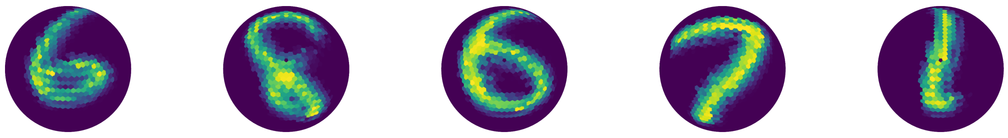

Figure A.1. *Validation task for the icosahedral neural network: shown are randomly selected examples of handwritten digits from the MNIST data set, projected onto the icosahedral grid, in this case the refinement level of the icosahedral grid $r = 4$. For the validation tasks, three versions of this data set are generated: One where the digits are not rotated ("N"), one in which rotations of the symmetry group of the icosahedron are applied ("I") and one in which we apply rotations of the group of rotations of the sphere("R").*

A. Supplementary Material

A.1. Implementation Details

We use pytorch (Paszke et al., 2019) to implement the CNNs and scikit-learn (Pedregosa et al., 2011) for the baseline models. Our code is publically available on GitHub.[5]

A.1.1. Validation Experiment

The validation experiment is a variation of one of the seminal tasks in ML: the classification of the MNIST data set of handwritten digits (LeCun et al., 2010). As a toy example, Cohen et al. (2019) project the digits onto the southern hemisphere of the icosahedral grid. Some examples are visualized in Figure A.1. Then, three types of data sets are produced: the digits can either be left as they are (“non-rotated”) - or the icosahedral grid can be rotated before the digits are projected onto it, either by a symmetry transformation of the icosahedron or by an unconstrained rotation (“fully-rotated”). Because the network architecture of Cohen et al. (2019) is equivariant to rotations of the icosahedron, the network should be able to classify digits that have been rotated by elements of this symmetry group with the same accuracy as non-rotated digits, even if during training it was never shown any rotated digits.

The data set creation process, our network architecture (Figure A.2), and the training procedure for this task follow Cohen et al. (2019) as closely as possible. We train for 60 epochs on the non-rotated and fully-rotated data sets and for one epoch on the icosahedrally rotated data set, which is 60 times larger than the other sets. We use a cross-entropy loss function and the Adam optimizer (Kingma & Ba, 2017) with a learning rate of 0.001 and the other parameters at their default values $\beta_{1,2} = (0.9, 0.999)$, $\epsilon = 10^{-8}$.

Our results in Table A.1 are comparable or better than those reported in Cohen et al., 2019. Notably, they demonstrate the equivariance of our implementation - the classification results for digits rotated by an icosahedral symmetry are almost identical to those of non-rotated digits.[6] Small differences between the results of our implementation and the implementation of Cohen et al., 2019 can partly be attributed to the fact that none of our hyperparameters were tuned - and to implementation details not described in Cohen et al. (2019), like the batch size or the used optimizer.

A.1.2. Isotope Emulation Experiment

UNet Training. To optimize the UNet models, we use the Adam optimizer (Kingma & Ba, 2017), with `lr` $= 0.001$ and the other parameters at their default values: $\beta_{1,2} = (0.9, 0.999)$, $\epsilon = 10^{-8}$. We use early stopping with a patience of 5, i.e. we abort training if no global minimum of the validation set loss is

[5] https://github.com/paleovar/isoEm/releases/v1.0.

[6] We did not fix a random seed, therefore, the results of N/N and N/I are not exactly identical.

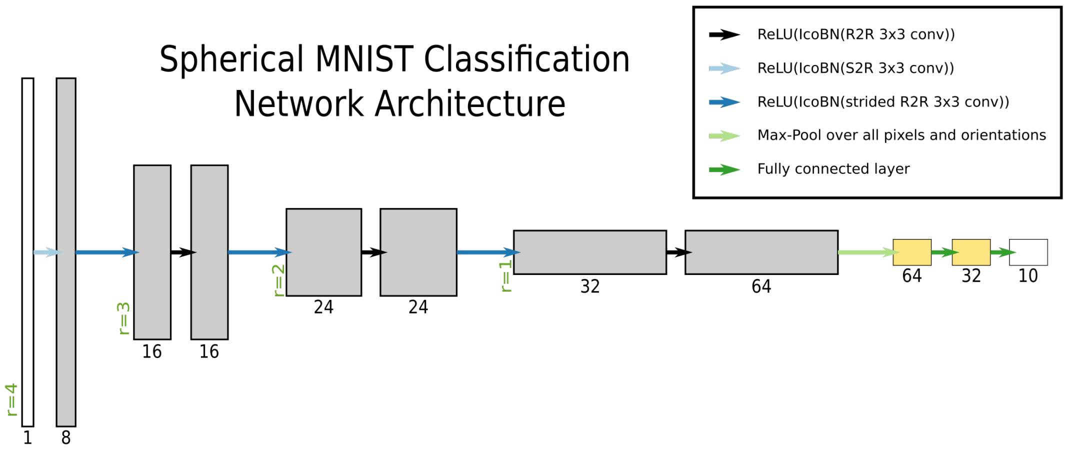
Spherical MNIST Classification Network Architecture
ReLU(IcoBN(R2R 3x3 conv))
ReLU(IcoBN(S2R 3x3 conv))
ReLU(IcoBN(strided R2R 3x3 conv))
Max-Pool over all pixels and orientations
Fully connected layer
r=4
r=3
r=2
r=1
1
8
16
16
24
24
32
64
64
32
10

Figure A.2. *Sketch of the architecture used to validate our implementation of the icosahedral network of Cohen et al. (2019). Spatial resolutions (green text) relate to the refinement level of the icosahedral grid. The architecture was chosen according to the details given in the supplementary material of the original paper. "S2R" and "R2R" are specific types of convolutional layers developed to maintain equivariance, and IcoBN is an adapted version of batch normalization. For details, see Cohen et al. (2019).*

Table A.1. *Classification accuracies (fraction of correctly classified to the total number of tested digits) for the icosahedral MNIST validation experiment (Cohen et al., 2019). "N", "I" and "R" indicate data sets, where either no rotations, rotations of the symmetry group of the icosahedron, or rotations of the symmetry group of the sphere were applied. Shown results are averages over three runs.*

Training set rot. type/Test set rot. type	N/N	N/I	N/R	I/I	I/R	R/R
Cohen et al.	**99.43**	**99.43**	**69.99**	**99.38**	66.26	99.31
Ours	99.23	99.34	68.57	99.27	**69.31**	**99.37**

reached in 5 consecutive epochs. As a result, the training runs are stopped after roughly 20 epochs. We use a batch size of 8. Training a model takes on the order of minutes to tens of minutes for a single run on a basic graphics card (NVIDIA GeForce GTX1650).

For both icosahedral and flat UNets, we use batch normalization (BN, Ioffe & Szegedy, 2015) and ReLU-activations after all but the last convolutional layer. We use 2x2 max-pooling to go to coarser spatial resolution and nearest neighbor interpolation (plate carrée projection) and a modified form of bilinear interpolation (icosahedral grid) to increase resolution. Data from skip connections and upsampling are simply concatenated.

Icosahedral UNet Approach. We create the icosahedral data set using an icosahedral grid at refinement level $r = 5$, meaning we apply 5 recursive steps in which each of the triangular faces of the icosahedron is divided into four smaller triangles, with the grid points subsequently projected back to the spherical surface. This results in a grid with $5 \times 2^{2r+1} + 2 = 10242$ vertices, while the iHadCM3 latitude-longitude grid encompasses 6816 grid points. Because the areas covered by the iHadCM3 grid cells vary with latitude, the resolution of the icosahedral grid is higher than the resolution of the iHadCM3 grid close to the equators, while close to the poles iHadCM3 data is better resolved than the icosahedral grid.

The architecture used for the icosahedral grid is visualized in Figure A.12. It is inspired by an architecture in Cohen et al. (2019). There is an implementation uncertainty remaining that relates to

Table A.2. *Effect of modifications to flat UNet architecture, globally averaged R^2 scores. Shown are standard deviations and mean over ten runs.*

Use CoordConv	Use area-weighted loss	Use cylindrical padding	R^2score
No	No	No	0.352 ± 0.015
No	No	Yes	0.357 ± 0.015
No	Yes	No	0.365 ± 0.005
Yes	No	No	0.367 ± 0.010
Yes	Yes	Yes	0.377 ± 0.005

the treatment of the vertices of the icosahedron at $r = 0$, i.e. the 12 corners of the unrefined grid. In opposition to all the other pixels in the icosahedral grid, these possess only five neighboring pixels (instead of six), thus introducing irregularities in the convolution patterns. Unable to extract the exact treatment of these corner pixels in Cohen et al. (2019), we made the choice to set these corner pixel values to zero in every layer. This will likely introduce biases at very coarse resolutions (i.e. r small).

We use the MSE-Loss from Equation (2) without weighting, because all pixels in the icosahedral grid cover an approximately equal area. Padding is done similarly to Cohen et al. (2019) in a way that asserts continuity between the faces of the icosahedral grid.

Flat UNet. The default architecture for the flat UNets is visualized in Figure A.11. By default, we use zero padding to keep the resolution before and after convolutions identical, in most experiments, however, only the latitudes get zero-padded, while we pad the longitudes cyclically to avoid discontinuities.

A.2. Investigating reasons for cross-prediction performance drops

We investigate the drops in R^2 score that occur when predicting with a model trained on iHadCM3 data on simulation data from other climate models. This experiment and its results were described in Section 3.4. As described in the main text, a possible explanation might be differences in the statistical connections between δ^{18}O and the predictor variables amongst the different climate models. To estimate these differences, we calculate the root of the squared differences in correlation coefficients between each model and iHadCM3 grid-box wise:

$$\left(\left(r^{\text{Model}}_{\text{d18O,prec}} - r^{\text{iHadCM3}}_{\text{d18O,prec}}\right)^2 + \left(r^{\text{Model}}_{\text{d18O,tsurf}} - r^{\text{iHadCM3}}_{\text{d18O,tsurf}}\right)^2\right)^{\frac{1}{2}} \tag{A.1}$$

Here r indicates the (temporal) Pearson correlation coefficients computed for each model and grid box. The results are visualized in panels (B2) to (D2) of Figure A.3.

Additionally, the generalization to other models might be impaired in regions where there are weak or no statistical relationships between δ^{18}O and the predictor variables in the iHadCM3 data. This may occur due to iHadCM3 misrepresenting the δ^{18}O relationships in the Earth system - or there might just be no strong connections between δ^{18}O and the predictor variables in these regions. In a simplistic approach, we assess this by plotting regions in which neither the correlation of temperature to δ^{18}O nor the correlation of precipitation amount and δ^{18}O exceeds an absolute value of 0.25 as hatches in Figure A.3 (B4) to (D4).

In panels (B4) to (D4), we plot the difference in cross-prediction performance between each model and iHadCM3, whose training set was used to train the emulator. We observe that especially for the ENSO region west of South America and over the North Atlantic Ocean (a region relevant for the North Atlantic Oscillation, see Wanner et al., 2001) a decline in performance coincides with changes in the correlation structure between the models. Over the southern oceans, there are both changes in correlation structure between the models and weak correlations for iHadCM3.

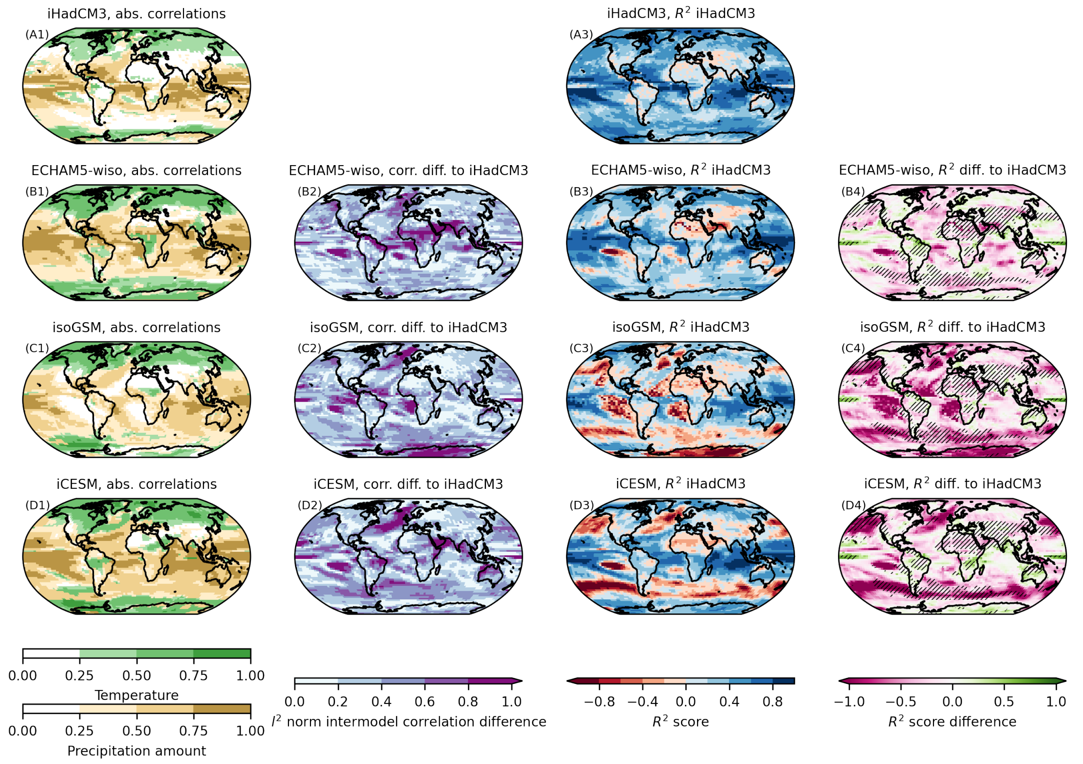
iHadCM3, abs. correlations
(A1)
iHadCM3, R^2 iHadCM3
(A3)
ECHAM5-wiso, abs. correlations
(B1)
ECHAM5-wiso, corr. diff. to iHadCM3
(B2)
ECHAM5-wiso, R^2 iHadCM3
(B3)
ECHAM5-wiso, R^2 diff. to iHadCM3
(B4)
isoGSM, abs. correlations
(C1)
isoGSM, corr. diff. to iHadCM3
(C2)
isoGSM, R^2 iHadCM3
(C3)
isoGSM, R^2 diff. to iHadCM3
(C4)
iCESM, abs. correlations
(D1)
iCESM, corr. diff. to iHadCM3
(D2)
iCESM, R^2 iHadCM3
(D3)
iCESM, R^2 diff. to iHadCM3
(D4)
0.00 0.25 0.50 0.75 1.00
Temperature
0.00 0.25 0.50 0.75 1.00
Precipitation amount
0.0 0.2 0.4 0.6 0.8 1.0
l^2 norm intermodel correlation difference
−0.8 −0.4 0.0 0.4 0.8
R^2 score
−1.0 −0.5 0.0 0.5 1.0
R^2 score difference

Figure A.3. *Investigating reasons for emulation quality decrease in cross-prediction experiments:* (A1) *to* (D1) *show the absolute correlations between* δ^{18}O *and the predictor variables. For each grid cell, only the larger of the two absolute correlation coefficients is shown.* (B2) *to* (D2) *show the differences in the correlation structure between the models as computed in Equation* (A.1). (A3) *to* (D3) *show the cross-prediction* R^2 *scores of the best deeper flat UNet model and are identical to the content of Figure* 5. (B4) *to* (D4) *show* R^2 *scores differences between each model and iHadCM3. In hatched regions, no correlation coefficient has an absolute value bigger than 0.25 in the iHadCM3 data set.*

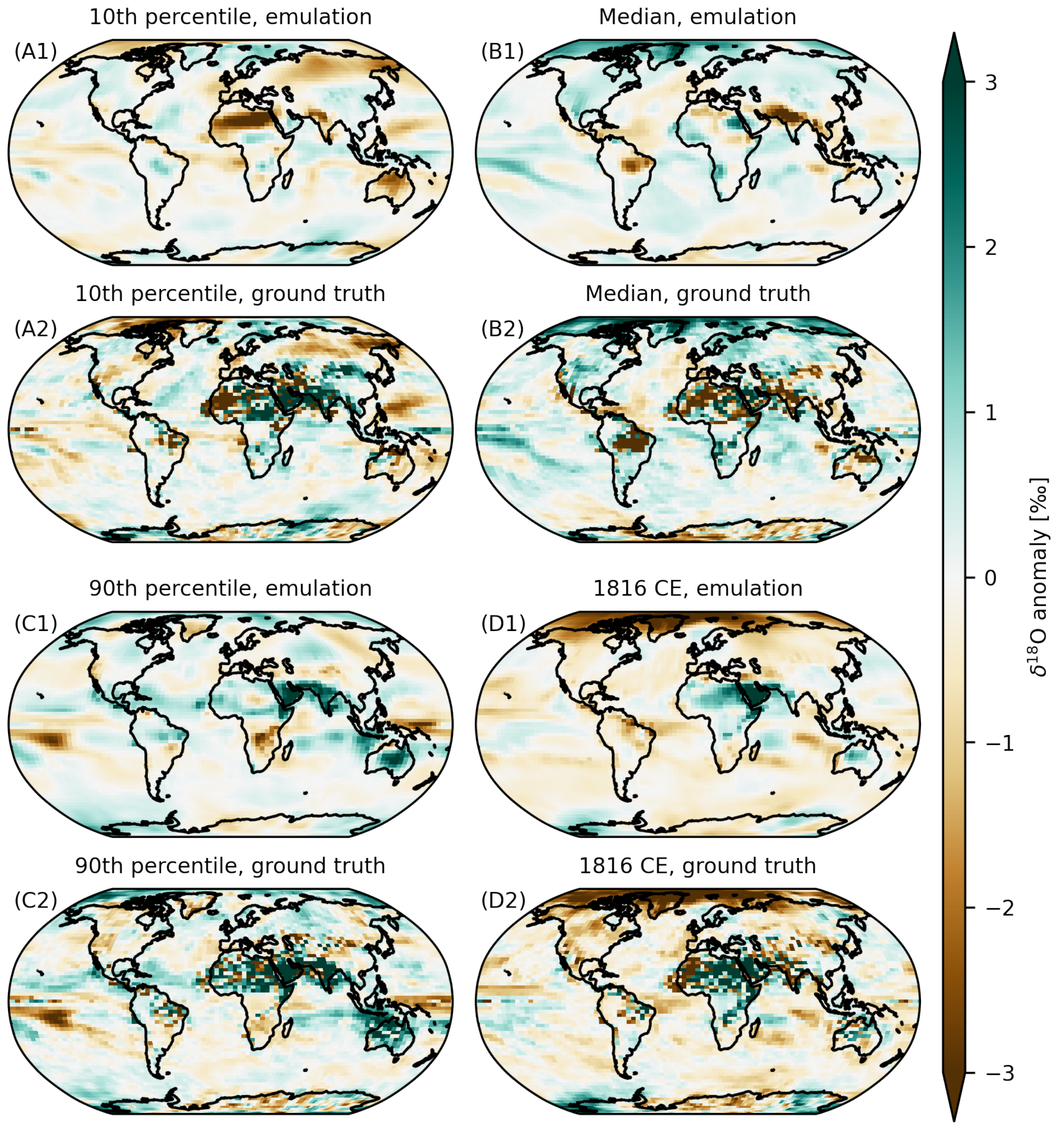
10th percentile, emulation
(A1)
Median, emulation
(B1)
10th percentile, ground truth
(A2)
Median, ground truth
(B2)
90th percentile, emulation
(C1)
1816 CE, emulation
(D1)
90th percentile, ground truth
(C2)
1816 CE, ground truth
(D2)
3
2
1
0
−1
−2
−3
δ^{18}O anomaly [‰]

Figure A.4. *Emulation results on iHadCM3 data set: We show anomalies produced by the ML-emulator ("emulation") and the "true" result in the iHadCM3 data set ("ground truth"). The anomalies are computed as the difference to the training set mean. We select time steps in which the emulator performs especially strong ("90th percentile") and weak ("10th percentile"), as measured by the anomaly correlation coefficient (ACC). Additionally, we show the time step, for which the ACC reaches its median value, and a year with a pronounced climatic anomaly: 1816 CE, the "year without a summer".*

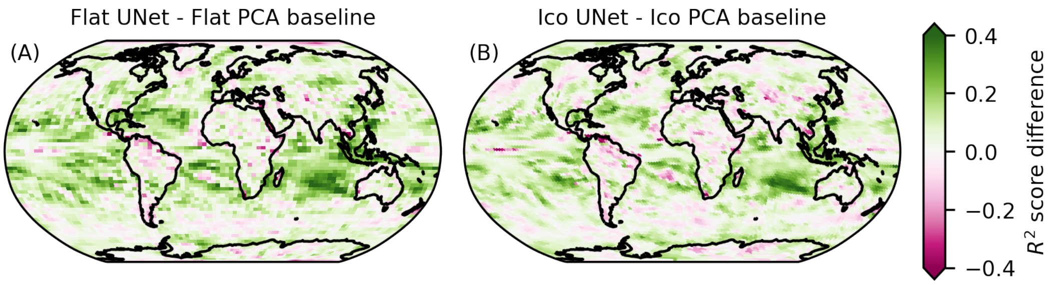

Figure A.5. *The difference in R^2 score between the UNet model and the PCA-based regression baseline for each grid type,*(A)*: plate carrée grid,* (B)*: icosahedral grid. On the plate carrée grid, we use the "modified" version of the flat UNet, including the modifications remedy distortions due to the spherical nature of the data. Green colors indicate better performance of the UNet compared to the baseline model. Before computing the differences, R^2 scores of ten runs are averaged for each configuration.*

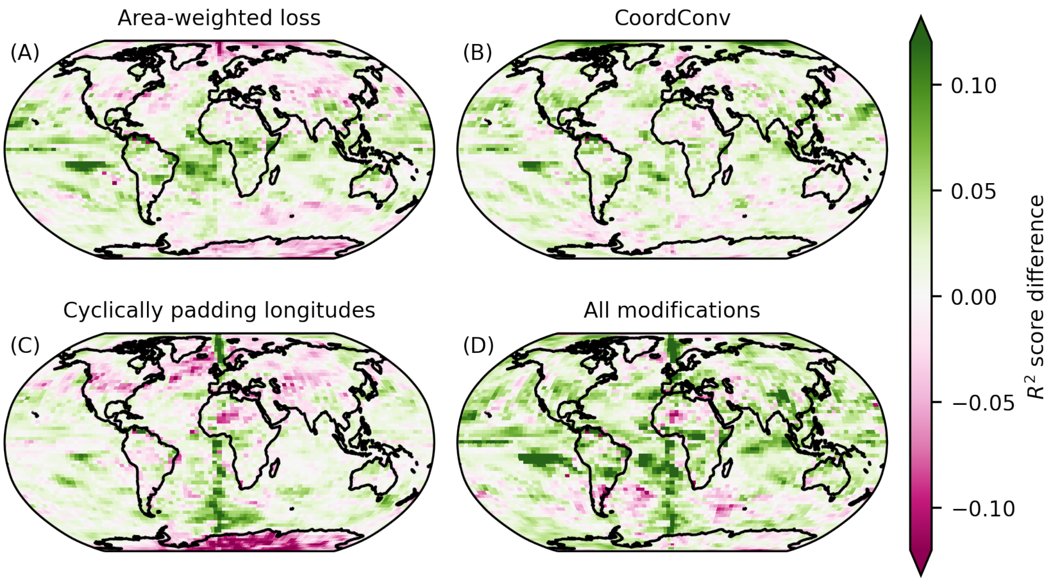

Figure A.6. *Effects of modifications to the "standard" UNet, which treats the longitude-latitude grid as a flat image: In each panel, we show the difference in R^2 score with respect to the unmodified version. We investigate the following modifications:* (A) *A loss function, in which the contribution of each grid box is weighted by the area it covers on Earth's surface.* (B) *CoordConv (Liu et al., 2018), a modification to CNNs, that allows the network to access the coordinates of locations in the image and break translational equivariance.* (C) *padding the longitudes cyclically instead of using zero padding and* (D) *the joint effect of the modifications. Shown in each panel are differences between averages over the R^2 scores of ten runs.*

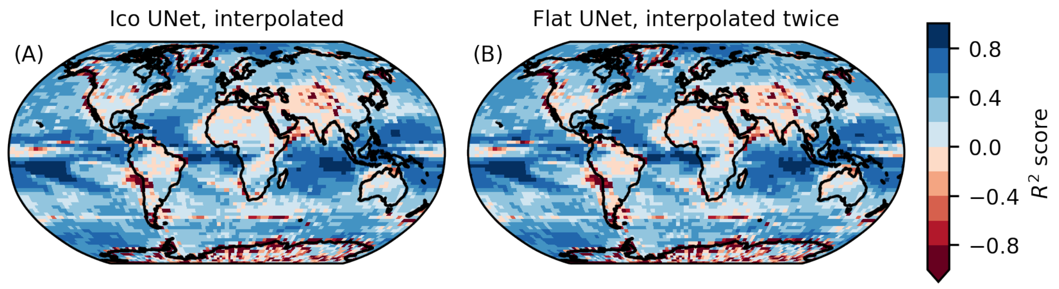

Figure A.7. *Interpolation between grids degrades results.* (A) *Performance of the icosahedral UNet architecture, here the results were interpolated to the flat grid.* (B) *results of the flat UNet architecture, after interpolating the predictions to the icosahedral grid and back to the plate carrée grid. The prediction quality is almost indistinguishable from the results of the icosahedral UNet. Additionally, the results are poor in regions, where we expect interpolations to have a negative effect: Over coastal and in the polar regions, where the icosahedral grid cells are larger than the iHadCM3 grid cells and therefore data partly gets "averaged" when interpolating from the finer iHadCM3 grid and to the coarser icosahedral grid and back. Thus, we can attribute a large part of the* R^2 *score differences to the interpolations between the grids.*

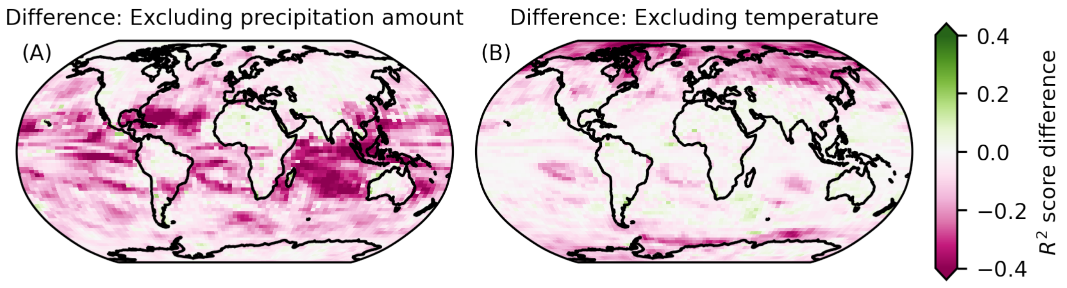
Difference: Excluding precipitation amount
(A)
Difference: Excluding temperature
(B)
0.4
0.2
0.0
−0.2
−0.4
R^2 score difference

Figure A.8. *Difference in emulation quality (*R^2 *score) between using only one of the predictor variables (surface temperature and precipitation amount) and using both simultaneously. Regions shaded in purple indicate a drop in performance when leaving out the corresponding predictor variable. Before computing the differences, averages over the* R^2 *scores of ten runs were formed for each configuration.*

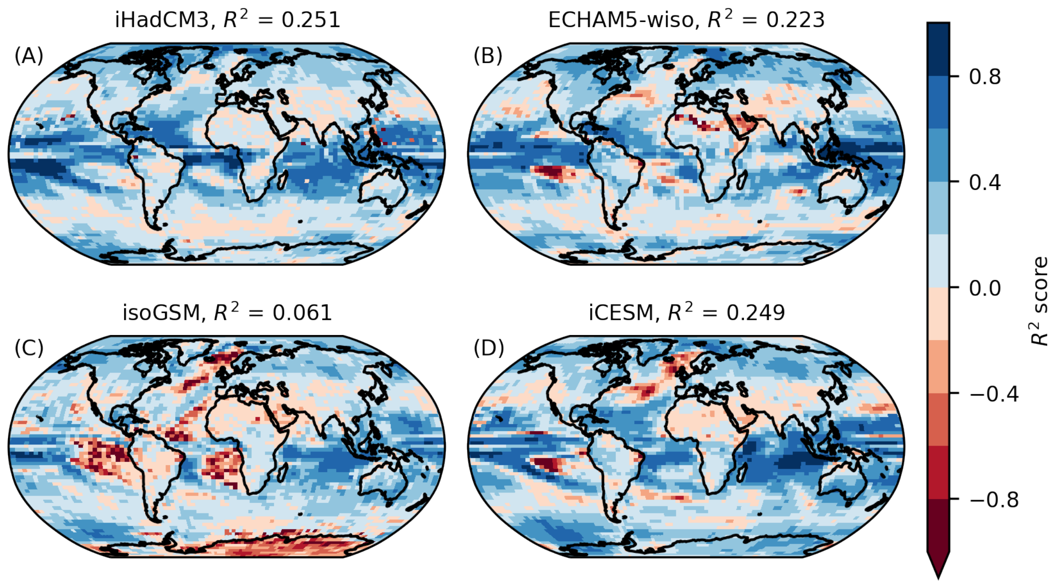

Figure A.9. *Results for the cross-prediction task with a pixel-wise linear regression model. The model was trained on the iHadCM3 data set, where the performance is worse than with the UNet model. For cross-predictions on other data sets, however, the emulation quality of the linear model is comparable to or even better than the performance of the UNet, as can be seen by comparing the globally averaged R^2scores in panels* (B) *to* (D) *to the corresponding panels of Figure 5.*

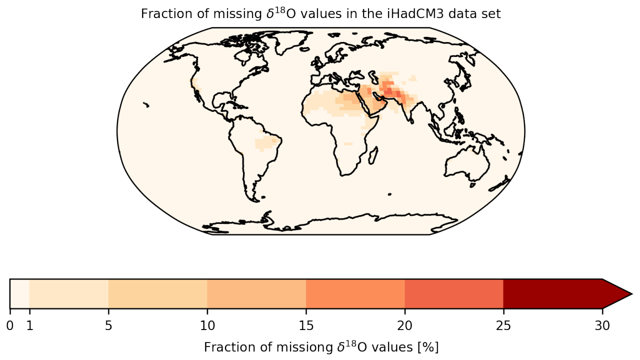
Fraction of missing δ^{18}O values in the iHadCM3 data set
0
1
5
10
15
20
25
30
Fraction of missiong δ^{18}O values [%]

Figure A.10. *Investigating missing δ^{18}O values: Spatial distribution of missing δ^{18}O data for iHadCM3 on monthly timescale. While for over 80% of the gridboxes, not a single value is missing, missing values are clustering mainly in hot and dry regions. For the worst grid box, δ^{18}O is missing in 25% of the time steps.*

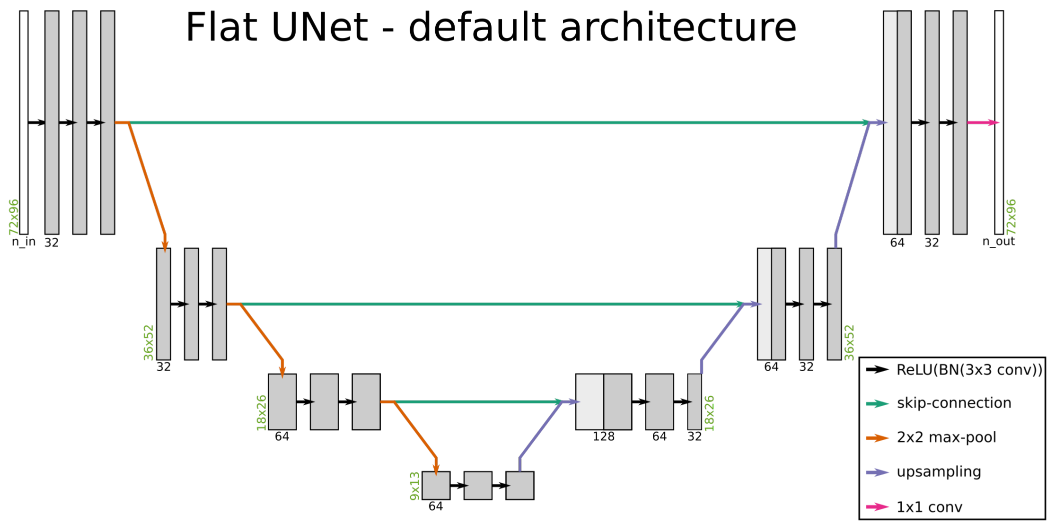

Figure A.11. *Sketch of our default flat UNet architecture. The parameters were chosen to roughly match in size with the icosahedral UNet architecture. Before the data is input into the network, it is resized in order to assure divisibility during down-and up-scaling, thus the input resolution (green) to the first layer is* 72×96 *instead of* 71×96 *as in the iHadCM3 grid. In the end, the output of the architecture is scaled back to the original grid. The number of computed features per layer is written below the corresponding data blocks (black). The number of input features* (n_{in}) *depends on the number of chosen variables (it equals 2 if using both temperature and precipitation amount), and the number of output features* n_{out} *equals* 1 *for all our applications. This graphic doesn't include the coordinate features that get appended to the data-tensors before each convolution, if CoordConv (Liu et al., 2018) is used.* BN *is short for batch normalization, and* 3x3 conv *indicates a convolutional layer with a* 3×3 *convolutional kernel.*

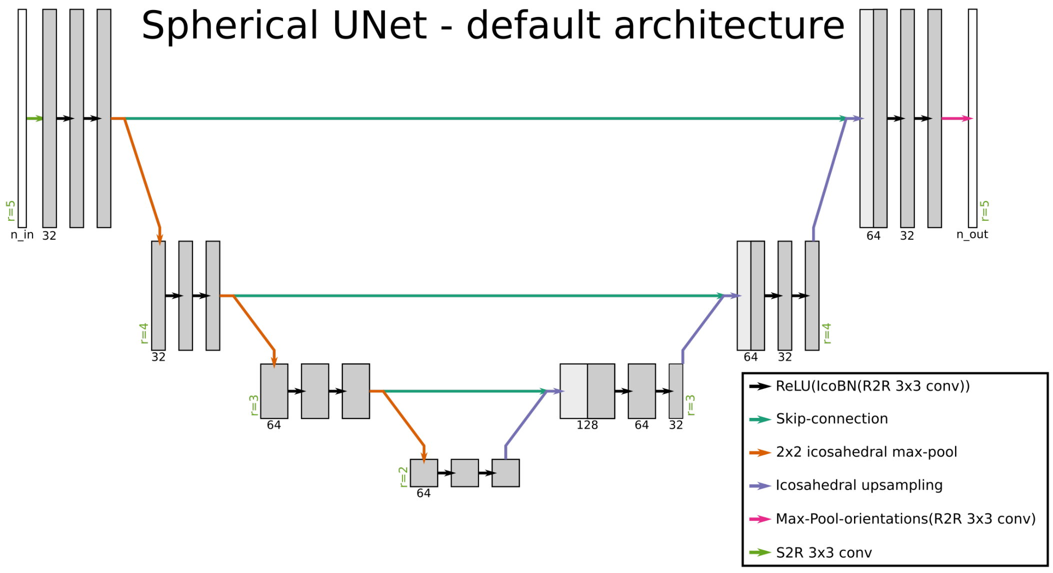

Figure A.12. *Sketch of the default icosahedral UNet architecture that takes into account the spherical nature of our data. The parameters are chosen similar to details given in the paper of Cohen et al., 2019. The icosahedral refinement level r is given in green, the number of features per layer in black below the corresponding blocks. "S2R" (scalar-to-regular) and "R2R" (regular-to-regular) are convolutional layers defined by Cohen et al., 2019 to achieve equivariance to a group of symmetry transformations.*