



Conditional updates of neural network weights for increased out of training performance

Jan Saynisch-Wagner and Saran Rajendran Sari

GFZ Helmholtz Centre for Geosciences, Potsdam, Germany

Correspondence: Jan Saynisch-Wagner (saynisch@gfz.de)

Received: 6 February 2026 – Discussion started: 23 February 2026

Revised: 28 July 2026 – Accepted: 19 September 2026 – Published: 2 October 2026

Abstract. This study proposes a method to enhance neural network performance when training data and application data are not very similar, e.g., out of distribution problems, as well as pattern and regime shifts. In contrast to previous approaches to out of distribution problems, which alter the inputs and outputs of an operator, our approach alters the operator itself. Here, we especially aim to alter the weights and biases of a neural network to mitigate the out of distribution problem. The method consists of three main steps: (1) Fine-tune a trained neural network towards reasonable subsets of the training data set and note down the resulting weight anomalies. (2) Choose reasonable predictors and derive a regression between the predictors and the noted weight anomalies. (3) Extrapolate the weights, and thereby the neural network, to the application data. We show and discuss this method in three nonlinear use cases from the climate sciences, which include successful temporal, spatial, and cross-regime extrapolations of neural networks.

1 Introduction

In physics, especially in nonlinear geosciences and climate sciences, the poor performance of neural networks (NN) when applied outside their training distribution or their trained dynamics poses a very strong limitation to their general applicability (Irrgang et al., 2021; Landsberg and Barnes, 2026).

In these fields, physical relations such as laws, dependencies, or sensitivities are commonly derived (or learned) under well-observed conditions and are then applied to less-observed conditions to gain knowledge about the latter. For example, results from lab or numerical model experiments are regularly applied to real-world problems or observations (e.g., Mehta et al., 2025); knowledge from our Earth and our Solar System is transferred to other planets and other star systems (e.g., Kvorka et al., 2026); learned relations that are derived today are transferred to the distant past, or to the future (e.g., Eyring et al., 2016; Wang et al., 2024; Koutsodendrakis et al., 2014). Usually, well-documented problems are transferred, e.g., across country borders (Sebastianelli et al., 2024; Kotz et al., 2021), across spatial scales (Ortega-Cisneros et al., 2025), from the well-observed upper ocean

to the deep ocean (Llovel et al., 2014; Lee and Gentemann, 2018), or from the lower latitudes to the poles (Irrgang et al., 2020); or a mixture of it all (Jung et al., 2024; van der Deure et al., 2025).

From a machine-learning perspective, this means that available training data are typically limited in their spatial and temporal extent, as observations or model simulations only cover specific time periods or regions. For example, Earth observing satellites often leave the higher latitudes unobserved depending on their inclination (e.g., Li et al., 2025); oceanographic Argo floats measure only the upper 2000 m of the marine water column (e.g., Wong et al., 2020); paleo records such as ice cores may exist for the past but only in regions where there is still ice today (e.g., Bereiter et al., 2015). In general, since the onset of satellite measurements, the majority of Earth observations cover only the last few decades; no measurements exist for the future, and comparatively few measurements exist for the distant past. Furthermore, the environmental conditions that exist on Earth are very heterogeneous (e.g., Peel et al., 2007), and most processes, for instance, the climate, are non-static in nature (e.g., Schannwell et al., 2024). The combination of spatiotemporally-localized

data distributions in combination with differing conditions between possible inference regions/times/environments and the application regions/times/environments makes it very difficult for NN to operate (e.g., Rasp et al., 2018; Hernanz et al., 2022; Hsieh, 2023).

In some cases, individual standardization or rescaling of the variables from the application data set is a valid option to mitigate the out of distribution (OOD) problem. However, in many real use cases, the standardization or rescaling removes essential information from the application data. This is true for nonlinear problems where standardizing or rescaling the input will lead to wrong outputs, as by the definition of non-linearity: $f(a+b) \neq f(a)+f(b)$ and $f(n \cdot a) \neq n \cdot f(a)$. Especially in the climate sciences, the conditions for contemporary artificial intelligence (AI) are challenging and raise the demand for new approaches (e.g., Eyring et al., 2024; Beucler et al., 2024; Fang et al., 2025; El Ghawi et al., 2025; Qin et al., 2025). For example, removing offsets and trends in variables such as sea level and temperature would discard information essential for predicting climate change, climate impacts, and tipping. Likewise, the amplitudes of extreme events should not be rescaled to fit those of the training data. In addition, standardization and rescaling require a certain amount of data points, which are not always available in the application data. Often, the learned relations have to be applied to one single (possibly extreme) event, to a few ice cores, to one other planet, moon, or star system, and so on, i.e., they pose so-called single-shot or few-shot problems in AI terminology. Consequently, in such cases, application-distribution-based standardization, rescaling, remapping, or basis transformation (e.g., Beucler et al., 2024) is not possible. Finally, even without any distribution shift or with normalization, the NN can still be confronted with unseen combinations of input values, which can pose a serious OOD problem.

Here, we propose a method to reduce this problem for NN by exploiting the NN's own sensitivity towards samples of the training data to predict NN weights and biases, which are intended to perform better in the OOD application realm (cf. Sect. 2, Method).

However, this paper does not discuss if and when NN are suitable choices for nonlinear OOD problems. Furthermore, we will not discuss (in depth) the many other methods that might be able to extrapolate the inputs or outputs of some NN to OOD regions. The reason is that many tasks do require NN, especially when these other approaches cannot be used. Examples of these tasks include reasoning and decision making, conditional generative AI, big data, and large classification problems such as image feature detection. These are applications in which extrapolating or averaging (cf. Hsieh, 2023) NN output will not give meaningful results. Our study focuses on problems where NN are used for various reasons (ease of use and implementation, existing workflows, compute limitations, invertibility, lack of alternatives, etc.), and these NN may be used on OOD data. Still, for completeness,

we provide a comparison of our experiment 1 results to several standard extrapolation methods in Appendix A, together with a brief discussion.

We think that AI applications in many fields of physics and other fields of science, such as medicine, biology, economic sciences, or social sciences, can benefit from the proposed method. Nonetheless, we demonstrate and evaluate the proposed methodology in three classes of experiments that are mainly motivated by climate sciences and Earth sciences (cf. Sect. 3, Experiment Design and Data). One experiment where NN weights are predicted through time, one experiment where NN weights are predicted through space, and one experiment where NN weights are predicted across regime boundaries. Within similar training and application data sets and domains, the chosen experiments do not represent any challenges to NN training (nor to basic mathematics and physics). The challenges we demonstrate and mitigate here arise only by training the NN in times and places that have very different conditions than the times and places the model has to operate on during application. Consequently, these experiments should be understood entirely as symbols or placeholders for more sophisticated AI tools.

2 Method

The proposed weight-prediction method for NN follows the pseudo-code given in Box 1. At its core, the approach applies existing extrapolation techniques to the weights of a NN instead of applying them to the input and output of NN as other approaches suggest (e.g., Hsieh, 2023; Beucler et al., 2024).

Our approach consists of 3 parts. After training of a suitable NN for the problem at hand with a training data set (Box 1, step 1), one has to do the following. (A) For each weight and bias of this hereafter termed “parent model”, collect the weight deviations arising from retraining the parent model with single data points or subsets of the training data set (Box 1, step 2). (B) For each weight and bias of the parent model, establish a dependency between said weight deviations and suitable predictors, e.g., a linear regression (Box 1, step 3). (C) For each weight and bias of the parent model, extrapolate the established dependency towards application data points and thus generate new application-data-tailored NN, the “child models” (Box 1, step 4). From now on, we summarize a group of child models originating from the same trained parent model (cf. *ParentModel*₀ in Box 1, step 1) under the singular “child model”. All models in one child model differ only due to the fact that their weights are predicted onto different application data points (cf. x_a and *ChildModel*_{*a*} in Box 1, step 4), until this step, they are based on exactly the same data. Consequently, if we use the plural from now on, we refer to different child model groups, i.e., groups originating from different *ParentModel*₀.

The reasoning why the above approach should work is that fine-tuning the *ParentModel*₀ further on a training data point

1. **Train:**
 - Train a suitable NN on the training data set $\mathbb{I}$, consisting of I data points x_i
 - Store $ParentModel_0$
2. **Forgetful online learning:**
 - Loop over a reasonable subset of training data points x_i , e.g., for all $x_i \in \mathbb{I}$:
 - (a) Fine-tune $ParentModel_0$ on a single training data point x_i , regularized towards $\mathbb{I}$. For example, by compiling the x_i -focused training data set $\mathbb{I}_i$, consisting of n copies of a single data point x_i and n random other training data points $x_{j \neq i} \in \mathbb{I}$.
Result: $ParentModel_i$
 - (b) Store $ParentModel_i$'s K weights (including biases) of all layers, nodes, and connections: W_{ik}
 - (c) Reload $ParentModel_0$
 - (d) Repeat from (a)
3. **Weight-Regression:**
 - Choose a suitable regression algorithm, e.g., linear regression $\mathbb{lm}()$
 - Choose (and prepare) a respective set of predictors/regressors $\mathbb{R}$, which correspond to the training data points x_i , used in step 2 above, and the application data point(s) x_a , used in step 4 below: R_i & $R_a \in \mathbb{R}$, e.g., time steps t_i & t_a
 - Loop over a suitable subset of the $ParentModel_0$'s K weights and biases, e.g., for all $k \in \mathbb{K}$:
 - (a) Fit chosen regression model, e.g., $RegressionModel_k = \mathbb{lm}(W_{ik}, R_i)$
 - (b) Store $RegressionModel_k$
 - (c) Repeat from (a)
4. **Weight-Prediction:**
 - Loop over a reasonable subset of application data points x_a , e.g., for all $x_a \in \mathbb{A}$:
 - (a) Predict all required weights for application data point x_a : $W_{ak} = RegressionModel_k(R_a)$
 - (b) Store $ChildModel_a$
 - (c) Repeat from (a)
5. **Actual Application:**
 - Apply child model to application data point(s): $ChildModel_a(x_a)$

Box 1. Pseudo-code of the weight-prediction approach.

x_i can result in a new model ($ParentModel_i$) whose weights and biases differ (a) only slightly and (b) in a meaningful way from $ParentModel_0$. Ideally, all $ParentModel_i$ form a point cloud around $ParentModel_0$ in NN parameter space, which encodes the respective sensitivities between this specific $ParentModel_0$'s weights and the input data. For this to be the case, the deviations between $ParentModel_0$ and the related $ParentModel_i$ have to be constrained in a way that no entirely new and therefore unrelated $ParentModel_i$ are created in the process. In other words, all $ParentModel_i$ should be closely yet meaningfully distributed around $ParentModel_0$. To achieve this, it should be theoretically sufficient to restrict the fine-tuning of $ParentModel_0$ (cf. Box 1, step 2a)

towards each x_i to a few update steps (epochs). However, we found that a regularized fine-tuning of $ParentModel_0$ towards individual training data points x_i works better and allows for more flexible fine-tuning approaches. To regularize the fine-tuning step, we add a representative fraction of the training data to the x_i fine-tuning batch. Naturally, the impact of such a regularization and the impact of the fine-tuning towards a single x_i in generating a new $ParentModel_i$ has to be weighted reasonably. Box 1, step 2a gives an example of how to achieve a suitable regularization easily in practice, e.g., we take n random samples from the training data combined with n duplicates of x_i to ensure equal weighting in the fine-tuning batch. Consequently, weighting in the given

example is 0.5 for each impact, i.e., equal impact from the entire training set ($\mathbb{I}$) and from a single data point x_i .

To choose a suitable approach for the $ParentModel_i$ generation (Box 1, step 2a), the distribution of the $ParentModel_i$ around $ParentModel_0$ can be easily plotted and analyzed, since this distribution does not depend on OOD data. The analysis could either focus on single weights or be based on a NN-wide integral view of weight differences between the $ParentModel_i$ and $ParentModel_0$.

Note, in Box 1, we show a total-valued form of the method. Naturally, handling only the weight anomalies or increments ΔW_{ik} is equally possible.

The exact execution of each given step of the core approach (Box 1) can vary substantially and should be adapted to the problem at hand and to domain knowledge if available. For example, two main choices have to be made during the approach. First, choices for suitable predictors for the weight-regression (Box 1, step 3) and the subsequent weight-prediction (step 4) have to be made. It is out of the scope of this paper to discuss these choices here in detail. Ideally, the choice should be heavily influenced by prior knowledge of the problem at hand. Especially, how the problem might behave outside the training data range. In principle, the predictors for the weights can be anything, e.g., (i) time, (ii) (subsets of) the original NN input, (iii) features derived from the original input, or (iv) additional information not used by the parent NN model. A careful selection and pruning of the predictors is advised. In our examples, we demonstrate the use of (ii) in experiment 2, and (iii) in experiments 1 and 3. Note that in experiment 2, the inputs of depth, latitude, and longitude are not necessary for the NN to calculate the correct sea water density as long as temperature, salinity, and pressure are given. However, they are promising weight-predictors. As we did not want to limit the parent NN by withholding information, we unified the inputs of parent NN training and weight-regression in experiment 2, i.e., the weight-predictors are deliberately included in the NN inputs.

Second, a choice for the weight-prediction method has to be made. Of these, many suitable exist, e.g., auto-regressive integrated moving average (ARIMA), eXtreme Gradient Boosting (XGBoost), Gaussian Processes, linear models (LM), generalized linear models (GLM, Nelder and Wedderburn, 1972), generalized additive model (GAM, Hastie and Tibshirani, 1986), support vector machines (SVM, Cortes and Vapnik, 1995), random forest (Ho, 1995), NN itself (cf. Chauhan et al., 2024), and many more. Again, it is not part of this paper to review regression methods and their rightful application. Nonetheless, we provide a comparison of these methods acting on our toy problems in Appendix A. In experiments 1 and 2, we chose LM for their ease of use and very fast computation. The latter is especially needed as we do regression and prediction for every weight and bias of every single node of the NN. However, if the NN is very large, low-rank adaptation (LoRA, Hu et al., 2021) or similar decompositions should be considered. We would not generally rec-

ommend using other purely data-driven methods like RF and NN as they suffer from poor OOD performance, as already stated (e.g., Rasp et al., 2018; Hernanz et al., 2022; Hsieh, 2023). Still, in some cases, we found improvements in our experiments by these methods. For instance, in experiment 3, we demonstrate the use of a NN for the weight-regression and weight-prediction tasks. In general, if you know a regression method that works well for your problem, domain, or data (cf. Figs. A1–A3), then this method could be a good candidate for extrapolating the weights of a NN (if the method is fast enough, that is).

3 Experiment Design and Data

3.1 Experiment 1: Tipping of the Atlantic Meridional Overturning Circulation

We base experiment 1 on van Westen et al. (2024), a paper that shows that critical weakening of the Atlantic Meridional Overturning Circulation (AMOC) strength can occur in complex climate models. This so-called AMOC tipping is triggered by forcing the northern Atlantic Ocean of the respective climate model with increased freshwater flux (e.g., related to the rapid melting of the Greenland Ice Sheet). The paper's supplementary material and the data we use here is located in the following repository: https://github.com/RenevanWesten/SA-AMOC-Collapse/tree/SA-AMOC-Collapse_v1.0 (last access: 1 August 2025). These hosing experiments are a common tool in climate sciences (e.g., Rahmstorf, 1996; Saynisch et al., 2016).

As input to the NN, we use 2d velocity fields representing vertical slices through the Atlantic Ocean along the latitude of 26° N, i.e., the maps of meridional velocity (VVEL). These model-based maps of horizontal velocity are always taken at the same location and are given as annual values. The VVEL data can be downloaded from the above-given repository from the folder: `/Data/CESM/Data/AMOC_section_26N/`. The files cover each 50 years of data; the first one is called `CESM_year_0001–0050.nc` (and so forth).

The target output of the NN is the strength of the AMOC given in Sverdrup ($1 \text{ Sv} = 10^6 \text{ m}^3 \text{ s}^{-1}$), i.e., a single value per input velocity map. The AMOC strength is defined as the total meridional volume transport at 26° N over the upper 1000 m (cf. Eq. (3) in van Westen et al., 2024). The respective AMOC strength time series is given in `/Data/CESM/Ocean/AMOC_transport_depth_0–1000m.nc`.

The challenge in experiment 1 is that the training is based only on data before the AMOC tipping point, occurring around year 1800 of the model data, cf. Fig. 1. However, the NN has to calculate the correct AMOC strength also during tipping and even after tipping, a serious OOD problem.

3.2 Experiment 2: Estimation of Sea Water Density

Experiment 2 is based on the highly nonlinear Equation Of State for sea water (EOS). For every desired oceanic location, the NN input is salinity (S), temperature (T), pressure (P), latitude (ϕ), longitude (λ), and depth (z). The target output is the local in-situ density (ρ) at this particular location. Consequently, the NN task is to convert 6 numbers into one. For data, we use the gridded observations of the World Ocean Atlas (Locarnini et al., 2024; Reagan et al., 2024), which are available under <https://www.ncei.noaa.gov/access/world-ocean-atlas-2023/> (last access: 1 September 2025). We use the 5° decadal averaged annual fields of S and T for every available latitude (36 levels), longitude (72 levels), and depth (102 levels) of the provided 3d ocean grid. These values represent long-term averages, and no temporal dimension is present in this experiment. We then calculate P and ρ by using the routines `swPressure( $z, \phi$ )` and `swRho( $S, T, P$ )` from the `oce` R-package (Kelley and Richards, 2025), which follow the Thermodynamic Equation Of Sea Water – 2010 (TEOS-10) standards (IOC et al., 2010).

Experiment 2 is motivated by the diminishing of oceanic monitoring capabilities with depth. Let's assume that we have found an interesting relation between our targets and the Argo float data. Unfortunately, the Argo floats cover only the upper 2000 m of the water column (Wong et al., 2020). However, we want to exploit our findings also below 2000 m. In a practical application case, this relation could estimate krill, CO₂, or nutrients. Here, we restrict ourselves to the estimation of sea water density.

In contrast to experiment 1 where knowledge has to be transferred over time (actually, also over to a different post-tipping dynamic), the challenge of experiment 2 is to transfer learned knowledge spatially. The training of the NN is restricted to a spatial (upper) subset of the world oceans. However, the trained NN has to redo the calculation for the entire ocean where OOD input and OOD output values occur. A further difficulty is that the EOS follows a complicated, highly nonlinear empirical polynomial (cf. Roquet et al., 2015).

3.3 Experiment 3: Uncertainty Estimation of Global Wind Velocity Reanalyses

The third experiment focuses on cross-regime extrapolation of learned uncertainties in global wind velocity reanalyses. The setup follows the concept proposed by Irrgang et al. (2020), who demonstrated that spatiotemporal uncertainty patterns learned at a single oceanic location can be generalized to the global ocean domain. Here, we extend this approach by transferring the learned uncertainty information from a single oceanic reference point to continental regions. Since wind dynamics over land are governed by different boundary-layer processes, surface roughness, and convective

regimes compared to those over the ocean, this setup represents a severe OOD scenario.

In contrast to Irrgang et al. (2020), who used three atmospheric reanalysis products as model input, we restrict the input data to two widely used reanalyses: ERA5 (Hersbach et al., 2023) from the European Centre for Medium-Range Weather Forecasts (ECMWF) and CFSv2 (Saha et al., 2014) from the National Centers for Environmental Prediction (NCEP). Data set preparation, NN model architecture, and NN model training follow the same strategy as in the reference study to ensure comparability. This configuration tests whether the learned sensitivities in wind velocity uncertainty over marine environments can meaningfully generalize to terrestrial conditions.

4 Results and Discussion

4.1 Experiment 1: Tipping of the Atlantic Meridional Overturning Circulation

Notable deviations in experiment 1 from the recipe given in Box 1:

- Skipped step 2c (reload *ParentModel*₀)
- EOF of the *ParentModel*₀ inputs are used as weight predictors

To predict AMOC strength from velocity maps, we use a standard shallow convolutional NN (CNN), with 2 convolutional/max pooling layers followed by two dense layers. The last layer has linear activation, and all other layers have the Exponential Linear Unit (ELU) activation. For the fine-tuning regularization (Box 1, step 2a), n is set to 200.

For the regression of the NN weights, the predictors are based on the decomposition into empirical orthogonal functions (EOF) of the NN's training data inputs. The principal components of the 4 leading EOF, i.e., 4 numbers per time step (respectively per velocity field), are used for the regression of the NN weights. Outside the training data, exactly the same EOF-basis as during training is used to decompose the velocity maps. Another possible set of predictors could be mean, maximum, minimum, and standard deviation of each individual velocity map, i.e., again 4 numbers per time step, or even time itself if a suitable low-dimensional bifurcation model were to be employed for the NN weight-prediction (cf. Rahmstorf, 1996). Note, however, that the sensitivity of the weights is not dictated by the domain dynamics alone but also depends on the involved activation functions (and not only the activation function from the node under consideration). For example, if a complex nonlinear change of model output is required by the application domain dynamics, it can, in principle, be realized by a less complex (e.g., linear) weight change when nonlinear activation functions are involved. Furthermore, the required weight changes can be distributed over several layers, which then will entangle all contributing activation functions.

As there is a clear order of the data in experiment 1, i.e., the temporal order, we skip the “reload *ParentModel*₀” step of the approach (cf. Box 1, step 2c). This way, the online learning is not forgetful, and the weights will evolve continuously along the time axis as the AMOC system does itself. This improves the results in this experiment as it reduces noise in the generation of the weight sensitivities W_{ik} . After establishing weight-predictor relations for every weight and bias of the CNN, including the CNN layers, the child model is predicted with sub-models (cf. *ChildModel*_{*a*}) for every single time step of the problem by using the same weight-predictor relations. Each of these sub-models is then applied to one single data point, i.e., time step, only.

Figure 1 shows the time series of the target value (black line), the output of the parent model (red line), and the output of the same model after the application of the weight-prediction approach (orange line), i.e., the child model. The training data starts at year 1, and the end of the training data (EOT) is shown by a vertical dashed line, here at 800 years. After the tipping of the AMOC sets in at around 1800 years, the parent model has clearly problems reproducing the black line. Its output is strongly biased towards the output it had learned during training. The child model surpasses the parent model in its performance during and after tipping.

Please note that since *ParentModel*₀ is not reset during step 2c of Box 1, at the end of step 2 we get a NN model that has seen the training data more often than *ParentModel*₀. Still, the performance of this model (not shown) is the same as that of *ParentModel*₀, and the improvements plotted in Fig. 1 must be attributed entirely to steps 3, 4, and 5 of our approach.

Since in general, the training of NN can have largely varying results, we repeated the experiment several hundred times with randomly varying EOT times, while the beginning of the training window remained fixed at year 1. As a consequence, the amount of data each NN is trained on varies, as well as how close the EOT comes to the actual tipping event. Furthermore, two weight-regression methods are tested. The results are aggregated in Fig. 2.

The figure shows root mean squared errors (RMSE) of parent models (yellow boxes), child models (green boxes), and their pairwise differences (blue boxes). For successful weight-prediction, the child models should show lower RMSE than the parent models, and the pairwise differences should show as positive values as possible. The variations summarized by each box are based on the rather random generation of parent models in combination with the random selection of an EOT between years 800 and 1800. In each panel, three horizontal subdivisions are shown where the RMSE are calculated for 3 different time windows: the full 2200 years of data including the training data (left), the time beyond each NN’s individual EOT, i.e., everything except the training data (middle), and a fixed time window covering years 1800–2200, i.e., the tipping event and after (right).

The two panels of Fig. 2 are both based on a linear model for the weight-regression. The standard R function `lm()` is used here. The difference between the two panels is the order of polynomials built from the predictors before weight-fitting. While the left panel uses polynomials of order 1, the right panel uses polynomials of order up to 2. As mentioned, the weight-regression is based on the EOF decomposition of the 2d meridional velocity fields that form the input for the NN itself. The predictors are always the 4 principal components associated with the 4 leading EOF as derived in the training time window.

One can see that in all cases the (green) child model boxes show lower RMSE than the (yellow) parent model boxes, leading to positive pairwise RMSE differences depicted as blue boxes. In addition, the performance gain of the child models compared to their parents increases when the models face conditions that are increasingly different from their training data. The average RMSE of parent and child boxes become larger from left to right in each of the two panels. From left to right, the RMSE in the subdivisions increasingly focuses on the OOD performance of the NN. However, the RMSE increase stronger in the parent models than in the child models, resulting in larger pairwise improvements when the tipping is approached and surpassed.

Since the regressions of polynomials of order 1 and 2 give comparable results (with a slight advantage of the order 1 results), this could be interpreted as a sign of robustness of the approach towards the regression method. However, polynomials of even higher-order (as far as tested) gave worse results (not shown). Here, the higher-order terms get random/noise-based slopes/gradients assigned to them during the regression (Box 1, step 3) as the higher-order terms have only a negligible influence within the training data where the regression is conducted. Nonetheless, these terms can still lead to significant yet wrong influences during and after tipping in step 4 of Box 1, the weight prediction. More research on how to stabilize the child generation is needed. A signal-to-noise ratio guided pruning of insignificant slopes returned by the regression step could help with this problem.

In general, as can be seen in Fig. 2, the approach’s success can vary considerably, resulting in even worse results from some child models. The same mechanism described above for higher polynomials is probably the reason for the poor results in some of the polynomials of order 1 and 2 cases, too. On average, the results are positive as the ensemble mean of the child models shows a clear improvement over the ensemble mean of the parent models. More research is needed to stabilize the results – either by identifying parent models unfit for the approach, or, if possible, by subjecting child models to forms of validation before their application. Until then, the ensemble approach should be considered.

In the following, we will discuss some noteworthy examples from the ensemble simulations aggregated in Fig. 2. Apart from the many good results that look more or less similar to Fig. 1, it can happen that the regression shows too little

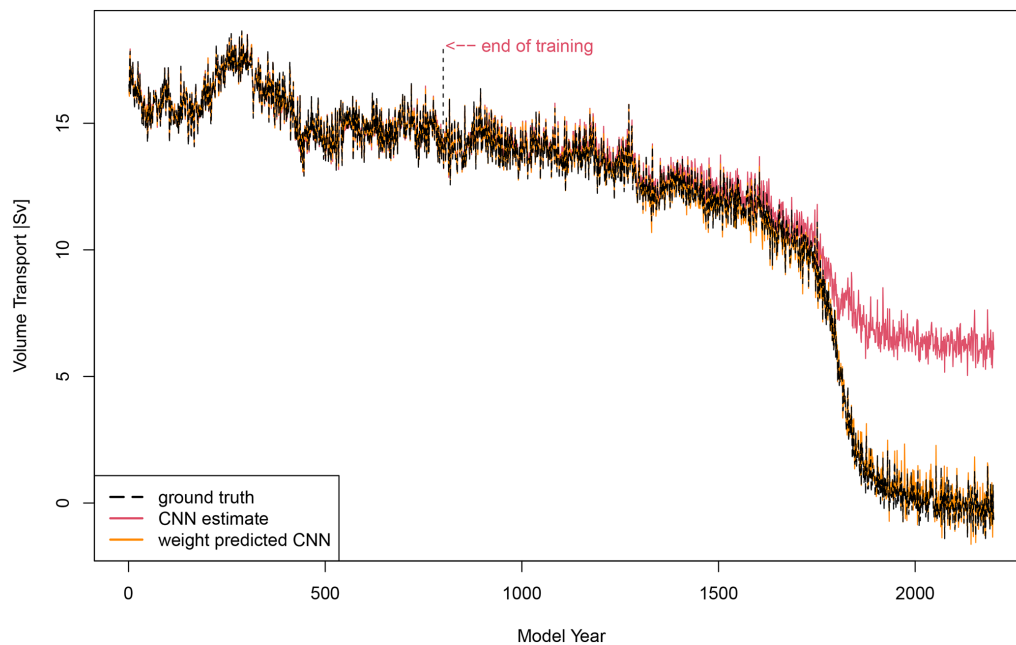


Figure 1. AMOC strength and an example of good performance of the weight-prediction method. Black Line: AMOC overturning decay and tipping, i.e., the target ground truth values. Red Line: Output of parent model trained until the dashed end of training (EOT) line. Orange Line: Output of weight-predicted child model. Online learning and subsequent fitting of EOF-to-weight relations used only data before the dashed EOT line.

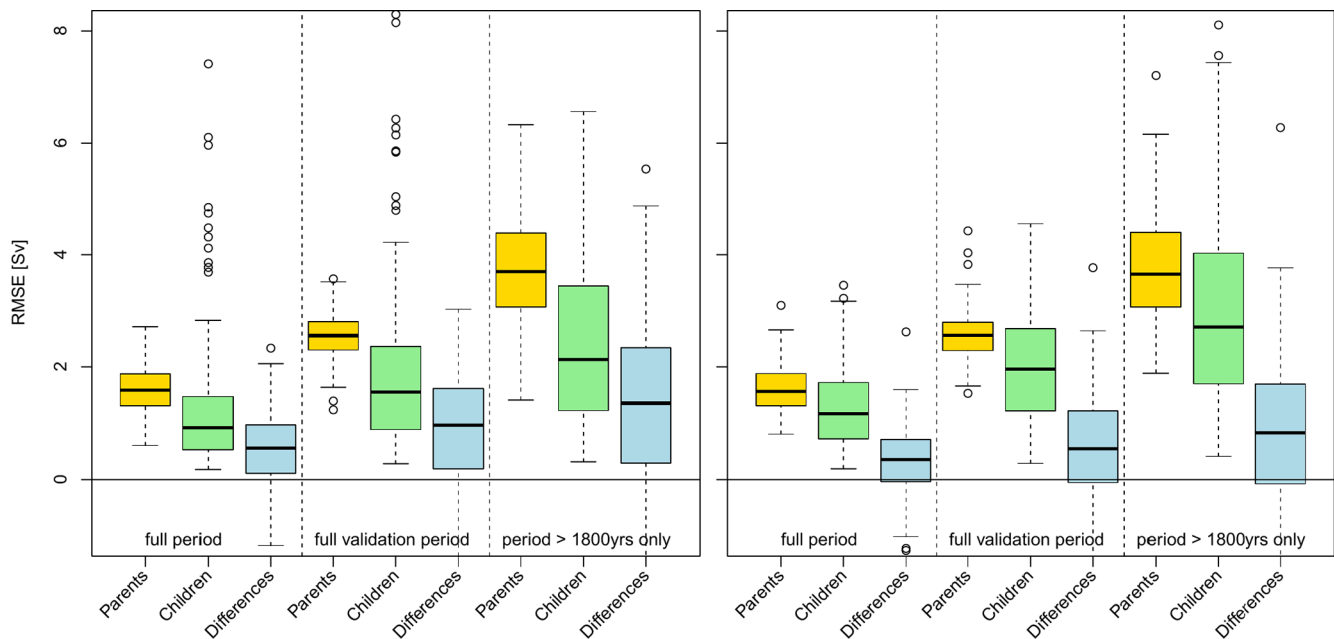


Figure 2. Impact of the choice of the weight-prediction method on the child model performance. Polynomials of order 1 (left panel) and order 2 (right panel) were fitted to the leading 4 principal components of the input variable (i.e., the meridional velocity maps). Yellow Boxes: RMSE of parent models. Green Boxes: RMSE of child models. Blue Boxes: Pairwise RMSE differences (i.e., unperturbed “parent” model minus corresponding weight-predicted “child” model). Positive values represent improvements due to the presented weight-prediction approach. Note that “full period” and “period > 1800” are the same for every simulation of the ensemble. The term “full validation period” refers to all data points after each simulation’s individual EOT (which is randomly chosen for every simulation before training). Each box is based on 300 parents/children/pairs.

sensitivity to the predictors and the parent, and child model outputs are very similar.

How it comes to this poor performance is rather unclear at the moment and needs more investigation. The most likely explanation is that the respective parent model sits either in a deep narrow minimum or a very flat part of the loss hyper-plane, and due to this, the proposed online learning resulted only in minimal variations of the weights (ΔW_{ik}). As the particular parent model is not sensitive to the online training step (Box 1, step 2), the weight-regression and the subsequent weight-prediction inherit this limitation. Still, the small induced weight variations are not pure noise, and the weight-prediction generates a better-performing child model, even if it is very similar to its parent (not shown).

Furthermore, in some rare cases, deterioration can happen when a child model is applied to data similar to the training conditions, as shown in Fig. 3. The child model (orange line) bends too early towards the post-tipping output and then overshoots. As a consequence, in the 500-year period following the EOT (at year 1200) the parent model (red line) slightly outperforms the child model (orange line). When calculating the RMSE for the entire post-EOT time, the child model still outperforms its parent model.

Figure 4 is based on the same results as Fig. 2, but here the RMSE are sorted by EOT. In addition, the figure shows only the RMSE for the tipping and the past tipping period. While in the view of Fig. 2, the polynomial of order 1 and order 2 based regressions did look very similar, here in Fig. 4, some large differences become evident. If trained only very far from the tipping point, the regression based on higher-order polynomials leads to wrong results. This strengthens the argument for the fail-mechanism already described, i.e., that higher-order terms get noise-based gradients assigned to them as they are not varying significantly during training and regression. When they then do start to vary during tipping, they will give large yet wrong contributions to the weight prediction, and this way will deteriorate the child model and its output.

In general, the skill of all models (parents and children) improves with larger EOT. This is not only due to more training data per se, but rather that the training data then also includes information located closer to the tipping point. Consequently, the models that are trained up-until tipping perform best, i.e., all yellow boxes go down from left to right in each panel of Fig. 4. On average, even as children and parents show increasingly similar performances with EOT closer to the tipping point, the on average better RMSE of the child models are never reached by the parent models.

A note on temporal data: In experiment 1, we shuffle the training data before applying the training/validation split. In some problems, this is not possible, either due to the choice of architecture (as LSTM) or not advised due to correlations within time (e.g., Schnaubelt, 2019; Bergmeir et al., 2018). If data shuffling is not an option but a test/validation split is still necessary, then the training data is pushed even further away

from the tipping point by the presence of the validation window. As a result, the parent models will show an even worse performance. Consequently, our weight-prediction approach will become more advantageous as it also uses the validation data located closer to the tipping point for the weight-regression.

4.2 Experiment 2: Estimation of Sea Water Density

Notable deviations in experiment 2 from the recipe given in Box 1:

- Quantile approach to sample only a fraction of the training data set $\mathbb{I}$ in step 2a
- No regularization towards $\mathbb{I}$ is applied during the fine-tuning step 2a (epochs are limited to 4)
- NN inputs (S , T , P , ϕ , λ , and z) are used as weight predictors as well

The NN of choice here is a simple two-layer feed-forward NN with ELU activation functions in the first layer and linear activation in the final layer. The regression of the NN weights uses the same set of variables as the NN input, i.e., S , T , P , ϕ , λ , and z (cf. Sect. 3.2). For the forgetful online learning (Box 1, step 2a), we fine-tune the parent model only on individual x_i without the regularization towards the entire training data set $\mathbb{I}$. The parent-to-child model differences (ΔW_{ik}) were limited by limiting the fine-tuning iterations (here: epochs = 4) only.

In experiment 2, the training data set is very large, and thus retraining the parent model on every training data point is prohibitively expensive. A random sub-sampling of the training set could lead to undersampling with respect to some of the predictors if the random draw is not sufficiently large. To be efficient and at the same time avoid undersampling the predictors, we sort the training data into quantiles (e.g., 300) for every predictor. Then we draw several x_i from every parameter-specific quantile interval for fine-tuning the parent model (Box 1, step 2). If it is suspected that special functions (or interactions) of predictors should play a role in the weight-regression, then sampling quantiles of the respective function-values (or interactions) is advised (e.g., one has not only to guarantee sufficient sampling of the predictors x and y , but also of $\log(x)$, y^2 , and $x \cdot y$ etc.).

As in experiment 1, individual *ChildModel_a* are generated for every single application data point x_a , here, every grid point of the 3d ocean grid.

Figure 5 shows the impact of the weight-prediction approach on a spatial OOD problem, where the parent model is only trained above the dashed line at 2000 m.

As a result, the weight-prediction approach improves the performance of the NN sea water density calculation in the deep ocean considerably. At depth, the improvements amount to 2 kg m^{-3} in the global average, which is substantial by oceanographic standards. Below the EOT of 2000 m,

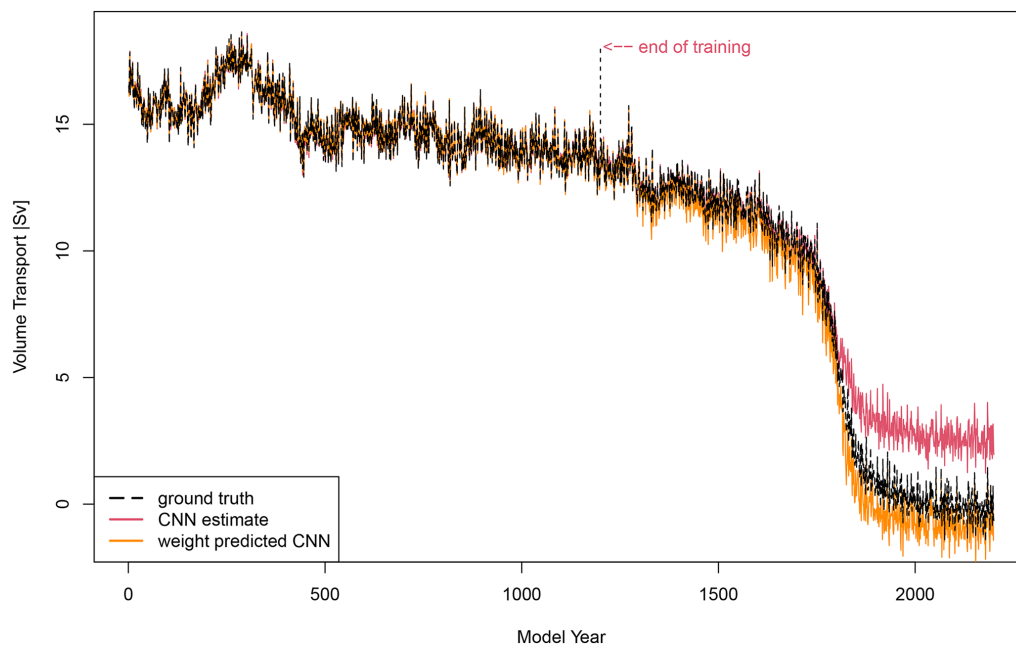


Figure 3. As Fig. 1, but with a sub-optimal example of child model performance. Still, the predicted child model (orange) outperforms the parent model (red) beyond the EOT threshold in the RMSE sense.

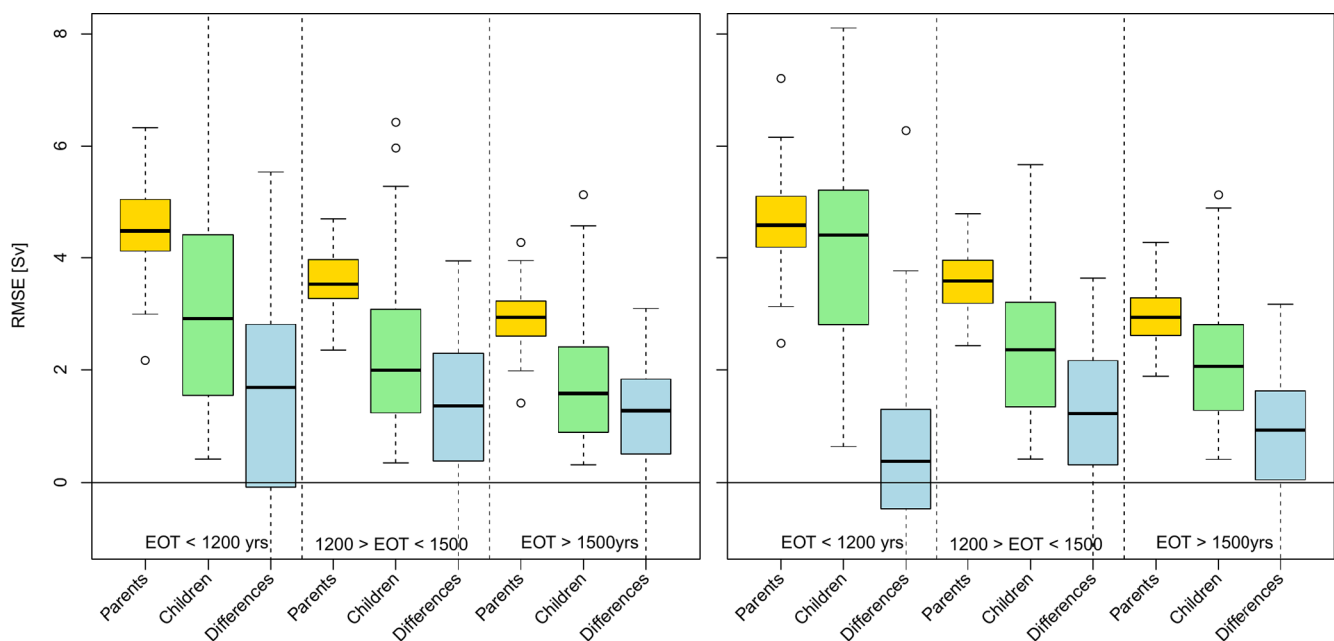


Figure 4. Impact of EOT on the RMSE of a fixed application range (1800 years, 2000 years). Polynomials of order 1 (left panel) and order 2 (right panel) were fitted to the leading 4 principal components of the input variable (i.e., the meridional velocity maps). Yellow Boxes: RMSE of parent models. Green Boxes: RMSE of child models. Blue Boxes: Pairwise RMSE differences (i.e., unperturbed “parent” model minus corresponding weight-predicted “child” model). Positive values represent improvements due to the presented weight-prediction approach. Each box is based on 100 parents/children/pairs.

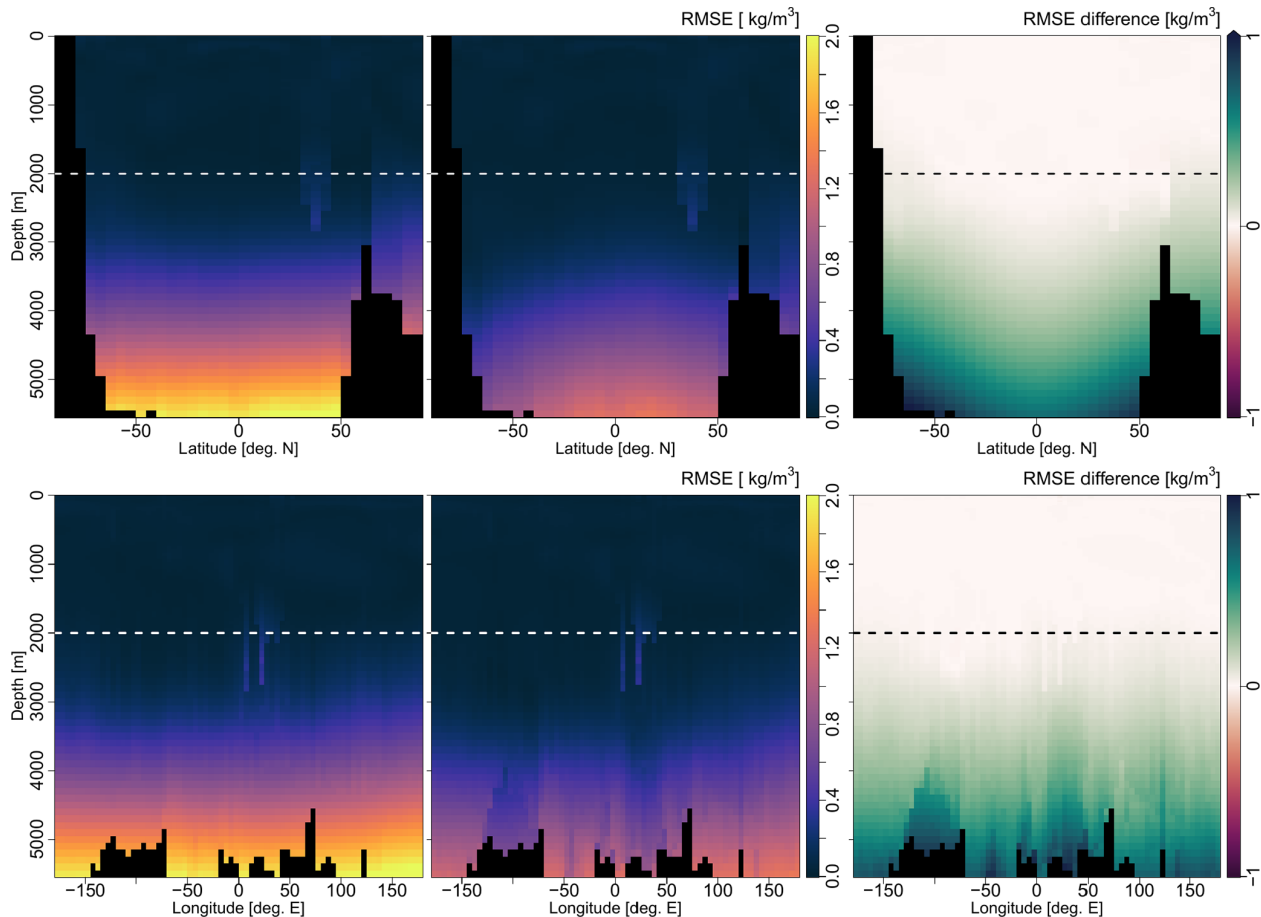


Figure 5. Depth-dependent zonally-averaged* (upper panels) and meridionally averaged* (lower panels) RMSE of NN-based sea water density estimation after training above the dashed line only. Left panels: Performance of the parent model. Middle panels: Performance of weight-predicted child model. Right panels: RMSE differences (parent minus child; the green color represents improvements by the child model). *: In giving every global output location equal weight, these averages represent a data-centered view and do not follow oceanographic standards.

the child model improves the RMSE of the parent model by 50 %–100 %. Regions where the parent model already performs well show less or no improvement. As already discussed in experiment 1, even slightly worse RMSE can be found if the approach is applied to the training data. In contrast to experiment 1, the results of experiment 2 appear to be very stable (not shown).

4.3 Experiment 3: Uncertainty Estimation of Global Wind Velocity Reanalyses

Notable deviations in experiment 3 from the recipe given in Box 1:

- A single NN is used instead of several *RegressionModel_k* in steps 3a & 4a
- Only the final layer of *ParentModel₀* is changed to generate the *ParentModel_i* and *ChildModel_a*

- EOF of the *ParentModel₀* inputs are used as weight predictors

As outlined in Sect. 3.3, this experiment assesses the proposed weight-prediction framework in the context of global wind velocity uncertainty estimation. The focus is on evaluating whether knowledge about uncertainty characteristics learned at one oceanic location can be extrapolated to terrestrial regions with different atmospheric dynamics. This task represents a demanding OOD test case for the proposed method.

Unlike the previous experiments, the child model generation (cf. Box 1, steps 3 and 4) is conducted by a NN, too. This choice of a NN-based child model generation underscores the significance of the underlying weight-prediction methodology over the regression model choice. The child model generation is implemented as a NN consisting of two fully connected hidden layers with 32 nodes each and parameter-specific output heads. Instead of predicting all

network weights and biases of the child model, we restrict the weight-regression and prediction to the final fully connected layer, while keeping the remaining parameters identical to those of the parent model. For this to work correctly, the not-to-be-predicted weights and biases have to be set as `trainable=False` already during the online-learning step (Box 1, step 2). The use of multiple output heads, together with a scaled mean-squared-error (MSE) loss, mitigates scaling disparities across the predicted weights and biases. All layers use ELU activation functions, and the Adam optimizer (learning rate = 0.001) is used.

The training and evaluation periods span 2012–2017, using six-hourly data of the 10 m zonal wind component, from two reanalysis products (ERA5 and CFSv2, cf. Sect. 3.3). Data from 2012–2016 are used for training, January–June 2017 for validation, and July–December 2017 for testing. All results reported below correspond to this testing period. From each of the two input data sets, the principal components of the four leading EOF, yielding eight predictor values in total per time step, are used as weight-regressors. For the regularization of the forgetful online learning (Box 1, step 2a), n is set to 200. The parent model is trained at a single oceanic grid point located at 180° E, 60° S, while the child model predictions are evaluated over the entire landmass to assess the method's ability to generalize beyond the training regime. As in the other experiments, individual *ChildModel_a* are generated for every single application data point x_a , that is, every grid point of the 2d globe.

Figure 6 summarizes the spatial distribution of the RMSE in zonal wind speed uncertainty prediction. Panel (a) shows the RMSE obtained by the child model trained only at the oceanic reference point (black dot), and panel (b) illustrates the parent minus child model RMSE differences. Blue areas in Fig. 6b mark regions where the proposed method outperforms the parent model, i.e., where RMSE are reduced by the weight-prediction method.

From these results, it is evident that the weight-prediction approach substantially enhances the wind velocity uncertainty estimation over continental regions, particularly where the parent model struggles the most. Notable improvements are observed over the southeast coast of Australia, eastern Brazil, the East African Highlands, the Indonesian archipelago, and southeastern Russia. In contrast, regions where the parent model already performs well show no major improvement, as expected. This applies to most of the oceanic region as well (not shown in the figure). This behavior suggests that the proposed method provides the greatest benefit under conditions of strong distributional shift, where the parent model's learned representations are least transferable (cf. experiments 1 and 2).

Quantitatively, the terrestrial global mean RMSE of the uncertainty estimates decreases from 0.1236 to 0.1104 m s^{-1} , corresponding to an average improvement of approximately 10.7%. Repeated experiments indicate that this value varies between −2% and 14%, with a mean

improvement of 5.25%, reflecting again the stochastic variability inherent in NN training. Despite this variability, the consistent tendency across repeated experiments confirms that the proposed method systematically improves the predictive performance of the parent model, particularly in regions where generalization is most challenging.

That a NN can be used to improve the OOD problem of another NN seems contradictory at first. Overall, the terrestrial input is equally OOD to the *ParentModel₀* as well as to the child-generating NN. One could ask why the child-generating NN is not limited by OOD, and why then not one NN alone can handle the problem. A full answer to these questions remains open to further research. So far, we can only argue that we, as users, know if an OOD task we gave to a NN remains generally the same but has to be adapted to perform outside the training realm. Consequently, splitting an OOD-experiment into problem-solving and problem-adaptation represents a form of structural implementation of prior knowledge. This understanding, the required task separation, and the subsequent training of both NN, a single NN cannot (yet) develop itself. Furthermore, as already discussed, a nonlinear regime shift in a NN's input and output data can be less nonlinear in the weight-space of the same NN when nonlinear activation functions are involved.

5 Summary and Final Remarks

In this study, a general method is presented that aims at improving neural network (NN) performance outside of their training data distribution (OOD). The method is proposed for problems where NN are used for one of the various reasons and re-normalization or standardization of the inputs or outputs is not possible due to limited data in few-shot problems or due to governing nonlinear relations, e.g., in climate sciences. Additionally, OOD is meant in a wider sense here. Even when all input variables have the same distribution within and out of training, a trained NN can operate on uncharted territory due to unseen value-combinations and interactions of the input variables.

After the training of a NN, which we call “parent model”, on a specific problem, the method requires the following three steps: First, the method collects the series of weight anomalies that arise by online-relearning or fine-tuning of the parent model towards subsets of the training data. Second, a relation (e.g., a linear regression) is to be established between these weight anomalies and suitable predictors (e.g., the input variables of the training data set itself). Third, the established relation is then used to generate new weights and thus a new NN that corresponds to an application data set by employing the same predictors but now with values that correspond to the application data set (e.g., the application data itself). In analogy to the parent model, we call this new model the “child model”. A child model can encompass a group of NN as a new NN can be generated for every single

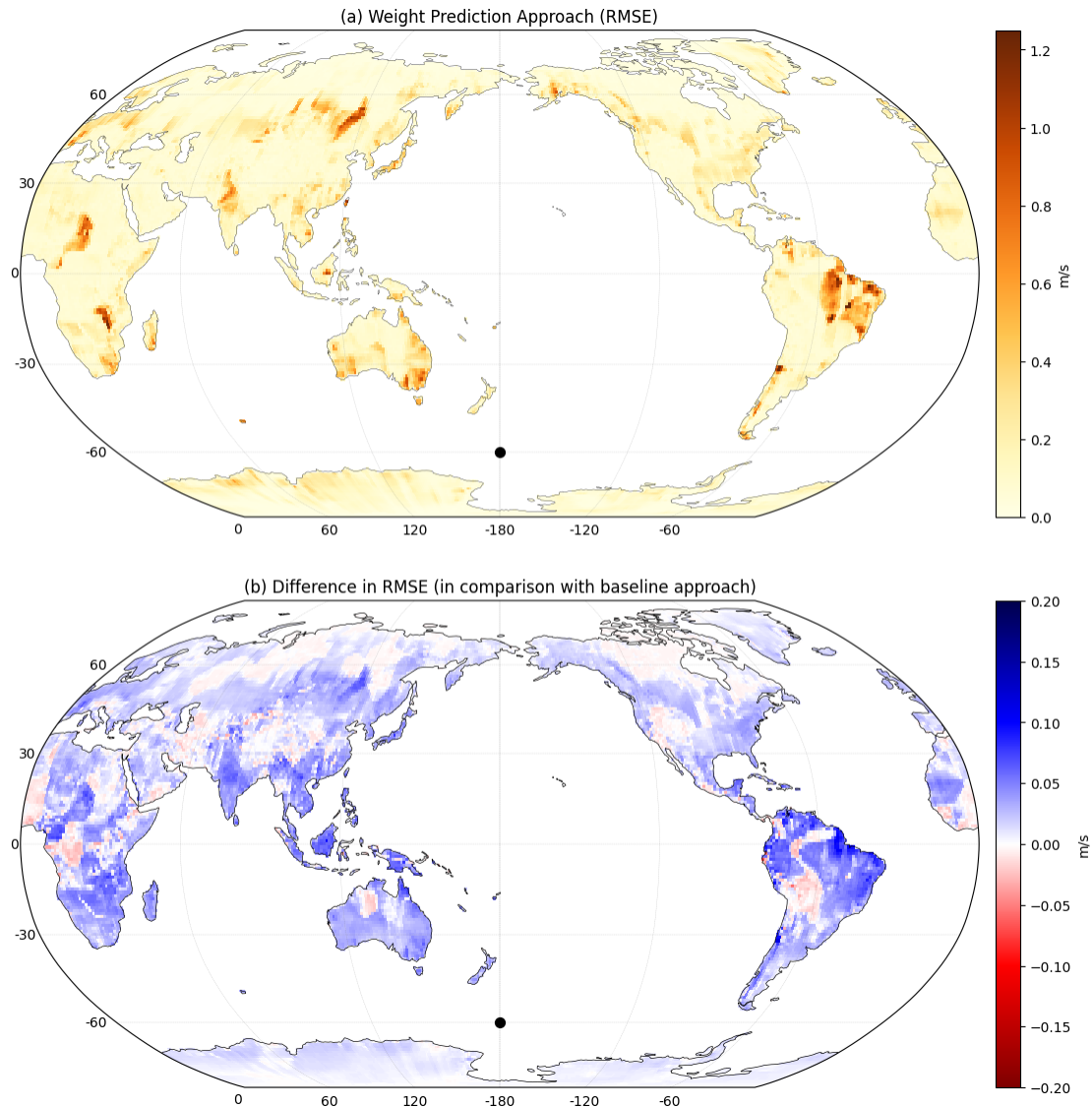


Figure 6. Spatial distribution of RMSE in the prediction of zonal wind-speed uncertainty using the proposed weight-prediction framework. **(a)** RMSE of the child model trained at the reference oceanic location (180° E, 60° S), represented by the black dot. **(b)** Difference in RMSE between the parent model and the child model. Blue regions indicate improvements.

application data point (time step or location). However, sub-setting is possible in all steps of the approach. For example, only one new NN could be generated per cluster of the application data set. Likewise, the weight sensitivities could be derived from quantiles or clusters of training data only.

Note, reasonable choices for the mentioned subsets should depend on the self-similarity the training data, respectively, the application data has. For example, if the training data shows notable clusters, fine-tuning on these clusters should be sufficient. Furthermore, if the application data is rather homogeneous, one does not need to calculate a single child model for every application data point.

In short, the approach is generating NN fine-tuned on subsets of the training data and then is extrapolating along these

NN to data-regions outside of the training data distribution. This way, the network's learned tasks can be transferred, e.g., to different geographic locations, future climates, or across (physical) regimes.

The method is demonstrated on 3 very different examples from Earth sciences: one temporal problem, one spatial problem, and one spatiotemporal problem. These examples and their execution demonstrate further the possible flexibility of the core approach in using linear, nonlinear, and NN-based regression methods; full and partial NN weight prediction; and different realizations of the online training step and its regularization.

In summary, our experiments demonstrate that the proposed weight-prediction framework enhances the extrapo-

lation capability of NN under strong temporal, spatial, and regime shifts. The method proves most effective where traditionally trained and therefore static models exhibit limited skill, underscoring its potential to extend NN capabilities to previously unseen regimes and conditions. As a result, NN can become adaptive and more dynamic operators.

Despite all these promising results, there are still some caveats, as choices have to be made that can heavily impact the method's performance. The choice of suitable predictors, e.g., if one does not choose the input variables themselves, has to be based on prior knowledge. This should pose no major problem for the experienced researcher. The same can be stated about the choice of whether all weights and biases of a NN or only specific subsets should be subjected to weight-regression and subsequent weight-prediction.

In contrast to these rather straightforward choices, the choice of the weight-regression and prediction method cannot be based on prior domain knowledge alone. The reason is that the weight sensitivities the proposed approach has to model are entwined with the network architecture and the employed activation functions. As a first guide, linear or slightly nonlinear regression methods should work fine for most cases.

The biggest drawback identified so far is that the weight-prediction method has inherited the notorious stochastic nature of NN training. From several parent NN (with identical architecture and hyperparameters) that are trained on the same data, not all produce better-performing offspring. To this issue, the most follow-up research will be dedicated. Although we already discussed possible reasons and solutions within the manuscript, we will have to understand the problem better to be able to identify and select parent models suitable for weight-regression or, in the long run, even foster the generation of more suitable parent models through adapted training procedures. Additionally, we see much room for improvement in the generation of the child models by predictor pruning, weight-regression validation, regression-ensembles, transformers, or other means.

As a conclusion, most steps of the approach may be called more or less experimental and can and will be improved in follow-up research. However, already now the method has to be considered very promising and is at least in our experiments robust in the ensemble-average sense.

As a further outlook, we see this method also as a promising tool for NN applications without OOD. During our many tests, we got the impression that the method can improve the performance of undertrained, overfitted, or by design undercomplex models as well.

Despite the fact that all given examples are from Earth sciences, the authors strongly believe that the demonstrated approach is useful in other fields, such as astrophysics, biology, social sciences, or technical applications. For example, in medicine, the approach could be used to transfer learned skills and knowledge to patient groups underrepresented or unrepresented in the training data.

Appendix A

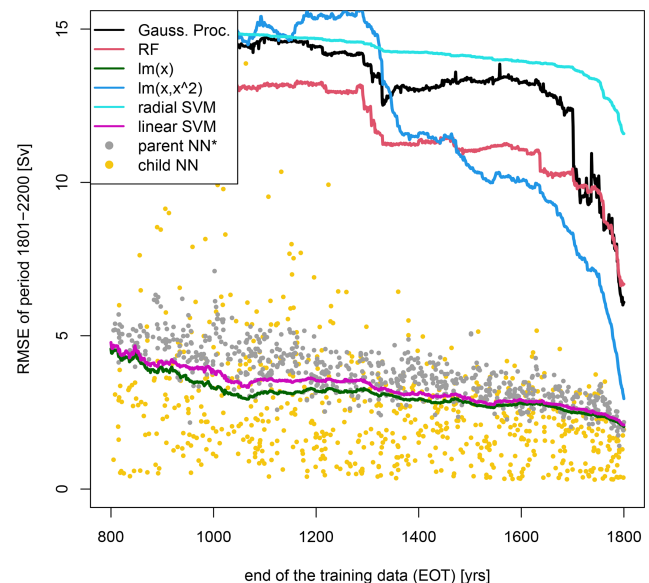


Figure A1. Performance comparison of parent and child NN to non-NN extrapolation methods in experiment 1. Impact of EOT on the RMSE of a fixed application range (1800 years, 2000 years). Lines show non-NN approaches: Gaussian processes (linear predictors), random forest (linear predictors), linear regression (linear predictors), linear regression (linear and squared predictors), linear and radial kernel SVM (linear predictors) are fitted to the leading 4 principal components of the meridional velocity maps. Dots show the parent NN and linear-fit-based child NN (cf. Fig. 4). * Note that the parent NN base their AMOC strength estimates on the velocity fields only, and not on the principal components, as the other methods do, i.e., the NN have to learn a suitable representation themselves, while the other methods benefit from prior knowledge.

To provide a baseline, we present and compare a range of standard linear and nonlinear methods used to extrapolate the targets of experiments 1 and 2, i.e., AMOC strength and sea water density, without any use of NN. The methods (Gaussian process, linear model, random forest, and support vector machine) were fitted to the same predictors as the weight prediction method, i.e., 4 leading EOF (experiment 1) and S , T , P , ϕ , λ , and z (experiment 2). Likewise, the respective OOD projections are based on the same predictors that our weight prediction method uses. Note that none of these methods outperform the child models in the OOD application cases. Most of the methods do not outperform the parent NN (*ParentModel₀*) either. However, even if a method exists that outperforms both the parent and the child NN in our specific use cases, our approach remains useful in all cases where NN are used for one of the various reasons, e.g., ease of use and implementation, existing NN workflows, compute limitations, invertibility requirements, or simple lack of alternatives (e.g., decision making, generative AI, image classification).

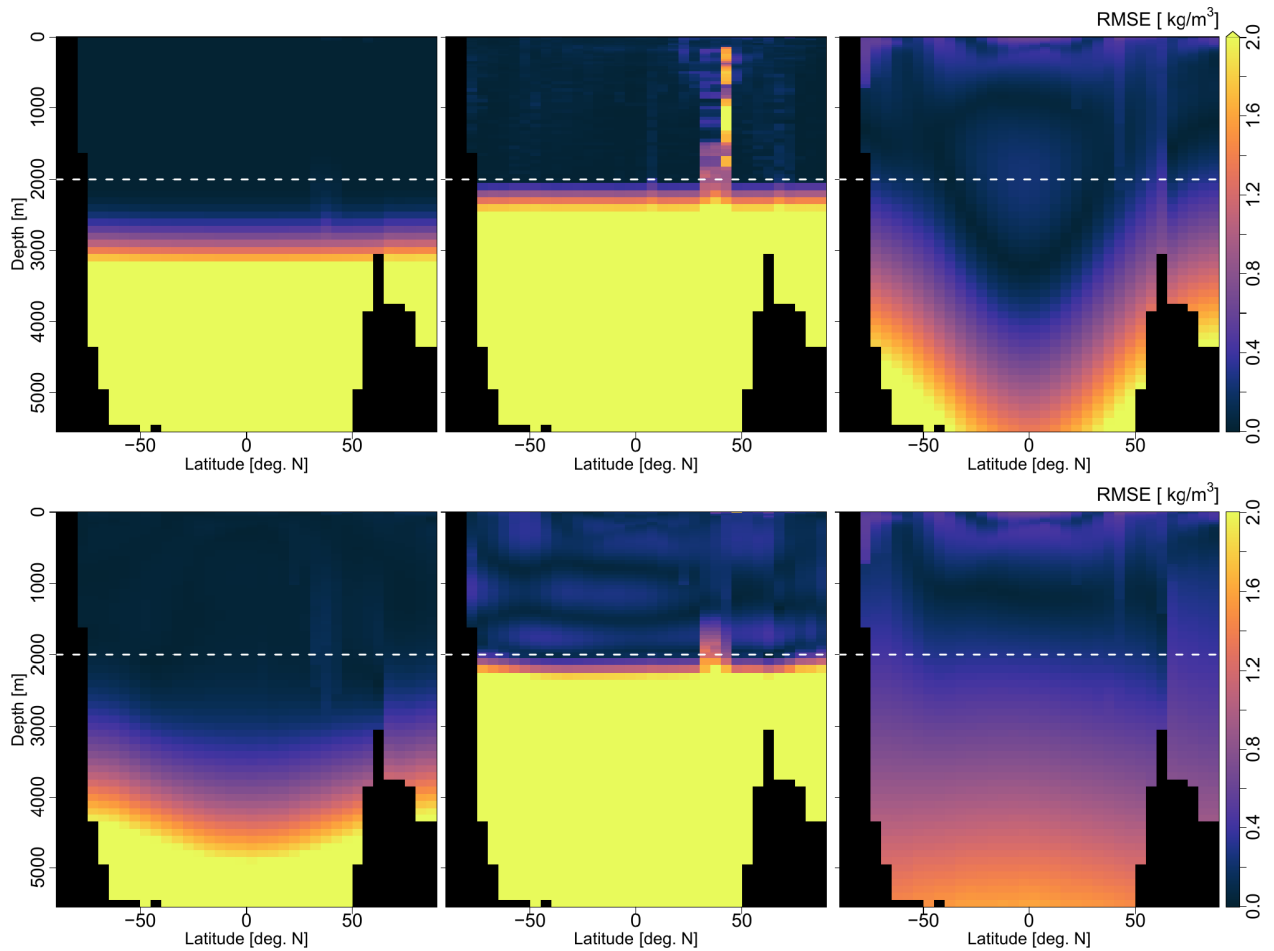


Figure A2. Non-NN model performance of experiment 2 (cf. Fig. 5, upper panel). Depth-dependent zonally-averaged* RMSE of non-NN sea water density estimation after training on ground truth data above the dashed line only. Upper row: Gaussian processes (linear predictors), random forest (linear predictors), linear regression (linear predictors). Lower row: Linear regression (squared predictors), radial kernel SVM (linear predictors), linear kernel SVM (linear predictors). Predictors in all cases are S , T , P , ϕ , λ , and z (cf. Sect. 3.2). * In giving every global output location equal weight, these averages represent a data-centered view and do not follow oceanographic standards.

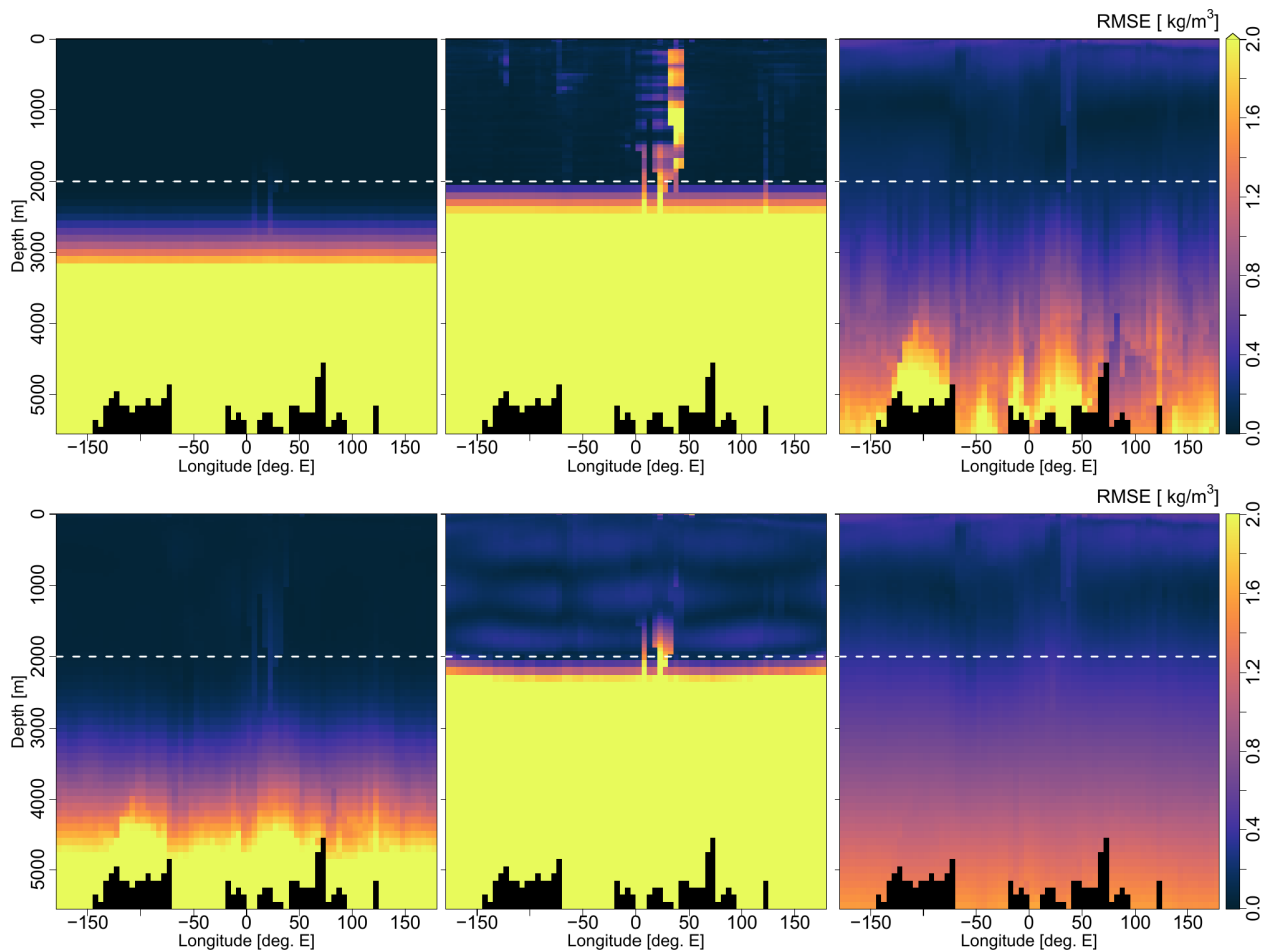


Figure A3. Non-NN model performance of experiment 2 (cf. Fig. 5, lower panel). Depth-dependent meridionally-averaged* RMSE of non-NN sea water density estimation after training on ground truth data above the dashed line only. Upper row: Gaussian processes (linear predictors), random forest (linear predictors), linear regression (linear predictors). Lower row: Linear regression (squared predictors), radial kernel SVM (linear predictors), linear kernel SVM (linear predictors). Predictors in all cases are S , T , P , ϕ , λ , and z (cf. Sect. 3.2). * In giving every global output location equal weight, these averages represent a data-centered view and do not follow oceanographic standards.

Code and data availability. The study was conducted entirely with open data. The AMOC data was taken from the supporting material of van Westen et al. (2024) available at https://github.com/RenevanWesten/SA-AMOC-Collapse/tree/SA-AMOC-Collapse_v1.0 (last access: 1 august 2025). The World Ocean Atlas data (Locarnini et al., 2024; Reagan et al., 2024) is available under <https://www.ncei.noaa.gov/access/world-ocean-atlas-2023/> (last access: 1 September 2025). The ECMWF-ERA5 climate data (Hersbach et al., 2023) is available from the Copernicus Climate Change Service (C3S) Climate Data Store (CDS) at <https://doi.org/10.24381/cds.adbb2d47>. The NCEP-CFSv2 data (Saha et al., 2014) is available from the NOAA Climate Prediction Center at <https://cfs.ncep.noaa.gov/> (last access: 1 October 2025). The study was conducted entirely with open software, namely Keras, Tensorflow, R, and Python. Exemplary source code for the weight prediction approach is available here: <https://doi.org/10.5281/zenodo.20488611> (Saynisch-Wagner and Sari, 2026).

Author contributions. JSW contributed to methodology development, experiment design, implementation, and interpretation. JSW wrote the first draft of the manuscript, financed the project, and did project supervision. SRS contributed to experiment design, method development, implementation, and refining the manuscript.

Competing interests. The contact author has declared that neither of the authors has any competing interests.

Disclaimer. Publisher's note: Copernicus Publications remains neutral with regard to jurisdictional claims made in the text, published maps, institutional affiliations, or any other geographical representation in this paper. The authors bear the ultimate responsibility for providing appropriate place names. Views expressed in the text are those of the authors and do not necessarily reflect the views of the publisher.

Financial support. This research has been supported by the Bundesministerium für Forschung, Technologie und Raumfahrt (BMFTR), within the PalMod III project, and the Helmholtz Association, within the Helmholtz International Berlin Research School in Data Science (HEIBRiDS).

The article processing charges for this open-access publication were covered by the GFZ Helmholtz Centre for Geosciences.

Review statement. This paper was edited by Adarsh Sankaran and reviewed by three anonymous referees.

References

- Bereiter, B., Eggleston, S., Schmitt, J., Nehrbass-Ahles, C., Stocker, T. F., Fischer, H., Kipfstuhl, S., and Chappellaz, J.: Revision of the EPICA Dome C CO₂ record from 800 to 600 kyr before present, *Geophys. Res. Lett.*, 42, 542–549, <https://doi.org/10.1002/2014GL061957>, 2015.
- Bergmeir, C., Hyndman, R., and Koo, B.: A note on the validity of cross-validation for evaluating autoregressive time series prediction, *Comput. Stat. Data An.*, 120, 70–83, <https://doi.org/10.1016/j.csda.2017.11.003>, 2018.
- Beucler, T., Gentine, P., Yuval, J., Gupta, A., Peng, L., Lin, J., Yu, S., Rasp, S., Ahmed, F., O’Gorman, P. A., Neelin, J. D., Lutsko, N. J., and Pritchard, M.: Climate-invariant machine learning, *Sci. Adv.*, 10, eadj7250, <https://doi.org/10.1126/sciadv.adj7250>, 2024.
- Chauhan, V. K., Zhou, J., Lu, P., Molaei, S., and Clifton, D. A.: A brief review of hypernetworks in deep learning, *Artif. Intell. Rev.*, 57, 1–29, <https://doi.org/10.1007/s10462-024-10862-8>, 2024.
- Cortes, C. and Vapnik, V.: Support-vector networks, *Mach. Learn.*, 20, 273–297, <https://doi.org/10.1007/BF00994018>, 1995.
- El Ghawi, R., Winkler, A., Reimers, C., Schall, A., Gensheimer, J., and Kraft, B.: Imitation or Identification: Limitations of Deep Learning in Extrapolating to Future Climate-Carbon Cycle Change, *Mach. Learn.*, 1, <https://doi.org/10.1088/3049-4753/ae2279>, 2025.
- Eyring, V., Bony, S., Meehl, G. A., Senior, C. A., Stevens, B., Stouffer, R. J., and Taylor, K. E.: Overview of the Coupled Model Intercomparison Project Phase 6 (CMIP6) experimental design and organization, *Geosci. Model Dev.*, 9, 1937–1958, <https://doi.org/10.5194/gmd-9-1937-2016>, 2016.
- Eyring, V., Collins, W. D., Gentine, P., Barnes, E. A., Barreiro, M., Beucler, T., Bocquet, M., Bretherton, C. S., Christensen, H. M., Dagon, K., Gagne, D. J., Hall, D., Hammerling, D., Hoyer, S., Iglesias-Suarez, F., Lopez-Gomez, I., McGraw, M. C., Meehl, G. A., Molina, M. J., Monteleoni, C., Mueller, J., Pritchard, M. S., Rolnick, D., Runge, J., Stier, P., Watt-Meyer, O., Weigel, K., Yu, R., and Zanna, L.: Pushing the frontiers in climate modelling and analysis with machine learning, *Nat. Clim. Change*, 14, 916–928, <https://doi.org/10.1038/s41558-024-02095-y>, 2024.
- Fang, S., Wang, Z., Kurths, J., and Fan, J.: Tipping Points and Cascading Transitions: Methods, Principles, and Evidences, *arXiv [preprint]*, <https://doi.org/10.48550/arXiv.2511.01168>, 2025.
- Hastie, T. and Tibshirani, R.: Generalized Additive Models, *Stat. Sci.*, 1, 297–310, <https://doi.org/10.1214/ss/1177013604>, 1986.
- Hernanz, A., García-Valero, J. A., Domínguez, M., and Rodríguez-Camino, E.: A critical view on the suitability of machine learning techniques to downscale climate change projections: Illustration for temperature with a toy experiment, *Atmos. Sci. Lett.*, 23, e1087, <https://doi.org/10.1002/asl.1087>, 2022.
- Hersbach, H., Bell, B., Berrisford, P., Biavati, G., Horányi, A., Muñoz Sabater, J., Nicolas, J., Peubey, C., Radu, R., Rozum, I., Schepers, D., Simmons, A., Soci, C., Dee, D., and Thépaut, J.-N.: ERA5 hourly data on single levels from 1940 to present, Copernicus Climate Change Service (C3S) Climate Data Store (CDS) [data set], <https://doi.org/10.24381/cds.adbb2d47>, 2023.
- Ho, T. K.: Random decision forests, in: Proceedings of 3rd international conference on document analysis and recognition, IEEE, Vol. 1, 278–282, <https://doi.org/10.1109/ICDAR.1995.598994>, 1995.
- Hsieh, W. W.: Improving Predictions by Nonlinear Regression Models from Outlying Input Data, *J. Environ. Inform.*, 41, <https://doi.org/10.3808/jei.202300493>, 2023.
- Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., and Chen, W.: LoRA: Low-rank adaptation of large language models, *arXiv [preprint]*, <https://doi.org/10.48550/arXiv.2106.09685>, 2021.
- IOC, SCOR, and IAPSO: The international thermodynamic equation of seawater – 2010: Calculation and use of thermodynamic properties, Tech. Rep. 56, UNESCO, Intergovernmental Oceanographic Commission, Manuals and Guides No. 56, UNESCO, 196 pp., https://www.teos-10.org/pubs/TEOS-10_Manual.pdf (last access: 1 August 2025), 2010.
- Irrgang, C., Saynisch-Wagner, J., and Thomas, M.: Machine Learning-Based Prediction of Spatiotemporal Uncertainties in Global Wind Velocity Reanalyses, *J. Adv. Model. Earth Sy.*, 12, e2019MS001876, <https://doi.org/10.1029/2019MS001876>, 2020.
- Irrgang, C., Boers, N., Sonnewald, M., Barnes, E. A., Kadow, C., Staneva, J., and Saynisch-Wagner, J.: Towards neural Earth system modelling by integrating artificial intelligence in Earth system science, *Nat. Mach. Intell.*, 3, 667–674, <https://doi.org/10.1038/s42256-021-00374-3>, 2021.
- Jung, H., Saynisch-Wagner, J., and Schulz, S.: Can explainable AI Offer a New Perspective for Groundwater Recharge Estimation?—Global-Scale Modeling Using Neural Network, *Water Resour. Res.*, 60, e2023WR036360, <https://doi.org/10.1029/2023WR036360>, 2024.
- Kelley, D. and Richards, C.: oce: Analysis of Oceanographic Data, r package version 1.8-4, <https://dankelley.github.io/oce/> (last access: 1 August 2025), 2025.
- Kotz, M., Wenz, L., Stechemesser, A., Kalkuhl, M., and Levermann, A.: Day-to-day temperature variability reduces economic growth, *Nat. Clim. Change*, 11, 319–325, <https://doi.org/10.1038/s41558-020-00985-5>, 2021.
- Koutsodendris, A., Pross, J., and Zahn, R.: Exceptional Agulhas leakage prolonged interglacial warmth during MIS 11c in Europe, *Paleoceanography*, 29, 1062–1071, 2014.

- Kvorka, J., Čadek, O., Šachl, L., and Velímský, J.: Convective flow in Ganymede's subsurface ocean: Implications for the induced magnetic field and topography, *Icarus*, 444, 116807, <https://doi.org/10.1016/j.icarus.2025.116807>, 2026.
- Landsberg, J. B. and Barnes, E. A.: Forecasting the future with yesterday's climate: Temperature bias in AI weather and climate models, *Geophys. Res. Lett.*, 53, e2025GL119740, <https://doi.org/10.1029/2025GL119740>, 2026.
- Lee, T. and Gentemann, C.: Satellite SST and SSS observations and their roles to constrain ocean models, in: *New Frontiers in Operational Oceanography*, edited by Chassignet, E., Pascual, A., Tintoré, J., and Verron, J., chap. 11, GODAE OceanView, 271–288, <https://doi.org/10.17125/gov2018.ch11>, 2018.
- Li, X.-M., Chang, L., Zhou, J., Sun, J., Lu, X., Han, X., Wang, J., Lu, H., Li, X., Wang, N., Qiu, Y., and Wu, Y.: Haishao-1 satellite: Low-inclination orbit spaceborne synthetic aperture radar, *The Innovation*, 6, <https://doi.org/10.1016/j.xinn.2025.100949>, <https://doi.org/10.1016/j.xinn.2025.100949>, 2025.
- Llovel, W., Willis, J. K., Landerer, F. W., and Fukumori, I.: Deep-ocean contribution to sea level and energy budget not detectable over the past decade, *Nat. Clim. Change*, 4, 1031–1035, 2014.
- Locarnini, R. A., Mishonov, A. V., Baranova, O. K., Reagan, J. R., Boyer, T. P., Seidov, D., Wang, Z., Garcia, H. E., Bouchard, C., Cross, S. L., Paver, C. R., and Dukhovskoy, D.: *World Ocean Atlas 2023, Volume 1: Temperature*, Tech. Rep. 89, NOAA Atlas NESDIS, <https://doi.org/10.25923/54bh-1613>, 2024.
- Mehta, A., Parsons, D., Lukas Holch, T., Berge, D., and Weidlich, M.: Convolution and Graph-based Deep Learning Approaches for Gamma/Hadron Separation in Imaging Atmospheric Cherenkov Telescopes, *Proceedings of Science, ICRC2025*, 1–8, <https://doi.org/10.22323/1.501.0752>, 2025.
- Nelder, J. A. and Wedderburn, R. W. M.: Generalized Linear Models, *J. Roy. Stat. Soc. Ser. A*, 135, 370–384, 1972.
- Ortega-Cisneros, K., Fierros-Arcos, D., Lindmark, M., Novaglio, C., Woodworth-Jefcoats, P., Eddy, T. D., Coll, M., Fulton, E., Oliveros-Ramos, R., Reum, J., Shin, Y.-J., Bulman, C., Capitani, L., Datta, S., Murphy, K., Rogers, A., Shannon, L., Whitehouse, G. A., Adekoya, E., Dias, B. S., Fuster-Alonso, A., Hansen, C., Husson, B., McGregor, V., Morell, A., Morzaria Luna, H.-N., Ouled-Cheikh, J., Ruzicka, J., Steenbeek, J., Stollberg, I., Subramaniam, R. C., Tulloch, V., Bryndum-Buchholz, A., Harrison, C. S., Heneghan, R., Maury, O., Pozo Buil, M., Schewe, J., Tittensor, D. P., Townsend, H., and Blanchard, J. L.: An Integrated Global-To-Regional Scale Workflow for Simulating Climate Change Impacts on Marine Ecosystems, *Earth's Future*, 13, e2024EF004826, <https://doi.org/10.1029/2024EF004826>, 2025.
- Peel, M. C., Finlayson, B. L., and McMahon, T. A.: Updated world map of the Köppen-Geiger climate classification, *Hydrol. Earth Syst. Sci.*, 11, 1633–1644, <https://doi.org/10.5194/hess-11-1633-2007>, 2007.
- Qin, R., Zhang, G., and Tang, Y.: On the Transferability of Semantic Segmentation for Very-High-Resolution Remote Sensing Data of Multi-City Environments, *Photogramm. Eng. Remote Sens.*, 91, 517–528, <https://doi.org/10.14358/PERS.24-00140R3>, 2025.
- Rahmstorf, S.: On the freshwater forcing and transport of the Atlantic thermohaline circulation, *Clim. Dynam.*, 12, 799–811, <https://doi.org/10.1007/s003820050144>, 1996.
- Rasp, S., Pritchard, M. S., and Gentine, P.: Deep learning to represent subgrid processes in climate models, *P. Natl. Acad. Sci. USA*, 115, 9684–9689, <https://doi.org/10.1073/pnas.1810286115>, 2018.
- Reagan, J. R., Seidov, D., Wang, Z., Dukhovskoy, D., Boyer, T. P., Locarnini, R. A., Baranova, O. K., Mishonov, A. V., Garcia, H. E., Bouchard, C., Cross, S. L., and Paver, C. R.: *World Ocean Atlas 2023, Volume 2: Salinity*, Tech. Rep. 90, NOAA Atlas NESDIS, <https://doi.org/10.25923/70qt-9574>, 2024.
- Roquet, F., Madec, G., McDougall, T. J., and Barker, P. M.: Accurate polynomial expressions for the density and specific volume of seawater using the TEOS-10 standard, *Ocean Model.*, 90, 29–43, <https://doi.org/10.1016/j.ocemod.2015.04.002>, 2015.
- Saha, S., Moorthi, S., Wu, X., Wang, J., Nadiga, S., Tripp, P., Behringer, D., Hou, Y.-T., ya Chuang, H., Iredell, M., Ek, M., Meng, J., Yang, R., Mendez, M. P., van den Dool, H., Zhang, Q., Wang, W., Chen, M., and Becker, E.: The NCEP Climate Forecast System Version 2, *J. Climate*, 27, 2185–2208, <https://doi.org/10.1175/JCLI-D-12-00823.1>, 2014.
- Saynisch, J., Petereit, J., Irrgang, C., Kuvshinov, A., and Thomas, M.: Impact of climate variability on the tidal oceanic magnetic signal — A model-based sensitivity study, *J. Geophys. Res.-Oceans*, 121, 5931–5941, <https://doi.org/10.1002/2016JC012027>, 2016.
- Saynisch-Wagner, J. and Sari, S. R.: Computational implementation accompanying: Conditional Updates of Neural Network Weights for Increased Out-of-Training Performance, *Zenodo* [code], <https://doi.org/10.5281/zenodo.20488611>, 2026.
- Schannwell, C., Mikolajewicz, U., Kapsch, M.-L., and Ziemann, F.: A mechanism for reconciling the synchronisation of Heinrich events and Dansgaard-Oeschger cycles, *Nat. Commun.*, 15, 2961, <https://doi.org/10.1038/s41467-024-47141-7>, 2024.
- Schnaubelt, M.: A comparison of machine learning model validation schemes for non-stationary time series data, Tech. Rep. 11/2019, FAU Discussion Papers in Economics, <https://www.econstor.eu/bitstream/10419/209136/1/1684440068.pdf> (last access: 1 December 2025), 2019.
- Sebastianelli, A., Spiller, D., Carmo, R., Wheeler, J., Nowakowski, A., Jacobson, L. V., Kim, D., Barlevi, H., Cordero, Z. E. R., Colón-González, F. J., Lowe, R., Ullo, S. L., and Schneider, R.: A reproducible ensemble machine learning approach to forecast dengue outbreaks, *Sci. Rep.*, 14, 3807, <https://doi.org/10.1038/s41598-024-52796-9>, 2024.
- van der Deure, T., Nogués-Bravo, D., Njotto, L. L., and Stensgaard, A.-S.: Climate Change Favors African Malaria Vector Mosquitoes, *Glob. Change Biol.*, 31, e70610, <https://doi.org/10.1111/gcb.70610>, 2025.
- Wang, J. Y., Nikolaou, N., an der Heiden, M., and Irrgang, C.: High-resolution modeling and projection of heat-related mortality in Germany under climate change, *Eur. J. Publ. Hlth.*, 34, ckae144.025, <https://doi.org/10.1093/eurpub/ckae144.025>, 2024.
- van Westen, R. M., Kliphuis, M., and Dijkstra, H. A.: Physics-based early warning signal shows that AMOC is on tipping course, *Sci. Adv.*, 10, eadk1189, <https://doi.org/10.1126/sciadv.adk1189>, 2024.
- Wong, A. P. S., Wijffels, S. E., Riser, S. C., Pouliquen, S., Hosoda, S., Roemmich, D., Gilson, J., Johnson, G. C., Martini, K., Murphy, D. J., Scanderbeg, M., Bhaskar, T. V. S. U., Buck, J. J. H., Merceur, F., Carval, T., Maze, G., Cabanes, C., André, X., Poffa, N., Yashayaev, I., Barker, P. M., Guinehut, S., Bel-

béoch, M., Ignaszewski, M., Baringer, M. O., Schmid, C., Lyman, J. M., McTaggart, K. E., Purkey, S. G., Zilberman, N., Alkire, M. B., Swift, D., Owens, W. B., Jayne, S. R., Hersh, C., Robbins, P., West-Mack, D., Bahr, F., Yoshida, S., Sutton, P. J. H., Cancouët, R., Coatanoan, C., Dobbler, D., Juan, A. G., Gourrion, J., Kolodziejczyk, N., Bernard, V., Bourlès, B., Claustre, H., D'Ortenzio, F., Le Reste, S., Le Traon, P.-Y., Rannou, J.-P., Saout-Grit, C., Speich, S., Thierry, V., Verbrugge, N., Angel-Benavides, I. M., Klein, B., Notarstefano, G., Poulain, P.-M., Vélez-Belchí, P., Suga, T., Ando, K., Iwasaka, N., Kobayashi, T., Masuda, S., Oka, E., Sato, K., Nakamura, T., Sato, K., Takatsuki, Y., Yoshida, T., Cowley, R., Lovell, J. L., Oke, P. R., van Wijk, E. M., Carse, F., Donnelly, M., Gould, W. J., Gowers, K., King, B. A., Loch, S. G., Mowat, M., Turton, J., Rama Rao, E. P., Ravichandran, M., Freeland, H. J., Gaboury, I., Gilbert, D., Greenan, B. J. W., Ouellet, M., Ross, T., Tran, A., Dong, M., Liu, Z., Xu, J., Kang, K., Jo, H., Kim, S.-D., and Park, H.-M.: Argo Data 1999–2019: Two Million Temperature-Salinity Profiles and Subsurface Velocity Observations From a Global Array of Profiling Floats, *Front. Mar. Sci.*, 7, <https://doi.org/10.3389/fmars.2020.00700>, 2020.