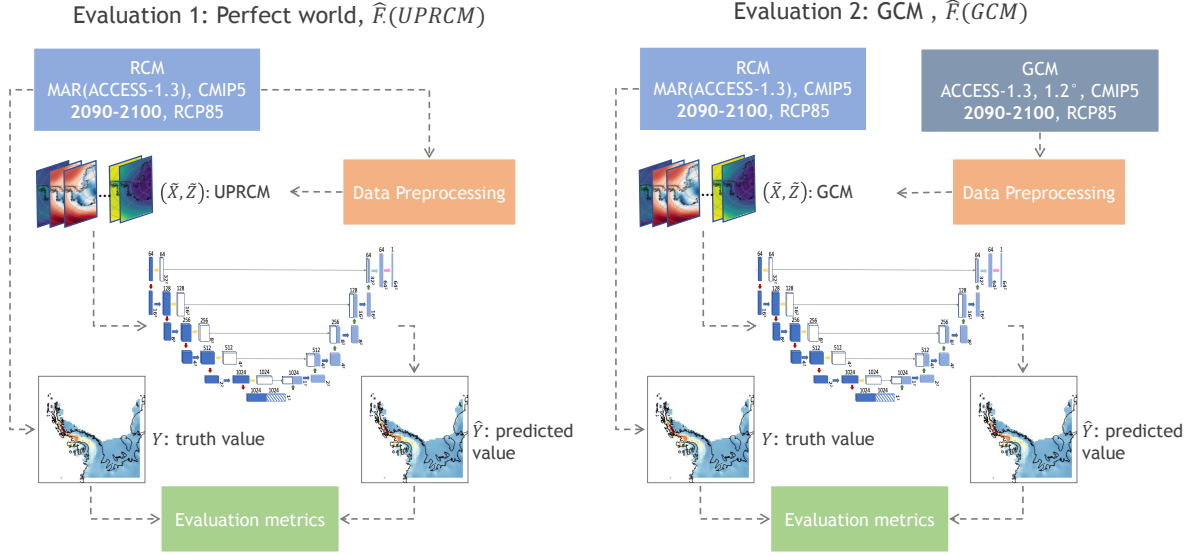


Figure 2.3: Evaluation frameworks of the RCM-emulator. Left: perfect world scenario where the emulator \hat{F}_U or \hat{F}_G makes regional SMB predictions using upscaled RCM features (UPRCM) as global-scale inputs - $\hat{F}_U(UPRCM)$. Right: global-scale features come from the GCM for predictions - $\hat{F}_U(GCM)$. Evaluation metrics used to compare predictions and the true SMB truth are described in subsection 2.4.1.

2.4 Evaluation

We separated the time-frame T of the climate simulations into an approximate 90%-10% split with a train and test period of $T_{train} = 1980 - 2090$ and $T_{test} = 2090 - 2100$, so that the RCM-emulator made predictions on features never seen during training. We arbitrarily chose to put the test period at the end of the climate model's time frame out of simplicity, but we could also have taken the test samples elsewhere as long as they were consecutive.

Evaluation of \hat{F}_U

To inspect how the emulator performs under conditions similar to those of its training, we first evaluated emulator \hat{F}_U in the perfect model with UPRCM features - $\hat{F}_U(UPRCM)$. Then, in a second step, we evaluated the emulator on GCM inputs, $\hat{F}_U(GCM)$. This assessed how the emulator trained on UPRCM could apply what it learned to GCM features. Furthermore, the performance of $\hat{F}_U(GCM)$ is also a good indicator of the presence of inconsistencies between UPRCM and GCM features.

Evaluation of \hat{F}_G

Because we trained this emulator on the GCM, we evaluated \hat{F}_G directly on test features from the GCM - $\hat{F}_G(GCM)$.

The two evaluation frameworks are illustrated in Figure 2.3. For each emulator \hat{F} , we assessed single month and average predictions made over the target domain. This way, we could examine which regions of the Antarctic Peninsula had the best reconstruction of SMB patterns in terms of details, precision, and intensity. Furthermore, we looked at individual time series of predicted SMB values for single points in the target region. We specifically chose these points to evaluate how the model handled different patterns and intensities of SMB.

2.4.1 Evaluation metrics

To evaluate the global performance of our machine learning model, we used different traditional statistics. For each RCM-emulator \hat{F} and point $p \in \mathcal{E}$, we compared the target SMB time series Y_p to the predicted values \widehat{Y}_p over the time period T_{test} (2090-2100). We used the Pearson correlation coefficient, RMSE, and Wasserstein distance as evaluation metrics. Correlation is a good indicator of the reconstruction of temporal patterns such as global synchrony and seasonality. RMSE and the Wasserstein distance evaluate the fitting of extreme values and the representation of monthly variability.

Pearson correlation coefficient

The Pearson correlation coefficient measures how two continuous time series change over time as a number between -1 (negatively correlated), 0 (uncorrelated), and 1 (perfectly correlated) [40].

$$r(Y_p, \widehat{Y}_p) = \frac{\text{cov}(Y_p, \widehat{Y}_p)}{\sigma(Y_p)\sigma(\widehat{Y}_p)} \quad \forall p \in \mathcal{E} \quad (2.9)$$

where $\text{cov}(\cdot)$ is the covariance and $\sigma(\cdot)$ is the standard deviation.

RMSE

Root Mean Squared Error (RMSE) measures the square root of the average squared differences between predicted and target observations. It is also defined as the square of the MSE:

$$\text{RMSE}(Y_p, \widehat{Y}_p) = \sqrt{\text{MSE}(Y_p, \widehat{Y}_p)} \quad (2.10)$$

$$= \sqrt{\frac{1}{T_{test}} \sum_t (\hat{y}_p^t - y_p^t)^2} \quad \forall p \in \mathcal{E} \quad (2.11)$$

where \hat{y}_p^t is predicted SMB value and y_p^t the true SMB value at location $p \in \mathcal{E}$ and time step $t \in T_{test}$.

Wasserstein distance

The Wasserstein distance measures the distance between two probability density functions $f(\cdot)$, in our case $f(Y_p)$ and $f(\widehat{Y}_p)$. It is the numerical cost of an optimal transportation problem i.e., the

cost of the optimal transport plan [57] for moving the mass in the predicted measure to match that in the target [12].

$$W\left(f(Y_p), f(\widehat{Y}_p)\right) = \sum_t |y_p^t - \hat{y}_p^t| \quad \forall p \in \mathcal{E} \quad (2.12)$$

where \hat{y}_p^t is the predicted SMB value and y_p^t the true SMB value at location $p \in \mathcal{E}$ and time step $t \in T_{test}$.

Chapter 3

Results

3.1 Prediction maps

To evaluate the performance of our RCM-emulator at reconstructing spatial structures of SMB values, we compared a prediction for a random month and the average predictions over the test period to the RCM truth. Prediction maps are compared to the truth using RMSE and results can be seen in Figure 3.1.

Unsurprisingly, in the first two columns in Figure 3.1, we see that compared to the low-resolution UPRCM map, the high-resolution RCM truth is more detailed and has a more complex spatial structure. As expected, because of their precipitation patterns, we see that the tip and west coast of the Antarctic Peninsula show high values of SMB in the RCM truth, some points reaching 10 mmWe/day. For the chosen random month of May 1980, except for higher intensity regions in the mainland, both $\hat{F}_U(\text{UPRCM})$ and $\hat{F}_G(\text{GCM})$ competently reproduce the spatial structure of the RCM truth. On average, both emulators' predictions also look very similar to the average true SMB. The emulators have a very small RMSE (0.27 and 0.30), which becomes even smaller when looking at the average prediction over the test period (0.09 and 0.08). On the other hand, $\hat{F}_U(\text{GCM})$, both for the random month and on average, is not able to reproduce the extreme values of SMB and creates a toned-down version of the truth. In particular, it underestimates the high magnitude SMB values on the west coast of the Antarctic Peninsula. This is reflected in its RMSE, which is twice as high as the other two models (0.53 for a random month and 0.23 on average).

Overall, these results hint that both emulators \hat{F}_U and \hat{F}_G , when evaluated on data similar to what they were trained on, have a solid capacity to reproduce the complex spatial structures of the RCM.

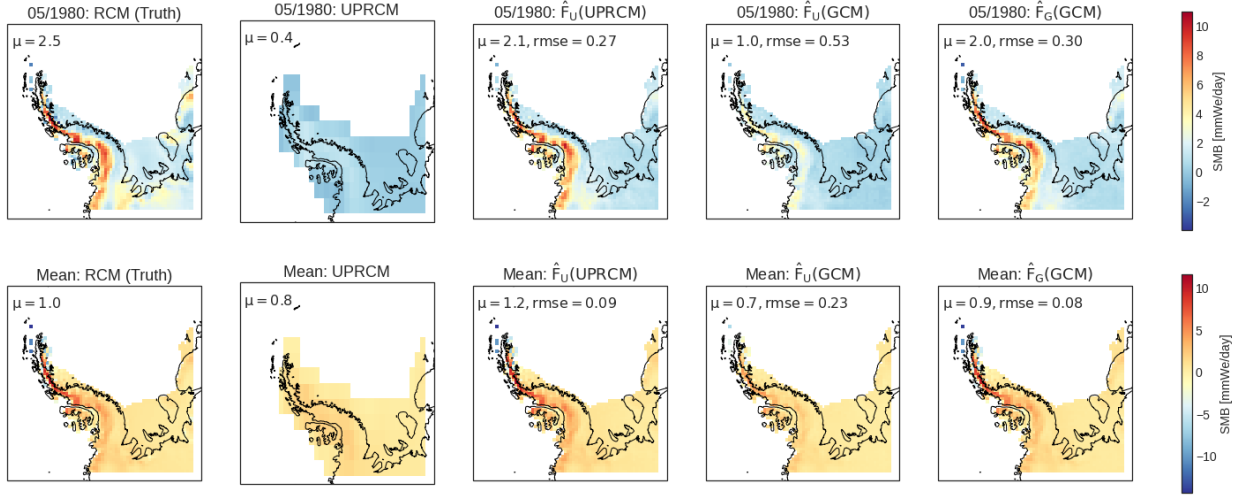


Figure 3.1: Surface mass balance (SMB) predictions of the RCM-emulator on a random month (05/1980) (top) and averaged over the test period (2090-2100) (bottom) over target domain \mathcal{E} . From left to right: SMB in RCM truth, UPRCM, $\hat{F}_U(\text{UPRCM})$, $\hat{F}_U(\text{GCM})$ and $\hat{F}_G(\text{GCM})$. Legend: spatial mean (μ) of SMB over \mathcal{E} and spatial RMSE (rmse) between the emulated and true RCM SMB pixel values.

3.2 Time-series of predictions

To assess how the RCM-emulator can predict different temporal patterns and intensities of SMB, we plotted the emulated time series for four geographical points (Figure 3.2a). P1 on the Larsen Ice Shelf has high precipitation levels, and SMB values oscillate annually between -5 and 5 mmWe/day. P2 on the west coast, has SMB values reaching low extremes of -15 mmWe/day. P3 on the east coast has low yearly precipitation and minor SMB variations (0-4 mmWe/day). P4 on the Ronne ice shelf, has a significantly drier climate (0-1.5 mmWe/day). Figure 3.2b presents each point's time series and annual SMB predictions. For each point, the RMSE, NRMSE, and Pearson correlation coefficient are calculated between the predicted and true SMB time series.

For each of these four points, $\hat{F}_U(\text{UPRCM})$ and $\hat{F}_G(\text{GCM})$ come very close to reproducing the temporal patterns of the RCM truth series, as is expected from the results of section 3.1. For P1 and P2, all emulators reproduce seasonality well, with high correlation values to the true SMB (close to 1 for P2). $\hat{F}_G(\text{GCM})$ is even a little bit better than $\hat{F}_U(\text{UPRCM})$ at emulating low drops and high peaks in SMB. For P3 and P4, both emulators show a good ability to reproduce the time series pattern, even when their behavior is less seasonal, like for P3. We notice for P4, that the emulators have difficulties reproducing multiple close peaks per year and tend to merge them into one prominent peak.

As was already seen in Figure 3.1, $\hat{F}_U(\text{GCM})$ tends to underestimate high amplitude values of SMB, and this is noticeable in the time series for all four points. The $\hat{F}_U(\text{GCM})$ predictions can reproduce the seasonal patterns (reflected in a high correlation) but produce a toned-down version of the true RCM time series (mirrored in a higher RMSE). This is observable also in the time series' distributions, where the peak of $\hat{F}_U(\text{GCM})$ is higher and centered around lower SMB values compared to $\hat{F}_U(\text{UPRCM})$ and $\hat{F}_G(\text{GCM})$, which are more spread out and closer to the

RCM truth. This pattern repeats in the annual SMB predictions for all emulators. $\hat{F}_U(\text{UPRCM})$ and $\hat{F}_G(\text{GCM})$ come very close to the true annual SMB values, with $\hat{F}_G(\text{GCM})$ again slightly better than $\hat{F}_U(\text{UPRCM})$ for almost all years. $\hat{F}_U(\text{GCM})$ on average consistently underestimates the truth by a factor of 2.

3.3 Evaluation metrics

We used three traditional machine learning metrics as described in subsection 2.4.1 to evaluate the overall performance of emulators \hat{F}_U and \hat{F}_G in reproducing the RCM truth. Results can be seen in Figure 3.3. We were particularly interested in seeing how the evaluation metrics differed for regions with high magnitudes of SMB (i.e., high precipitation levels), such as the tip and west coast of the Antarctic Peninsula, and dryer regions, such as the east coast and mainland.

Pearson correlation coefficient: As seen in the box-plot of the correlation values, on average, SMB predictions from $\hat{F}_U(\text{UPRCM})$ and $\hat{F}_G(\text{GCM})$ have higher correlation values to the RCM truth than $\hat{F}_U(\text{GCM})$. This is especially flagrant on the tip and west coast of the Antarctic Peninsula, where $\hat{F}_U(\text{UPRCM})$ and $\hat{F}_G(\text{GCM})$ show the highest correlation to the target, with values close to 1. The east coast of the Peninsula has the lowest correlation for all models but especially for $\hat{F}_U(\text{GCM})$. We suspect this is because the model's predictions underestimate the temporal SMB patterns of dry regions.

Wasserstein distance and RMSE: $\hat{F}_U(\text{GCM})$ has the highest Wasserstein distance and RMSE values, which indicates that the density probability functions of its emulated SMB series are farther from the RCM truth. The Antarctic Peninsula has especially high differences in distributions, with outliers with values up to 10 for the Wasserstein distance and 14 for RMSE. This hints that the $\hat{F}_U(\text{GCM})$ emulator is not able to predict high SMB magnitudes when given GCM inputs while trained on UPRCM.

Overall, we see that according to these evaluation metrics, $\hat{F}_U(\text{UPRCM})$ and $\hat{F}_G(\text{GCM})$ perform very similarly and are consistently better than $\hat{F}_U(\text{GCM})$. While temporal patterns and seasonality are best reconstructed in regions of high precipitation, such as the tip and west coast of the Antarctic Peninsula, the emulators' predictions tend to underestimate extreme (high and low) values. This is especially visible for $\hat{F}_U(\text{GCM})$ and in dry regions, such as the inland and east Peninsula.

3.4 Bias between UPRCM and GCM variables

To assess the bias and inconsistencies between the large-scale (low-resolution) and local-scale (high-resolution) atmospheric variables, we calculated the temporal and spatial correlation between UPRCM and GCM features (Equation 2.7 and 2.8). Results are illustrated in Figure 3.4.

Temporal correlation

For most atmospheric variables, the time series of UPRCM and GCM features are highly positively correlated, with values very close to one (Figure 3.4a). However, the two wind variables (EW, NW) show inconsistencies between large and local-scale time series over the mainland and the Antarctic Peninsula, with minimal correlation values reaching 0.2. This suggests that, except for the winds, there is almost no inconsistency in the seasonal patterns between regional high-resolution and global low-resolution variables.

Spatial correlation

Atmospheric variables like temperature, specific humidity, radiation, and precipitation show significant changes in spatial correlation between UPRCM and GCM features (Figure 3.4b). The most flagrant variables are shortwave downward radiation (SWD) and precipitation (PR). SWD has an annual spatial correlation pattern strongly oscillating between approximately 0.2 in winter and 0.8 in summer. On the other hand, PR has a very poor spatial correlation over the whole test period, with a maximum of only 0.4. Looking at some individual months of those variables, we see that for SWD, the spatial correlation is low between UPRCM and GCM maps during winter because the GCM predicts higher radiation values in the mainland of Antarctica than UPRCM (Figure 3.4c). However, because there is very little radiation, spatial correlation is high for the summer months. For precipitation, the spatial patterns in the UPRCM and GCM are very different, with the UPRCM being much more precise in its predictions. For example, for November 2093, the UPRCM shows a local high-intensity precipitation event on the tip of the Antarctic Peninsula, while the GCM predicts a more vague pattern of lower intensity over the water in the west. We also notice a streak pattern on the left boundary of the RCM for all variables, but this is a known problem as the boundaries of climate models are usually imprecise.

Overall, we found problems in consistency between large GCM and local-scale RCM variables. As mentioned in subsection 2.2.2, these inconsistencies might confuse an RCM-emulator, and this might justify the need for the perfect model framework for training a model.

3.5 Computational efficiency

Using early stopping as described in section 2.3, RCM-emulators \hat{F}_U and \hat{F}_G trained for 31 and 40 epochs before stopping. Training took approximately 4 minutes, with roughly 6 seconds per epoch. A GPU is no longer needed once the model is trained, and making predictions on test data takes 15 seconds on a CPU, which is nearly instantaneous. This computational time is significantly smaller than running an RCM simulation which can take multiple weeks to calculate on a super-computer. Note that our computational time does not include data preparation for the RCM-emulator. But if the data is readily obtainable and a correct data processing pipeline is in place, selecting the target region and creating the UPRCM takes only a few hours on a CPU.

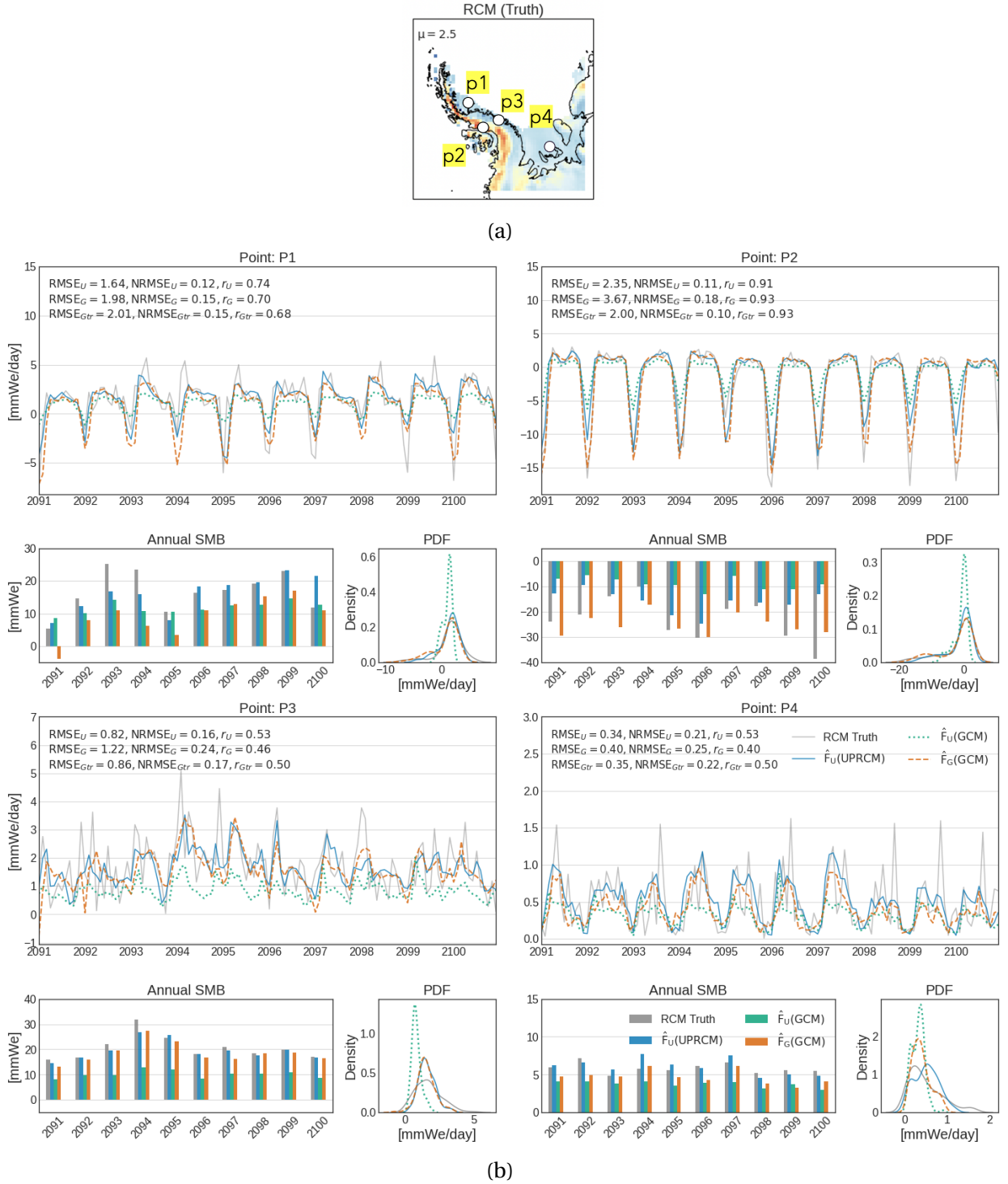


Figure 3.2: SMB predictions of RCM-emulators $\hat{F}_U(\text{UPRCM})$ (blue line), $\hat{F}_U(\text{GCM})$ (dotted green) and $\hat{F}_G(\text{GCM})$ (dashed orange) compared to RCM truth (grey line) over test period (2090-2100). (b) Time-series, daily probability density functions (PDF), and bar-plots of annual sums of SMB predictions for four different geographical points (a) in target domain \mathcal{E} . Legend: Pearson correlation coefficient (r), RMSE (RMSE), and normalized RMSE (NRMSE) between the time series of emulated and true SMB. For these metrics $U = \hat{F}_U(\text{UPRCM})$, $G = \hat{F}_U(\text{GCM})$ and $Gtr = \hat{F}_G(\text{GCM})$.

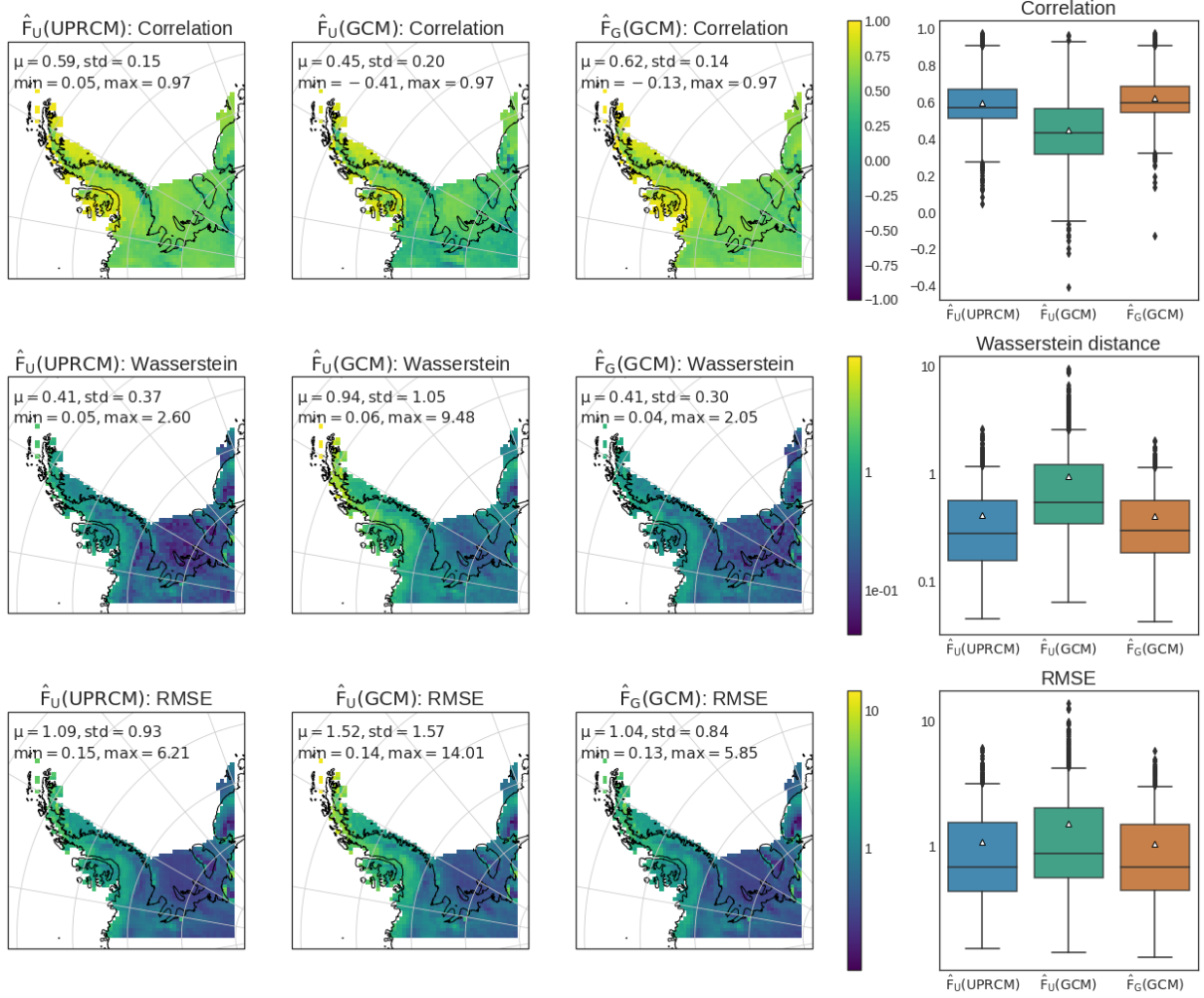
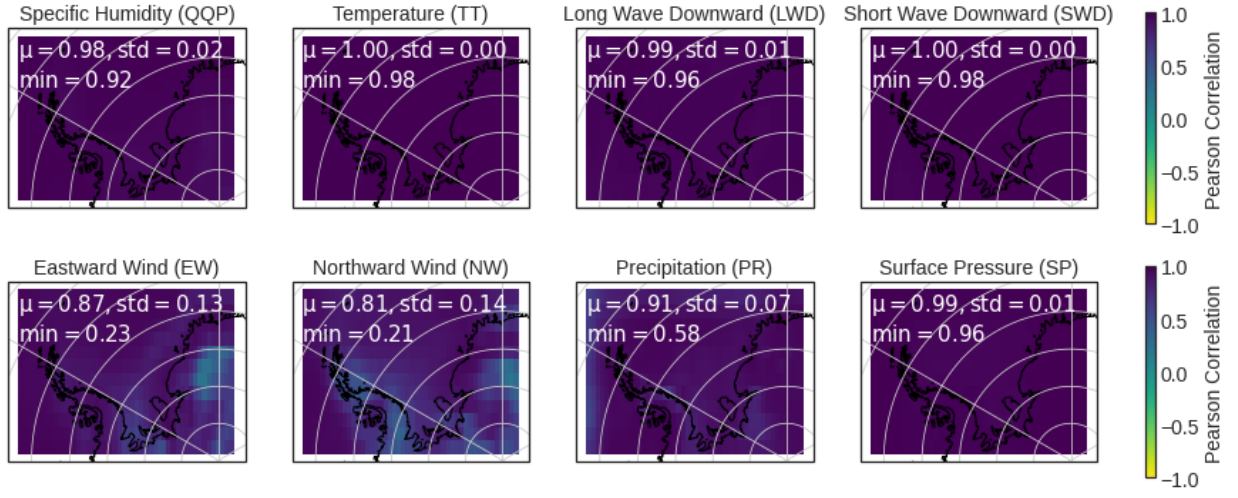
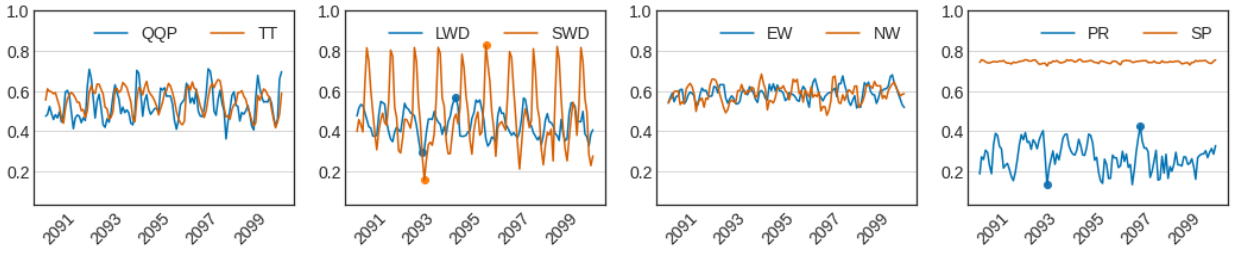


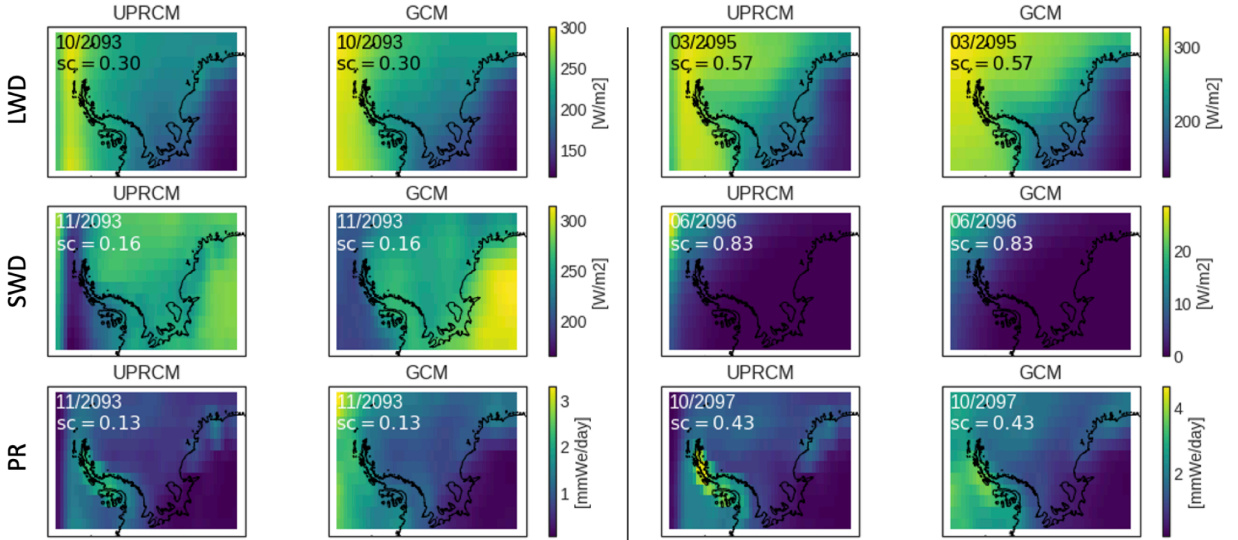
Figure 3.3: Evaluation of predictions from RCM-emulators $\hat{F}_U(\text{UPRCM})$, $\hat{F}_U(\text{GCM})$ and $\hat{F}_G(\text{GCM})$ over target domain \mathcal{E} and test period (2090-2100). At each position, $p \in \mathcal{E}$, the time series of the true SMB are compared to predicted SMB values. Right: box-plot of evaluation metrics, from lower to upper quartile, with a line at the median and triangle at the mean. From top to bottom: Pearson correlation coefficient, Wasserstein distance, and RMSE as in section 2.4.1. Legend: spatial mean (μ), standard deviation (std), minimum (min) and maximum (max) of metrics over \mathcal{E} .



(a) Temporal correlation



(b) Spatial correlation



(c) Spatial patterns for LWD, SWD and PR for example months

Figure 3.4: Temporal (a) and spatial (b, c) correlation between UPRCM and GCM variables given as input to RCM-emulator over target domain \mathcal{E} and test period (2090-2100). (a) Pearson correlation coefficient between UPRCM and GCM time series for each point p in \mathcal{E} . Legend: mean (μ), standard deviation (std) and minimum (min) of correlation values over \mathcal{E} . (b) Spatial correlation between UPRCM and GCM variable maps over \mathcal{E} at each time step. (c) Example of months with lowest (left) and highest (right) spatial correlation (sc) between UPRCM and GCM for long/short-wave downward radiation (LWD/SWD) and precipitation (PR). Chosen months are illustrated with dots on the time series in (b).

Chapter 4

Discussion

4.1 Extending an emulator of near-surface temperature to SMB

In this thesis, we extended the machine learning model that emulated near-surface temperature over the south of France, proposed by Doury et al. [9] to emulate SMB over the Antarctic Peninsula. In doing so, we faced different challenges because SMB is notoriously difficult to observe and represent compared to temperature [27].

Since temperature is a variable present in both the GCM and RCM, it was amidst the low-resolution features given as input to the temperature emulator from Doury et al. [9]. In our case, SMB was only available in the RCM. Thus, our emulator could reconstruct a high-resolution image of SMB given only a limited number of other atmospheric variables that did not include a low-resolution image of SMB. This might explain why the SMB patterns of some regions in the Antarctic Peninsula, especially those with less seasonal patterns (e.g., P3 in Figure 3.2), might be more challenging for the model to reconstruct. We rely on the assumption that our model has a clear physical relationship to learn between our eight large-scale atmospheric variables and local-scale SMB, but this might not be true for all points in the Peninsula. There might be regions where the factors influencing SMB are more complex than the variables we have and where the model might benefit from additional information, such as other large-scale atmospheric variables, ice sheet topography, or past SMB patterns. This could be explored in further work.

For their temperature emulator, Doury et al. had seven atmospheric variables at different pressure levels at their disposition, resulting in 19 input features. Unfortunately, because our chosen dataset had only eight surface-level variables, we could only train our model on eight features. Doury et al. found that while more inputs in their model seemed to improve the quality of the emulated series, having a "cheaper" model with fewer inputs showed satisfying results and could be regarded as a suitable option [9]. While we ideally wished to have more variables to help our emulator reconstruct regional SMB over the Antarctic Peninsula, having a "cheaper" model at least gave a higher computational gain (4 minutes of training compared to 2 hours in [9]).

Ice sheet SMB varies highly across multiple space and time scales [27]. For example, as was discussed in section 3.2, on the Antarctic Peninsula, while some regions have high seasonal amplitude changes in SMB, others have (almost) none. This was not the case for the near-surface temperature emulated in [9], where, even though their target domain (southern France with the Pyrenees) covered different climates, its temperature fluctuated in approximately the same annual amplitude range for all points in their domain. Therefore, opposite to the model developed by Doury et al., our model had to acclimate different scales in SMB values over the Peninsula. For this, we used a normalized RMSE loss and attention mechanisms, but our emulator still better reconstructs regions with high seasonal change. The results hint that NRMSE and attention might not be enough to account for different scales in target values and should be adapted in further work.

4.2 Necessity of the perfect model framework

As outlined in section 2.3, we trained two RCM-emulators separately; one followed the perfect model framework (\hat{F}_U) and another directly trained with large-scale GCM data (\hat{F}_G).

Doury et al. state that the perfect model framework is necessary to learn the RCM downscaling function without any interfering biases between the GCM and RCM. However, to be useful, the RCM-emulator should give accurate reconstructions of SMB when given large-scale GCM variables as input. Otherwise, the model would not be valuable if it needs upscaled RCM features to make predictions. Nevertheless, while the predictions of $\hat{F}_U(\text{GCM})$ follow the correct temporal patterns of the RCM truth, they consistently underestimate the truth and produce a toned-down version of the true SMB time series. We suspect this is because, under the perfect model framework, the emulator cannot learn GCM-RCM inconsistencies during its training. Consequently, when the model is given GCM inputs while trained on UPRCM, inconsistencies are preserved and appear in the local reconstructions. This problem was also found by Doury et al. for their temperature emulator [9].

As shown in section 3.4, we find significant bias and inconsistencies between RCM and GCM variables. For example, the two wind variables showed high temporal discrepancies between large and local-scale simulations. This might happen if there is an offset in the RCM time series compared to the GCM or if the patterns are completely different. Furthermore, we also found spatial inconsistencies for most atmospheric variables, especially for precipitation and downward radiation. Some of this bias might exist for problematic reasons, such as inconsistent forcings or boundary conditions when RCM simulations are computed. We suspect this might be the reason for the streaks visible on the left boundary of RCM images in Figure 3.4c. However, we assume that the rest of the inconsistencies exist because the RCM adjusts the low-resolution simulation for valid reasons i.e., for a more fine-scaled simulation of the physical processes [9, 26, 33, 38, 53].

We trained the second emulator \hat{F}_G directly with large-scale GCM inputs to see whether it could account for the RCM-GCM inconsistencies. $\hat{F}_G(\text{GCM})$ misses some precision compared

to $\hat{F}_U(\text{UPRCM})$ in emulating SMB patterns for regions with minor SMB variations (e.g. P4 in Figure 3.2, SMB smaller than 1.5 mmWe/day). But for the other points, $\hat{F}_G(\text{GCM})$ comes very close to the RCM truth. It is also performing exceedingly well in terms of predicting annual SMB values and is consistently better than $\hat{F}_U(\text{GCM})$.

Under the assumption that most of the GCM-RCM inconsistencies exist for valid reasons, it seems like the perfect model framework may not be necessary to train our RCM-emulator. When trained with low-resolution GCM data, the model can predict SMB values close to the truth, as if it learned the underlying RCM-GCM inconsistencies and dynamics.

4.3 Usability of the RCM-emulator

For our RCM-emulator to be practical, it has to be faster and easier to use than computing an RCM simulation. We showed in section 3.5 that training the emulator and making predictions is very fast, with predictions made nearly instantaneously and training in 4 minutes. The primary time bottle-neck for a user might be getting the correct data and pre-processing it to fit the desired training setting. In the following, we will explore how easy the model is to use.

Transparency

Neural networks have a reputation for being a black box algorithm with a decision process that is hard to understand. This is not different in our case. Because we give all available variables to our model without making any uphill feature selection, it would be interesting to know which physical features carry the most weight in the model's decision. Even though there are new ways to try to open the black box [17, 52], it remains a challenging problem for which we, unfortunately, did not have more time to explore. While traditionally, computing an RCM allows the user to have complete control over things such as variables, forcings, and boundary conditions, using a machine learning emulator offers less transparency.

Randomness and reproducibility

Training a neural network while relying on a GPU creates several sources of randomness [43, 61]. To make our model reproducible, we used a fixed seed, and in this setting, our model always gives the same predictions. Nevertheless, another seed might provide very different results, so to reproduce our results, one must be careful to use the same random settings. To assess the robustness of our RCM-emulator, one could run it several times using different seeds and observe the average results. Due to time constraints, this was omitted.

Specificity of the RCM-emulator

One limitation of our RCM-emulator is that it focused only on reconstructing one target domain, the Antarctic Peninsula. If one wanted to expand it to other areas, one would have to re-train the model on data of this specific region. We suspect that applying our emulator elsewhere would

not work otherwise, as it would be faced with images very different from its training. However, we believe that because the emulator works well for the Peninsula, an area that shows high variability in SMB patterns, it will perform just as well for other domains of Antarctica once it has been correctly trained. In further work, we could also envision training the emulator on a more extensive dataset containing several regions of Antarctica or using techniques like transfer learning and domain adaptation so that the model can reproduce new areas.

Furthermore, if one wanted to extend the emulator to other climate model simulations, one would also need to re-train it for those climate models. One exception might be if they are similar to ACCESS 1.3 and MAR(ACCESS1.3). For example, one could try making predictions with our emulator on another future RCP simulation of MAR(ACCESS1.3), such as RCP4.5.

Data availability

The ACCESS 1.3 GCM data is openly available to the public on the Australian NCI website [37]. We had to ask for the MAR(ACCESS1.3) RCM data from the Geo-science Institute of the University of Grenoble. Therefore, while acquiring GCM data is easy, getting access to RCM simulations might be more challenging for a lambda user.

Code availability

The trained RCM-emulators, notebooks, and python scripts for pre-processing and machine learning are available on our GitHub repository [16]. For any questions or additional information about the code, email: marijn.vandermeer@bluewin.ch.

Chapter 5

Conclusion

In this thesis, we built a machine learning model called the RCM-emulator that reconstructs local-scale surface mass balance (SMB) values over the Antarctic Peninsula by learning the downscaling function of a regional climate model (RCM). This means that our model, when given large-scale (low-resolution) atmospheric variables from a global climate model (GCM), can reconstruct a local-scale (high-resolution) image of SMB from an RCM. Compared to running an RCM, the emulator is designed to be computationally faster.

Our model is an extension of a U-Net model proposed by Doury et al. that emulated near-surface temperature over the south of France [9]. We extended their U-net by adding convolutional block attention mechanisms in the encoder of the U-Net and replaced traditional convolutional operations with depthwise-separable convolutions. Furthermore, to account for the different scales of SMB values across the Antarctic Peninsula, we used a normalized RMSE.

For training our model, we proposed two frameworks. The first is a perfect model framework where the emulator received low-resolution inputs from the RCM upscaled to GCM resolution instead of the GCM. For this, we created large-scale features from RCM with interpolation (UPRCM). This setting was designed so that the model learned the downscaling function undisturbed by potential GCM-RCM inconsistencies and biases. In the second framework, we trained the model on large-scale features directly sourced from the GCM to evaluate whether it could make accurate predictions despite those inconsistencies.

Training our model takes under four minutes on a GPU, and predictions are almost instantaneous, but data gathering and pre-processing can be time-consuming. However, our emulator is significantly faster than an RCM simulation, which can take several weeks on a supercomputer.

We evaluated the first emulator, trained in the perfect model setting, on large-scale UPRCM features and, in a second step, directly with variables from the GCM. While the emulator evaluated on UPRCM features can almost perfectly reproduce the high-resolution SMB truth, predictions made with GCM features consistently underestimate it. This is not surprising as we find high

spatial and temporal bias while exploring the consistency between GCM and RCM features. Because the perfect model framework does not allow the emulator to learn large-scale GCM-RCM inconsistencies, they are conserved when making predictions with GCM variables, leading to underestimation.

The second emulator, trained directly on GCM features, could reproduce detailed high-resolution SMB maps. The model can make correct annual SMB predictions and reconstruct the temporal patterns of individual SMB time series and global spatial structures over the Antarctic Peninsula. Its only limitation is producing an accurate representation of points in dry climates where the normalized RMSE loss is insufficient to account for the small SMB values. In further work, one could focus on these climates to improve the emulator's predictions. Assuming that RCM-GCM inconsistencies exist for a valid reason and that we wish the emulator to account for them, our results indicate that a perfect model framework might not be needed and that our model can learn them.

In conclusion, we built a machine learning RCM-emulator that can make fast and fine-scaled SMB reproductions over the Antarctic Peninsula from GCM data. Therefore, this emulator can be an interesting tool for providing low-cost local-scale information on the evolution of ice sheets during climate change.

Bibliography

- [1] C. Agosta, X. Fettweis, and R. Datta. “Evaluation of the CMIP5 models in the aim of regional modelling of the Antarctic surface mass balance”. In: *The Cryosphere* 9.6 (2015), pp. 2311–2321. DOI: 10.5194/tc-9-2311-2015. URL: <https://tc.copernicus.org/articles/9/2311/2015/>.
- [2] Cécile Agosta, Vincent Favier, Gerhard Krinner, Hubert Gallée, and Christophe Genton. *Evolution of the Antarctic surface mass balance by high-resolution downscaling and impact on sea-level change for the next centuries*. eng. 2012.
- [3] *Antarctic Ice Sheet*. https://en.wikipedia.org/wiki/Antarctic_ice_sheet. Accessed: 2020-06-27.
- [4] *Antarctic Peninsula*. https://en.wikipedia.org/wiki/Antarctic_Peninsula. Accessed: 2020-06-27.
- [5] *Attention U-Net A detailed explanation of the Attention U-Net*. <https://towardsdatascience.com/a-detailed-explanation-of-the-attention-u-net-b371a5590831>. Accessed: 2020-07-10.
- [6] Daohua Bi, Martin Dix, S. Marsland, Siobhan O’Farrell, Harun Rashid, P. Uotila, Hirst, Eva Kowalczyk, Golebiewski, Arnold Sullivan, Yizhong Yan, Hannah, Charmaine Franklin, Zhongmin Sun, Vohralik, Watterson, R. Fiedler, Mark Collier, and Kamal Puri. “The ACCESS coupled model: Description, control climate and evaluation”. In: *Australian Meteorological and Oceanographic Journal* 63 (Dec. 2012), pp. 41–64. DOI: 10.22499/2.6301.004.
- [7] J. Bolibar, A. Rabatel, I. Gouttevin, C. Galiez, T. Condom, and E. Sauquet. “Deep learning applied to glacier evolution modelling”. In: *The Cryosphere* 14.2 (2020), pp. 565–584. DOI: 10.5194/tc-14-565-2020. URL: <https://tc.copernicus.org/articles/14/565/2020/>.
- [8] Mark Collier and Peter Uhe. “The Centre for Australian Weather and Climate Research CMIP5 datasets from the ACCESS1.0 and ACCESS1.3 coupled climate models”. In: (Dec. 2012).
- [9] Antoine Doury, Samuel Somot, Sébastien Gadat, Aurélien Ribes, and Lola Corre. “Regional Climate Model Emulator based on Deep Learning: Concept and First Evaluation of a Novel Hybrid Downscaling Approach”. In: (July 2021). DOI: 10.21203/rs.3.rs-725819/v1.
- [10] S. Draggan. *Antarctic Peninsula*. Ed. by National Council for Science and the Environment. Cleveland: Encyclopedia of Earth, 2009.

- [11] Peter Fretwell, Hamish Pritchard, David Vaughan, Jonathan Bamber, N. Barrand, Robin Bell, Cesidio Bianchi, R. Bingham, D. Blankenship, Gino Casassa, G. Catania, D. Callens, H. Conway, Alison Cook, H. Corr, Detlef Damaske, Volkmar Damm, Fausto Ferraccioli, Rene Forsberg, and Achille Zirizzotti. *Bedmap2: Improved ice bed, surface and thickness datasets for Antarctica*. Feb. 2013. DOI: 10.5194/tcd-6-4305-2012.
- [12] Charlie Frogner, Chiyuan Zhang, Hossein Mobahi, Mauricio Araya-Polo, and Tomaso A. Poggio. “Learning with a Wasserstein Loss”. In: *CoRR* abs/1506.05439 (2015). arXiv: 1506.05439. URL: <http://arxiv.org/abs/1506.05439>.
- [13] Jeremy Fyke, Olga Sergienko, Marcus Löfverström, Stephen Price, and Jan T. M. Lenaerts. “An Overview of Interactions and Feedbacks Between Ice Sheets and the Earth System”. In: *Reviews of Geophysics* 56.2 (2018), pp. 361–408. DOI: <https://doi.org/10.1029/2018RG000600>. eprint: <https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2018RG000600>. URL: <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2018RG000600>.
- [14] Hubert Gallée, Cécile Agosta, Luc Gentil, Vincent Favier, and Gerhard Krinner. “A Downscaling Approach Toward High-Resolution Surface Mass Balance Over Antarctica”. In: *Surv Geophys* 32 (Sept. 2011), pp. 507–518. DOI: 10.1007/s10712-011-9125-3.
- [15] M. Geyer, D. Salas Y Melia, E. Brun, and M. Dumont. “The Greenland ice sheet: modelling the surface mass balance from GCM output with a new statistical downscaling technique”. In: *The Cryosphere Discussions* 7 (2013), pp. 3163–3207. DOI: 10.5194/tcd-7-3163-2013. URL: <https://tc.copernicus.org/preprints/7/3163/2013/>.
- [16] *GitHub with code*. <https://github.com/marvande/master-thesis>. Accessed: 2020-07-05.
- [17] Riccardo Guidotti, Anna Monreale, Franco Turini, Dino Pedreschi, and Fosca Giannotti. “A Survey Of Methods For Explaining Black Box Models”. In: *CoRR* abs/1802.01933 (2018). arXiv: 1802.01933. URL: <http://arxiv.org/abs/1802.01933>.
- [18] Charlotte Hartley. *Antarctic ice sheet collapse could add 3 meters to sea-level rise*. July 2020. URL: <https://www.science.org/content/article/antarctic-ice-sheet-collapse-could-add-3-metres-sea-level-rise>.
- [19] *How U-net works*. <https://developers.arcgis.com/python/guide/how-unet-works/>. Accessed: 2020-07-11.
- [20] Jeremy Howard et al. *fastai*. <https://github.com/fastai/fastai>. 2018.
- [21] Z. Hu, P. Kuipers Munneke, S. Lhermitte, M. Izeboud, and M. van den Broeke. “Improving surface melt estimation over the Antarctic Ice Sheet using deep learning: a proof of concept over the Larsen Ice Shelf”. In: *The Cryosphere* 15.12 (2021), pp. 5639–5658. DOI: 10.5194/tc-15-5639-2021. URL: <https://tc.copernicus.org/articles/15/5639/2021/>.
- [22] *Ice sheet*. https://en.wikipedia.org/wiki/Ice_sheet. Accessed: 2020-07-27.
- [23] *Ice sheet mass balance and sea level (ismass)*. <https://www.scar.org/science/ismass/home/>. Accessed: 2020-07-07.

- [24] C. Kittel, C. Amory, C. Agosta, N. C. Jourdain, S. Hofer, A. Delhasse, S. Doutreloup, P.-V. Huot, C. Lang, T. Fichet, and X. Fettweis. “Diverging future surface mass balance between the Antarctic ice shelves and grounded ice sheet”. In: *The Cryosphere* 15.3 (2021), pp. 1215–1236. DOI: 10.5194/tc-15-1215-2021. URL: <https://tc.copernicus.org/articles/15/1215/2021/>.
- [25] Remko Klaver, Rein Haarsma, Pier Luigi Vidale, and Wilco Hazeleger. “Effective resolution in high resolution global atmospheric models for climate studies”. In: *Atmospheric Science Letters* 21.4 (2020), e952. DOI: <https://doi.org/10.1002/asl.952>. eprint: <https://rmets.onlinelibrary.wiley.com/doi/pdf/10.1002/asl.952>. URL: <https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/asl.952>.
- [26] René Laprise, R. Elía, Daniel Caya, Sébastien Biner, Philippe Lucas-Picher, Emilia Diaconescu, Martin Leduc, Adelina Alexandru, Leo Separovic, and Canadian Diagnostics. “Challenging some tenets of Regional Climate Modelling”. In: *Meteorology and Atmospheric Physics* 100 (Jan. 2008), pp. 3–22. DOI: 10.1007/s00703-008-0292-9.
- [27] Jan T. M. Lenaerts, Brooke Medley, Michiel R. van den Broeke, and Bert Wouters. “Observing and Modeling Ice Sheet Surface Mass Balance”. In: *Reviews of Geophysics* 57.2 (2019), pp. 376–420. DOI: <https://doi.org/10.1029/2018RG000622>. eprint: <https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2018RG000622>. URL: <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2018RG000622>.
- [28] Jan T. M. Lenaerts, Miren Vizcaino, Jeremy Fyke, Leo van Kampenhout, and Michiel R. van den Broeke. “Present-day and future Antarctic ice sheet climate and surface mass balance in the Community Earth System Model”. English. In: *Climate Dynamics* 47 (Sept. 2016). Report, pp. 1367+. ISSN: 09307575. URL: <https://link.gale.com/apps/doc/A470431762/AONE?u=anon~6373b9ed&sid=googleScholar&xid=35ff5624>.
- [29] Hao Li, Zheng Xu, Gavin Taylor, and Tom Goldstein. “Visualizing the Loss Landscape of Neural Nets”. In: *CoRR* abs/1712.09913 (2017). arXiv: 1712.09913. URL: <http://arxiv.org/abs/1712.09913>.
- [30] Lin Liu. “A Review of Deep Learning for Cryospheric Studies”. In: *Deep Learning for the Earth Sciences*. John Wiley Sons, Ltd, 2021. Chap. 17, pp. 258–268. ISBN: 9781119646181. DOI: <https://doi.org/10.1002/9781119646181.ch17>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781119646181.ch17>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119646181.ch17>.
- [31] *MAR Modele Atmospherique Regional*. https://mar.cnrs.fr/index.php?option_smdi=accueil&idm=1. Accessed: 2020-06-27.
- [32] S. Marshak. *Essentials of Geology*. Essentials of Geology v. 2. W.W. Norton, 2003. ISBN: 9780393924114. URL: <https://books.google.com/gi/books?id=BCTnzAEACAAJ>.
- [33] Vasubandhu Misra. “Addressing the Issue of Systematic Errors in a Regional Climate Model”. In: *Journal of Climate* 20.5 (2007), pp. 801–818. DOI: 10.1175/JCLI4037.1. URL: <https://journals.ametsoc.org/view/journals/clim/20/5/jcli4037.1.xml>.

- [34] Mathieu Morlighem, E. Rignot, Tobias Binder, Donald Blankenship, Reinhard Drews, Graeme Eagles, Olaf Eisen, Fausto Ferraccioli, René Forsberg, Peter Fretwell, Vikram Goel, Jamin Greenbaum, Gudmundur Gudmundsson, Jingxue Guo, Veit Helm, Coen Hofstede, Ian Howat, Angelika Humbert, Wilfried Jokar, and Duncan Young. “Deep glacial troughs and stabilizing ridges unveiled beneath the margins of the Antarctic ice sheet”. In: *Nature Geoscience* 13 (Feb. 2020), pp. 1–6. DOI: 10.1038/s41561-019-0510-8.
- [35] Richard H. Moss, Jae A. Edmonds, Kathy A. Hibbard, Martin R. Manning, Steven K. Rose, Detlef P. van Vuuren, Timothy R. Carter, Seita Emori, Mikiko Kainuma, Tom Kram, Gerald A. Meehl, John F. B. Mitchell, Nebojsa Nakicenovic, Keywan Riahi, Steven J. Smith, Ronald J. Stouffer, Allison M. Thomson, John P. Weyant, and Thomas J. Wilbanks. “The next generation of scenarios for climate change research and assessment”. In: *Nature* 463.7282 (Feb. 2010), pp. 747–756. ISSN: 1476-4687. DOI: 10.1038/nature08823. URL: <https://doi.org/10.1038/nature08823>.
- [36] R. Mottram, N. Hansen, C. Kittel, J. M. van Wessem, C. Agosta, C. Amory, F. Boberg, W. J. van de Berg, X. Fettweis, A. Gossart, N. P. M. van Lipzig, E. van Meijgaard, A. Orr, T. Phillips, S. Webster, S. B. Simonsen, and N. Souverijns. “What is the surface mass balance of Antarctica? An intercomparison of regional climate model estimates”. In: *The Cryosphere* 15.8 (2021), pp. 3751–3784. DOI: 10.5194/tc-15-3751-2021. URL: <https://tc.copernicus.org/articles/15/3751/2021/>.
- [37] *NCL*. <https://esgf.nci.org.au/search/esgf-nci/>. Accessed: 2020-06-27.
- [38] M. Noguer, R. Jones, and J. Murphy. “Sources of systematic errors in the climatology of a regional climate model over Europe”. In: *Climate Dynamics* 14.10 (Aug. 1998), pp. 691–712. ISSN: 1432-0894. DOI: 10.1007/s003820050249. URL: <https://doi.org/10.1007/s003820050249>.
- [39] Ozan Oktay, Jo Schlemper, Loïc Le Folgoc, Matthew C. H. Lee, Mattias P. Heinrich, Kazunari Misawa, Kensaku Mori, Steven G. McDonagh, Nils Y. Hammerla, Bernhard Kainz, Ben Glocker, and Daniel Rueckert. “Attention U-Net: Learning Where to Look for the Pancreas”. In: *CoRR* abs/1804.03999 (2018). arXiv: 1804.03999. URL: <http://arxiv.org/abs/1804.03999>.
- [40] *Pearson correlation coefficient*. https://en.wikipedia.org/wiki/Pearson_correlation_coefficient. Accessed: 2020-07-27.
- [41] *Quick Facts on Ice Sheets*. <https://nsidc.org/cryosphere/quickfacts/icesheets.html>. Accessed: 2020-07-05.
- [42] *Quick Facts on Ice Shelves*. <https://nsidc.org/cryosphere/quickfacts/iceshelves.html>. Accessed: 2020-07-21.
- [43] *Reproducibility*. <https://pytorch.org/docs/stable/notes/randomness.html>. Accessed: 2020-07-25.

- [44] E. Rignot, I. Velicogna, M. R. van den Broeke, A. Monaghan, and J. T. M. Lenaerts. “Acceleration of the contribution of the Greenland and Antarctic ice sheets to sea level rise”. In: *Geophysical Research Letters* 38.5 (2011). DOI: <https://doi.org/10.1029/2011GL046583>. eprint: <https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2011GL046583>. URL: <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2011GL046583>.
- [45] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015. DOI: 10.48550/ARXIV.1505.04597. URL: <https://arxiv.org/abs/1505.04597>.
- [46] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-Net: Convolutional Networks for Biomedical Image Segmentation”. In: *CoRR* abs/1505.04597 (2015). arXiv: 1505.04597. URL: <http://arxiv.org/abs/1505.04597>.
- [47] Emilia Sanchez-Gomez, S. Somot, and M. Déqué. “Ability of an ensemble of regional climate models to reproduce weather regimes over Europe-Atlantic during the period 1961-2000”. In: *Climate Dynamics* 33.5 (Oct. 2009), pp. 723–736. DOI: 10.1007/s00382-008-0502-7.
- [48] Emilia Sanchez-Gomez and Samuel Somot. “Impact of the internal variability on the cyclone tracks simulated by a regional climate model over the Med-CORDEX domain”. In: *Climate Dynamics* 51.3 (Aug. 2018), pp. 1005–1021. DOI: 10.1007/s00382-016-3394-y.
- [49] R. Sellevold, L. van Kampenhout, J. T. M. Lenaerts, B. Noël, W. H. Lipscomb, and M. Vizcaino. “Surface mass balance downscaling through elevation classes in an Earth system model: application to the Greenland ice sheet”. In: *The Cryosphere* 13.12 (2019), pp. 3193–3208. DOI: 10.5194/tc-13-3193-2019. URL: <https://tc.copernicus.org/articles/13/3193/2019/>.
- [50] H. Seroussi, S. Nowicki, A. J. Payne, H. Goelzer, W. H. Lipscomb, A. Abe-Ouchi, C. Agosta, T. Albrecht, X. Asay-Davis, A. Barthel, R. Calov, R. Cullather, C. Dumas, B. K. Galton-Fenzi, R. Gladstone, N. R. Golledge, J. M. Gregory, R. Greve, T. Hattermann, M. J. Hoffman, A. Humbert, P. Huybrechts, N. C. Jourdain, T. Kleiner, E. Larour, G. R. Leguy, D. P. Lowry, C. M. Little, M. Morlighem, F. Pattyn, T. Pelle, S. F. Price, A. Quiquet, R. Reese, N.-J. Schlegel, A. Shepherd, E. Simon, R. S. Smith, F. Straneo, S. Sun, L. D. Trusel, J. Van Breedam, R. S. W. van de Wal, R. Winkelmann, C. Zhao, T. Zhang, and T. Zwinger. “ISMIP6 Antarctica: a multi-model ensemble of the Antarctic ice sheet evolution over the 21st century”. In: *The Cryosphere* 14.9 (2020), pp. 3033–3070. DOI: 10.5194/tc-14-3033-2020. URL: <https://tc.copernicus.org/articles/14/3033/2020/>.
- [51] Andrew Shepherd, Lin Gilbert, Alan S. Muir, Hannes Konrad, Malcolm McMillan, Thomas Slater, Kate H. Briggs, Aud V. Sundal, Anna E. Hogg, and Marcus E. Engdahl. “Trends in Antarctic Ice Sheet Elevation and Mass”. In: *Geophysical Research Letters* 46.14 (2019), pp. 8174–8183. DOI: <https://doi.org/10.1029/2019GL082182>. eprint: <https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2019GL082182>. URL: <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2019GL082182>.

- [52] Ravid Shwartz-Ziv and Naftali Tishby. “Opening the Black Box of Deep Neural Networks via Information”. In: *CoRR* abs/1703.00810 (2017). arXiv: 1703.00810. URL: <http://arxiv.org/abs/1703.00810>.
- [53] Silje Lund Sørland, Christoph Schär, Daniel Lüthi, and Erik Kjellström. “Bias patterns and climate change signals in GCM-RCM model chains”. In: *Environmental Research Letters* 13.7 (July 2018), p. 074017. DOI: 10.1088/1748-9326/aacc77. URL: <https://doi.org/10.1088/1748-9326/aacc77>.
- [54] Karl E. Taylor, Ronald J. Stouffer, and Gerald A. Meehl. “An Overview of CMIP5 and the Experiment Design”. In: *Bulletin of the American Meteorological Society* 93.4 (2012), pp. 485–498. DOI: 10.1175/BAMS-D-11-00094.1. URL: <https://journals.ametsoc.org/view/journals/bams/93/4/bams-d-11-00094.1.xml>.
- [55] Kevin Trebing and Siamak Mehrkanoon. “SmaAt-UNet: Precipitation Nowcasting using a Small Attention-UNet Architecture”. In: *CoRR* abs/2007.04417 (2020). arXiv: 2007.04417. URL: <https://arxiv.org/abs/2007.04417>.
- [56] É. Vignon, M.-L. Roussel, I. V. Gorodetskaya, C. Genthon, and A. Berne. “Present and Future of Rainfall in Antarctica”. In: *Geophysical Research Letters* 48.8 (2021). e2020GL092281. DOI: <https://doi.org/10.1029/2020GL092281>. eprint: <https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2020GL092281>. URL: <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2020GL092281>.
- [57] Cédric Villani. “Optimal Transport: Old and New”. In: 2008.
- [58] *Weather nowcasting*. [https://en.wikipedia.org/wiki/Nowcasting_\(meteorology\)#cite_note-OMM-1](https://en.wikipedia.org/wiki/Nowcasting_(meteorology)#cite_note-OMM-1). Accessed: 2020-07-08.
- [59] *Why are glaciers and sea ice melting?* <https://www.worldwildlife.org/pages/why-are-glaciers-and-sea-ice-melting>. Accessed: 2020-07-27.
- [60] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. “CBAM: Convolutional Block Attention Module”. In: *CoRR* abs/1807.06521 (2018). arXiv: 1807.06521. URL: <http://arxiv.org/abs/1807.06521>.
- [61] Donglin Zhuang, Xingyao Zhang, Shuaiwen Leon Song, and Sara Hooker. “Randomness In Neural Network Training: Characterizing The Impact of Tooling”. In: *CoRR* abs/2106.11872 (2021). arXiv: 2106.11872. URL: <https://arxiv.org/abs/2106.11872>.

Appendix A

Data processing

A.1 Pre-processing of the GCM

All pre-processing of the RCM and GCM data was done on Pangeo, a community platform for Big Data geoscience. First, we downloaded the ACCESS 1.3 GCM data from the Australian NCI website [37]. From their database, we chose the dataset with atmospheric variables (Amon) from the CMIP5 and r1i1p1 ensemble. As a time frame, we decided to use the historical and future RCP8.5 simulations, which are monthly mean aggregations of daily values. Now, the data can also be directly downloaded with the wget script on our GitHub.

From this GCM dataset, we chose the eight variables as seen in Table 2.1. To restrict the dataset size and because we were only interested in the region of Antarctica, we cut the data so that its latitude was between -40 and -90 °. Because the GCM is in latitude-longitude coordinates, but the RCM is in polar stereographic coordinates, we reprojected the GCM to the same projection system as the RCM. We did this by upsampling the RCM stereographic grid to cover approximately the same resolution as the GCM (68×206 km) and then interpolated the GCM on that grid. We changed the units for the following variables to coincide with the RCM:

- Precipitation (PR): from $\text{kgm}^{-2}\text{s}^{-1}$ to mmWe/day
- Surface pressure (SP): from pa to hpa
- Temperature (TT): from K to C
- Specific humidity (QQP): from g/g to g/Kg

Finally, the GCM variables were smoothed by a 3×3 moving average filter to follow the same pre-processing procedure as for the temperature emulator in [9] (c.f. subsection 2.2.2).

A.2 Pre-processing of the RCM

We acquired the MAR(ACCESS 1.3) RCM data from the Geoscience Institute of the University of Grenoble. Because ACCESS 1.3 was in monthly frequency, we did a monthly mean aggregation on the RCM. We also changed the spatial units from km to m to be in accordance with GCM units.

Variables like the winds, temperature and specific humidity were initially provided for seven pressure levels (200, 500, 600, 700, 800, 850, and 925 hpa) while we needed surface-level values to coincide with the GCM. Each pressure level had NaN values where it intersected with the surface. So, for each point p in the RCM domain, we took the last non-NaN value on the highest possible pressure level as the surface value (Figure A.1). Furthermore, RCM wind variables were given as x and y-components, while GCM winds were northward and eastward. Therefore RCM wind components needed to be reprojected to become north/eastward winds. First we calculated a grid that gave the latitude (lat) and longitude (lon) coordinates for each point $(i, j) \in X, Y \subset \mathcal{E}$ in the RCM polar stereographic grid \mathcal{E} . Then, for each point (i, j) we applied Algorithm 1.

Algorithm 1 Transformation of wind x/y-components into north/eastward

```

1: GEddxx = 90
2:  $\Delta\phi = 90 - \text{GEddxx}$ 
3:  $dr = \pi/180$ 
4: for  $i, j \in X, Y$  do
5:    $\phi \leftarrow -1 * dr * (\text{lon}[i, j] + \Delta\phi)$ 
6:    $cphi \leftarrow \cos(-\phi)$ 
7:    $sphi \leftarrow \sin(-\phi)$ 
8:    $\text{windEast}[i, j] \leftarrow cphi * \text{windX}[i, j] - sphi * \text{windY}[i, j]$        $\triangleright$  Eastward wind component
9:    $\text{windNorth}[i, j] \leftarrow sphi * \text{windX}[i, j] + cphi * \text{windY}[i, j]$    $\triangleright$  Northward wind component
10: end for
[1]
```

Finally, to create GCM-like low-resolution UPRCM features from the RCM, we reprojected the RCM on the GCM grid by interpolation. Then, we used the same moving average filter for the GCM to smooth the data (Figure A.2). Finally, because there is no precipitation variable in the RCM, we created one by adding the snowfall and rainfall variables.

$$\text{PR} = \text{SF} + \text{RF}$$

A.3 Architecture selection

We measured the impact of adding convolutional block attention mechanisms (CBAM) and depth-wise convolutions (DSC) to the U-Net. We compared the performance of the U-Net with and without CBAM and DSC modules and measured the evaluation metrics from subsection 2.4.1 (Table A.1). We calculated the percentage of improvement in metrics when going from the basic

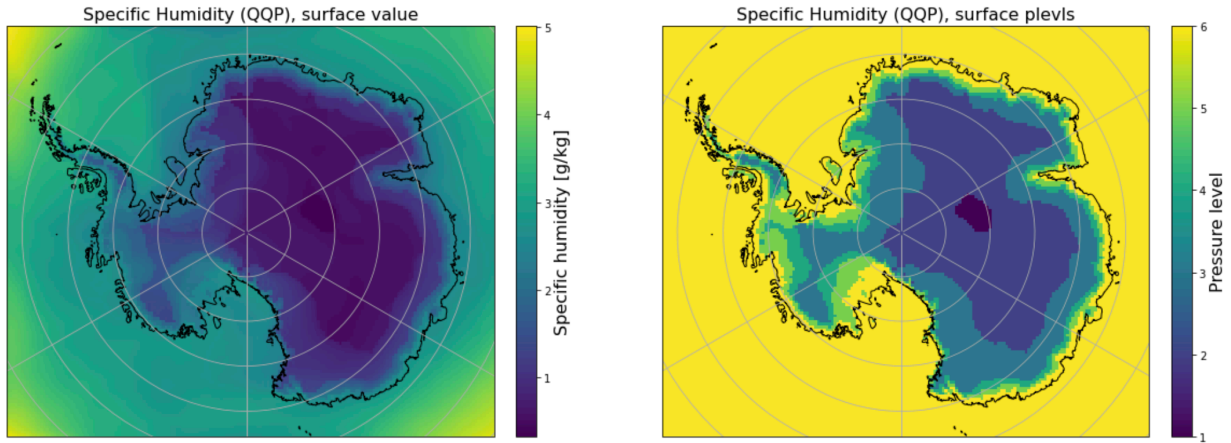


Figure A.1: Illustration of the process of changing the specific humidity (QQP) variable from values at different pressure levels to surface level values (left). Right: pressure at surface over Antarctica.

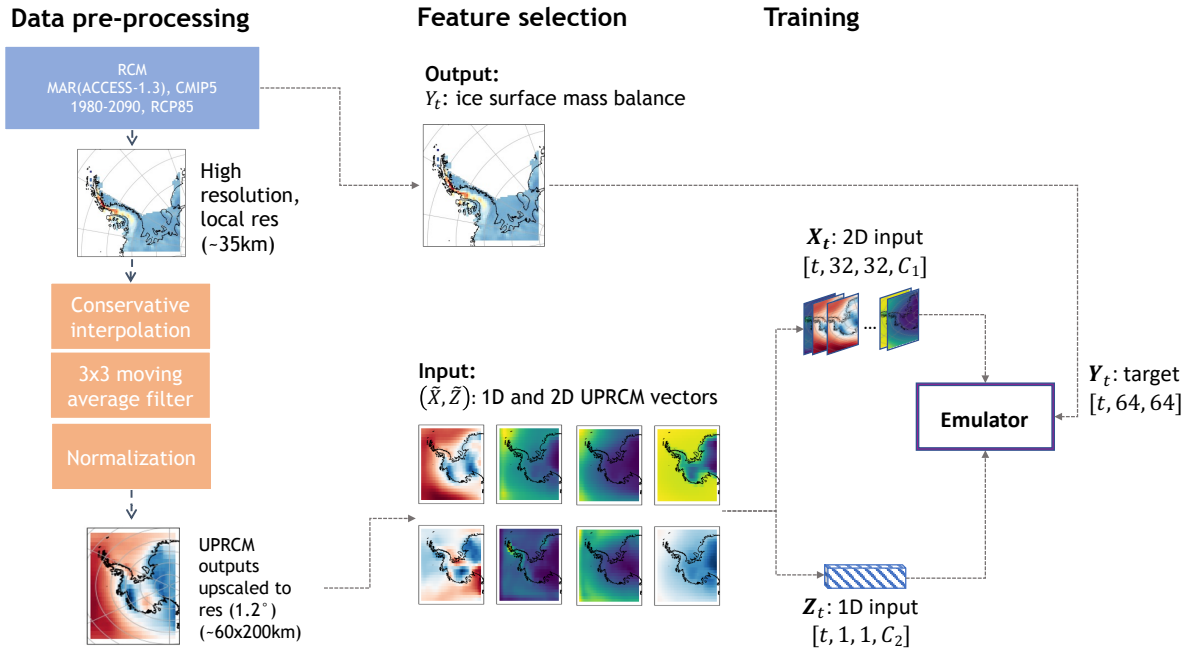


Figure A.2: RCM data flow to create UPRCM data.

RCM-emulator to the U-Net with CBAM and DSC for $\hat{F}_U(\text{UPRCM})$ and $\hat{F}_G(\text{GCM})$. The greatest metric improvement was for the Wasserstein distance, which decreased by 26% and 28%. CBAM and DSC had a smaller impact on the rest of the metrics. RMSE decreased by 9% and 7%, and NRMSE by 10% and 7%. The Pearson correlation coefficient was slightly better for the basic model with a -1% and -2%. Regarding computational time, training with CBAM and DSC gave a 3 minutes gain. Although the improvement in predictions was not that impressive when adding CBAM and DSC, we still decided to use them in our architecture because of the impact on the Wasserstein distance and the smaller architecture.

Table A.1

	LR	Without Attention			Attention - $\hat{F}_U(\text{UPRCM})$			Attention - $\hat{F}_G(\text{GCM})$		
		0.1	0.01	0.005	0.1	0.01	0.005	0.1	0.01	0.005
RMSE	min	0.1711	0.1439	0.1494	0.1686	0.1483	0.1389	0.1687	0.1432	0.1463
	mean	1.2909	1.1904	1.2475	1.1919	1.0931	1.0734	1.1982	1.0978	1.1054
	median	0.8682	0.7428	0.7454	0.7629	0.6981	0.6904	0.7156	0.6967	0.7081
	max	7.0078	6.1929	8.5094	7.1384	5.9493	6.2150	8.7846	6.5332	7.3255
NRMSE	min	0.0023	0.0019	0.0019	0.0023	0.0020	0.0019	0.0023	0.0019	0.0020
	mean	0.0172	0.0159	0.0167	0.0159	0.0146	0.0143	0.0160	0.0147	0.0148
	median	0.0116	0.0099	0.0099	0.0102	0.00932	0.0092	0.0096	0.0093	0.0095
	max	0.0936	0.0827	0.1136	0.0953	0.0794	0.0830	0.1173	0.0872	0.0978
Wasserstein	min	0.0782	0.0262	0.0283	0.0508	0.0379	0.0269	0.0417	0.0386	0.0334
	mean	0.7013	0.5653	0.4979	0.5722	0.3742	0.3663	0.4897	0.3575	0.3585
	median	0.5559	0.3284	0.2572	0.4127	0.2595	0.2680	0.3322	0.2554	0.2392
	max	3.4777	3.0174	4.5005	3.1355	2.3569	2.5277	5.0815	2.6017	2.2302
Pearson	min	0.0421	-0.0038	0.0858	-0.1127	0.0720	0.0725	0.0009	0.0493	0.0469
	mean	0.5616	0.5988	0.5749	0.5263	0.5818	0.5893	0.5789	0.5824	0.5879
	median	0.5651	0.5839	0.5685	0.536743	0.563422	0.5733	0.5715	0.5625	0.5689
	max	0.9641	0.9662	0.9499	0.9742	0.9766	0.9715	0.9623	0.9601	0.9794

Table A.1. Comparison of three models according to four evaluation metrics calculated between final SMB predictions and RCM-truth (subsection 2.4.1). From top to bottom: RMSE, normalized RMSE (NRMSE), Wasserstein distance, and Pearson correlation coefficient. From left to right: U-Net without attention mechanisms and DSC, U-Net trained on UPRCM and evaluated on UPRCM ($\hat{F}_U(\text{UPRCM})$), and U-Net trained on GCM and evaluated on GCM ($\hat{F}_G(\text{GCM})$). All models were trained for 50 epochs with early stopping, NRSME loss, and three initial learning rates: 0.1, 0.01, and 0.005.