Data Mining and Knowledge Discovery (2023) 37:136–166
https://doi.org/10.1007/s10618-022-00877-6

ConvMOS: climate model output statistics with deep learning

Michael Steininger[1] · **Daniel Abel**[2] · **Katrin Ziegler**[2] · **Anna Krause**[1] · **Heiko Paeth**[2] · **Andreas Hotho**[1]

Received: 9 December 2021 / Accepted: 5 October 2022 / Published online: 27 October 2022
© The Author(s) 2022

Abstract

Climate models are the tool of choice for scientists researching climate change. Like all models they suffer from errors, particularly systematic and location-specific representation errors. One way to reduce these errors is model output statistics (MOS) where the model output is fitted to observational data with machine learning. In this work, we assess the use of convolutional Deep Learning climate MOS approaches and present the ConvMOS architecture which is specifically designed based on the observation that there are systematic and location-specific errors in the precipitation estimates of climate models. We apply ConvMOS models to the simulated precipitation of the regional climate model REMO, showing that a combination of per-location model parameters for reducing location-specific errors and global model parameters for reducing systematic errors is indeed beneficial for MOS performance. We find that ConvMOS models can reduce errors considerably and perform significantly better than three commonly used MOS approaches and plain ResNet and U-Net mod-

Responsible editor: Albrecht Zimmermann.

✉ Michael Steininger
steininger@informatik.uni-wuerzburg.de

Daniel Abel
daniel.abel@uni-wuerzburg.de

Katrin Ziegler
katrin.ziegler@uni-wuerzburg.de

Anna Krause
anna.krause@informatik.uni-wuerzburg.de

Heiko Paeth
heiko.paeth@uni-wuerzburg.de

Andreas Hotho
hotho@informatik.uni-wuerzburg.de

[1] Chair of Computer Science X (Data Science), University of Würzburg, Würzburg, Germany

[2] Chair of Physical Geography, University of Würzburg, Würzburg, Germany

els in most cases. Our results show that non-linear MOS models underestimate the number of extreme precipitation events, which we alleviate by training models specialized towards extreme precipitation events with the imbalanced regression method DenseLoss. While we consider climate MOS, we argue that aspects of ConvMOS may also be beneficial in other domains with geospatial data, such as air pollution modeling or weather forecasts.

Keywords Neural networks · Climate · Model output statistics

1 Introduction

An important source of information for the prospective effects of climate change are numerical climate models such as general circulation models (GCMs) and regional climate models (RCMs). However, these models often exhibit systematic errors and deficiencies in representations of climate processes which limit the quality of the resulting projections. Especially the hydrological cycle is subject to uncertainty, amplifying this problem for precipitation. It is therefore common to apply model output statistics (MOS), which are statistical post-processing techniques to reduce these errors. MOS correct the modeled precipitation to correspond more closely to observational data. With climate change becoming a more and more severe issue, we believe that it is important for the data mining community to contribute to the global effort towards assessing and combating climate change by further improving MOS performance both in the mean and for extreme events. Better MOS allows us to study future climate conditions and effects of climate change more accurately (Paeth 2011).

Currently used climate MOS approaches typically rely on standard methods from statistics and machine learning like Linear Regression (Paeth 2011) and Random Forests (RFs) (Noor et al. 2019). For each location of interest a separate model instance is fitted to reduce errors in precipitation. These models are either local when they use large-scale atmospheric conditions at that specific location or non-local, when they also consider conditions at locations nearby.

In this work, we aim to further bridge the gap between climate science and machine learning by assessing the use of convolutional Deep Learning climate MOS approaches and designing our novel climate MOS architecture ConvMOS which considers the nature of typical errors present in precipitation estimates of climate models: (i) location-specific errors stemming from poor grid point representation of land surface characteristics, e.g. topography (Paeth 2011) or great lakes (Samuelsson et al. 2010) and (ii) systematic errors originating from the use of simplified climate processes, as is often the case for cloud and rainfall formation (Paeth 2011). To efficiently reduce both types of errors, ConvMOS—as shown in Fig. 1—combines per-location model parameters, which learn to reduce errors specific to a location, and global model parameters, which learn spatial precipitation patterns to effectively reduce systematic errors in climate model outputs. Our architecture composition studies (Sect. 5.2 and Appendix A.2) show that such parameter combinations improve climate MOS performance in practice. We also consider and evaluate other popular CNN architectures for climate MOS, namely ResNets (He et al. 2016) and U-Net (Ronneberger et al. 2015).

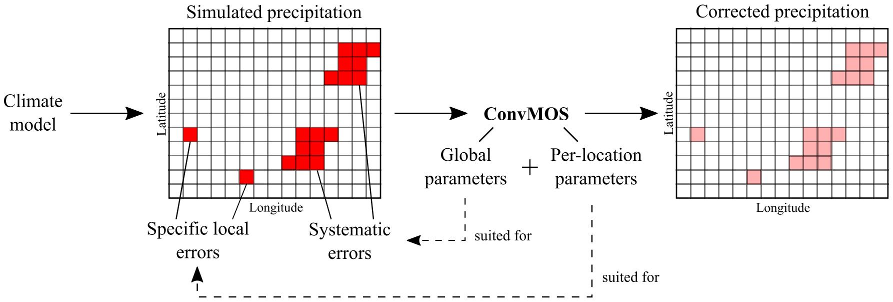

Fig. 1 ConvMOS: Systematic and location-specific errors in climate model outputs are reduced with our Deep Learning architecture that combines global and per-location parameters

We apply the approaches to correcting simulated precipitation of the RCM REMO (Majewski 1991; Jacob 2001; Jacob et al. 2001) and show that ConvMOS models reduce errors considerably, providing significantly better performance than the commonly used MOS approaches local Linear Regression (Eden and Widmann 2014), non-local Principal Component Regression (Eden and Widmann 2014), and non-local Random Forest (Sa'adi et al. 2017; Noor et al. 2019) in most cases. Additionally, we find that ConvMOS models typically perform better in comparison to plain ResNets or U-Net. Our results also show that all considered non-linear Deep Learning models underestimate the number of extreme precipitation events more than REMO and linear approaches. To alleviate this, we train ConvMOS models specialized towards estimating extreme precipitation events with the imbalanced regression method DenseLoss (Steininger et al. 2021), showing that such MOS models are better at estimating extreme precipitation events. Additional analysis is provided in the Appendix, where we analyze the training duration of the considered MOS techniques as well as MOS results over time. For this, we find no clear temporal error trends in our setting, suggesting that MOS approaches do not necessarily have to be updated over time. While we validated our approach on climate MOS, we argue that aspects of the ConvMOS architecture may also be beneficial for other applications with geospatial data, which is especially common in environmental tasks. Code and REMO data is available.[1]

We make the following contributions:

- We present a novel convolutional Deep Learning architecture for climate MOS *ConvMOS*, consisting of *local* and *global* network modules.
- We show with architecture composition studies (Sect. 5.2 and Appendix A.2) that the combination of per-location and global model parameters does indeed improve climate MOS performance.
- We compare ConvMOS to three commonly used climate MOS approaches and two popular CNN models, finding that our approach performs significantly better in most metrics.

[1] https://github.com/SteiMi/convmos An early version of this work was presented at NeurIPS 2020 Tackling Climate Change with Machine Learning Workshop (Steininger et al. 2020).

- We assess ConvMOS models specialized at estimating extreme precipitation events with the imbalanced regression method DenseLoss to allow for improved estimates for extreme events.

In Sect. 2 we discuss related research. Sect. 3 describes the data we used. We describe our proposed ConvMOS architecture for climate MOS in Sect. 4. In Sect. 5 we describe our experimental evaluation and its results. Sects. 6, and 7 discuss this work and consider its broader impact, respectively. Finally, Sect. 8 provides a conclusion.

2 Related work

The following introduces related prior work on spatio-temporal modeling, the climate MOS task considered in this work, and fully convolutional models which are related to the architecture proposed in this work.

2.1 Spatio-temporal modeling

In this work, we consider a combination of a climate model with machine learning techniques in order to provide spatio-temporal predictions of precipitation. While this is a standard approach in this particular domain, there are also other approaches to spatio-temporal modeling.

One approach is modeling spatio-temporal autocorrelation. Specific techniques include LASSO-VAR (Cavalcante et al. 2017), training Multilayer Perceptrons (MLPs) with entropy-based criteria (Ceci et al. 2019), or suitable feature extraction techniques in conjunction with tree models (Corizzo et al. 2021). There are also models which combine non-parametric tree models with parametric models for distribution tails in order to improve forecasting of extreme values (Gonçalves et al. 2021), which is similar in goal but different in technique to our experiment using a sample weighting technique for better extreme value estimation in Sect. 5.7.

Spatio-temporal forecasts are also often modeled with Deep Learning in domains like air pollution prediction, with approaches that combine temporal LSTM (Long Short-Term Memory) (Hochreiter and Schmidhuber 1997) layers with, for example, spatial attention (Shi et al. 2021), nearest neighbor approaches (Qin et al. 2019), or convolutional neural networks (CNNs) (Zhang et al. 2020). This combination of different model types for spatial and temporal aspects bears some resemblance to the approach proposed in this work, where local and global model parameters are combined to model different spatial aspects (location-specific and global, systematic errors).

The difference between the climate MOS task considered in this work and the aforementioned spatio-temporal modeling approaches is, that, strictly speaking, we do not consider climate MOS to be a forecasting task from a machine-learning-view. The temporal dynamics required for forecasts are entirely handled by the climate model. A MOS approach does not directly need to forecast future states, but only adjust the current state provided by the climate model. One may incorporate the time dimension in climate MOS approaches, but it is uncommon and may not necessarily be beneficial, which is why this work focuses on traditional non-temporal climate MOS. Nonetheless, spatio-temporal models may benefit from also consider a combination of global and local parameters for their spatial and maybe even temporal parts in order to efficiently learn both global, systematic and location- or time-specific patterns.

2.2 Climate model output statistics

There are two approaches to climate MOS—distribution-wise and event-wise MOS. Distribution-wise MOS corrects the simulated variable's distribution by mapping distribution characteristics (e.g. means) to the observed distribution. Event-wise MOS links simulated and observed time series through statistical models, which generally performs better than distribution-wise MOS (Eden and Widmann 2014). Thus, this work considers event-wise MOS.

A simple approach used by Eden and Widmann (2014) is local Linear Regression where an individual Linear Regression is fitted per location of interest, which has shown to work reasonably well. Most other works propose non-local MOS approaches, where for each location the MOS is aware of climatic conditions at nearby locations. This can lead to a large number of predictors for the MOS, which is why dimensionality reduction techniques, e.g. principal component analysis (PCA), are often applied (Paeth 2011; Eden and Widmann 2014; Sa'adi et al. 2017; Noor et al. 2019). Non-local MOS has been done with a range of machine learning models namely Linear or Principal Component Regression (Paeth 2011; Eden and Widmann 2014), Random Forests (RFs) (Sa'adi et al. 2017; Noor et al. 2019), Support Vector Machines (SVMs) (Sa'adi et al. 2017; Pour et al. 2018; Ahmed et al. 2019), and Multilayer Perceptrons (Moghim and Bras 2017).

While these methods have proven to be effective, we believe that there is considerable potential in exploring advanced Deep Learning techniques. Especially CNNs (LeCun et al. 1998) have shown proficiency in tasks with geospatial data, where each input "pixel" relates to a geographic location on Earth and provides information on the state there like prior precipitation for precipitation forecasts (Shi et al. 2017) or land-usage for air pollution estimation (Steininger et al. 2020). This indicates potential for novel non-local climate MOS with this type of neural network.

2.3 Fully convolutional networks

A core aspect of the ConvMOS architecture is the use of fully convolutional networks. These are neural networks that consist solely of convolutional layers.

Fully convolutional networks were first introduced for semantic segmentation of images in the computer vision domain (Long et al. 2015). They are useful for tasks where both the input and the output are image-like, meaning that pixels or cells are arranged in a grid. This is the case in computer vision tasks like semantic segmentation or instance segmentation (He et al. 2017). A particularly notable fully convolutional network is U-Net (Ronneberger et al. 2015) that was proposed for biomedical image segmentation and has been applied to many problems like image-to-image translation since (Kandel et al. 2020).

Fully convolutional networks are also suitable for geospatial environmental machine learning tasks like climate MOS, since the locations of a study area can be arranged in an image-like grid with the different environmental variables (e.g. precipitation) being channels of this image. One domain where they have shown good results is statistical downscaling of climate data by improving its spatial resolution through fully convolutional super-resolution CNNs (Vandal et al. 2017; Liu et al. 2020). Similarly, fully convolutional networks have been used successfully for precipitation nowcasting, which is short-term forecasting of rainfall (Agrawal et al. 2019). These positive results for similarly structured data suggests that this model type can also be beneficial for climate MOS. We believe that their ability to learn spatial patterns is also well suited for efficiently reducing systematic errors in climate models. Recent work outside of the climate domain in the related field of post-processing ensemble weather forecasts has also shown promising results by applying fully convolutional CNNs and locally connected networks that are not translation invariant (Grönquist et al. 2021). Thus, using CNNs in combination with per-location model parameters, which can reduce location-specific errors, is a promising approach for use in climate MOS.

3 Dataset

For evaluation we use the model and observational data presented next.

Model Data We use daily data of the regional climate model (RCM) REMO (hydrostatic version REMO2015) (Majewski 1991; Jacob 2001; Jacob et al. 2001) for the period 2000 to 2015. REMO is based on the Europa Modell (Majewski 1991) and the model physics of the GCM ECHAM4 (Roeckner 1996) with further improvements (e.g. Hagemann (2002); Semmler (2002); Kotlarski (2007)). The reanalysis ERA-Interim (0.75° × 0.75°) (Dee 2011; Berrisford et al. 2011) is used as forcing data, providing the lateral boundary conditions. The atmosphere's vertical resolution is represented by 27 hybrid levels with increasing distance to the atmosphere's top. In lower levels they follow the topography (Teichmann 2010). Our study area spans over an extended German region with 0.11° resolution covering the area from -1.43° to 22.22° E and 42.77° to 57.06° N (GER-11). This grid does not have 215 × 130 cells as one would think based on area and resolution but instead 121 × 121 cells since the grid is not axially parallel to latitudes or longitudes due to REMO's usage of rotated coordinates for numerical reasons (Lüthi and Heinzeller 2017). We use 23 MOS predictors (see Table 1), which all stem from REMO except for the elevation from the GTOPO dataset (0.009° × 0.009°) (DAAC 1996; Gesch et al. 1999). REMO also uses GTOPO's elevation.

Observational Data For observational data we use the gridded dataset E-OBS (Haylock et al. 2008) version 19.0e. It is based on an ensemble of interpolated station data and is therefore subject to some uncertainty, as station density varies in space and time (Cornes et al. 2018). Our predictand is E-OBS's daily precipitation sums at 0.1° resolution. Both model and observational data are interpolated bilinearly to the same 0.11° grid (Schulzweida 2019).

Table 1 MOS predictors

Predictor	Height levels	Predictor	Height levels
Temperature (K)	2 m a. s.; 100;200;500;850;950 hPa	Geopotential height (m)	100; 200; 500; 850; 950 hPa
Min. temperature (K)	2 m a. s.	Total precipitation (mm)	–
Max. temperature (K)	2 m a. s.	Specific humidity ($\mathrm{kg\,kg^{-1}}$)	100; 200; 500; 850; 950 hPa
U-wind ($\mathrm{m\,s^{-1}}$)	10 m a. s.	Sea level pressure (Pa)	–
V-wind ($\mathrm{m\,s^{-1}}$)	10 m a. s.	Elevation (m)	–

Total precipitation is the sum of snowfall, convective, and large scale precipitation. “a. s.” stands for “above surface”

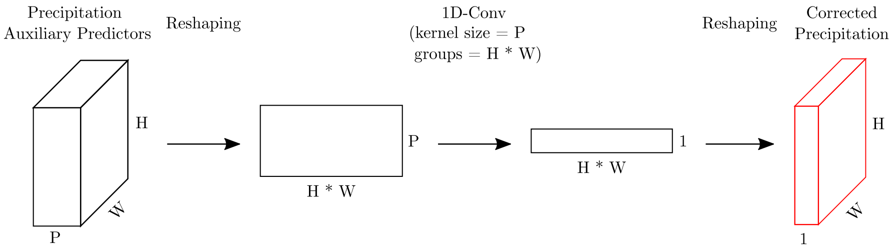

Fig. 2 Local network module's structure. H and W represent study area height and width. P is the number of predictors. Depiction is not to scale

4 ConvMOS

To explore the combination of global and per-location model parameters with CNNs as MOS we propose the architecture ConvMOS.

4.1 Idea

The basic idea of ConvMOS is derived from two sources of error in climate models: First, location-specific errors which often stem from poor grid point representation of topography. This representation can lead to abrupt topographic elevation, potentially influencing processes affecting precipitation (Paeth 2011; Eden and Widmann 2014). Second, systematic errors originating from parameterization, which replaces too complex or too small-scale processes with simpler variants. Cloud and rainfall formation is based on parameterization, leading to precipitation overestimation over land (Paeth 2011).

To efficiently reduce both types of errors, we propose a model consisting of both per-location model parameters, which can learn the characteristics of a specific location, and global model parameters, which can learn spatial precipitation patterns to efficiently help reduce systematic errors in climate models. Thus, we define two modules: local network and global network.

4.2 Local network

The local network module contains individual model parameters for each location in the study area, allowing it to reduce specific local errors. For ease of integration into the neural network architecture, we do not use a separate model (e.g. a linear regression) per location. Instead, we implement this through reshaping and a linearly activated 1D CNN, as is depicted in Fig. 2. The input at each time of size (predictors, height, width) is first reshaped so it has the dimensions (height $*$ width, predictors). In conjunction with setting the kernel size equal to the number of predictors, this allows us to group the convolution for each input channel (i.e. each location) so that each location is convolved with its own set of filters for all predictors. Thus, each location has its own model parameters, in which location characteristics can be encoded. The 1D CNN output is of shape (height $*$ width, 1) which we reshape to (1, height, width), giving

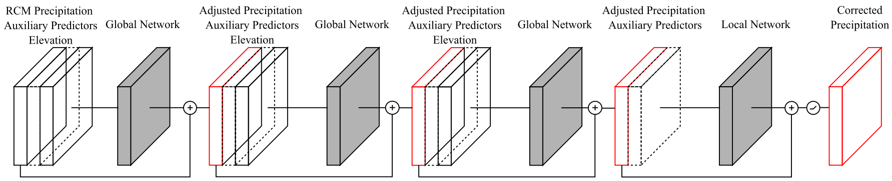

Fig. 3 ConvMOS architecture with the composition ConvMOS-gggl, having three global and one local network module

us the output of the local network module. This output can be interpreted as a grid with per-location precipitation residuals. This module is not provided with elevation data as it would be static across all times for each location.

This approach allows us to integrate per-location model parameters seamlessly into a Deep Learning model. The naive alternative of using separate models per location is harder to implement concurrently within an Deep Learning architecture running on a GPU. For Deep Learning libraries such as PyTorch (Paszke et al. 2019), which we use in this work, our module is simply another convolutional layer. This allows for efficient training and inference.

4.3 Global network

The global network learns spatial patterns in precipitation and other predictors. This can be done efficiently with CNNs (Vandal et al. 2017). The module contains a 2D CNN with four layers which should be well suited for learning useful filters which can reduce systematic errors across the study area. In addition to the local modules' predictors, the global network is also provided with elevation data for each location. In contrast to the per-location model parameters, this information is not static for the filters of the 2D CNN since the filters are applied for all locations across the study area. Starting from the first layer, the layers have 4, 8, 16, and 1 filters and kernel sizes of 9, 1, 5, and 3, respectively. Each convolutional layer has its padding parameter set to half its kernel size (rounded down to the nearest whole number) which leads to each layer's output having the same width and height as its input. All layers use a stride and a dilation of 1. The first three layers use the ReLU (Nair and Hinton 2010) activation function while the last layer is activated linearly. As with the local network, this module also outputs a grid of precipitation residuals.

4.4 Architecture

The ConvMOS architecture consists of sequentially concatenated instances of global and local network modules. Figure 3 depicts an example of a ConvMOS model. ConvMOS expects a 3D input with dimensions (predictors, height, width) for each time step. The data is sequentially passed through the modules (depicted in gray) where each module adjusts the precipitation input with the goal of reducing the error. The architecture employs so called "shortcut connections" for each module where each module's output is added to its precipitation input, which eases training for neural networks (He et al. 2016). In this work, we employ the depicted model with three global networks followed by a local network module, which is the result of our architecture composition study described in Sect. 5.2. We call this exact architecture composition ConvMOS-gggl. The global networks aim to reduce any systematic errors across the study area. Finally, the local network corrects any specific local errors and makes sure that the systematic corrections of the global network are not introducing new local errors. As precipitation cannot be negative we use a ReLU after the final shortcut connection to force positive values. The architecture is fitted with the Adam optimizer (Kingma and Ba 2014), the mean squared error (MSE) as the loss function, a learning rate of 0.001, and a batch size of 128. Only errors at locations where observational data is available were incorporated for the MSE. Training is conducted for at most 100000 epochs. Early stopping is used to stop training when the validation MSE is not improving for more than 40 consecutive epochs, preventing considerable overfitting (Caruana et al. 2001).

5 Experiment

To evaluate ConvMOS models, we apply them to the data described in Sect. 3. After defining our experimental setup, we evaluate our hypothesis regarding the benefit of combined per-location model parameters and global model parameters while also finding ConvMOS's best architecture composition for use in the experiment. We also apply standard ResNet and U-Net CNN models in addition to three commonly used MOS approaches, a local Linear Regression, a non-local Principal Component Regression approach and a non-local Random Forest method, for comparison and evaluate them for general and seasonal performance. Thereafter, we assess ConvMOS models specialized towards estimating extreme precipitation events using the imbalanced regression method DenseLoss. Additional analysis can be found in the Appendix, where we analyze the training duration of the considered MOS approaches and evaluate MOS results over time, finding no clear temporal error trends which suggests that—at least for the climate model and timespan considered in this work—MOS approaches do not necessarily have to be updated over time.

5.1 Experimental setup

We split the 16 years of daily data into a training (2000–2009), a validation (2010), and a test set (2011–2015). All predictors are standardized based on the training set so that they have a mean of zero and a standard deviation of one. Target values are not standardized and metrics are thus also computed on non-standardized data. The hyperparameter values presented in this work for the local and global network modules of our ConvMOS architecture were selected based on preliminary tests using the validation set. For evaluation, we use a number of common MOS metrics, namely root-mean-squared error (RMSE), normalized RMSE (NRMSE), Pearson Correlation, Skill score (Perkins et al. 2007), R^2, and Bias to assess different performance aspects. NRMSE divides the RMSE for each location in the study area by the difference between the maximum and minimum observed precipitation there, which we then multiply by 100 to receive a percentage. Skill score measures the common area between the probability density function of the observed precipitation and the simulated precipitation. To this end, data is binned (we use bins of 1 mm width as Perkins did) and the Skill score is both distributions' cumulative minimum value of each binned value. Thus, a perfect Skill score would be 1. R^2 describes the proportion of variance explained by a model with 1 being a perfect score. Models with R^2 lower than 0 fit worse than the data's mean. The Bias metric is the mean error. A positive value indicates overestimation of precipitation while a negative value indicates the opposite. MOS approaches with non-deterministic fitting methods, i.e. ConvMOS, ResNets, U-Net and the non-local Random Forest, are trained 20 times since performance may differ per fitted instance. All reported mean Correlations use Fisher's z-transformation (Silver and Dunlap 1987). When we report significant differences in the following, we confirmed this with a Wilcoxon signed-rank test (Wilcoxon 1945) and a significance level of 0.05.

5.2 Architecture composition study

The key idea behind ConvMOS is the combination of per-location model parameters and global model parameters which is why the architecture allows for different combinations of sequentially connected local and global network modules. In order to test whether this combination is beneficial and to find the best module arrangement we evaluate a number of composition candidates. We train 20 instances per composition on the training set and test them on the validation set. To allow for early stopping we remove the 2009 data from the training set, evaluate the model after each epoch on this data and stop training when the MSE in 2009 does not improve for more than 40 epochs in a row.

Table 2 shows mean metrics on the validation set for all study area locations available in observational data (i.e. land points) of each architecture composition sorted by RMSE. The architecture ConvMOS-gggl shows the best performance, surpassing all other tested compositions in terms of RMSE, NRMSE, Correlation, and R^2. Compared to ConvMOS-glgl with the second lowest RMSE, ConvMOS-gggl's RMSE and Bias are not significantly different but its NRMSE, Correlation and R^2 are significantly better. ConvMOS-gggl's Skill score is not significantly different from the best model for that metric (ConvMOS-glll) as well as its Bias, which is also not significantly different from the model with lowest Bias (ConvMOS-ggl). Overall, we consider ConvMOS-gggl to provide the best performance, which is why we choose this composition for our experiment. We find that, considering the results of ConvMOS-ggl and ConvMOS-gl, additional global network modules at the model's front reduces errors further, presumably since more complex spatial patterns can be learned. ConvMOS-ggl performs significantly better than ConvMOS-gl in all metrics. ConvMOS-gggl is significantly better than ConvMOS-ggl only in RMSE, NRMSE, and Correlation. This suggests diminishing improvements with more global modules. The results also show that the key idea behind ConvMOS—the combination of per-location and global model parameters—can indeed improve performance in terms of RMSE, NRMSE,

Table 2 Mean validation set metrics per architecture composition sorted by RMSE (left side $<$ right side), rounded to three decimal places

Mod.	RMSE	NRMSE	Cor.	Skill	R^2	Bias	Mod.	RMSE	NRMSE	Cor.	Skill	R^2	Bias
gggl	**3.282**	**8.817**	**.748**	.843	**0.511**	.035	lgll	3.349	9.008	.735	.824	0.489	.053
glgl	3.292	8.863	.746	.838	0.506	.067	gl	3.351	8.976	.735	.837	0.496	.051
gglg	3.298	9.442	.746	.825	−0.389	.064	llgl	3.351	9.032	.734	.817	0.483	.054
ggl	3.299	8.850	.745	.844	0.509	**.015**	llgg	3.356	9.652	.737	.821	−0.340	.078
ggll	3.302	8.861	.745	.844	0.507	.034	gggg	3.362	11.327	.744	.827	−7.511	.047
glgg	3.306	9.746	.747	.825	−1.058	.067	ggg	3.365	11.345	.744	.826	−7.555	.054
lggl	3.309	8.980	.743	.827	0.462	.054	llg	3.381	9.488	.730	.813	0.047	.091
glg	3.310	9.471	.745	.825	−0.419	.061	lg	3.382	9.579	.731	.816	−0.150	.077
gllg	3.318	9.445	.743	.825	−0.312	.071	lllg	3.386	9.467	.729	.818	0.094	.064
lglg	3.334	9.512	.740	.829	−0.235	.066	gg	3.404	11.543	.737	.824	−8.338	.085
lggg	3.344	9.885	.740	.824	−1.012	.084	g	3.486	11.701	.722	.816	−8.327	.101
lgg	3.347	9.717	.739	.822	−0.574	.066	llll	3.786	10.044	.649	.795	0.375	.189
lgl	3.347	9.004	.735	.821	0.489	.061	lll	3.790	10.054	.648	.793	0.374	.195
glll	3.348	8.968	.736	**.846**	0.496	.046	ll	3.792	10.060	.647	.792	0.373	.197
gll	3.348	8.969	.736	.843	0.496	.036	l	3.802	10.085	.644	.784	0.371	.202

Bold values indicate the best value

"Mod." is modules, "Cor." is Correlation, "g"/"l" is global/local network. RMSE and Bias are in mm, NRMSE is in %

Correlation, and R^2. The best architecture composition ConvMOS-gggl consists of a combination of different modules. Furthermore, it performs significantly better in terms of the aforementioned four metrics than the best composition consisting of solely local or global modules, namely ConvMOS-gggg. Compositions consisting solely of local modules or global modules typically perform worse than combinations of both. Additionally, we find that having a local network module as the final module provides relatively good NRMSE and R^2 values. We hypothesize that the global module's filters adjust precipitation in a similar way everywhere, leading to low performance for these metrics in some areas, e.g. when there is only relatively minor precipitation. An additional architecture composition study with U-Nets as global modules in the Appendix (see Appendix A.2) further confirms most findings presented here.

5.3 Standard climate MOS approaches

For comparison, we also evaluate standard climate MOS approaches. Similar to prior work (Paeth 2011; Sa'adi et al. 2017; Noor et al. 2019), we preprocess the standard MOS methods' predictors to reduce dimensionality and remove potentially unhelpful information. Like Sa'adi et al. (2017) and Noor et al. (2019) we use supervised PCA (Bair et al. 2006). For each location, we select the best predictors based on a univariate regression. Local MOS models choose from 23 predictors for a specific location while non-local models have another 23 predictors per considered nearby location (i.e. $11 \times 11 \times 23 = 2783$ predictors when considering locations at most 5 cells away). The number of retained predictors is set according to an exhaustive grid search at each location that considers choosing the 1 to 30 best predictors with our validation data. Then, PCA reduces the dimensionality of these predictors, keeping the first components that explain at least 95 % of the variance (Sa'adi et al. 2017). All non-Deep-Learning MOS methods described in the following use this preprocessing scheme.

Local Linear Regression (Lin) For each cell in the study area, a separate Linear Regression is fitted where simulated precipitation is the predictor and observed precipitation is the predictand. This approach is local in that each Linear Regression is unaware of conditions in nearby cells (Eden and Widmann 2014).

Non-local Principal Component Regression (NL PCR) Instead of only using large-scale conditions at a specific location for a Linear Regression, we provide all available predictors at each nearby location (at most 5 cells away in either direction) on the grid. This is feasible with the help of the supervised PCA which reduces the dimensionality of the predictors (Eden and Widmann 2014).

Non-local Random Forest (NL RF) For the non-local Random Forest, we provide all available predictors of each location ±5 cells away, as with NL PCR. The supervised PCA applied for preprocessing is also what Sa'adi et al. (2017) and Noor et al. (2019) used. Each location in our study area has its own RF instance for MOS which uses scikit-learn's RF (Pedregosa et al. 2011). Since RF performance depends considerably on its hyperparameters, we look for optimal values with a random search. For each cell we train 20 RF instances on the training set with hyperparameter values sampled randomly from the search space shown in Table 6. Each instance is evaluated on the validation set. The RF instance with the best R^2 is then applied on the test set.

5.4 Standard deep learning approaches

To further put our results in perspective, we also apply some common Deep Learning architectures. Suitable architectures allow mapping an input image to a new output image of the same size since this is structurally similar to our task of mapping an input climate to a precipitation output with the same spatial dimensions. In our experiment, we consider two commonly used architectures, namely ResNet (He et al. 2016) and U-Net (Ronneberger et al. 2015).

ResNet ResNets are popular models in computer vision which is why it is interesting to see how such a general architecture fares for climate MOS. ResNets are available with different numbers of layers. In our experiment, we used ResNet18, ResNet34, ResNet50, and ResNet101. We omit ResNet152 as its memory requirements are too large for most GPUs available to us when trained on our task and we also found no performance gains between larger and smaller ResNets anyways. The ResNets are adapted for our task by changing the number of input features in the first convolutional layer from 3 to 23 (one per predictor), removing the softmax activation necessary for classification, and replacing the final fully connected layer with one that maps to 121 x 121 (height x width) outputs. Training is conducted in the same way as for ConvMOS (i.e. same learning rate, optimizer, early stopping, loss, batch size).

U-Net Another important architecture for image-to-image tasks is U-Net. This architecture has already shown its proficiency in the related task of post-processing ensemble weather forecasts (Grönquist et al. 2021). Because of this similarity, we use their U-Net variant that differs from the standard U-Net in a few aspects: (i) Up-convolutions are replaced with bilinear interpolation followed by a 3 x 3 convolution with stride 1 to avoid checkerboard artefacts. (ii) U-Net's five levels are reduced to three levels to avoid overfitting. (iii) The number of filters per convolution are halved as they observed no improved performance with more filters.

Training is conducted in the same way as for ConvMOS (i.e. same learning rate, optimizer, early stopping, loss, batch size).

We also evaluate the use of this U-Net within the ConvMOS architecture by using it as a global network module instead of the one presented in Sect. 4.3. For this approach, we sequentially connect one global network module (here a U-Net) and one local network module, which is the resulting composition of the architecture composition study in the Appendix (see Appendix A.2). This is similar to the model proposed by Grönquist et al. (2021) for their weather forecasting task but with a ConvMOS local network module after the U-Net instead of their locally connected network. This approach is denoted as *ConvMOS-UNet* or short *CM-UNet* in the following.

5.5 Results

Table 3 shows mean metrics on the test set for all study area locations available in observational data. All MOS approaches improve all metrics considerably when compared to applying no MOS, except for the Skill score. This suggests that REMO's precipitation distribution at land locations is already rather close to that of the observations with a Skill score of 0.93 and can barely be improved by MOS methods. All

Table 3 Test set mean metrics for all locations having observational data

Metric MOS	RMSE (mm)	NRMSE (%)	Cor.	Skill	R^2	Bias (mm)
REMO raw	5.32	15.83	0.49	**0.91**	−28.24	0.31
Lin	3.51	8.03	0.58	0.47	0.33	−0.03
NL PCR	3.37	7.80	0.62	0.81	0.36	0.02
NL RF	3.39 ± 0.00	7.82 ± 0.00	0.61	0.82	0.36 ± 0.00	0.03 ± 0.00
ResNet18	3.03 ± 0.01	7.04 ± 0.03	0.71	0.60	0.47 ± 0.01	−0.06 ± 0.07
ResNet34	3.06 ± 0.02	7.10 ± 0.04	0.71	0.61	0.46 ± 0.01	−0.07 ± 0.09
ResNet50	3.04 ± 0.01	7.05 ± 0.03	0.71	0.61	0.47 ± 0.00	−0.10 ± 0.10
ResNet101	3.03 ± 0.02	7.04 ± 0.04	0.71	0.64	0.47 ± 0.01	−0.04 ± 0.08
U-Net	2.97 ± 0.02	8.37 ± 0.12	**0.74**	0.82	−5.60 ± 0.88	−0.03 ± 0.08
CM-UNet	**2.92 ± 0.01**	7.01 ± 0.11	**0.74**	0.70	0.13 ± 0.22	**0.01 ± 0.10**
ConvMOS	2.93 ± 0.02	**6.77 ± 0.05**	0.73	0.89	**0.51 ± 0.02**	−0.10 ± 0.05

Bold values indicate the best value
Values rounded to two decimal places. Std. dev. for Correlation (always 0.00) and Skill score (between 0.00 and 0.03) omitted for brevity

Deep-Learning-based MOS approaches perform better than standard approaches in terms of RMSE, NRMSE, Correlation and R^2, except for U-Net's NRMSE as well as U-Net's and CM-UNet's R^2. We find that U-Net and, to a lesser extent, CM-UNet struggle at some locations as can be seen in the Appendix' Fig. 7 for NRMSE. These low performance locations typically have very low precipitation, with which these models in particular have issues. The two ConvMOS models combining local and global model weights—CM-UNet and ConvMOS—tend to perform best. CM-UNet provides significantly better RMSE than all other approaches except for ConvMOS. CM-UNet's correlation is also significantly better than all other MOS methods except for U-Net, while ConvMOS is also only closely behind. For NRMSE, Skill score, and R^2, ConvMOS is significantly better than all other MOS approaches. This indicates that ConvMOS-based approaches can estimate precipitation more accurately than all considered comparison methods. ConvMOS's Skill score is close but still reduced slightly by 0.02 compared to REMO's. ConvMOS shows less Bias than REMO but it seems to have a tendency to underestimate precipitation as most approaches do. CM-UNet tends to show the lowest Bias. We also ran this experiment with precipitation as the only climate predictor as some prior work has done (Eden and Widmann 2014; Noor et al. 2019) but found all considered methods to perform worse without additional predictors.

Figure 4 visualizes RMSEs for all locations with observational data across the study area for all assessed approaches using each method's best instance with regard to test RMSE. Similarly as in Table 3, all MOS methods can reduce errors from the original REMO output. Especially precipitation in the Alps and other mountainous regions is improved considerably. We find that CNN approaches tend to provide lower errors compared to other MOS methods in general but also for seemingly difficult areas. All standard MOS approaches show a bright yellow spot near the border between Italy and

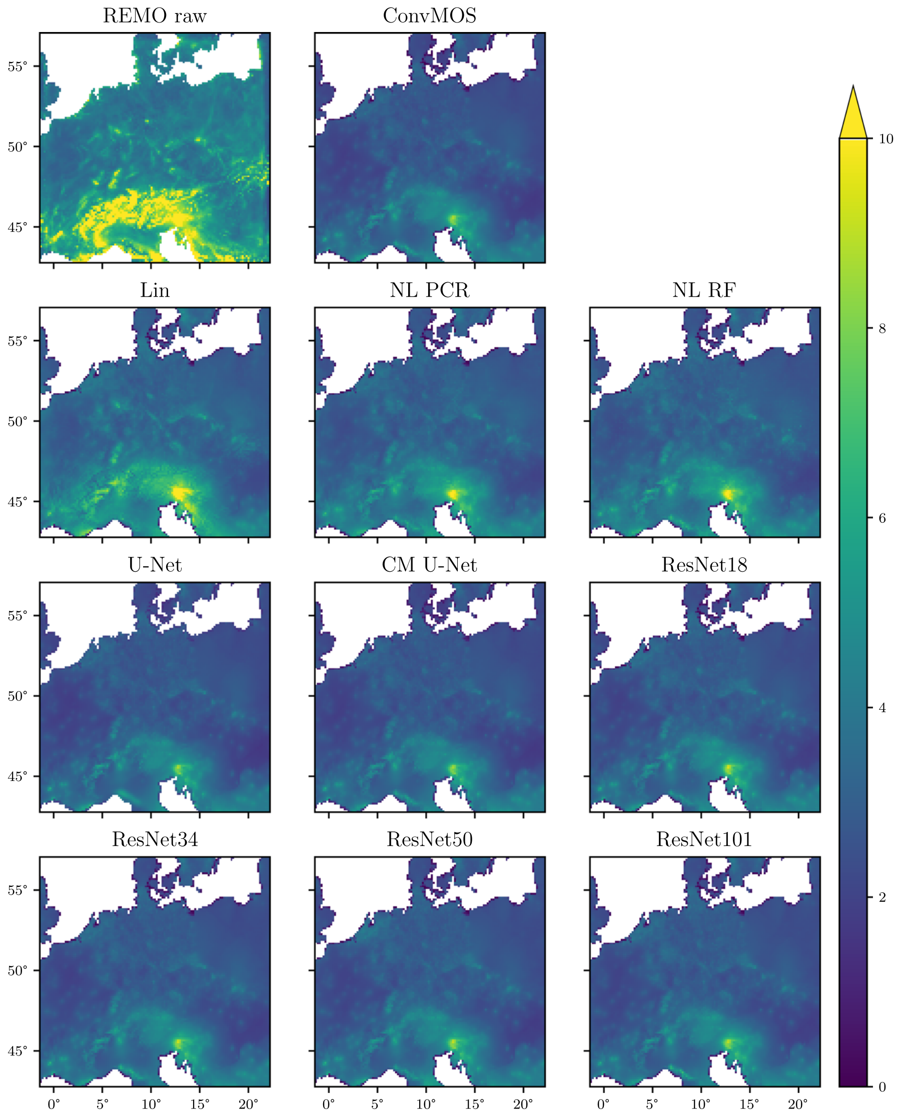

Fig. 4 RMSE of precipitation in mm for the test set across the study area. *Note* REMO raw has cells with far larger RMSE than 10 mm but we limited the colorbar's extent for better visibility of general performance

Slovenia indicating high error and difficulty there, that is less pronounced for CNN models. In the observational E-OBS data for this area we noticed that there tends to be higher precipitation during the test time frame compared to the training time frame. We hypothesize that especially the standard MOS approaches have difficulties due to this shift in the precipitation distribution there.

Figure 5 depicts the daily precipitation distributions on the test set for all locations with observational data for E-OBS's observed precipitation, REMO's precipitation,

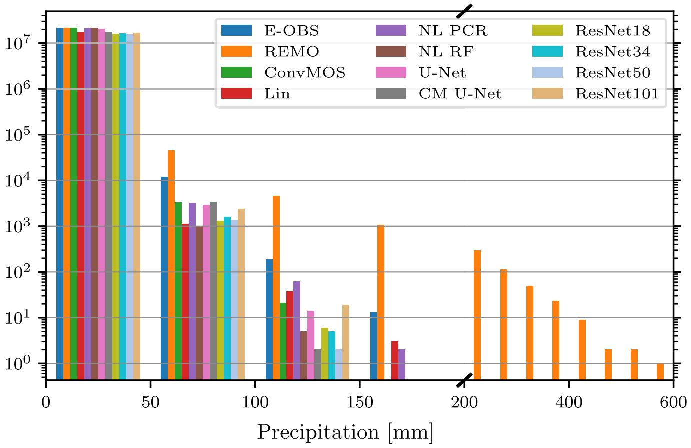

Fig. 5 Daily precipitation distributions on the test set. The 12 bins are 50 mm wide, starting at 0 mm. The y-axis (number of events) is scaled logarithmically and the x-axis (precipitation) is compressed over 200 mm for brevity

and the outputs of each MOS method's best model instance (i.e. lowest test RMSE). It shows that REMO often simulates considerably more precipitation than ever observed despite the good mean Skill score per location, showing the weaknesses of its hydrological cycle. All MOS approaches underestimate the number of high precipitation events ($\geq$ 50 mm). NL RF and all Deep Learning models are particularly conservative about their estimates, showing considerably fewer events with more than 100 mm than both linear MOS approaches and E-OBS. This indicates room for improvement when considering relatively rare extreme precipitation events with non-linear MOS.

5.6 Seasonal results

To assess whether the MOS approaches fitted with training data covering entire years exhibit seasonal anomalies, we also evaluate them per season. Table 4 shows the mean RMSE per season on the test set for all study area locations available in observational data. The seasons are DJF (December–February), MAM (March–May), JJA (June–August), and SON (September–November).

The seasonal results show that all MOS methods reduce errors across the year. REMO seems to have more problems estimating precipitation during summer and autumn (i.e. JJA and SON) for this study area which also results in larger RMSE of MOS outputs in these seasons. As with the overall RMSE, CM-UNet and ConvMOS are providing the best RMSE across the seasons with similarly low standard deviations. ConvMOS is slightly better in the more difficult JJA and SON seasons while CM-UNet is best during DJF and MAM. This difference is statistically significant for all seasons. All Deep Learning models that do not combine local and global weights are better than the standard approaches but worse than ConvMOS and CM-UNet. NL PCR and NL RF have similar performance and both tend to perform better than Lin.

Table 4 Seasonal RMSE in mm for all locations with observational data

Season MOS	DJF	MAM	JJA	SON
REMO raw	4.59	4.50	6.28	5.13
Lin	2.64	3.04	4.22	3.74
NL PCR	2.44	2.89	4.20	3.48
NL RF	2.48 ± 0.00	2.94 ± 0.00	4.18 ± 0.00	3.51 ± 0.00
ResNet18	2.14 ± 0.03	2.62 ± 0.01	3.77 ± 0.01	3.18 ± 0.02
ResNet34	2.18 ± 0.03	2.64 ± 0.01	3.79 ± 0.03	3.22 ± 0.02
ResNet50	2.13 ± 0.02	2.62 ± 0.01	3.77 ± 0.01	3.19 ± 0.02
ResNet101	2.13 ± 0.03	2.61 ± 0.02	3.77 ± 0.02	3.18 ± 0.02
U-Net	2.11 ± 0.02	2.54 ± 0.02	3.68 ± 0.03	3.09 ± 0.03
CM-UNet	**2.06 ± 0.02**	**2.51 ± 0.01**	3.64 ± 0.02	**3.04 ± 0.02**
ConvMOS	2.09 ± 0.02	2.52 ± 0.01	**3.63 ± 0.02**	**3.04 ± 0.02**

Bold values indicate the best value

Values rounded to two decimal places. "DJF" is December–February, "MAM" is March–May, "JJA" is June–August, "SON" is September–November

5.7 Focusing on extreme precipitation estimation

Our results show that non-linear models underestimate the number of extreme precipitation events more severely than REMO and linear approaches (see Fig. 5). These events can have negative effects on society and the environment like floods (Kundzewicz 2003), impact on plants (Zeppel et al. 2014) or increased disease spread (Chen et al. 2012). As such, it can be of interest to train models that perform particularly well for estimating the number and intensity of extreme events. Thus, we adapt ConvMOS-gggl to improve extreme precipitation estimation as it is among the best models in our experiment. In the following, we consider daily precipitation of at least 50 mm as extreme which is also the threshold at which the German Meteorological Service gives out a stage 3 precipitation warning for very dangerous weather (Deutscher Wetterdienst 2021).

One technique for training regression models with more emphasis on performance for rare data points in comparison to common data points is *DenseLoss* (Steininger et al. 2021). It estimates the target variable's density function from the training data points and gives each training data point a weight based on each sample's target value density. These weights are higher for samples in relatively rare parts of the target value range (i.e. extreme precipitation samples) in comparison to samples from more common parts of the target value range (i.e. precipitation closer to 0 mm). A sample's weight influences how much the error of that sample influences model training, leading to models better suited for estimating rare data points such as samples with extreme precipitation. The magnitude of weighting differences between samples with different rarity is configured through α. Through preliminary tests on the validation set, we found $\alpha = 1.0$ to provide the lowest RMSE for extreme samples which is why we set α to one. DenseLoss' minimal weight threshold ϵ is set to 10^{-6} as in the original paper. We modify the early stopping procedure to consider the validation MSE of extreme samples only and train 20 instances of ConvMOS with DenseLoss, which we call ConvMOS-DL in the following. DenseLoss shifts the model's focus towards extreme precipitation events due to which we expect model performance for non-extreme samples to degrade to some extent while extreme data points are estimated more accurately.

To asses performance for extreme precipitation events, we split all samples into two bins, evaluating all occurrences of at least 50 mm separately from those with lower precipitation in the test set. The test set contains 12240 extreme and 21331874 non-extreme samples. Due to the rarity of extreme events we can not calculate meaningful mean metrics per cell but instead report mean metrics over all samples of a bin. In addition to the RMSE, we also evaluate how well a model can distinguish between extreme and non-extreme samples. To this end, we calculate a recall per bin and the balanced accuracy, which is defined as the mean of the extreme and non-extreme recalls. We consider a prediction accurate if it is lower than 50 mm for non-extreme samples and at least 50 mm for extreme samples. Table 5 shows RMSE and recall for REMO's raw output, ConvMOS, and ConvMOS-DL for non-extreme and extreme samples as well as the models' balanced accuracies. As expected, ConvMOS performs better in terms of both metrics for non-extreme data points in comparison to ConvMOS-DL and REMO raw, while the model using DenseLoss is still better than the raw REMO output. For extreme precipitation events, we see significantly better performance with ConvMOS-DL compared to ConvMOS. ConvMOS-DL can correctly identify on average 20.99 % of the extreme samples while ConvMOS only identifies 11.94 % correctly. REMO raw is closer to ConvMOS-DL's recall on extreme samples with 20.03 % but has considerably higher RMSE. When considering balanced accuracy, we find that ConvMOS-DL can distinguish best between extreme and non-extreme samples while REMO is similarly skilled in this aspect. Improved prediction of extreme values with DenseLoss can also be seen in a histogram, where the distribution is visibly closer to the observed precipitation (see Appendix). All in all, we find that DenseLoss can be used to train climate MOS models better suited for the analysis of extreme precipitation. Such models provide lower general performance but can distinguish better between extreme and non-extreme events while also showing lower errors for extreme precipitation events.

6 Discussion

In this work, we have shown that convolutional climate MOS and especially ConvMOS models can improve the quality of precipitation data significantly. However, we also found that especially non-linear approaches tended to perform poorly for the estimation of extreme precipitation events. We were able to alleviate this by training models specialized for extreme events with DenseLoss but ideally we could train models that perform well for both extreme and non-extreme precipitation events. Approaches to consider in the future for this may be uncertainty quantification methods which explicitly model uncertainty and, thus, may provide estimates that better follow the desired distribution (Abdar et al. 2021). It remains to be seen whether such techniques help the estimates' distribution to become closer to the real distribution while keeping metrics like RMSE at similarly low or even lower levels as reported here.

Table 5 Test set mean metrics for extreme/non-extreme observed precipitation ($\geq$ or $<$ 50 mm)

Metric	$<$ 50 mm		$\geq$ 50 mm		Balanced Acc. [%]
MOS	RMSE [mm]	Acc. Recall [%]	RMSE [mm]	Acc. Recall [%]	
REMO raw	5.75	99.77	47.48	20.03	59.90
ConvMOS	**2.94 ± 0.01**	**100.00 ± 0.00**	35.29 ± 0.76	11.94 ± 2.13	55.97 ± 1.07
ConvMOS-DL	4.15 ± 0.12	99.99 ± 0.00	**32.01 ± 0.64**	**20.99 ± 2.19**	**60.49 ± 1.09**

Bold values indicate the best value
"Acc." is accuracy. Values rounded to two decimal places

The MOS methods evaluated in this work only consider the spatial but not directly the temporal aspect of this task. The climate state at a particular time is dependent on the previous states and in our case only the climate model takes this into account. It is possible that including information of earlier time steps within the climate MOS models can help improve performance even further. It would therefore be interesting to consider this for future work.

As usual with machine learning techniques, it is often important to set suitable hyperparameters to achieve decent performance with a specific estimator. While it is feasibly possible to optimize the hyperparameters even for each location individually with the NL RF baseline, it is considerably more complex to tune Deep Learning models due to the enormous number of hyperparameters to consider and the dependencies between hyperparameters (e.g. CNN kernel sizes affect the output tensor shape, which can affect the structure of all following layers). For this reason, we only conducted limited hyperparameter tuning for ConvMOS and CM-UNet (e.g. architecture composition studies) and no tuning for the baseline ResNet and U-Net architectures. While the performance for all Deep Learning approaches and especially the ResNets and U-Nets may be further improved to some extent, this does not affect the main point of this work, namely that a combination of global and location-specific model parameters is beneficial as shown in both architecture composition studies. We furthermore believe that using pre-defined ResNets and U-Nets from prior work is an interesting baseline as these are likely models a practitioner would use, especially if the hardware and time is not available for more involved hyperparameter searches when conducting a climate study.

In contrast to reducing errors with climate MOS after running a climate model, a different approach to improving climate data is to directly reduce the source of errors in climate models. Uncertainties in climate models are primarily caused by the approximation of complex, high resolution processes through so-called parametrizations. To this end, there is work on learning better parametrizations with Deep Learning techniques, but they are not good enough yet to be used in practice (Rasp et al. 2018). Until these problems are solved, climate MOS methods like those considered in this work can be used as an effective tool for correcting climate model outputs.

7 Broader impact

The experiments conducted in this work consider climate MOS specifically and show that ConvMOS's combination of global and local model parameters are beneficial for the estimation quality. However, we believe that other domains may also benefit from aspects of ConvMOS's architecture. Location-specific parameters allow for the implicit encoding of a location's special characteristics during training, which we suspect to also be beneficial for other domains with geospatial data, where models like CNNs with their global model parameters are generally used on their own. Such data is common in environmental machine learning tasks like air pollution modeling or weather forecasting. For example, air pollution forecasting approaches like the one proposed by Zhang et al. (2020) use a CNN-based spatial feature extractor where each input "pixel" corresponds to a specific location that has its specific characteristics. We believe that the combination of the existing CNN-based model for the efficient extraction of spatial features with a model containing per-location weights is likely to improve the overall model, as it is now able to encode location-specific characteristics that may be important for air pollution modeling.

Within the climate domain, this work provides a powerful new tool with ConvMOS. We hope to promote the application of ConvMOS through our publicly available code. This allows researchers conducting climate studies to apply our technique in order to provide them with more accurate data.

Besides the methodological and practical impact, we hope to foster more interest with our work in the data mining and machine learning community towards novel contributions for environmental tasks. Environmental issues like climate change are among the most pressing issues of our time and we believe that our community can provide important contributions for understanding, mitigation, and adaption of and to these processes, as is laid out in more detail in Rolnick et al. (2022).

8 Conclusion

In this work, we assessed convolutional Deep Learning climate MOS approaches and presented our ConvMOS architecture that is built specifically to reduce location-specific errors as well as systematic errors in climate model outputs. We applied ConvMOS models to the output of the RCM REMO in order to reduce errors in its simulated precipitation. In our architecture composition study, we showed that the combination of per-location model parameters and global model parameters is beneficial for MOS performance. Furthermore, our MOS approach is able to improve daily precipitation data considerably while also providing significantly better performance than three commonly used MOS approaches and plain ResNet and U-Net models in most cases. We also showed that issues of non-linear Deep Learning MOS for estimating extreme precipitation events can be alleviated by training models specialized for extreme events with the imbalanced regression method DenseLoss. Improvements in MOS allow for more accurate climate data especially at high spatial resolutions which allows us to better assess the effects of climate change. While ConvMOS is designed with climate MOS in mind, we believe that the architecture's combination of location-specific and global model parameters can also be beneficial for other tasks with geospatial data (e.g. air pollution modeling, weather forecasting), which opens interesting opportunities for future work.

Acknowledgements This research was conducted in the BigData@Geo project supported by the European Regional Development Fund (ERDF).

Funding Open Access funding enabled and organized by Projekt DEAL.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

A Appendix

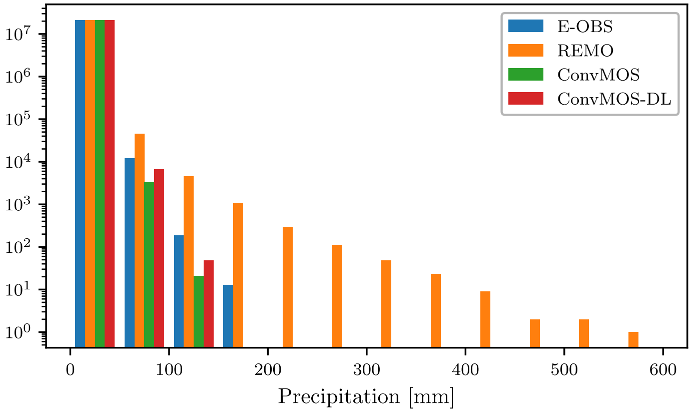

Fig. 6 Daily precipitation distribution on the test set for E-OBS's observations, the estimates of REMO, ConvMOS, and ConvMOS-DL. The 12 bins begin at 0 mm and are 50 mm wide. The y-axis (occurrences) is scaled logarithmically

Table 6 RF hyperparameter search space. "HP" stands for hyperparameter

HP	Range	HP	Range	HP	Range
n_estimators	10–2000	min_samples_split	2–10	max_depth	10–110
max_features	0.01–1.0	min_samples_leaf	1–10	bootstrap	T or F

A.1 Model training time

Applying MOS can provide more accurate climate data but it comes with an additional time burden due to the MOS's training procedure. To quantify this time burden fairly, we fit each MOS approaches five times on the same hardware—in contrast to the cluster of heterogeneous hardware used in the main experiment—and measure the training time.

All Deep Learning models (i.e. ResNets, U-Nets, CM-UNet, ConvMOS) are trained on a single Nvidia RTX 2080 TI GPU (Graphics Processing Unit), which is relatively affordable consumer hardware in comparison to expensive data center GPUs. These models are implemented in PyTorch 1.7.1 (Paszke et al. 2019) with CUDA 11.0. The other models (i.e. Lin, NL PCR, NL RF) are fitted using 15 cores of an AMD Epyc 7502P processor, which is not a standard consumer but a more expensive data center CPU (Central Processing Unit). These non-GPU models are implemented in Scikit-learn 0.23.2 (Pedregosa et al. 2011).

Table 7 shows the mean training duration in hours per MOS approach in this training duration experiment. Both NL RF and NL PCR stand out with relatively long training duration. As with Lin, NL RF and NL PCR fit one model per location, which is time

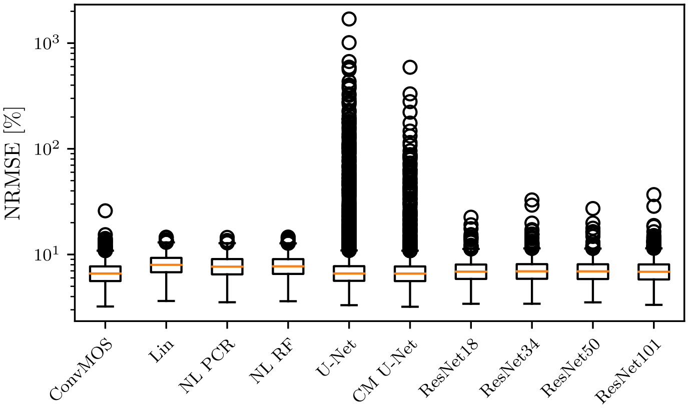

Fig. 7 Mean test NRMSEs per location. U-Net and CM U-Net show high NRMSE on mostly low-precipitation locations in contrast to all other models

Table 7 Mean training duration in hours per MOS approach

MOS	Train duration [h]	Hardware	MOS	Train duration [h]	Hardware
Lin	0.03 ± 0.00	CPU	ResNet50	0.49 ± 0.05	GPU
NL PCR	58.16 ± 25.33	CPU	ResNet101	0.62 ± 0.05	GPU
NL RF	93.32 ± 1.47	CPU	U-Net	0.42 ± 0.03	GPU
ResNet18	0.38 ± 0.03	GPU	CM-UNet	0.44 ± 0.05	GPU
ResNet34	0.49 ± 0.04	GPU	ConvMOS	1.14 ± 0.25	GPU

consuming for large study areas like the one used here with 121 × 121 locations. However, NL RF's and NL PCR's long training times are mostly due to supervised PCA. NL RF takes longer than NL PCR due to the higher model complexity and the per-location hyperparameter tuning, which we employ for optimal performance (see Sect. 5.3). All Deep Learning approaches are trained in under two hours. ConvMOS's training duration is comparatively long and shows high variance. Regardless, there is no large practical difference between these Deep Learning approaches with regard to training duration since all train relatively quickly. All in all, we consider these training times—except for NL PCR and NL RF—minor in comparison to the time needed for the climate simulations of the climate model, which usually takes multiple days. The Deep Learning approaches are faster in settings with large study areas while providing better performance, as seen in this work's main experiment.

A.2 Architecture composition study for CM-UNet

ConvMOS's architecture composition study shows that a combination of local and global modules is beneficial. We further confirm this and optimize CM-UNet's module composition by conducting the architecture composition study again with U-Net as the global module (CM-UNet). The experimental setup is the same except for the different global modules and the batch size of 64 instead of 128 due to GPU memory

Table 8 Mean validation metrics per CM-UNet composition sorted by RMSE (left < right), rounded to three decimal places. "Mod." is modules, "Cor." is Correlation, "g"/"l" is global/local network. RMSE, Bias in mm. NRMSE in %

Mod.	RMSE	NRMSE	Cor.	Skill	R^2	Bias	Mod.	RMSE	NRMSE	Cor.	Skill	R^2	Bias
gl	**3.264**	8.858	.751	.842	0.475	−0.009	glg	3.303	9.522	.749	.830	−0.416	0.056
ggll	3.269	8.883	.750	.836	0.461	0.019	glgg	3.306	9.777	.750	.827	−1.048	0.072
glll	3.269	**8.823**	.751	**.845**	**0.501**	0.002	lggg	3.307	10.211	.749	.827	−2.458	0.038
gglg	3.270	9.372	.751	.833	−0.290	**−0.001**	g	3.310	10.918	.753	.832	−5.774	0.040
gll	3.271	8.856	.751	.839	0.484	0.040	lgg	3.313	10.262	.749	.823	−2.590	0.013
gggl	3.272	9.029	.751	.842	0.337	0.022	gg	3.313	10.894	.753	.826	−5.479	0.039
ggl	3.272	8.978	.751	.832	0.393	0.050	lg	3.320	10.279	.747	.821	−2.679	0.037
glgl	3.274	9.029	.751	.834	0.345	0.020	llgg	3.327	10.158	.744	.821	−2.073	0.048
lgll	3.277	8.968	.749	.833	0.405	0.027	ggg	3.329	10.957	.752	.824	−5.529	0.079
lgl	3.277	9.092	.749	.837	0.255	−0.010	llg	3.339	10.102	.743	.827	−1.877	0.005
lggl	3.286	9.182	.748	.834	0.156	0.027	lllg	3.343	9.997	.741	.832	−1.532	−0.004
llgl	3.299	9.177	.745	.833	0.213	0.019	llll	3.787	10.050	.648	.795	0.375	0.188
gggg	3.301	10.888	**.754**	.830	−5.586	0.014	lll	3.789	10.053	.648	.794	0.374	0.190
lglg	3.302	9.674	.747	.824	−0.762	0.034	ll	3.793	10.063	.647	.792	0.373	0.196
gllg	3.303	9.550	.748	.829	−0.507	0.041	l	3.802	10.086	.643	.783	0.370	0.200

Bold values indicate the best value

limitations with compositions consisting of three or four U-Nets. We do not expect the change in batch size to affect the comparison considerably.

Table 8 shows validation set mean metrics for all locations with observational data of each CM-UNet architecture composition sorted by RMSE. The composition CM-UNet-gl provides the lowest RMSE as well as Correlation, Skill score, and Bias that are not significantly different to the composition with the best value in the respective metric. While it is not best in NRMSE and R^2, we choose this composition for our experiments due to its low RMSE and it being among the best compositions with regard to the other metrics. Again, we find that the combination of per-location and global model parameters can improve performance in terms of RMSE, NRMSE and R^2, where CM-UNet-gl provides significantly better performance in comparison to the best composition consisting solely of global or local modules, namely CM-UNet-gggg. Compositions without both global and local modules typically perform worse than combinations of both. An exception are Correlations, where CM-UNet-gggg performs best significantly but performs subpar especially for NRMSE and R^2. This study confirms again that a local network as the final module provides relatively good NRMSE and R^2.

A.3 Estimation quality over time

This work considers MOS where temporal climate dynamics are entirely modeled by the climate model. Daily precipitations are adjusted disregarding time. Since MOS use training data from a certain time range, it is interesting to consider error trends with increasing distance to this time period. Distributions produced by climate models may change over time, possibly leading to issues for MOS. We investigate this by analyzing the test set performance over time.

Figure 8 visualizes daily RMSE of precipitation over the test set time range for REMO, ConvMOS, and NL PCR, smoothed with a moving average window of 14

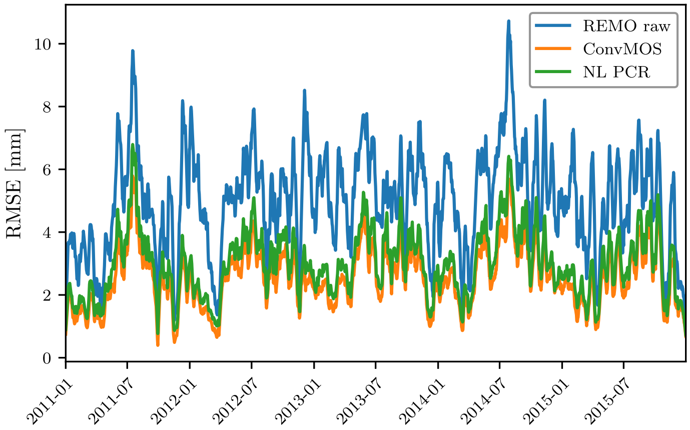

Fig. 8 RMSE of precipitation in mm for the test set across the study area over time. The graph is smoothed using a moving average window of 14 days

Table 9 Test set mean RMSE in mm per year for all locations having observational data. Values rounded to two decimal places

Year MOS	2011	2012	2013	2014	2015
REMO raw	4.93	5.20	5.45	5.72	4.86
Lin	3.41	3.32	3.55	3.80	3.31
NL PCR	3.26	3.20	3.44	3.67	3.11
NL RF	3.27 ± 0.00	3.19 ± 0.00	3.45 ± 0.00	3.70 ± 0.00	3.15 ± 0.00
ResNet18	2.93 ± 0.02	2.83 ± 0.02	3.08 ± 0.02	3.31 ± 0.01	2.85 ± 0.02
ResNet34	2.97 ± 0.02	2.87 ± 0.02	3.11 ± 0.02	3.34 ± 0.02	2.86 ± 0.02
ResNet50	2.92 ± 0.02	2.84 ± 0.01	3.08 ± 0.02	3.31 ± 0.02	2.86 ± 0.02
ResNet101	2.92 ± 0.03	2.84 ± 0.02	3.07 ± 0.02	3.32 ± 0.02	2.84 ± 0.02
U-Net	2.87 ± 0.03	2.78 ± 0.02	3.00 ± 0.02	3.21 ± 0.02	2.80 ± 0.02
CM-UNet	**2.82 ± 0.02**	**2.73 ± 0.02**	**2.96 ± 0.02**	**3.17 ± 0.01**	**2.76 ± 0.02**
ConvMOS	**2.82 ± 0.02**	2.75 ± 0.01	**2.96 ± 0.02**	3.18 ± 0.02	**2.76 ± 0.02**

Bold values indicate the best value

Table 10 Test set mean RMSE relative to REMO's RMSE in % per year for all locations having observational data. Values rounded to percentages

Year MOS	2011	2012	2013	2014	2015
REMO raw	100	100	100	100	100
Lin	69	64	65	67	68
NL PCR	66	62	63	64	64
NL RF	66 ± 0	61 ± 0	63 ± 0	65 ± 0	66 ± 0
ResNet18	59 ± 0	55 ± 0	57 ± 0	58 ± 0	59 ± 0
ResNet34	60 ± 0	55 ± 0	57 ± 0	58 ± 0	59 ± 0
ResNet50	59 ± 0	55 ± 0	57 ± 0	58 ± 0	59 ± 0
ResNet101	59 ± 1	55 ± 0	56 ± 0	58 ± 0	58 ± 0
U-Net	58 ± 1	**53 ± 0**	55 ± 0	56 ± 0	58 ± 1
CM-UNet	**57 ± 0**	**53 ± 0**	**54 ± 0**	**55 ± 0**	**57 ± 0**
ConvMOS	**57 ± 0**	**53 ± 0**	**54 ± 0**	56 ± 0	**57 ± 0**

Bold values indicate the best value

days to reduce noise. It shows that MOS RMSE mostly follows REMO's RMSE but on a lower level. We find no noticeable trend in RMSE for the models depicted, as well as the other MOS approaches. We come to the same conclusion when considering Table 9, which shows the absolute RMSE per year for all MOS approaches and REMO, and Table 10, which shows the relative RMSE per year for each MOS approach as percentages of REMO's RMSE. Especially the latter table shows for all MOS techniques only small RMSE fluctuations of at the very most 5 % relative to REMO's RMSE, suggesting that MOS error trends follow REMO's error trends.

While the limited timespan available does not allow for a conclusive answer regarding error trends for longer timespans, the data does suggest that time difference between training data and test data may have no or only a minor influence on errors. Nevertheless, when considering longer timespans than five years, the climate model output's distribution may change to such an extent, that there may be a noticeable effect. We suggest to analyze this in climate studies that apply MOS in order to detect potential issues with distributional shifts.

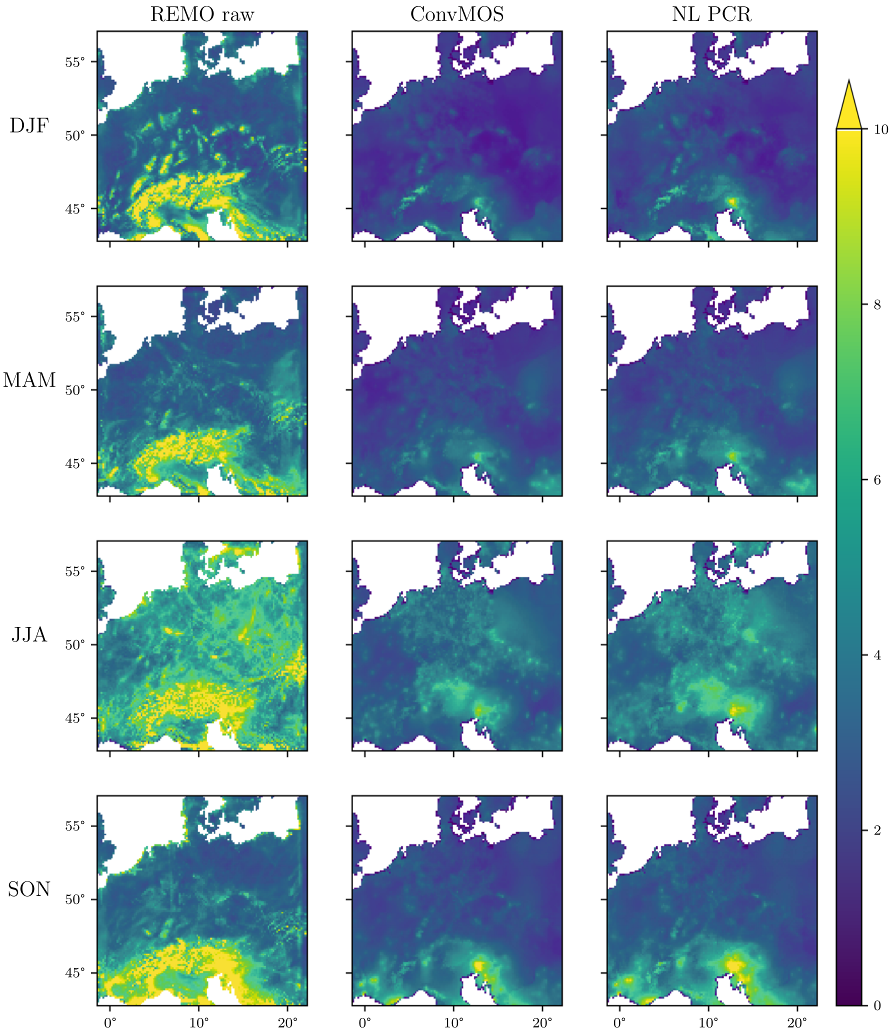

Fig. 9 Test RMSE in mm per location and season (DJF is December, January, and February; MAM is March, April, and May; JJA is June, July, and August; SON is September, October, and November). REMO has larger RMSEs than 10 mm but the colorbar's extent is limited to better show general performance

A.4 Seasonal results over the study area

Seasonal results are visualized in Fig. 9 for REMO's raw output, ConvMOS, and NL PCR, which is overall the best standard MOS approach in terms of RMSE. We show the same model instances as in Fig. 4. During all seasons, we find REMO's largest errors in mountainous regions like the Alps. This also shows in the RMSE of both ConvMOS's and NL PCR's output where these areas often continue to have more pronounced errors. The season with the largest error JJA shows more evenly distributed large RMSE values across the study area compared to the other seasons, resulting also in comparatively large RMSE in the MOS outputs. The relatively large overall RMSE values of season SON concentrate in the Alps and the Mediterranean coast, while RMSE for cells north of the Alps seem similar to those during seasons DJF and MAM. Matching the findings of Table 4, we tend to see lower RMSE with ConvMOS in comparison to NL PCR. For example, the latter has more difficulties in reducing the large errors near the border between Italy and Slovenia and we also often see slightly larger RMSEs north of the Alps in comparison to our approach. These results show that ConvMOS can be better than standard MOS approaches at improving precipitation estimates regardless of the season.

References

Abdar M et al (2021) A review of uncertainty quantification in deep learning: techniques, applications and challenges. In: Information fusion

Agrawal S, Barrington L, Bromberg C, Burge J, Gazen C, Hickey J (2019) Machine learning for precipitation nowcasting from radar images. arXiv:1912.12132

Ahmed K, Shahid S, Nawaz N, Khan N (2019) Modeling climate change impacts on precipitation in arid regions of Pakistan: a non-local model output statistics downscaling approach. Theor Appl Climatol 137:1–2

Bair E, Hastie T, Paul D, Tibshirani R (2006) Prediction by supervised principal components. In: JASA 101.473, pp 119–137

Berrisford P et al (2011). The ERA-interim archive. Version 2.0. In: ECMWF

Caruana R, Lawrence S, Giles CL (2001) Overfitting in neural nets: backpropagation, conjugate gradient, and early stopping. In: NeurIPS

Cavalcante L, Bessa RJ, Reis M, Browell J (2017) LASSO vector autoregression structures for very short-term wind power forecasting. Wind Energy 20(4):657–675

Ceci M, Corizzo R, Malerba D, Rashkovska A (2019) Spatial autocorrelation and entropy for renewable energy forecasting. Data Min Knowl Discov 33(3):698–729

Chen M-J, Lin C-Y, Wu Y-T, Wu P-C, Lung S-C, Su H-J (2012) Effects of extreme precipitation to the distribution of infectious diseases in Taiwan, 1994–2008. PLoS ONE 7(6):e34651

Corizzo R, Ceci M, Fanaee-T H, Gama J (2021) Multi-aspect renewable energy forecasting. Inf Sci 546:701–722

Cornes RC, van der Schrier G, van den Besselaar EJ, Jones PD (2018) An ensemble version of the E-OBS temperature and precipitation data sets. J Geophys Res Atmosp 123(17):9391–9409

DAAC, EDC (1996) GTOPO 30 Database. In: US Geological Survey

Dee D et al (2011) The ERA-Interim reanalysis: configuration and performance of the data assimilation system. Q J R Meteorol Soc 137:553–597

Deutscher Wetterdienst (2021) Warnkriterien. https://www.dwd.de/DE/wetter/warnungen_aktuell/kriterien/warnkriterien.html

Eden JM, Widmann M (2014) Downscaling of GCM-simulated precipitation using model output statistics. JCLI 27(1):312–324

Gesch DB, Verdin KL, Greenlee SK (1999) New land surface digital elevation model covers the earth. Eos 80(6):69–70. https://doi.org/10.1029/99EO00050

Gonçalves C, Cavalcante L, Brito M, Bessa RJ, Gama J (2021) Forecasting conditional extreme quantiles for wind energy. Electr Power Syst Res 190:106636

Grönquist P et al (2021) Deep learning for post-processing ensemble weather forecasts. Philos Trans R Soc A 379(2194):20200092

Hagemann S (2002). An improved land surface parameter dataset for global and regional climate models. https://doi.org/10.17617/2.2344576

Haylock MR, Hofstra N, Klein Tank AMG, Klok EJ, Jones PD, New M (2008) A European daily high-resolution gridded data set of surface temperature and precipitation for 1950–2006. JGR Atmosp 113(20):D20119

He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. CVPR 2016:770–778

He K, Gkioxari G, Dollár P, Girshick R (2017) Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp 2961–2969

Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780

Jacob D (2001) A note to the simulation of the annual and inter-annual variability of the water budget over the Baltic Sea drainage basin. Meteorol Atmosp Phys 77(1–4):61–73

Jacob D et al (2001) A comprehensive model inter-comparison study investigating the water budget during the BALTEX-PIDCAP period. Meteorol Atmosp Phys 77(1–4):19–43

Kandel ME et al (2020) Phase imaging with computational specificity (PICS) for measuring dry mass changes in sub-cellular compartments. Nat Commun 11(1):1–10

Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv:1412.6980

Kotlarski S (2007) A Subgrid Glacier Parameterisation for Use in Regional Climate Modelling. Ph.D. thesis. Hamburg, p 178

Kundzewicz ZW (2003) Extreme precipitation and floods in the changing world. IAHS Publ 281:32–39

LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. In: IEEE 86.11, pp 2278–2324

Liu Y, Ganguly AR, Dy J (2020) Climate downscaling using YNet: a deep convolutional network with skip connections and fusion. In: KDD 2020

Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3431–3440

Lüthi D, Heinzeller D (2017) Leitfaden zur Nutzung dynamischer regionaler Klimamodelle. In: promet 99, p 49

Majewski D (1991) The Europa-modell of the Deutscher Wetterdienst. In: ECMWF "numerical methods in atmospheric models" 2, pp 147–191

Moghim S, Bras RL (2017) Bias correction of climate modeled temperature and precipitation using artificial neural networks. J Hydrometeorol 18:1867–1884

Nair V, Hinton GE (2010) Rectified linear units improve restricted Boltzmann machines. In: ICML

Noor M, Ismail T bin, Ullah S, Iqbal Z, Nawaz N, Ahmed K (2019) A non-local model output statistics approach for the downscaling of CMIP5 GCMs for the projection of rainfall in Peninsular Malaysia. In: JWCC

Paeth H (2011) Postprocessing of simulated precipitation for impact research in West Africa. Part I: model output statistics for monthly data. Clim Dyn 36(7–8):1321–1336

Paszke A et al (2019) PyTorch: an imperative style, high-performance deep learning library. In: NeurIPS. Curran Associates, Inc., pp 8024–8035

Pedregosa F et al (2011) Scikit-learn: machine learning in python. In: JMLR

Perkins S, Pitman A, Holbrook N, McAneney J (2007) Evaluation of the AR4 climate models' simulated daily maximum temperature, minimum temperature, and precipitation over Australia using probability density functions. J Clim 20(17):4356–4376

Pour SH, Shahid S, Chung E-S, Wang X-J (2018) Model output statistics downscaling using support vector machine for the projection of spatial and temporal changes in rainfall of Bangladesh. Atmos Res 213:149–162

Qin Z, Cen C, Guo X (2019) Prediction of air quality based on KNN-LSTM. J Phys Conf Ser 1237:4

Rasp S, Pritchard MS, Gentine P (2018) Deep learning to represent subgrid processes in climate models. PNAS 115(39):9684–9689

Roeckner E et al (1996) The atmospheric general circulation model ECHAM4: model description and simulation of present-day climate. Max-Planck-Institute of Meteorology, Technical report Hamburg, p 171

Rolnick D et al (2022) Tackling climate change with machine learning. ACM Comput Surv. https://doi.org/10.1145/3485128

Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. In: MICCAI. Springer, pp 234–241

Sa'adi Z, Shahid S, Chung E-S, Ismail T (2017) Projection of spatial and temporal changes of rainfall in Sarawak of Borneo Island using statistical downscaling of CMIP5 models. Atmos Res 197:446–460

Samuelsson P, Kourzeneva E, Mironov D (2010) The impact of lakes on the European climate as simulated by a regional climate model. Boreal Environ Res 15:113–129

Schulzweida U (2019). CDO. https://doi.org/10.5281/zenodo.3539275

Semmler T (2002) Der Wasser- und Energiehaushalt der arktischen Atmosphre. PhD thesis. Hamburg, pp 1–123

Shi X et al (2017) Deep learning for precipitation nowcasting: a benchmark and a new model. In: NeurIPS, pp 5617–5627

Shi G, Leung Y, Zhang JS, Fung T, Du F, Zhou Y (2021) A novel method for identifying hotspots and forecasting air quality through an adaptive utilization of spatio-temporal information of multiple factors. Sci Total Environ 759:143513

Silver NC, Dunlap WP (1987) Averaging correlation coefficients: should Fisher's z transformation be used? J Appl Psychol 72(1):146

Steininger M, Abel D, Ziegler K, Krause A, Paeth H, Hotho A (2020) Deep learning for climate model output statistics. arXiv:2012.10394

Steininger M, Kobs K, Davidson P, Krause A, Hotho A (2021) Density-based weighting for imbalanced regression. Mach Learn 110(8):2187–2211

Teichmann C (2010) Climate and air pollution modelling in south America with focus on megacities. Ph.D. thesis. Hamburg, p 167

Vandal T, Kodra E, Ganguly S, Michaelis A, Nemani R, Ganguly AR (2017) Deepsd: generating high resolution climate change projections through single image super-resolution. KDD 2017:1663–1672

Wilcoxon F (1945) Individual comparisons by ranking methods. Biomet Bull 1(6):80–83

Zeppel M, Wilks JV, Lewis JD (2014) Impacts of extreme precipitation and seasonal changes in precipitation on plants. Biogeosciences 11:11

Zhang Q, Lam JC, Li VO, Han Y (2020) Deep-AIR: a hybrid CNN-LSTM framework for Fine-grained air pollution forecast. arXiv:2001.11957

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.