Deep Learning Regional Climate Model Emulators: a comparison of two downscaling training frameworks

Marijn van der Meer[1,2,3], Sophie de Roda Husman[3], Stef Lhermitte[3,4]

[1]Laboratory of Hydraulics, Hydrology, and Glaciology (VAW), ETH Zürich, Zurich, Switzerland
[2]Swiss Federal Institute for Forest, Snow, and Landscape Research (WSL), Birmensdorf, Switzerland
[3]Department of Geoscience & Remote Sensing, Delft University of Technology, Delft, the Netherlands
[4]Department of Earth & Environmental Sciences, KU Leuven, Leuven, Belgium

Key Points:

- We developed a computationally fast machine learning emulator to downscale a global climate model to regional resolution.
- The emulator reproduces regional high-resolution surface mass balance predictions over the Antarctic Peninsula from a global climate model.
- The imperfect model framework outperforms the perfect model framework in the application success of the deep learning emulator.

Corresponding author: Marijn van der Meer, vandermeer@vaw.baug.ethz.ch

Abstract

Regional climate models (RCMs) have a high computational cost due to their higher spatial resolution compared to global climate models (GCMs). Therefore, various downscaling approaches have been developed as a surrogate for the dynamical downscaling of GCMs. This study assesses the potential of using a cost-efficient machine learning alternative to dynamical downscaling by using the example case study of emulating surface mass balance (SMB) over the Antarctic Peninsula. More specifically, we determine the impact of the training framework by comparing two training scenarios: (1) a perfect and (2) an imperfect model framework. In the perfect model framework, the RCM-emulator learns only the downscaling function; therefore, it was trained with upscaled RCM features at GCM resolution. This emulator accurately reproduced SMB when evaluated on upscaled RCM features, but its predictions on GCM data conserved RCM-GCM inconsistencies and led to underestimation. In the imperfect model framework, the RCM-emulator was trained with GCM features and downscaled the GCM while exposed to RCM-GCM inconsistencies. This emulator predicted SMB close to the truth, showing it learned the underlying inconsistencies and dynamics. Our results suggest that a deep learning RCM-emulator can learn the proper GCM to RCM downscaling function while working directly with GCM data. Furthermore, the RCM-emulator presents a significant computational gain compared to an RCM simulation. We conclude that machine learning emulators can be applied to produce fast and fine-scaled predictions of RCM simulations from GCM data.

Plain Language Summary

Over the last century, climate scientists have tried to deepen their understanding of the behavior of climate processes through two types of computer climate simulations: global (GCMs) and regional (RCMs) climate models. GCMs cover the whole planet but do not contain fine spatial details, whereas RCMs provide highly detailed information but cover small areas and come at a high additional computational cost. Therefore, we imitated regional models from global models using machine learning to facilitate their faster development.

To test our machine learning framework, we focused on the Antarctic Peninsula and aimed to reproduce the surface mass balance of ice formation and loss. We trained our model to learn the relationship between a group of low-resolution images of climate variables and a high-resolution image from surface mass balance images in the same region.

Our results show that the machine learning model is fast and could recreate regional images of ice sheet processes from global data almost identical to existing on-site observations. This is a good start for further usage of machine learning emulators.

In conclusion, we can make fast and detailed reproductions of surface mass balance processes at regional scales from globally accessible climate data using machine learning.

1 Introduction

Numerous climate models have been developed to understand and predict the behavior of different climate phenomena. Climate models simulate climate variables in different parts of the world over time and space. Their complexity is a compromise between computational costs, the resolution of pixels, and the domain covered (Doury et al., 2022). Depending on the spatial resolution and domain, two types of models are typically defined: global (GCMs) and regional climate models (RCMs).

GCMs are simulations that cover the entire world. Since they have global domains, their spatial resolution is typically low (50-300 km), which complicates capturing the effects of local forcings and the fine-scale representation of heterogeneous surface regions (Kittel

et al., 2021; Seroussi et al., 2020). On the other hand, RCMs are a dynamic downscaling of GCMs, and their driving data is typically derived from GCMs directly (e.g., Giorgi and Bates (1989), Box and Rinke (2003), Fettweis et al. (2017), and Kotlarski et al. (2015)). RCMs have a higher spatial resolution (1-50 km) than GCMs but cover a limited globe area. Due to the RCMs' higher spatial resolution, they come with a high computational cost and time (usually several weeks on supercomputers). Furthermore, while RCMs eliminate most of the low-resolution bias from the GCM inconsistencies, they can still misrepresent key small-scale processes due to their coarse resolutions (Sellevold et al., 2019).

This study explores the potential of using a more cost-efficient machine learning alternative to dynamical downscaling by using the example case study of emulating surface mass balance (SMB) over the Antarctic Peninsula. SMB is the net balance between inputs and outputs of mass on top of the ice sheet (Lenaerts et al., 2019). It is a key input to essential climate variables when observing the Antarctic Ice Sheet and is typically obtained from RCMs after dynamical downscaling. Changes in the surface mass of Antarctica impact the global mass balance and, therefore, the ice dynamics and sea-level rise (Mottram et al., 2021). Currently, however, it is challenging to model SMB accurately because it varies strongly across multiple scales of space and time. Moreover, SMB is impacted by complex interactions between the atmosphere and the ice sheet surface, large-scale atmospheric circulations, and ice sheet topography (Lenaerts et al., 2019). For a fine-scale representation of Antarctica, such as its edges or peripheral ice, the resolution of a GCM is too coarse (Kittel et al., 2021; Seroussi et al., 2020). In addition, GCMs typically do not correctly incorporate critical polar physical processes, such as snow melt, albedo feedback, etc. (Kittel et al., 2021; Lenaerts et al., 2017). Polar-oriented RCMs, such as the Modèle Atmosphérique Régional (MAR), tackle the problem of low spatial resolution of GCMs over Antarctica and give a significantly more robust evaluation of mass and energy fluxes at the surface, but at a high computational cost (Fyke et al., 2018; Kittel et al., 2021).

One generally used alternative to the dynamical downscaling of GCMs is empirical statistical downscaling. Using observational data, statistical downscaling methods estimate statistical relationships between regional climate variables and global-scale predictors. Local climate changes are simulated by applying those relationships to the outputs of GCMs (Sellevold et al., 2019; Doury et al., 2022). In this line, Agosta et al. (2012) and Ghilain et al. (2022) developed a statistical downscaling of Antarctic SMB components from large-scale atmospheric forcings. Similarly, in Greenland, Sellevold et al. (2019) and Geyer et al. (2013) used an elevation class method to downscale SMB. Statistical downscaling can also be combined in a hybrid model with RCMs, e.g., Gallée et al. (2011) used a cascade of atmospheric models from large to local-scale to simulate high-resolution SMB over Antarctica. However, statistical downscaling approaches are limited because of (1) their dependence on observational data, (2) their need for a high-quality calibration dataset, and (3) their stationary statistical assumption of large/local-scale relationship that is required to remain constant under climate change (Dayon et al., 2015; Erlandsen et al., 2020; Doury et al., 2022).

More recently, novel statistical methods that use machine learning have been proposed to downscale GCMs. The machine learning RCM-emulator receives low-resolution global inputs and outputs a high-resolution image of a regional predictor. The emulator is designed to save computational costs and augment the ensemble of RCM simulations by combining the advantages of dynamical and empirical statistical downscaling (Doury et al., 2022). Machine learning surrogates of computationally expensive and complex RCMs are still a novel and recent approach in the cryosphere community. Nevertheless, machine learning has already been harnessed in other applications that model ice variables and dynamics, e.g., Bolibar et al. (2020), Hu et al. (2021), and Jouvet et al. (2022).

This study proposes two SMB emulators to downscale a GCM, using a (1) perfect and (2) imperfect model framework. The first emulator was trained following the perfect model

framework developed by Doury et al. (2022), where upscaled RCM features (UPRCM) are used as low-resolution inputs. The perfect model framework evaluates how the RCM-emulator performs when it only has to learn the downscaling function of the RCM. For this approach, the RCM-emulator needs perfect spatial and temporal consistency between the global climate variable inputs and local-scaled SMB images it recreates. Such an alignment cannot be guaranteed when using variables from a GCM and RCM because they stem from different models, and an RCM can generate sub-GCM-grid variability. The perfect model framework provides a perfect alignment by bypassing GCM/RCM variability and replacing GCM variables with an upscaled RCM (Sanchez-Gomez et al., 2009; Sanchez-Gomez & Somot, 2018). Nevertheless, we expect this framework to have limited use in the study case of SMB because of large differences in RCM and GCM simulations over the Antarctic Peninsula (Bozkurt et al., 2021). Therefore, we explore the potential of an alternative called the imperfect model framework, where the RCM-emulator is trained on coarse input features directly from the GCM. In this imperfect training framework, we aim to analyze whether the model could learn the underlying dynamics, despite inconsistencies between GCM and RCM simulations.

In this study, we explore the downscaling potential of the two machine learning frameworks by applying them to SMB over the Antarctic Peninsula. We first present the data, machine learning architecture, and frameworks that define, train, and evaluate both RCM-emulators in Section 2. Then, Section 3 shows the evaluation results of the emulators in the case study. In the end, Sections 4 and 5 discuss the results of the two frameworks and draw conclusions about the emulator and training frameworks for future use. Appendix A provides additional information about the data pre-processing pipeline, and Appendix B machine learning background about certain acronyms.

2 Materials and Methods

This study aimed to build an RCM-emulator $\hat{F}$ that uses a neural network architecture to estimate the downscaling function F in the following equation:

$$Y = \mathrm{F}(X) \quad X \subset \mathcal{D}, Y \subset \mathcal{E} \tag{1}$$

where X are low-resolution variables from a GCM over an input domain $\mathcal{D}$, and Y is a high-resolution surface variable of an RCM over a target domain $\mathcal{E}$.

2.1 Data and pre-processing

2.1.1 Choice of climate models

The goal of the RCM-emulator was to reproduce monthly SMB predictions from MAR(ACCESS1.3), a regional downscaling by MAR of the ACCESS1.3 GCM. This GCM is from the Coupled Model Intercomparison Project - Phase 5 (CMIP5) (Bi et al., 2012; Taylor et al., 2012). The RCM and its corresponding GCM were selected as the climate simulations for the emulator for two reasons. First, MAR accurately models physical processes in polar regions such as SMB, air-snow interactions, and atmospheric circulation over ice sheets (Donat-Magnin et al., 2021). Secondly, Kittel et al. (2021) and Agosta et al. (2015) showed that MAR(ACCESS1.3) outperformed other climate models when comparing predictions of the current Antarctic climate to ERA-Interim data.

The MAR simulations cover the period of 1980-2006 and future climate projections under a high-emission scenario (RCP8.5) from 2006-2100 (Moss et al., 2010). The RCM grid is in south polar stereographic coordinates and has a resolution of 35×35 km. In contrast, the GCM resolution is of 1.25° latitude by 1.875° longitude (approximately 68× 206 km) (Bi et al., 2012; Collier & Uhe, 2012). The GCM was projected to south polar stereographic coordinates to have it in the same projection system as the RCM (c.f. Appendix A for more details).

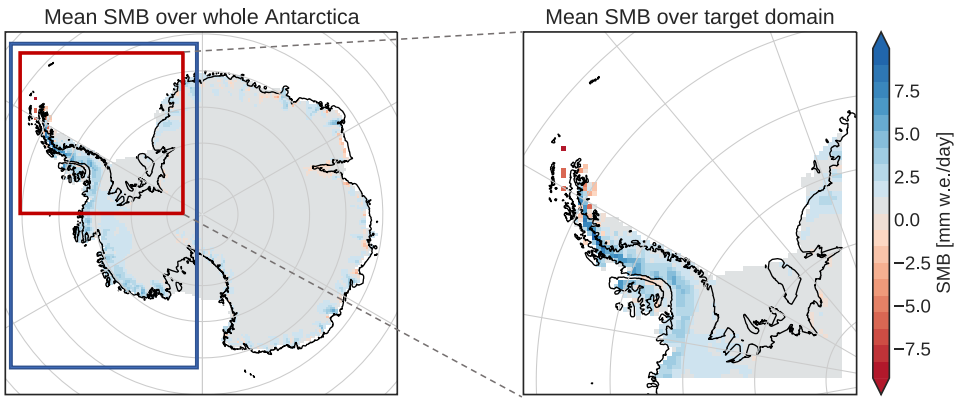

Figure 1: Target domain $\mathcal{E}$ (in dark red) and input domain $\mathcal{D}$ (in blue) for the RCM-emulator. The mean daily SMB values from 1980 to 2100 above Antarctica (left) and the Antarctic Peninsula (right) are illustrated underneath.

2.1.2 Target and input domain

The target domain $\mathcal{E}$ chosen for the RCM-emulator is a grid box that has a domain size of 64×64 pixels (at a resolution of 35×35 km, so 2240×2240 km) and covers approximately 5 million square kilometers. The target domain covers an area centered on the Antarctic Peninsula and extends to the Ronne-Filchner ice shelf in West Antarctica (Figure 1). The target domain is mountainous, with its highest peaks rising to about 3'000 m, and major ice shelves include Larsen C, Wilkins, and Ronne-Filchner. Precipitation varies significantly within the target domain. For example, the northern tip of the Antarctic Peninsula has the highest yearly precipitation levels (35-50 cm), and its west and northeast coasts also reach 35 cm/year. However, along the Antarctic Peninsula's east coast and Antarctica's interior, the climate is drier, with 10-15 cm/year of precipitation (Vignon et al., 2021; Draggan, 2009).

The heterogeneity of climate variables, like precipitation, over the target domain leads to a high temporal and spatial variability in SMB values. This shows when looking at mean daily SMB values (Figure 1). For example, dry inland points have minor variations of SMB, with maximum daily values under 2 mm water equivalent per day (mm w.e./day), while a point on the west coast of the Antarctic Peninsula can reach low extremes of -20 mm w.e./day. This substantial variation in SMB values with different annual patterns provides a unique use case to test the RCM-emulator in complex environments.

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

2.1.3 Input features of the RCM-emulator

As input features, the RCM-emulator receives a two-dimensional array X and a one-dimensional array Z. X is an array that contains images of normalized monthly means ($\tilde{V}_{t,x} \in \mathbb{R}^{\mathcal{D}}$) of eight different climate variables $x \in C_1$ at near surface level over domain $\mathcal{D}$ and T

months. Table 1 shows an overview of the eight climate variables chosen for this study. For each climate variable x and month t, each image $V_{t,x}$ is normalized according to its own spatial mean and standard deviation before providing them as inputs to the RCM-emulator:

$$\tilde{V}_{t,i,j,x} = \frac{V_{t,i,j,x} - \bar{V}_{t,x}}{\sigma(V_{t,x})} \quad \forall (i,j) \in \mathcal{D}, t \in T, x \in C_1 \tag{2}$$

where $V_{t,i,j,x}$ is the pixel at location (i,j), and $\bar{V}_{t,x}$ and $\sigma(V_{t,x})$ are the spatial mean and standard deviation of image V for variable x at time step t, respectively. Using spatial normalization of $V_{t,x}$ transforms each pixel of an image so that they are on a similar scale.

Overall, the input feature X contains $T*C_1$ normalised images of dimension $\mathcal{D}$ and is described by the following equation:

$$X = \left[\tilde{V}_{t,x} \quad \forall t \in T, x \in C_1\right] \in \mathbb{R}^{T \times \mathcal{D} \times C_1} \tag{3}$$

Z is a one-dimensional temporal encoding of the eight climate variables and includes the time series of spatial means $\bar{V}_{t,x}$ and standard deviations $\sigma(V_{t,x})$ for each $x \in C_1$ and $t \in T$ (Table 1). Because the climate variable images $V_{t,x}$ in X are normalized at each time step by their spatial mean, they no longer carry any temporal encoding. Providing Z to the RCM-emulator gives it access to this additional information. Following the same procedure as Doury et al. (2022), each element $Z_{t,x}$ in Z ($\bar{V}_{t,x}$ or $\sigma(V_{t,x})$) is normalized according to a reference period (T_{ref} =1980-2000):

$$\tilde{Z}_{t,x} = \frac{Z_{t,x} - \bar{Z}_{T_{\text{ref}},x}}{\sigma(Z_{T_{\text{ref}},x})} \quad t \in T, x \in C_1 \tag{4}$$

where $\bar{Z}_{T_{\text{ref}},x}$ and $\sigma(Z_{T_{\text{ref}},x})$ are, respectively, the temporal mean and standard deviation of the arrays of spatial means or standard deviations of $V_{t,x}$ for climate variable x and over the reference period $t \in T_{\text{ref}}$. Z also includes a cosine and sine vector

$$\cos_t = \cos\left(\frac{2\pi t}{12}\right);\ \sin_t = \sin\left(\frac{2\pi t}{12}\right) \quad \forall t \in T \tag{5}$$

to encode information about the month of the year. Overall, this gives the following equation for Z:

$$Z = \left[\tilde{Z}_{t,x}, \cos_t, \sin_t \quad \forall t \in T, x \in C_1\right] \in \mathbb{R}^{T \times C_2} \tag{6}$$

where $C_2 = 2 * C_1 + 2$. Figure 2 shows an example of X and Z for one time step t.

2.2 Model

2.2.1 Architecture

The following section goes into the details of the architecture of the RCM-emulator model and will use several machine-learning terms. Appendix B provides additional machine learning background information.

The RCM-emulator receives an eight-channeled 32 × 32 low-resolution image at time step t (where each channel is a climate variable) and its corresponding temporal encoding Z_t, and outputs a one-channeled 64×64 high-resolution image of SMB values predicted by the RCM-emulator at time step t (Figure 2).

The RCM-emulator's architecture (Figure 2) is a combination of the U-Net emulator developed by Doury et al. (2022) and the SmaAt-UNet by Trebing et al. (2021). Our U-Net model is equipped with convolutional block attention mechanisms (CBAM) and depthwise-separable convolutions (DSC) instead of regular convolutional operations. DSCs are designed to reduce the number of parameters and make the model faster (Trebing et al., 2021; Chollet, 2017).

Table 1: Two and one-dimensional input features* given to the RCM-emulator at time-step $t \in T$.

Variable Name	**Notation**	**Units**	**Dimensions**
2D variables			
Northward Wind	NW	$[\mathrm{ms}^{-1}]$	$\mathcal{D}$
Eastward Wind	EW	$[\mathrm{ms}^{-1}]$	$\mathcal{D}$
Shortwave Downward Radiation	SWD	$[\mathrm{Wm}^{-2}]$	$\mathcal{D}$
Longwave Downward Radiation	LWD	$[\mathrm{Wm}^{-2}]$	$\mathcal{D}$
Specific Humidity	QQP	[g/Kg]	$\mathcal{D}$
Temperature	TT	[°C]	$\mathcal{D}$
Precipitation	PR	[mmWe/day]	$\mathcal{D}$
Pressure	SP	[hPa]	$\mathcal{D}$
1D variables			
Spatial mean of 2D variables	$\bar{V}_{x,t}$		C_1
Spatial std of 2D variables	$\sigma\left(V_{x,t}\right)$		C_1
Seasonal Indicators	$\cos_t$, $\sin_t$		2

*Each feature is a daily output of a climate variable at near-surface level over domain $\mathcal{D}$. The frequency of variables is monthly after a monthly mean aggregation.

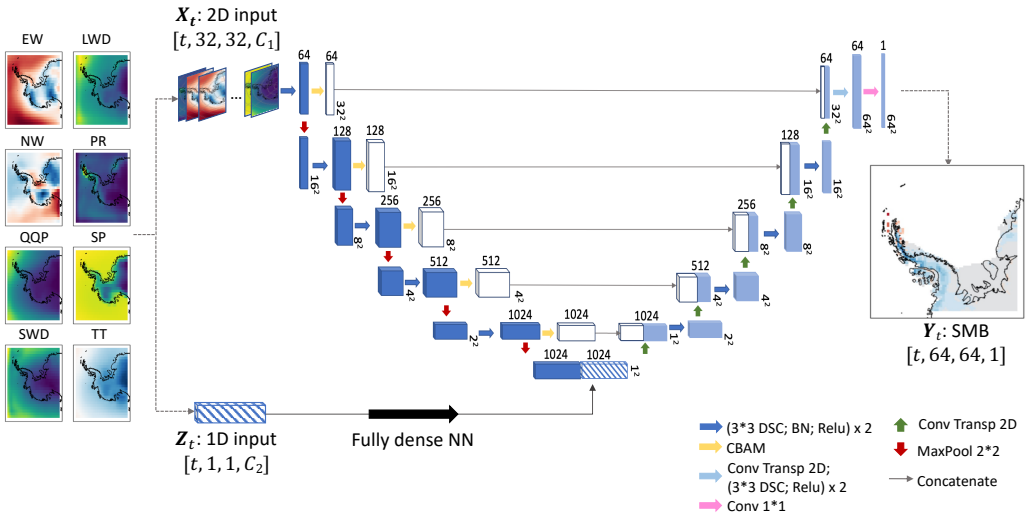

Figure 2: A conceptual overview of RCM-emulator at time step t. Left: Climate variables from the GCM (Table 1) over their input domain. The low-resolution 2D input variables X_t and corresponding 1D variable Z_t are created from these and given as input to the RCM-emulator. Middle: scheme of the U-Net architecture used for the RCM-emulator. The low-resolution 2D input variables X_t go through the left side of the U-Net (encoder) to reduce their size (and increase the number of channels) before being concatenated with Z_t at the bottom. Z_t first goes through a fully dense neural network to increase its number of channels to the same number as encoded X_t. The resulting feature map then goes through the right part of the U-Net (decoder) to reach the same resolution as the target SMB image. Right: high-resolution surface mass balance (SMB) map Y_t over the target domain produced by the RCM-emulator. Abbreviations of U-Net operators: DSC (depthwise-separable convolutions), BN (batch normalization), CBAM (convolutional block attention mechanisms), Conv Transp 2D (2D transposed convolution), MaxPool (max pooling), ReLU (rectified linear unit) (c.f. Appendix B for more information). Each block is a feature map with a number of channels (on top) and a size (on the right). Color code of blocks: convoluted and downsampled image (dark blue), attention feature maps (white), convoluted and upsampled image (light blue), Z_t through a fully dense neural network (striped blue).

A U-Net model (first proposed by Ronneberger (2017)) is a U-shaped convolutional neural network, divided into a downsampling (encoder) section that forms the left side and an upsampling (decoder) on the right. In convolutional neural networks, an input array goes through a series of layers where filters are applied. The output of each filter is a feature map (a multi-dimensional array of numerical values representing the learned features of an input image), which is then used as input to the next layer of the neural network.

The encoder of our RCM-emulator consists of double DSCs followed by a max pooling filter; this reduces the size and increases the number of channels of the low-resolution 2D input variables X_t. At each layer of the encoder, double DSCs duplicate the number of channels, and max pooling downsamples the image by halving its size. Each feature map from the double convolution also goes through a CBAM filter. CBAMs create an attention feature map highlighting important regions over the channels and spatial dimension of a feature map (Trebing et al., 2021). Note that the input to the next

layer of the encoder is not the attention feature map but the convoluted and downsampled image of the previous layer; the attention maps are used in the decoder (see below). This way, the original image features are preserved throughout the encoder layers. Overall, the encoder learns an abstract representation of the input images at each layer; the deeper it goes, the more general features are extracted.

At the bottom of the U-Net, encoded spatial information from X_t (i.e., the result of X_t going through the encoder) and temporal information from Z_t are concatenated. First, a fully dense neural network is applied to the corresponding 1D input Z_t of X_t to reach the same number of channels as the output of the last layer of the encoder. Then, it is concatenated with the previous feature map of the encoder at the bottom of the U-Net. 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 (Doury et al., 2022; Sha et al., 2020).

The decoder is built out of three parts that repeat at each layer. First, a 2D transposed convolution operation upsamples the feature maps by doubling their size. Secondly, the resulting feature map is concatenated with the previous encoder's attention feature map via skip connections. Lastly, double DSCs halve the number of channels. At the decoder's end, an additional up-sampling layer and a 1×1 convolution are added to reach the target image size. This allows the decoder to create a high-resolution image of the same size as the target SMB from the RCM.

2.2.2 RCM-emulators in (im)perfect model frameworks

This study proposes and compares two training scenarios to use the RCM-emulator architecture to downscale GCMs: (1) a perfect and (2) an imperfect model framework. The two frameworks differ in the climate model used to source the low-resolution variables to train the RCM-emulator.

Perfect model framework:
The first RCM-emulator ($\hat{\mathrm{F}}_{\mathrm{P}}$) was trained following a perfect model framework (Doury et al., 2022). In the perfect model framework, the low-resolution training inputs of $\hat{\mathrm{F}}_{\mathrm{P}}$ are upscaled features from the same RCM as the high-resolution SMB target. The performance of $\hat{\mathrm{F}}_{\mathrm{P}}$ evaluates how the emulator performs when it has to learn only the downscaling function F of the RCM (Equation 1). For this purpose, the RCM-emulator is fed with low-resolution inputs and high-resolution targets that are perfectly aligned and show high spatial and temporal correlation. Consequently, the perfect model framework avoids learning relationships between local/large-scale features that are RCM/GCM specific and is a solution to circumvent potential large-scale inconsistencies between GCM and RCM variables (Sanchez-Gomez et al., 2009; Sanchez-Gomez & Somot, 2018).

To test the effect of the perfect model framework, we created upscaled RCM features (UP-RCM) from the RCM that have the same spatial resolution as GCMs. First, RCM features were upscaled to GCM resolution using conservative interpolation (Pletzer & Fillmore, 2015). Then, a 3×3 moving average filter smoothed the upscaled RCM features. This filter removes local-scale information that might remain after the upscaling (Doury et al., 2022; Klaver et al., 2020).

Imperfect model framework:
For the second RCM-emulator ($\hat{\mathrm{F}}_{\mathrm{I}}$), the low-resolution training inputs are GCM features. This imperfect model framework allows for spatial and temporal inconsistencies between the RCM output and GCM input during training. The performance of $\hat{\mathrm{F}}_{\mathrm{I}}$ assesses whether the RCM-emulator can learn to downscale from the GCM to RCM despite inconsistencies. One potential advantage of the imperfect model framework is that it learns both the downscaling function and a GCM/RCM relationship, so it can be used to generate RCM output from GCM output directly.

Bias and inconsistencies:
Since the difference between the perfect and imperfect model framework depends mainly on the differences between upscaled RCM and GCM, two correlation statistics were used to assess the presence of inconsistencies between upscaled RCM and GCM features. First, for each atmospheric variable $x \in C_1$ and point $p = (i, j)$ in the input domain $\mathcal{D}$, the Pearson correlation coefficient was calculated between the GCM and upscaled RCM time series (Appendix B2). Secondly, for each $x \in C_1$ and time step $t \in T$, the spatial correlation (sc) between GCM (G_t^x) and upscaled RCM images (U_t^x) was computed:

$$\mathrm{sc}\left(G_t^x, U_t^x\right) = \frac{\mathrm{cov}(G_t^x, U_t^x)}{\sigma(G_t^x)\sigma(U_t^x)} \quad \forall t \in T, x \in C_1. \tag{7}$$

2.3 Training

Every observation given to the RCM-emulator comprises features X_t and Z_t for monthly time step $t \in T$ (Figure 2). X_t is an array of dimension $32 \times 32 \times 8$, where 32×32 is the spatial size (number of pixels) of the input domain $\mathcal{D}$, and 8 is the number of different atmospheric variables chosen as predictors. Z_t is the corresponding temporal encoding of X_t and of dimension 18 (c.f. Section 2.1.3).

To address the high spatiotemporal variability in SMB values over the target domain, we used a normalized RMSE (NRMSE) loss function. Normalizing the RMSE facilitates comparing datasets with different magnitudes and large variability, as in our case. For each time step t, the NRMSE was calculated between the predicted SMB maps $\hat{Y}^t$ and the target SMB Y^t from the RCM over all positions in the target domain $p \in \mathcal{E}$:

$$\mathrm{NRMSE}\left(Y^t, \hat{Y}^t\right) = \frac{\sqrt{\frac{1}{P}\sum_p (\hat{y}_p^t - y_p^t)^2}}{Y_{\max} - Y_{\min}} \quad \forall t \in T \tag{8}$$

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

Both RCM-emulators were trained using a batch size of 100 (i.e., the number of samples propagated through the neural network before updating the internal model parameters) and over a maximum of 50 epochs (i.e., the number of passes the whole training dataset takes through the neural network). We used early stopping (Prechelt, 1998), and the perfect and imperfect models converged, respectively, at 30 and 32 epochs. In addition, we used a learning-rate scheduler (ReduceLROnPlateau module from PyTorch) that adjusted the learning rate between epochs and reduced the learning rate on loss plateaus, starting with an initial learning rate of $\mathrm{LR}_0 = 0.005$. The batch size and initial learning rate were chosen from hyperparameter tuning. The RCM-emulators were trained on a graphics processing unit (GPU), which took approximately 4 minutes. A GPU was no longer needed once the model was trained, and making predictions on test data took 15 seconds on a central processing unit (CPU).

2.4 Evaluation

The last ten years of the time frame of the climate simulations were separated into a test period $T_{test} = 2090 - 2100$ (120 samples). The remaining time was separated using a random 20%/80% split into a validation (266 samples) and training set (1066 samples). These were used during the training of the model to calculate validation and training metrics (Figure B1). These time frames separated input features X and Z into training, validation, and testing features. The testing features were not seen by the RCM-emulators during training and were only used for evaluating the models' performance afterward. The test period was arbitrarily chosen to be at the end of the climate models' time frame, but it could also have been taken elsewhere as long as they were consecutive.

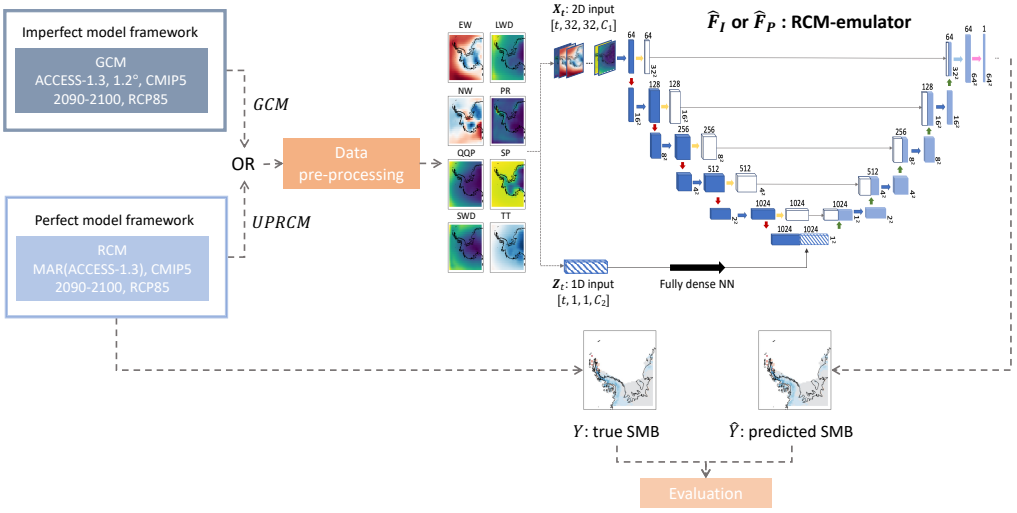

Figure 3: Evaluation setting of the RCM-emulators $\hat{\mathrm{F}}_{\mathrm{P}}$ and $\hat{\mathrm{F}}_{\mathrm{I}}$. Left: selection of the climate model used for low-resolution inputs to evaluate $\hat{\mathrm{F}}_{\mathrm{P}}$ and $\hat{\mathrm{F}}_{\mathrm{I}}$. Either GCM or upscaled RCM (UPRCM) features are used. Upper left: The imperfect model framework where $\hat{\mathrm{F}}_{\mathrm{P}}$ or $\hat{\mathrm{F}}_{\mathrm{I}}$ make high-resolution SMB predictions ($\hat{Y}$) using low-resolution features that come from the GCM. Lower left: the perfect model scenario where low-resolution features for the predictions of $\hat{\mathrm{F}}_{\mathrm{P}}$ are UPRCM. The evaluation metrics used to compare predictions and the target SMB from the RCM (Y) were the root mean square error, Wasserstein distance, and Pearson correlation coefficient. The data pre-processing pipeline is described in Section 2.2.2. The architecture of the RCM-emulator is defined in Figure 2.

For the evaluation of the RCM-emulator from the two training frameworks, their predictions of SMB over the test period were compared to the target SMB from the RCM using different statistics: Pearson correlation coefficient, Root Mean Square Error (RMSE), and Wasserstein distance (c.f. Appendix B2). For each emulator and point in the target domain $p \in \mathcal{E}$, we compared the target SMB time series Y_p to the predicted values $\widehat{Y_p}$ over the test period. Correlation is a good indicator of the reconstruction of temporal patterns such as synchrony and seasonality. RMSE and Wasserstein distance evaluate the fitting of extreme values and the representation of monthly variability.

2.4.1 Evaluation of the (im)perfect model framework

The following three types of tests made to evaluate the predictions of the RCM-emulator from the two training frameworks are illustrated in Figure 3.

Perfect model framework - $\hat{\mathrm{F}}_{\mathrm{P}}$:
The performance of the RCM-emulator $\hat{\mathrm{F}}_{\mathrm{P}}$ trained in the perfect model framework on upscaled RCM was evaluated twice. First, we evaluated the predictions made by $\hat{\mathrm{F}}_{\mathrm{P}}$ with upscaled RCM test features - $\hat{\mathrm{F}}_{\mathrm{P}}$(UPRCM). This assessed how the emulator performs when tested in conditions similar to its training, i.e., on input data from the same climate model. In a second step, we evaluated the predictions made by $\hat{\mathrm{F}}_{\mathrm{P}}$ with GCM inputs - $\hat{\mathrm{F}}_{\mathrm{P}}$(GCM). This considers how the RCM-emulator trained on upscaled RCM performs when receiving GCM data as input, i.e., how it generalizes to new distributions. To be useful, the RCM-emulator should give accurate reconstructions of SMB when re-

ceiving GCM variables as input. Furthermore, the accuracy of the $\hat{F}_P$(GCM) predictions is also an indicator of the presence of inconsistencies between upscaled RCM and GCM features.

Imperfect model framework - $\hat{F}_I$:
Emulator $\hat{F}_I$, trained in the imperfect model framework with the GCM, was evaluated once. Its predictions made with test features from the GCM - $\hat{F}_I$(GCM) were compared to the target SMB.

For each of the three types of evaluations $\hat{F}_P$(UPRCM), $\hat{F}_P$(GCM), and $\hat{F}_I$(GCM), we analyzed single month and average predictions made over the test period (Section 3.1). In addition, we examined which regions of the target domain had the best reconstructions of SMB patterns in terms of precision (RMSE, Wasserstein distance) and temporal synchrony (Pearson correlation) (Section 3.2.1). Furthermore, we assessed the presence of inconsistencies between upscaled RCM and GCM features to evaluate the need for the perfect or imperfect model framework (Section 3.2.2). Finally, we compared the target SMB to the time series of predicted SMB values for four points in the target domain (Section 3.3). We specifically chose these four points to evaluate how the RCM-emulators handled different patterns and intensities of SMB.

3 Results

3.1 Emulated SMB fields

To evaluate the performance of RCM-emulators $\hat{F}_P$ and $\hat{F}_I$ at reconstructing spatial structures of SMB values in the (im)perfect model framework, we compared a prediction for a random month and the average predictions over the test period (2090-2100) to the target (RCM) SMB using RMSE (Figure 4a). In addition, we visualize the bias of the models by plotting the difference in mean and standard deviation compared to the target SMB over the test period (Figure 4b).

Compared to the low-resolution upscaled RCM map, the high-resolution RCM is more detailed and shows more complex spatial structures (Figure 4a). In both RCM and upscaled RCM maps, the tip and west coast of the Antarctic Peninsula have high values of SMB (with maximum values of 10 mm w.e./day). For the random month of May 1980 (first row in Figure 4a), both $\hat{F}_P$(UPRCM) and $\hat{F}_I$(GCM) accurately reproduce the spatial structure of the target RCM (RMSE of 0.27 and 0.29, respectively), except for high-value SMB regions in the mainland, south of the Ronne-Filchner Ice Shelf. On average, over the test period (second row in Figure 4a), both predictions of $\hat{F}_P$(UPRCM) and $\hat{F}_I$(GCM) are very similar to the average target SMB (RMSE of 0.09 and 0.15, respectively).

However, when the perfect model framework makes predictions from GCM features, it cannot reproduce the extreme values of SMB (lower/higher than -5/5 mm w.e./day). In particular, $\hat{F}_P$(GCM) underestimates the high magnitude SMB values on the west coast of the Antarctic Peninsula (Figure 4b). This is reflected in its RMSE in Figures 4a, which is twice as high as the other two evaluations (0.53 for the random month and 0.22 on average).

Overall, these results hint at the fact that both the imperfect and perfect model framework RCM-emulators, when evaluated on data similar to what they were trained on, have a solid capacity to reproduce the complex spatial structures of the RCM.

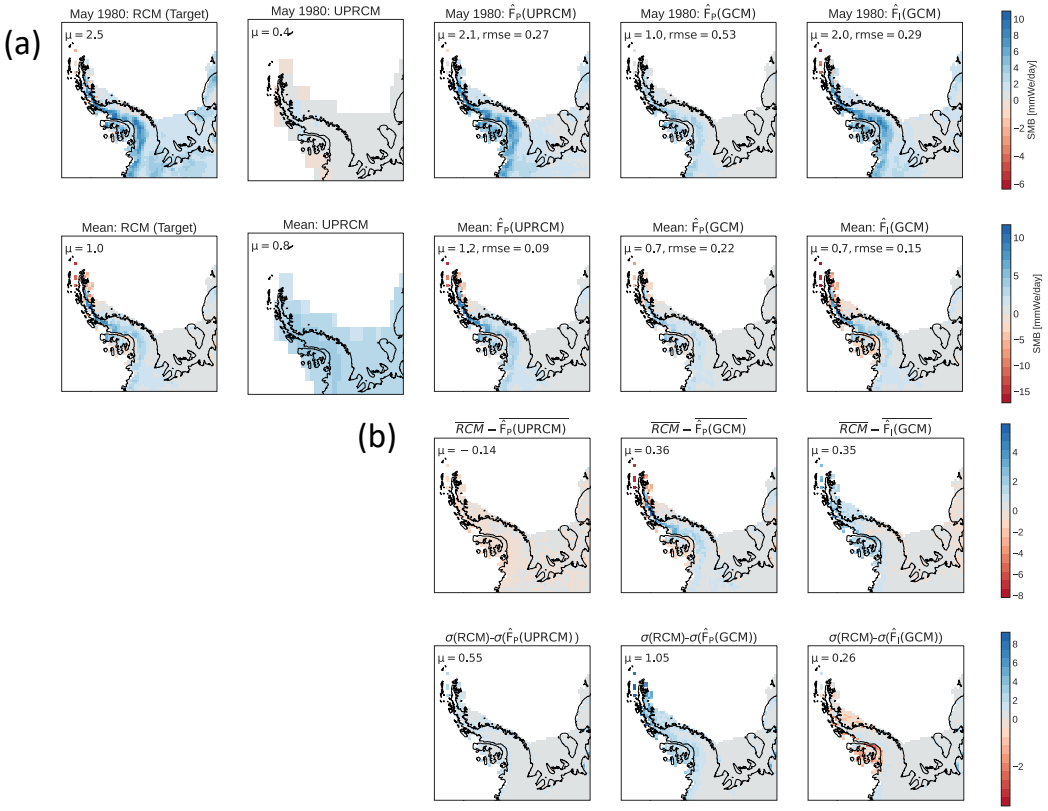

Figure 4: (a) Surface mass balance (SMB)predictions of the RCM-emulators on a random month (May 1980) (top) and averaged over the test period (2090-2100) (bottom) over target domain $\mathcal{E}$. From left to right: SMB in target RCM, upscaled RCM, $\hat{\mathrm{F}}_{\mathrm{P}}$(UPRCM) - trained in the perfect model framework and evaluated on upscaled RCM, $\hat{\mathrm{F}}_{\mathrm{P}}$(GCM) - trained in the perfect model framework and evaluated on GCM, and $\hat{\mathrm{F}}_{\mathrm{I}}$(GCM) - trained in the imperfect model framework and evaluated on GCM. (b) Difference between average (top) or standard deviation (bottom) of RCM and SMB predictions of the RCM-emulators over the test period. Legend: spatial mean (μ) of SMB over domain $\mathcal{E}$ and spatial RMSE (rmse) between the emulated and target RCM SMB pixel values.

3.2 Performance of the RCM-emulators

3.2.1 Evaluation metrics

Three statistical metrics (Appendix B2) were used to evaluate the overall performance of RCM-emulators $\hat{\mathrm{F}}_{\mathrm{P}}$ and $\hat{\mathrm{F}}_{\mathrm{I}}$ in the perfect and imperfect model framework, respectively (Figure 5). In addition to the general performance, we were also interested in seeing how the evaluation metrics differed for regions with high magnitudes of SMB, such as the tip and west coast of the Antarctic Peninsula, and dryer regions, such as the east coast and mainland over the Ronne-Filchner Ice Shelf.

Pearson correlation coefficient:
Figure 5d shows the box plot of the correlation values between SMB predictions of RCM-emulators $\hat{\mathrm{F}}_{\mathrm{P}}$ and $\hat{\mathrm{F}}_{\mathrm{I}}$, and the target SMB. On average, SMB predictions from $\hat{\mathrm{F}}_{\mathrm{P}}$(UPRCM) and $\hat{\mathrm{F}}_{\mathrm{I}}$(GCM) have higher correlation values to the target RCM than $\hat{\mathrm{F}}_{\mathrm{P}}$(GCM) (0.59, 0.62 and 0.45, respectively). This is especially visible on the tip and west coast of the

Antarctic Peninsula, where $\hat{F}_P$(UPRCM) and $\hat{F}_I$(GCM) have the highest correlation to the target (0.97 and 0.97, respectively) (Figure 5b-c). On the other hand, the east coast of the Antarctic Peninsula has the lowest correlation for all models but especially for $\hat{F}_P$(GCM). We suspect this is related to the regional precipitation amounts, as the RCM-emulators seem to perform less well for particularly dry regions.

Wasserstein distance and RMSE:
Figure 5h and Figure 5l show that $\hat{F}_P$(GCM) has the highest Wasserstein distance and RMSE values amongst all models, which indicates that the density probability functions of its emulated SMB series are the furthest from the target RCM. The Antarctic Peninsula has a particularly high Wasserstein distance and RMSE, with outliers values up to 10 and 14, respectively (Figure 5f-j). This hints that the RCM-emulator trained in the perfect model framework is not able to predict extreme SMB magnitudes when given GCM inputs, i.e, it does not generalize well to a new distribution.

According to these three evaluation metrics, $\hat{F}_P$(UPRCM) and $\hat{F}_I$(GCM) perform very similarly, are close to the target RCM, and are consistently better than $\hat{F}_P$(GCM). This indicates that the imperfect model framework should be preferred. Pearson correlation shows that temporal patterns and seasonality are best reconstructed in regions of high precipitation, such as the tip and west coast of the Antarctic Peninsula. Still, the RCM-emulators' predictions tend to underestimate extreme (high and low) SMB values, which is especially visible for $\hat{F}_P$(GCM) and in dry regions, such as the inland and east Antarctic Peninsula.

3.2.2 Inconsistencies between upscaled RCM and GCM variables

To assess the inconsistencies between the large-scale (low-resolution) and local-scale (high-resolution) atmospheric variables, temporal (Figure 6a) and spatial (Figure 6b-c) correlation were calculated between upscaled RCM and GCM features. As spatial and temporal inconsistencies might confuse an RCM-emulator, their presence could justify the need for the perfect model framework for training the emulator. Furthermore, this provided information on the severity of inconsistencies the RCM-emulator had to accommodate in the imperfect model framework to downscale the GCM.

Temporal correlation:
For most atmospheric variables in Figure 6a, the time series of upscaled RCM and GCM features are highly positively correlated, with values very close to one, indicating that every pixel shows a high temporal consistency between RCM and GCM for the atmospheric variables. However, the two wind variables (east/north-ward wind) show inconsistencies between global GCM and regional upscaled RCM time series over the mainland and the Antarctic Peninsula, with minimal Pearson correlation values of -0.2, indicating inconsistencies in temporal behavior of wind in RCM and GCM. Figure 6a suggests that, except for the winds, there is temporal consistency in the seasonal patterns between regional high-resolution and global low-resolution variables. This indicates that for most of the variables, the RCM-emulator in the imperfect model framework learns the downscaling function between GCM and RCM images that are well aligned in time.

Spatial correlation:
Figure 6b shows the spatial correlation (Equation 7) between upscaled RCM and GCM features, indicating how well the spatial patterns in atmospheric variables between RCM and GCM correspond. Atmospheric variables, like temperature, specific humidity, radiation, and precipitation, differ significantly in spatial patterns between the upscaled RCM and GCM models. Their spatial correlation shows large variability over time and often reaches low correlation values. Shortwave downward radiation and precipitation are the variables that show the largest spatial inconsistency between RCM and GCM. Shortwave downward radiation has an annual spatial correlation pattern strongly oscillating between approximately 0.1 in Austral summer (Nov-Feb) and 0.8 in Austral win-

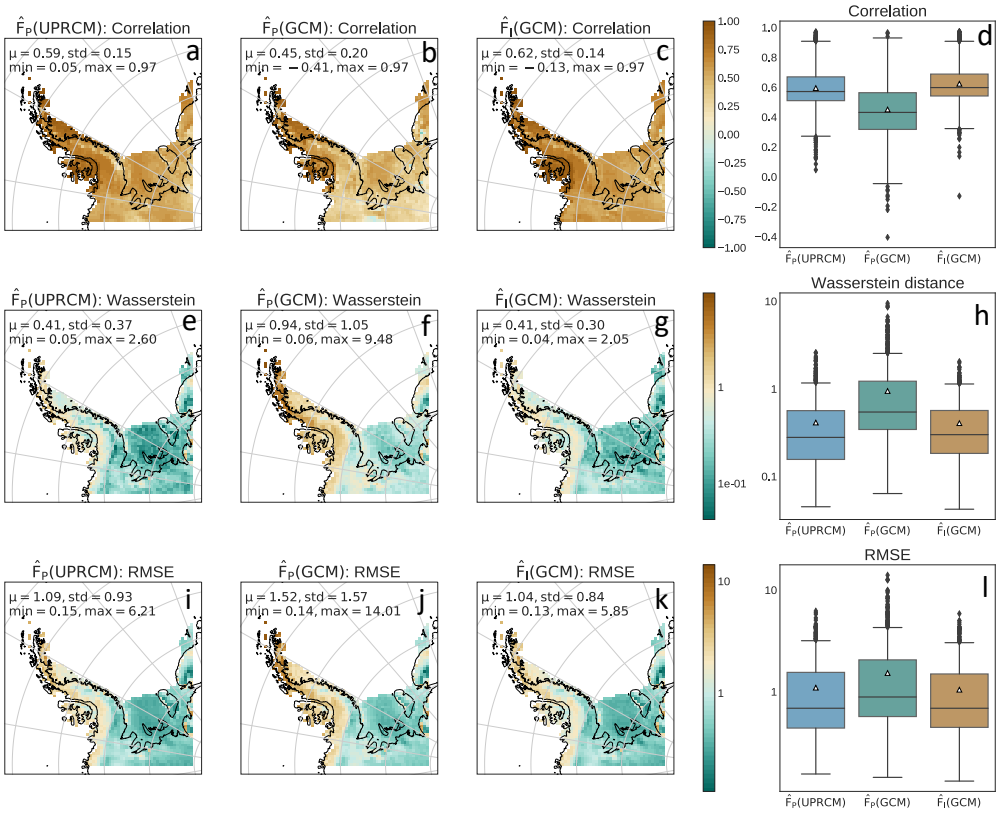

Figure 5: Evaluation metrics on predictions from RCM-emulators over target domain $\mathcal{E}$ and test period (2090-2100). For each position $p \in \mathcal{E}$, the time series of predicted SMB values are compared to the target SMB using three different metrics (from top to bottom): Pearson correlation coefficient, Wasserstein distance, and RMSE. Target SMB time series are compared to predictions made by (from left to right): $\hat{F}_P$(UPRCM) - trained in the perfect model framework and evaluated on upscaled RCM, $\hat{F}_P$(GCM) - trained in the perfect model framework and evaluated on GCM, and $\hat{F}_I$(GCM) - trained in the imperfect model framework and evaluated on GCM. Legend: spatial mean (μ), standard deviation (std), minimum (min) and maximum (max) of metrics over $\mathcal{E}$. Right: box-plot of evaluation metrics over all positions $p \in \mathcal{E}$, from lower to upper quartile, with a line at the median and a triangle at the mean.

ter (June-Aug). On the other hand, precipitation has a very poor spatial correlation over the whole test period, with a maximum of only 0.38, indicating that the spatial precipitation patterns in RCM and GCM are highly inconsistent.

Figure 6c shows individual examples of precipitation and short/long-wave downward radiation to illustrate the biases and inconsistencies in spatial patterns between large-scale GCM and upscaled RCM. It shows that the spatial correlation of shortwave radiation is low between upscaled RCM and GCM maps during Austral summer because the GCM predicts higher radiation values in the mainland of Antarctica than upscaled RCM. On the other hand, spatial correlation is high for the Austral winter months; however, we suspect this is only because there was very little radiation at that time. This great difference between Austral winter and summer months explains the strong yearly oscillation of spatial correlation for shortwave radiation. The spatial patterns in the upscaled RCM and GCM are also very different for precipitation. The upscaled RCM is much more

detailed in its predictions, showing upscaled representations of higher resolution. For example, for November 2093, the upscaled RCM 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 Bellingshausen Sea. Overall, these results show that in the imperfect model framework, the RCM-emulator is faced with strong spatial inconsistencies between the RCM and GCM when learning the downscaling function.

3.3 Time series of predictions

Figure 7 shows the emulated time series for four geographical points to assess how well the RCM-emulators can predict different temporal patterns and intensities of SMB. P1 on the Larsen Ice Shelf has high precipitation levels, and SMB values oscillate annually between -5 and 5 mm w.e./day. P2, on the west coast of the Antarctic Peninsula, has SMB values reaching low extremes of -15 mm w.e./day. P3 on the east coast of the Antarctic Peninsula has low yearly precipitation and minor SMB variations (0-4 mm w.e./day). P4 on the Ronne Ice Shelf has a significantly drier climate (0-1.5 mm w.e./day).

For each of these four points, $\hat{\mathrm{F}}_{\mathrm{P}}$(UPRCM) and $\hat{\mathrm{F}}_{\mathrm{I}}$(GCM) come very close to reproducing the temporal patterns of the target RCM series. For P1 and P2, $\hat{\mathrm{F}}_{\mathrm{P}}$(UPRCM) and $\hat{\mathrm{F}}_{\mathrm{I}}$(GCM) reproduce the seasonality well, with high correlation values to the target SMB (for P1: 0.74 and 0.68; for P2: 0.91 and 0.93, respectively). $\hat{\mathrm{F}}_{\mathrm{I}}$(GCM) even outperforms $\hat{\mathrm{F}}_{\mathrm{P}}$(UPRCM) at emulating low drops in SMB. For P3 and P4, RCM-emulators can reproduce the time series pattern, even when the time series' behavior is less seasonal, like for P3. We notice for P4 that $\hat{\mathrm{F}}_{\mathrm{P}}$(UPRCM) and $\hat{\mathrm{F}}_{\mathrm{I}}$(GCM) have difficulties reproducing multiple close peaks per year and tend to merge them into one prominent peak.

When the perfect model framework makes predictions from the GCM, it tends to underestimate the high amplitude values of SMB. This is also noticeable in the time series for all four points in Figure 7b. The predictions of $\hat{\mathrm{F}}_{\mathrm{P}}$(GCM) can reproduce the seasonal patterns (reflected in a high correlation of 0.93 for P2, for example) but produce a version of the target RCM time series where extremes are underestimated (resulting in a higher RMSE). This is also visible in the probability density functions, where the peak of $\hat{\mathrm{F}}_{\mathrm{P}}$(GCM) is higher and centered around lower SMB values compared to $\hat{\mathrm{F}}_{\mathrm{P}}$(UPRCM) and $\hat{\mathrm{F}}_{\mathrm{I}}$(GCM), which reach extreme values and resemble more the target RCM. This pattern repeats in the annual SMB predictions for all RCM-emulators. $\hat{\mathrm{F}}_{\mathrm{P}}$(UPRCM) and $\hat{\mathrm{F}}_{\mathrm{I}}$(GCM) come very close to the target annual SMB values for all four points, with $\hat{\mathrm{F}}_{\mathrm{I}}$(GCM) again slightly better than $\hat{\mathrm{F}}_{\mathrm{P}}$(UPRCM) for almost all years. $\hat{\mathrm{F}}_{\mathrm{P}}$(GCM) consistently underestimates the truth by an approximate factor of two.

4 Discussion

This study compared two hybrid statistical frameworks that downscale GCMs to RCMs using deep learning. We explored the downscaling potential of these frameworks by applying them to a complex climate variable, SMB, over the Antarctic Peninsula. The two RCM-emulators were developed based on the same U-Net architecture but with a different training framework. (1) $\hat{\mathrm{F}}_{\mathrm{P}}$ followed the perfect model framework with local/large-scale training features that both came from the RCM. For this, RCM features were upscaled to GCM resolution (UPRCM) to serve as large-scale training inputs to the model. On the other hand, (2) $\hat{\mathrm{F}}_{\mathrm{I}}$ was trained in the imperfect model framework where the large-scale features came from the GCM.

4.1 Performance of the RCM-emulators

In the perfect model framework, $\hat{\mathrm{F}}_{\mathrm{P}}$(UPRCM) competently reproduces the spatial and temporal pattern of the target RCM. However, when tested on GCM features, $\hat{\mathrm{F}}_{\mathrm{P}}$(GCM)

is not able to reproduce the extreme values of SMB and creates a less extreme version of the truth by underestimating the high magnitude SMB values (Figure 4).

We also find significant inconsistencies between RCM and GCM variables(Figure 6a). For example, the two wind variables showed high temporal discrepancies between large and local-scale simulations. Temporal inconsistencies might occur 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 shortwave downward radiation. Some of this might exist because of errors during the computation of RCM simulations, such as inconsistent forcings or boundary conditions. However, we assume that the rest of the inconsistencies exist because the RCM adjusts the low-resolution simulation, i.e., for a more fine-scaled simulation of the physical processes (Doury et al., 2022; Sørland et al., 2018; Misra, 2007; Noguer et al., 1998; Laprise et al., 2008). These inconsistencies might confuse an RCM-emulator, which could justify the need for the perfect model framework for training a model.

To see whether an RCM-emulator could offset the inconsistencies between the RCM and GCM, we trained the second emulator $\hat{\mathrm{F}}_{\mathrm{I}}$ in the imperfect model framework directly with large-scale GCM inputs. $\hat{\mathrm{F}}_{\mathrm{I}}$(GCM) misses some precision compared to $\hat{\mathrm{F}}_{\mathrm{P}}$(UPRCM) in emulating SMB patterns for regions with minor SMB variations (e.g., P4 in Figure 7, SMB smaller than 1.5 mm w.e./day). But for the other points, $\hat{\mathrm{F}}_{\mathrm{I}}$(GCM) comes very close to the target RCM. It is also performing exceedingly well in terms of predicting annual SMB values and is consistently better than $\hat{\mathrm{F}}_{\mathrm{P}}$(GCM).

Our GCM downscaling method works well, even when tested on a complex variable, such as SMB, that is not originally present in the GCM. The RCM-emulator has a solid capacity to reproduce the spatial and temporal patterns of the RCM, but only for $\hat{\mathrm{F}}_{\mathrm{P}}$(GCM) and $\hat{\mathrm{F}}_{\mathrm{I}}$(GCM), i.e., when evaluated on data similar to the model's training set. However, $\hat{\mathrm{F}}_{\mathrm{P}}$(GCM) is not able to generalize to GCM data and reproduce SMB correctly.

4.2 Implications of our main results

The results of this study show that machine learning provides a valuable alternative to dynamical/statistical downscaling, especially for climate variables that are not in the GCM. Furthermore, the RCM-emulator's process is very fast, i.e., approximately 4 minutes for 30 epochs of training on a GPU and instantaneous predictions on CPU. This computational time is significantly smaller than running an RCM simulation which can take multiple weeks to calculate on a super-computer.

Our results also show that the choice of training framework for the RCM-emulator matters. Doury et al. (2022) 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, an 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. In this study, while $\hat{\mathrm{F}}_{\mathrm{P}}$ performs well when evaluated on an upscaled RCM, it underperforms on real GCM features, resulting in large biases and inconsistencies. The predictions of $\hat{\mathrm{F}}_{\mathrm{P}}$(GCM) follow the correct temporal patterns of the target RCM, but they consistently underestimate the target SMB time series. We suspect this is because the RCM-emulator cannot learn to compensate for RCM-GCM inconsistencies during its training under the perfect model framework. Consequently, when the model is given GCM inputs while trained on upscaled RCM, inconsistencies are preserved and appear in the local reconstructions. Therefore in this study case, the perfect model framework underperforms when used in a practical framework on real GCMs. However, when trained in the imperfect framework with low-resolution GCM data, the RCM-emulator can predict SMB values close to the truth, as if it learned the underlying RCM-GCM inconsistencies and dynamics.

4.3 Limitations of machine learning

Although machine learning methods show strong potential to be surrogates of computationally expensive climate models, they also come with important limitations. First, neural networks, as used here, have a reputation for being a black box algorithm and thus having a decision process that is hard to understand. Still, there are recent developments to make them more transparent (Rocca & Perna, 2022; Guidotti et al., 2019; Savage, 2022). Secondly, training a neural network while relying on a GPU creates several sources of randomness, and reproducibility depends on using the exact same settings as the authors of a framework (Feng & Hao, 2020; Scardapane & Wang, 2017). Lastly, machine learning models remain very specific, and their training dataset determines the application range. So, learning from small amounts of data and applying what a model learned to new domains (Transfer Learning) remains a significant challenge in machine learning (Dube et al., 2018). Therefore, despite the advantage of the imperfect framework, $\hat{\mathrm{F}}_{\mathrm{I}}(\mathrm{GCM})$ has two major limitations: it is potentially region and model specific. This means the model would need to be retrained to make predictions on another region or RCM/GCM combination. Furthermore, it remains to be tested how the model would perform on a different climate scenario or a related GCM. We suspect this will depend significantly on the similarity of this new setting to the original training data.

4.4 Broader implications of this study

This study has shown the potential of using deep learning methods to downscale GCMs. Machine learning provides an RCM-emulator that can make very fast and fine-scaled reproductions of an RCM variable, even when that variable is not present in the GCM. However, we also illustrated the importance of choosing the model's training framework. In cases of significant inconsistencies between RCM/GCMs, the perfect model framework is limited and does not work well. On the other hand, we have shown that it is possible with the imperfect model framework to make accurate predictions directly from GCM data. This deep learning emulator provides low-cost local-scale information while learning the underlying RCM-GCM inconsistencies and dynamics.

5 Conclusion

This study aimed to explore the potential of using machine learning surrogate frameworks instead of the computationally expensive dynamical downscaling of GCMs. We built a deep learning RCM-emulator that learned the downscaling function of an RCM and tested the emulator by reconstructing local-scale SMB values over the Antarctic Peninsula. This means that the RCM-emulator, when given large-scale (low-resolution) atmospheric variables from a GCM, can reconstruct a local-scale (high-resolution) image of SMB from an RCM. Compared to running an RCM, the RCM-emulator is designed to be computationally faster.

The RCM-emulator's architecture is a U-Net model with convolutional block attention mechanisms. The U-net has depthwise-separable convolutions instead of normal convolutions to make a smaller and more efficient network.

We proposed two training scenarios for the RCM-emulator: the perfect and imperfect model framework. The two frameworks differ in their source of low-resolution variables to train the RCM-emulator. In the perfect model framework, the RCM-emulator receives upscaled RCM features as low-resolution inputs. This setting was designed for the emulator to learn the RCM downscaling function undisturbed by potential RCM-GCM inconsistencies. In the imperfect model framework, the RCM-emulator is trained on large-scale features directly sourced from the GCM to evaluate whether it can make accurate predictions despite RCM-GCM inconsistencies.

We evaluated the emulator trained in the perfect model setting twice: (1) on large-scale upscaled RCM features and (2) directly with variables from the GCM. While the emulator evaluated on upscaled RCM 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 inconsistencies between GCM and RCM features. Because the perfect model framework focuses only on the downscaling function of the RCM, it does not allow the RCM-emulator to learn large-scale RCM-GCM inconsistencies. Thus they are conserved when making predictions with GCM variables, leading to underestimation by the RCM-emulator.

The second RCM-emulator, trained in the imperfect model framework directly on GCM features, reproduces detailed and accurate high-resolution SMB maps. This RCM-emulator makes correct annual SMB predictions and reconstructs the temporal patterns of individual SMB time series and global spatial structures over the Antarctic Peninsula. The performance of the RCM-emulator in the imperfect model framework hints that it can accurately downscale the GCM despite RCM-GCM inconsistencies and that a perfect model framework might not be helpful.

Training the emulator takes under four minutes on a GPU, and predictions are almost instantaneous. Hence, the RCM-emulator is significantly faster than an RCM simulation, which can take several weeks on a supercomputer. However, machine learning surrogates are limited by their specificity to their training dataset, and transferring their knowledge to other domains or climate models remains challenging.

In conclusion, we built a machine learning surrogate model for the dynamic downscaling of GCMs. The RCM-emulator can make fast and fine-scaled SMB reproductions over the Antarctic Peninsula from GCM data. Therefore, this RCM-emulator can be an interesting tool for providing low-cost, high-resolution information on climate variables.

Appendix A Data pre-processing

A1 Pre-processing of the GCM

All RCM and GCM data pre-processing was done on Pangeo, a community platform for Big Data geoscience. First, the ACCESS 1.3 GCM data was downloaded from the Australian NCI website (`https://esgf.nci.org.au/search/esgf-nci/`). 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. This data can also be directly downloaded using the wget script on our GitHub (`https://github.com/marvande/RCM-Emulator`).

From this GCM dataset, we chose the eight variables as seen in Table 1. Next, we cut the data so that its latitude was between -40 and -90° so that it only contained the region of Antarctica. Because the GCM is in latitude-longitude coordinates, but the RCM is in polar stereographic coordinates, the GCM was reprojected to the RCM coordinate system. For this, the RCM stereographic grid was upsampled to cover approximately the same resolution as the GCM (68×206 km), and then the GCM was interpolated on that grid. Finally, the GCM variables were smoothed by a 3×3 moving average filter to follow the same pre-processing procedure as Doury et al. (2022) for their near-surface temperature emulator.

A2 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.

Variables like wind, 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 as the surface value on the highest possible pressure level.

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. 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.

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. Finally, because there is no precipitation variable in the RCM, we created one by adding the snowfall and rainfall variables (PR = SF + RF).

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

GEddxx = 90
$\Delta\phi = 90 - \text{GEddxx}$
$dr = \pi/180$
for $i, j \in X, Y$ **do**
    $\phi \leftarrow -1 * dr * (\text{lon}[i, j] + \Delta\phi)$
    cphi $\leftarrow cos(-\phi)$
    sphi $\leftarrow sin(-\phi)$
    windEast$[i, j] \leftarrow$ cphi $*$ windX$[i, j] -$ sphi $*$ windY$[i, j]$ ▷ Eastward wind component
    windNorth$[i, j] \leftarrow$ sphi $*$ windX$[i, j] +$ cphi $*$ windY$[i, j]$ ▷ Northward wind
end for
[1]

Appendix B Machine learning background

B1 Machine learning terms

- Attention: technique developed for neural networks to simulate cognitive attention so that the network pays more attention to the essential parts of the data, even if they are small. Attention mechanisms enhance some parts of the input data while reducing others (Soydaner, 2022).
- Batch normalization: standardizes the inputs given to the next layer in a deep neural network for each mini-batch. Usually, the mean and standard deviation of each input variable to a layer per mini-batch are calculated during training, and these statistics are then used to perform the standardization (Ioffe & Szegedy, 2015).
- Convolutional block attention mechanisms (CBAM): attention module for convolutional neural networks. CBAMs sequentially create two attention maps from a feature map along the channels and space dimensions. Then the attention maps are multiplied with the feature map (Woo et al., 2018; Trebing et al., 2021).
- Depth-wise separable convolutions (DSC): DSCs are divided into depth and point operations. First (depth-wise) convolutions are applied to one channel at a time (as opposed to traditional convolutions, where it's applied over all channels). Then, in a second step (point-wise), convolutions of 1×1 are applied to all channels. The advantage of DSCs over traditional convolutions is that they are computationally cheaper, i.e., they require fewer calculations and have a smaller number

of parameters, i.e., they reduce the risk of overfitting (Trebing et al., 2021; Chollet, 2017).

- (Max) pooling layers: usually added after a convolution layer and downsample a feature map by reducing its size (the number of channels is maintained). They independently apply a function (typically taking the average or maximum) over patches in each feature map. More specifically, max-pooling layers calculate the maximum value for each area (e.g., 2×2) of the feature map, and thus the size of the feature map is divided by the kernel size (e.g., 2).
- Rectified linear activation function (ReLU): a piecewise linear function that outputs an input feature map if it is positive. Otherwise, it returns zero.

B2 Evaluation metrics

For the evaluation of the RCM-emulator from the (im)perfect training frameworks, their predictions of SMB over the test period were compared to the target SMB from the RCM using different statistics: Pearson correlation coefficient, Root Mean Square Error (RMSE), and Wasserstein distance. For each emulator and point in the target domain $p \in \mathcal{E}$, we compared the target SMB time series Y_p to the predicted values $\widehat{Y_p}$ over the test period T_{test}.

- 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)

$$\mathrm{r}\left(Y_p, \widehat{Y_p}\right) = \frac{\mathrm{cov}(Y_p, \widehat{Y_p})}{\sigma(Y_p)\sigma(\widehat{Y_p})} \quad \forall p \in \mathcal{E} \tag{B1}$$

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

$$\mathrm{RMSE}\left(Y_p, \widehat{Y_p}\right) = \sqrt{\mathrm{MSE}\left(Y_p, \widehat{Y_p}\right)} \tag{B2}$$

$$= \sqrt{\frac{1}{T_{test}} \sum_t (\hat{y}_p^t - y_p^t)^2} \qquad \forall p \in \mathcal{E} \tag{B3}$$

  where $\hat{y}_p^t$ is SMB value predicted by the RCM-emulator and y_p^t the target SMB value at location $p \in \mathcal{E}$ and time step $t \in T_{test}$.
- 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 for moving the mass in the predicted measure to match that in the target

$$\mathrm{W}\left(f(Y_p), f(\widehat{Y_p})\right) = \sum_t |y_p^t - \hat{y}_p^t| \quad \forall p \in \mathcal{E} \tag{B4}$$

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

Appendix C Open Research

C1 Data availability

The ACCESS 1.3 GCM data was obtained from the Australian NCI website (`https://esgf.nci.org.au/search/esgf-nci/`). The MAR(ACCESS1.3) RCM data is from Kittel

et al. (2021). The MAR version used for the present work is tagged as v3.11.1, and the MAR outputs used in this study are available on Zenodo (`https://doi.org/10.5281/zenodo.4459259`; Kittel, 2021). In addition, the pre-processed GCM/RCM data to run the code and the saved PyTorch RCM-emulator models are available on Zenodo (`https://doi.org/10.5281/zenodo.7875882`).

C2 Code availability

The RCM-emulator architecture was implemented in PyTorch 1.11, and the machine learning training was done on Google Colab's GPU (NVIDIA Tesla K80). The up-to-date working versions of these experiments and source code are available on Zenodo (`https://doi.org/10.5281/zenodo.7875967`). All scripts needed to obtain and process input data (as described in Appendix A) can be found under the following directory (RCM-Emulator/scr/Preprocessing/). All scripts for training and evaluating the RCM-emulator are located in the (RCM-Emulator/scr/Machine-Learning/) directory. Model results are published in this directory (RCM-Emulator/results/).

Additional information about the code and data is also available via email (vandermeer@vaw.baug.ethz.ch).

Acronyms

CBAM Convolutional block attention mechanisms
DSC Depth-wise separable convolutions
GCM Global Climate Model
MAR Modèle Atmosphérique Régional
RCM Regional Climate Model
SMB Surface mass balance
UPRCM RCM upscaled to GCM resolution
GPU Graphics Processing Unit
CPU Central Processing Unit

Notation

$\boldsymbol{\mathcal{D}}$ Input Domain (dim: $[\![1, I]\!] \times [\![1, J]\!]$)
$\boldsymbol{\mathcal{E}}$ Target Domain (dim: $[\![1, K]\!] \times [\![1, L]\!]$)
$\boldsymbol{(i, j)}$ Spatial indexes over input grid (dim: $\mathcal{D}$)
$\boldsymbol{(k, l)}$ Spatial indexes over target grid (dim: $\mathcal{E}$)
$\boldsymbol{X}$ Input: 2D variables over $\mathcal{D}$ (dim: $T \times [\![1, I]\!] \times [\![1, J]\!] \times C_1$)
$\boldsymbol{Z}$ Input: 1D variables over $\mathcal{D}$ (dim: $T \times C_2$)
$\boldsymbol{Y}$ Target: SMB over $\mathcal{E}$ (dim: $T \times [\![1, K]\!] \times [\![1, L]\!]$)
$\boldsymbol{t}$ Monthly temporal index (dim: T)
$\boldsymbol{x}$ 2D variables index (dim: C_1)
$\boldsymbol{z}$ 1D variables index (dim: C_2)
$\boldsymbol{F}$ Downscaling function of the RCM
$\hat{\boldsymbol{F}}$ Emulator: estimation of F
$\hat{\mathbf{F}}_{\mathbf{P}}\mathbf{(UPRCM)}$ Emulator trained on UPRCM, prediction on UPRCM (dim: $\mathcal{E}$)
$\hat{\mathbf{F}}_{\mathbf{P}}\mathbf{(GCM)}$ Emulator trained on UPRCM, prediction on GCM (dim: $\mathcal{E}$)
$\hat{\mathbf{F}}_{\mathbf{I}}\mathbf{(GCM)}$ Emulator trained on GCM, prediction on GCM (dim: $\mathcal{E}$)

Acknowledgments

We want to express our special thanks to Christoph Kittel for providing us with the MAR(ACCESS1.3) RCM dataset and for his support. Likewise, we would like to thank Antoine Doury et al.'s team for providing information about their near-surface temperature emulator. Finally, we would also like to thank Prof. Daniel Farinotti for proofreading the manuscript and for his constructive feedback. This manuscript resulted from MSc thesis research by Marijn van der Meer at TUDelft/EPFL, supervised by Sophie de Roda Husman, Stef Lhermitte, and Martin Jaggi. Sophie de Roda Husman and Stef Lhermitte received support from the Nederlandse Organisatie voor Wetenschappelijk Onderzoek (NWO) under grant no. OCENW.GROOT.2019.091.

References

Agosta, C., Favier, V., Krinner, G., Gallée, H., & Genthon, C. (2012). Evolution of antarctic surface mass balance by high-resolution downscaling and impact on sea-level changes. *XXXII SCAR and Open Science Conference*.

Agosta, C., Fettweis, X., & Datta, R. (2015). Evaluation of the cmip5 models in the aim of regional modelling of the antarctic surface mass balance. *The Cryosphere*, *9*(6), 2311–2321. doi: 10.5194/tc-9-2311-2015

Bi, D., Dix, M., Marsland, S., O'Farrell, S., Rashid, H., Uotila, P., ... Puri, K. (2012). The access coupled model: Description, control climate and evaluation. *Australian Meteorological and Oceanographic Journal*, *63*, 41-64. doi: 10.22499/2.6301.004

Bolibar, J., Rabatel, A., Gouttevin, I., Galiez, C., Condom, T., & Sauquet, E. (2020). Deep learning applied to glacier evolution modelling. *The Cryosphere*, *14*(2), 565–584. doi: 10.5194/tc-14-565-2020

Box, J. E., & Rinke, A. (2003). Evaluation of greenland ice sheet surface climate in the hirham regional climate model using automatic weather station data. *Journal of Climate*, *16*(9), 1302 - 1319. doi: 10.1175/1520-0442(2003)16⟨1302:EOGISS⟩2.0.CO;2

Bozkurt, D., Bromwich, D., Carrasco, J., & Rondanelli, R. (2021). Temperature and precipitation projections for the antarctic peninsula over the next two decades: contrasting global and regional climate model simulations. *Climate Dynamics*, *56*, 1-22. doi: 10.1007/s00382-021-05667-2

Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*. IEEE. doi: 10.1109/cvpr.2017.195

Collier, M., & Uhe, P. (2012). The centre for australian weather and climate research cmip5 datasets from the access1.0 and access1.3 coupled climate models.

Dayon, G., Boé, J., & Martin, E. (2015). Transferability in the future climate of a statistical downscaling method for precipitation in france. *Journal of Geophysical Research: Atmospheres*, *120*(3), 1023-1043. doi: https://doi.org/10.1002/2014JD022236

Donat-Magnin, M., Jourdain, N. C., Kittel, C., Agosta, C., Amory, C., Gallée, H., ... Chekki, M. (2021). Future surface mass balance and surface melt in the amundsen sector of the west antarctic ice sheet. *The Cryosphere*, *15*(2), 571–593. doi: 10.5194/tc-15-571-2021

Doury, A., Somot, S., Gadat, S., Ribes, A., & Corre, L. (2022). Regional climate model emulator based on deep learning: Concept and first evaluation of a novel hybrid downscaling approach. *Climate Dynamics*. doi: 10.1007/s00382-022-06343-9

Draggan, S. (2009). *Antarctic peninsula* (N. C. for Science & the Environment, Eds.). Cleveland: Encyclopedia of Earth.

Dube, P., Bhattacharjee, B., Petit-Bois, E., & Hill, M. (2018). Improving transfer-

ability of deep neural networks. *CoRR*, *abs/1807.11459*. doi: 10.1007/978-3-030-30671-7_4

Erlandsen, H. B., Parding, K. M., Benestad, R., Mezghani, A., & Pontoppidan, M. (2020). A hybrid downscaling approach for future temperature and precipitation change. *Journal of Applied Meteorology and Climatology*, *59*(11), 1793 - 1807. doi: 10.1175/JAMC-D-20-0013.1

Feng, Y., & Hao, L. (2020). Testing randomness using artificial neural network. *IEEE Access*, *8*, 163685–163693. doi: 10.1109/access.2020.3022098

Fettweis, X., Box, J. E., Agosta, C., Amory, C., Kittel, C., Lang, C., ... Gallée, H. (2017). Reconstructions of the 1900–2015 greenland ice sheet surface mass balance using the regional climate mar model. *The Cryosphere*, *11*(2), 1015–1033. doi: 10.5194/tc-11-1015-2017

Fyke, J., Sergienko, O., Löfverström, M., Price, S., & Lenaerts, J. T. M. (2018). An overview of interactions and feedbacks between ice sheets and the earth system. *Reviews of Geophysics*, *56*(2), 361-408. doi: https://doi.org/10.1029/2018RG000600

Gallée, H., Agosta, C., Gential, L., Favier, V., & Krinner, G. (2011). A downscaling approach toward high-resolution surface mass balance over antarctica. *Surveys in Geophysics*, *32*, 507-518. doi: 10.1007/s10712-011-9125-3

Geyer, M., Salas Y Melia, D., Brun, E., & Dumont, M. (2013). The greenland ice sheet: modelling the surface mass balance from gcm output with a new statistical downscaling technique. *The Cryosphere Discussions*, *7*, 3163–3207. doi: 10.5194/tcd-7-3163-2013

Ghilain, N., Vannitsem, S., Dalaiden, Q., Goosse, H., De Cruz, L., & Wei, W. (2022). Large ensemble of downscaled historical daily snowfall from an earth system model to 5.5 km resolution over dronning maud land, antarctica. *Earth System Science Data*, *14*(4), 1901–1916. doi: 10.5194/essd-14-1901-2022

Giorgi, F., & Bates, G. T. (1989). The climatological skill of a regional model over complex terrain. *Monthly Weather Review*, *117*(11), 2325 - 2347. doi: 10.1175/1520-0493(1989)117⟨2325:TCSOAR⟩2.0.CO;2

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., & Pedreschi, D. (2019). A survey of methods for explaining black box models. *ACM Computing Surveys*, *51*(5), 1–42. doi: 10.1145/3236009

Hu, Z., Kuipers Munneke, P., Lhermitte, S., Izeboud, M., & van den Broeke, M. (2021). Improving surface melt estimation over the antarctic ice sheet using deep learning: a proof of concept over the larsen ice shelf. *The Cryosphere*, *15*(12), 5639–5658. doi: 10.5194/tc-15-5639-2021

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, *abs/1502.03167*.

Jouvet, G., Cordonnier, G., Kim, B., Lüthi, M., Vieli, A., & Aschwanden, A. (2022). Deep learning speeds up ice flow modelling by several orders of magnitude. *Journal of Glaciology*, *68*(270), 651–664. doi: 10.1017/jog.2021.120

Kittel, C., Amory, C., Agosta, C., Jourdain, N. C., Hofer, S., Delhasse, A., ... Fettweis, X. (2021). Diverging future surface mass balance between the antarctic ice shelves and grounded ice sheet. *The Cryosphere*, *15*(3), 1215–1236. doi: 10.5194/tc-15-1215-2021

Klaver, R., Haarsma, R., Vidale, P. L., & Hazeleger, W. (2020). Effective resolution in high resolution global atmospheric models for climate studies. *Atmospheric Science Letters*, *21*(4), e952. doi: https://doi.org/10.1002/asl.952

Kotlarski, S., Lüthi, D., & Schär, C. (2015). The elevation dependency of 21st century european climate change: an rcm ensemble perspective. *International Journal of Climatology*, *35*(13), 3902-3920. doi: https://doi.org/10.1002/joc.4254

Laprise, R., Elía, R., Caya, D., Biner, S., Lucas-Picher, P., Diaconescu, E., ... Diagnostics, C. (2008). Challenging some tenets of regional cli-

mate modelling. *Meteorology and Atmospheric Physics*, *100*, 3-22. doi: 10.1007/s00703-008-0292-9

Lenaerts, J., Lhermitte, S., Drews, R., Ligtenberg, S., Berger, S., Helm, V., ... Pattyn, F. (2017). Meltwater produced by wind–albedo interaction stored in an east antarctic ice shelf. *Nature Climate Change*, *7*, 58–62. doi: 10.1038/nclimate3180

Lenaerts, J., Medley, B., van den Broeke, M., & Wouters, B. (2019). Observing and modeling ice sheet surface mass balance. *Reviews of Geophysics*, *57*(2), 376-420. doi: https://doi.org/10.1029/2018RG000622

Misra, V. (2007). Addressing the issue of systematic errors in a regional climate model. *Journal of Climate*, *20*(5), 801 - 818. doi: 10.1175/JCLI4037.1

Moss, R. H., Edmonds, J. A., Hibbard, K. A., Manning, M. R., Rose, S. K., van Vuuren, D. P., ... Wilbanks, T. J. (2010, 01). The next generation of scenarios for climate change research and assessment. *Nature*, *463*(7282), 747-756. doi: 10.1038/nature08823

Mottram, R., Hansen, N., Kittel, C., van Wessem, J. M., Agosta, C., Amory, C., ... Souverijns, N. (2021). What is the surface mass balance of antarctica? an intercomparison of regional climate model estimates. *The Cryosphere*, *15*(8), 3751–3784. doi: 10.5194/tc-15-3751-2021

Noguer, M., Jones, R., & Murphy, J. (1998, 01). Sources of systematic errors in the climatology of a regional climate model over europe. *Climate Dynamics*, *14*(10), 691-712. doi: 10.1007/s003820050249

Pletzer, A., & Fillmore, D. (2015). Conservative interpolation of edge and face data on n dimensional structured grids using differential forms. *Journal of Computational Physics*, *302*, 21–40. doi: 10.1016/j.jcp.2015.08.029

Prechelt, L. (1998). Early stopping - but when? In *Lecture notes in computer science* (pp. 55–69). Springer Berlin Heidelberg. doi: 10.1007/3-540-49430-8_3

Rocca, M. L., & Perna, C. (2022). Opening the black box: Bootstrapping sensitivity measures in neural networks for interpretable machine learning. *Stats*, *5*(2), 440–457. doi: 10.3390/stats5020026

Ronneberger, O. (2017). Invited talk: U-net convolutional networks for biomedical image segmentation. In *Informatik aktuell* (pp. 3–3). Springer Berlin Heidelberg. doi: 10.1007/978-3-662-54345-0_3

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

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

Savage, N. (2022). Breaking into the black box of artificial intelligence. *Nature*. doi: 10.1038/d41586-022-00858-1

Scardapane, S., & Wang, D. (2017). Randomness in neural networks: an overview. *WIREs Data Mining and Knowledge Discovery*, *7*(2), e1200. doi: https://doi.org/10.1002/widm.1200

Sellevold, R., van Kampenhout, L., Lenaerts, J. T. M., Noël, B., Lipscomb, W. H., & Vizcaino, M. (2019). Surface mass balance downscaling through elevation classes in an earth system model: application to the greenland ice sheet. *The Cryosphere*, *13*(12), 3193–3208. doi: 10.5194/tc-13-3193-2019

Seroussi, H., Nowicki, S., Payne, A. J., Goelzer, H., Lipscomb, W. H., Abe-Ouchi, A., ... Zwinger, T. (2020). Ismip6 antarctica: a multi-model ensemble of the antarctic ice sheet evolution over the 21st century. *The Cryosphere*, *14*(9), 3033–3070. doi: 10.5194/tc-14-3033-2020

Sha, Y., II, D. J. G., West, G., & Stull, R. (2020). Deep-learning-based gridded downscaling of surface meteorological variables in complex terrain.

part i: Daily maximum and minimum 2-m temperature. *Journal of Applied Meteorology and Climatology*, *59*(12). doi: https://doi.org/10.1175/JAMC-D-20-0057.1

Sørland, S. L., Schär, C., Lüthi, D., & Kjellström, E. (2018). Bias patterns and climate change signals in GCM-RCM model chains. *Environmental Research Letters*, *13*(7), 074017. doi: 10.1088/1748-9326/aacc77

Soydaner, D. (2022, may). Attention mechanism in neural networks: where it comes and where it goes. *Neural Computing and Applications*, *34*(16), 13371–13385. doi: 10.1007/s00521-022-07366-3

Taylor, K. E., Stouffer, R. J., & Meehl, G. A. (2012). An overview of cmip5 and the experiment design. *Bulletin of the American Meteorological Society*, *93*(4), 485 - 498. doi: 10.1175/BAMS-D-11-00094.1

Trebing, K., Stanczyk, T., & Mehrkanoon, S. (2021). SmaAt-UNet: Precipitation nowcasting using a small attention-UNet architecture. *Pattern Recognition Letters*, *145*, 178–186. doi: 10.1016/j.patrec.2021.01.036

Vignon, E., Roussel, M.-L., Gorodetskaya, I. V., Genthon, C., & Berne, A. (2021). Present and future of rainfall in antarctica. *Geophysical Research Letters*, *48*(8), e2020GL092281. doi: https://doi.org/10.1029/2020GL092281

Woo, S., Park, J., Lee, J., & Kweon, I. S. (2018). CBAM: convolutional block attention module. *CoRR*, *abs/1807.06521*.

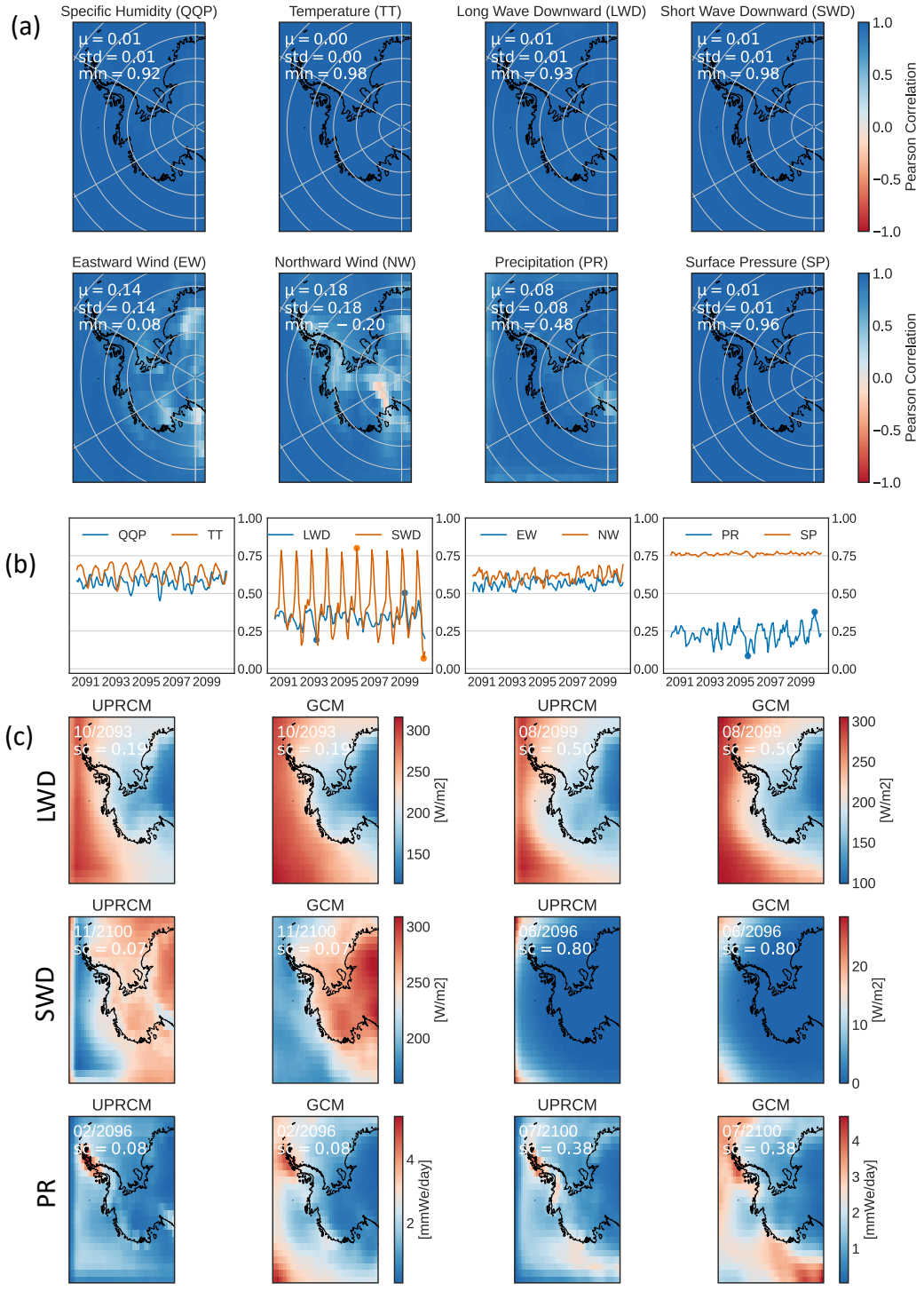

Figure 6: Temporal (a) and spatial (b, c) correlation between time series (a) and images (b, c) of upscaled RCM and GCM variables given as input to RCM-emulators over input domain $\mathcal{D}$ and test period (2090-2100). (a) Pearson correlation coefficient between upscaled RCM and GCM time series for each point in $\mathcal{E}$. Legend: mean (μ), standard deviation (std) and minimum (min) of correlation values over $\mathcal{E}$. (b) Spatial correlation between upscaled RCM and GCM variables over $\mathcal{E}$ at each time step. Legend: specific humidity (QQP), temperature (TT), long/short wave downward radiation (LWD/SWD), eastward/northward wind (EW/NW), precipitation (PR), and surface pressure (SP). (c) Example of months with lowest (left) and highest (right) spatial correlation (sc) between upscaled RCM 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).

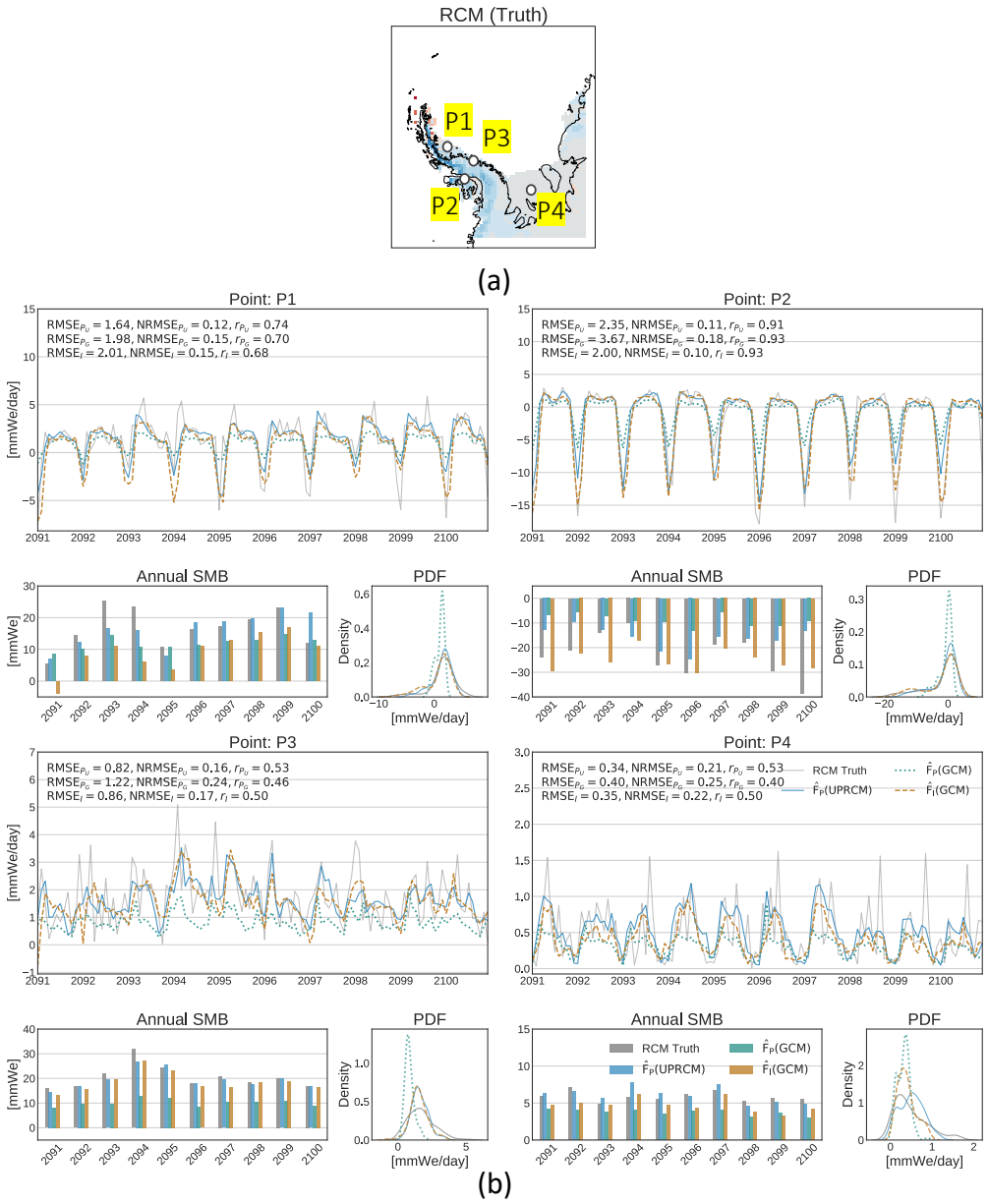

Figure 7: SMB predictions of RCM-emulators $\hat{F}_P$(UPRCM) (blue line), $\hat{F}_P$(GCM) (dotted green) and $\hat{F}_I$(GCM) (dashed orange) compared to target RCM (grey line) over test period (2090-2100). (b) Time series, daily probability density functions (PDF), and bar-plots of annual sums of 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 target SMB. For these metrics $P_U = \hat{F}_P$(UPRCM), $P_G = \hat{F}_P$(GCM) and $I = \hat{F}_I$(GCM).

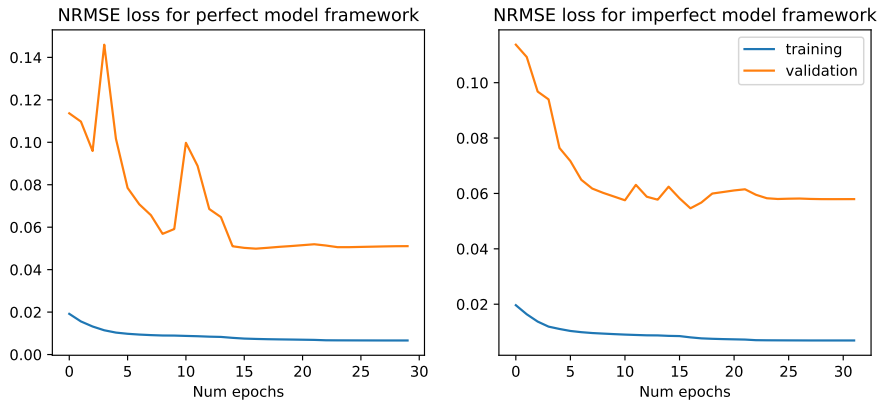

Figure B1: Training (blue) and validation NRMSE loss (orange) in the perfect (left) and imperfect model framework (right). Models trained over a maximum of 50 epochs (with early stopping) with a batch size of 100.