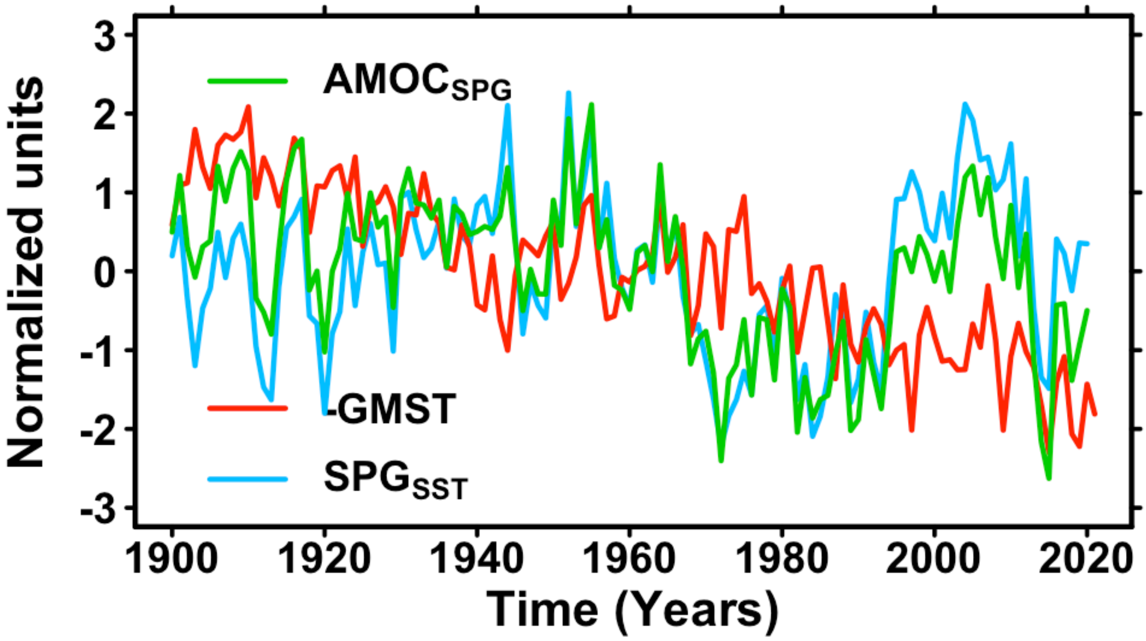
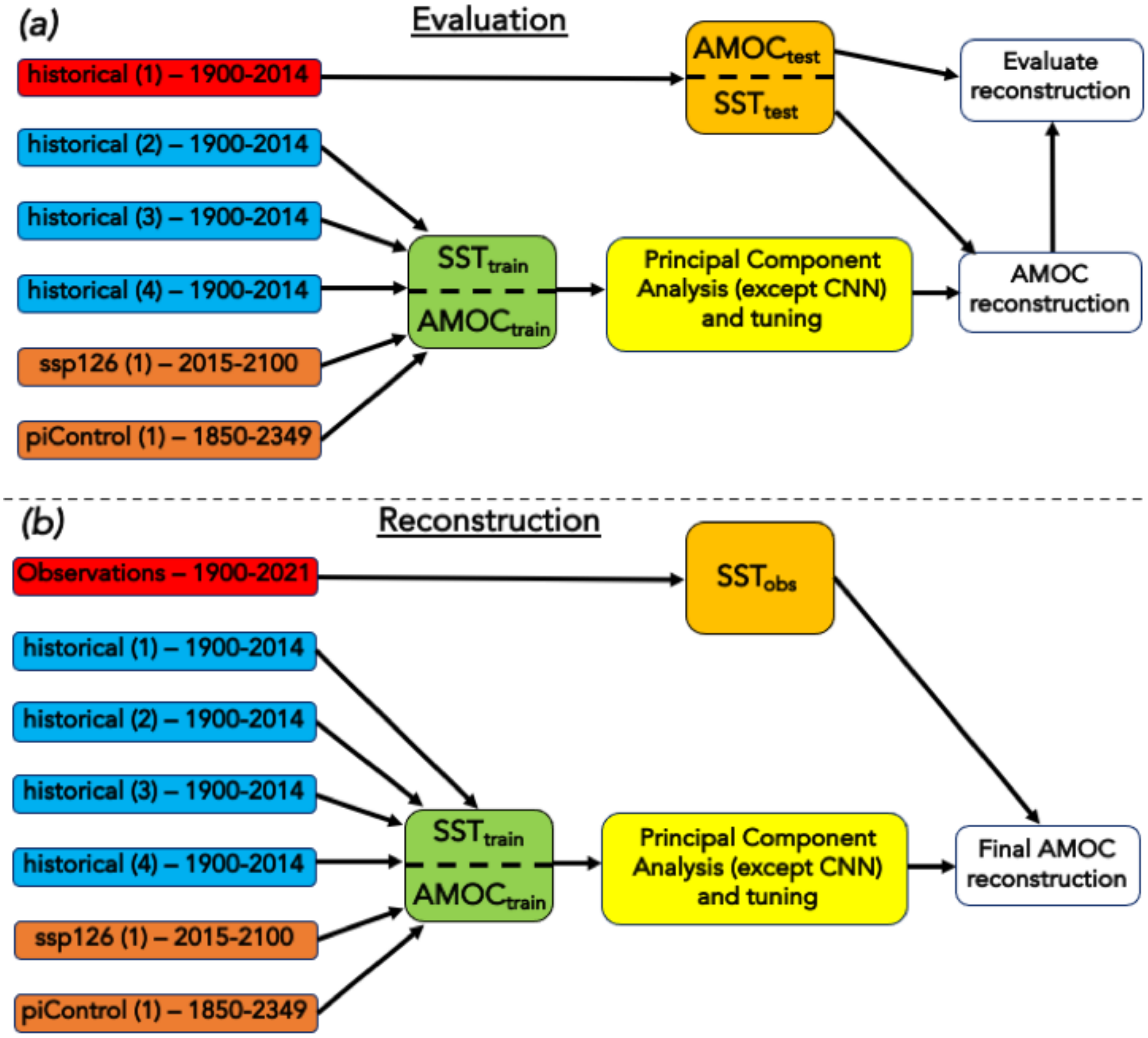
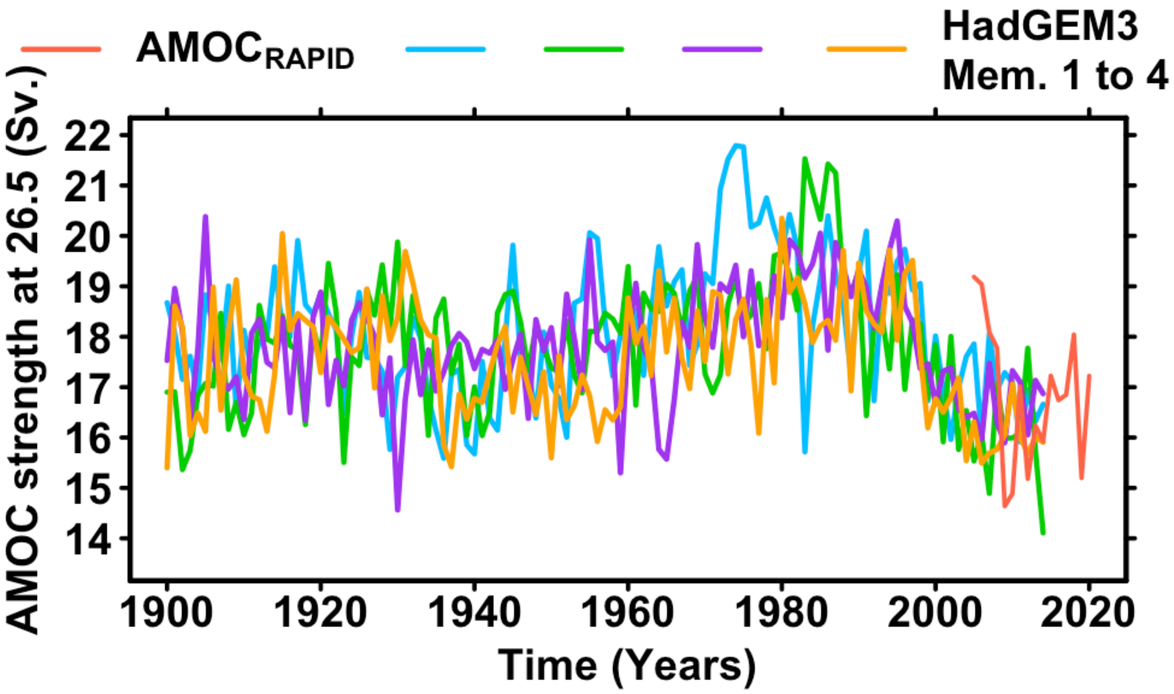


Supplementary Figure 1: State-of-the-art Atlantic Meridional Overturning Circulation (AMOC) index calculation. Subpolar gyre AMOC index (AMOC$_{SPG}$, green) calculated as Global Mean SST anomalies (GMST, red, multiplied by -1) subtracted from area-averaged subpolar gyre SST (SPG$_{SS}$, blue), see Methods. All time series are normalized for graphical representation.

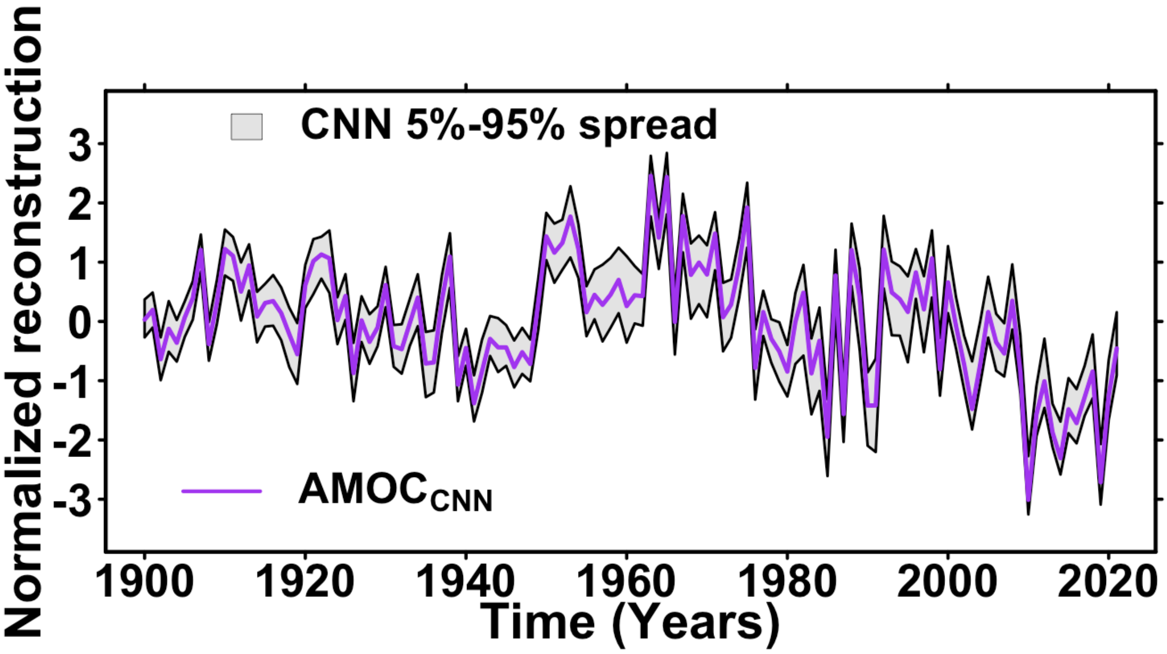


(a)
Evaluation
historical (1) – 1900-2014
historical (2) – 1900-2014
historical (3) – 1900-2014
historical (4) – 1900-2014
ssp126 (1) – 2015-2100
piControl (1) – 1850-2349
AMOC_{test} / SST_{test}
SST_{train} / AMOC_{train}
Principal Component Analysis (except CNN) and tuning
Evaluate reconstruction
AMOC reconstruction
(b)
Reconstruction
Observations – 1900-2021
historical (1) – 1900-2014
historical (2) – 1900-2014
historical (3) – 1900-2014
historical (4) – 1900-2014
ssp126 (1) – 2015-2100
piControl (1) – 1850-2349
SST_{obs}
SST_{train} / AMOC_{train}
Principal Component Analysis (except CNN) and tuning
Final AMOC reconstruction

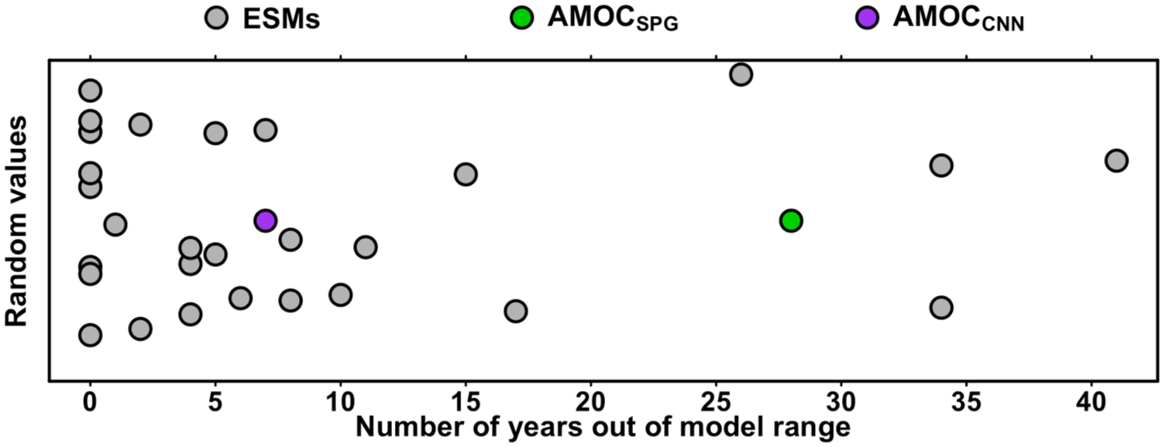
Supplementary Figure 2: Machine learning framework of the study. a. Scheme of the evaluation of the machine learning methods (Fig. 2-3). The scheme here shows the evaluation for the first historical simulation member. This evaluation is done for all four historical simulation members. **b.** Scheme of the final Atlantic Meridional Overturning Circulation reconstruction from sea surface temperature observations.



Supplementary Figure 3: Comparison of Atlantic Meridional Overturning Circulation (AMOC) in 4 historical HADGEM3 runs and direct measurements. Red line is the timeseries of direct AMOC observations from RAPID[4,6], and blue, green, purple, and orange lines indicate historical AMOC timeseries as simulated by the HadGEM3 Earth System Model (Supplementary Fig. 2).



Supplementary Figure 4: Sensitivity analysis of the Convolutional Neural Network (CNN) method applied to real observations. Purple line: The Atlantic Meridional Overturning Circulation (AMOC) index reconstructed as the median from 500 CNN reconstructions (AMOC$_{CNN}$, Methods). Grey shaded. area: 5-95% envelop from the same 500 CNN reconstructions (Methods).



Supplementary Figure 5: Distribution of out-of-range years in historical Earth System Model (ESM) simulations. Each dot represents the number of times (x-axis, in years) a timeseries from Fig. 5a. lies outside of the CMIP6 ESM range described by 28 single-member simulations (Supplementary Table 4). We used random values on the y-axis to ease the graphical representation. Grey dots: ESM values. Green dot: value obtained for the subpolar gyre Atlantic Meridional Overturning Circulation (AMOC) index (AMOC$_{SPG}$, Methods). Purple dot: value obtained for the AMOC reconstructed from convolutional neural network (AMOC$_{CNN}$, Methods).

Model name	Modelling center (country)	Experiment (period)	Members
BCC-CSM2-MR	BCC (China)	historical (1900-2014)	r1i1p1f1, r2i1p1f1, r3i1p1f1
CESM2	NCAR (USA)	historical (1900-2014)	r1i1p1f1, r2i1p1f1, r3i1p1f1
CanESM5	CCCma (Canada)	historical (1900-2014)	r1i1p2f1, r2i1p2f1, r3i1p2f1
E3SM-2-0	DOE (USA)	historical (1900-2014)	r1i1p1f1, r2i1p1f1, r3i1p1f1
FGOALS-f3-L	CAS (China)	historical (1900-2014)	r1i1p1f1, r2i1p1f1, r3i1p1f1
GISS-E2-2-H	NASA GISS (USA)	historical (1900-2014)	r1i1p1f1, r2i1p1f1, r3i1p1f1
HadGEM3-GC31-MM	MOHC (UK)	historical (1900-2014)	r1i1p1f3, r2i1p1f3, r3i1p1f3
INM-CM5-0	INM (Russia)	historical (1900-2014)	r1i1p1f1, r2i1p1f1, r3i1p1f1
IPSL-CM6A-LR	IPSL (France)	historical (1900-2014)	r1i1p1f1, r2i1p1f1, r3i1p1f1
KACE-1-0-G	NIMS-KMA (Korea)	historical (1900-2014)	r1i1p1f1, r2i1p1f1, r3i1p1f1
MIROC-ES2H	JAMSTEC (Japan)	historical (1900-2014)	r1i1p4f1, r2i1p4f1, r3i1p4f1
MIROC6	JAMSTEC (Japan)	historical (1900-2014)	r1i1p1f1, r2i1p1f1, r3i1p1f1
MPI-ESM1-2-HAM	MPI (Germany)	historical (1900-2014)	r1i1p1f1, r2i1p1f1, r3i1p1f1
MPI-ESM1-2-HR	MPI (Germany)	historical (1900-2014)	r1i1p1f1, r2i1p1f1, r3i1p1f1
NorCPM1	NCC (Norway)	historical (1900-2014)	r1i1p1f1, r2i1p1f1, r3i1p1f1
NorESM2-MM	NCC (Norway)	historical (1900-2014)	r1i1p1f1, r2i1p1f1, r3i1p1f1
UKESM1-0-LL	MOHC (UK)	historical (1900-2014)	r1i1p1f2, r2i1p1f2, r3i1p1f2

Supplementary Table 1: List of 51 historical simulations from 17 CMIP6 Earth System Models for the bias analysis presented in Fig. 1.

Model name	Modelling center (country)	Experiment (period)	Member
HadGEM3-GC31-MM	MOHC (UK)	historical (1900-2014)	r1i1p1f3
HadGEM3-GC31-MM	MOHC (UK)	historical (1900-2014)	r2i1p1f3
HadGEM3-GC31-MM	MOHC (UK)	historical (1900-2014)	r3i1p1f3
HadGEM3-GC31-MM	MOHC (UK)	historical (1900-2014)	r4i1p1f3
HadGEM3-GC31-MM	MOHC (UK)	piControl (1850-2349)	r1i1p1f1
HadGEM3-GC31-MM	MOHC (UK)	ssp1-2.6 (2015-2100)	r1i1p1f3

Supplementary Table 2: List of the 6 HadGEM3 simulations used for training CNN (Figs. 2-4) and other machine learning methods (Figs. 2).

CNN model	Batch size	Initial learning rate	Number of epochs
Historical 1 excluded	32	0.0005	4000
Historical 2 excluded	128	0.0001	4000
Historical 3 excluded	256	0.001	4000
Historical 4 excluded	64	0.0005	3000
All HadGEM3 simulations	64	0.0005	4000

Supplementary Table 3: List of tuned parameters for the five CNN models of the study.

Model name	Modelling center (country)	Experiment (period)	Members
ACCESS-CM2	MOHC (UK)	historical (1900-2014) ssp2-4.5 (2015-2021)	r1i1p1f1 r1i1p1f1
ACCESS-ESM1-5	MOHC (UK)	historical (1900-2014) ssp2-4.5 (2015-2021)	r1i1p1f1 r1i1p1f1
CESM2	NCAR (USA)	historical (1900-2014) ssp2-4.5 (2015-2021)	r1i1p1f1 r1i1p1f1
CESM2-WACCM	NCAR (USA)	historical (1900-2014) ssp2-4.5 (2015-2021)	r1i1p1f1 r1i1p1f1
CIESM	THU (China)	historical (1900-2014) ssp2-4.5 (2015-2021)	r1i1p1f1 r1i1p1f1
CMCC-CM2-SR5	CMCC (Italy)	historical (1900-2014) ssp2-4.5 (2015-2021)	r1i1p1f1 r1i1p1f1
CMCC-ESM2	CMCC (Italy)	historical (1900-2014) ssp2-4.5 (2015-2021)	r1i1p1f1 r1i1p1f1
CNRM-CM6-1	CNRM (France)	historical (1900-2014) ssp2-4.5 (2015-2021)	r1i1p1f1 r1i1p1f1
CNRM-CM6-1-HR	CNRM (France)	historical (1900-2014) ssp2-4.5 (2015-2021)	r1i1p1f1 r1i1p1f1
CNRM-ESM2-1	CNRM (France)	historical (1900-2014) ssp2-4.5 (2015-2021)	r1i1p1f1 r1i1p1f1
CanESM5	CCCma (Canada)	historical (1900-2014) ssp2-4.5 (2015-2021)	r1i1p1f1 r1i1p1f1
CanESM5-CanOE	CCCma (Canada)	historical (1900-2014) ssp2-4.5 (2015-2021)	r1i1p1f1 r1i1p1f1
EC-Earth3	EC-Earth Consortium (EU)	historical (1900-2014) ssp2-4.5 (2015-2021)	r1i1p1f1 r1i1p1f1
EC-Earth3-Veg	EC-Earth Consortium (EU)	historical (1900-2014) ssp2-4.5 (2015-2021)	r1i1p1f1 r1i1p1f1
FGOALS-f3-L	CAS (China)	historical (1900-2014) ssp2-4.5 (2015-2021)	r1i1p1f1 r1i1p1f1
FGOALS-g3	CAS (China)	historical (1900-2014) ssp2-4.5 (2015-2021)	r1i1p1f1 r1i1p1f1
GFDL-ESM4	NOAA GFDL (USA)	historical (1900-2014) ssp2-4.5 (2015-2021)	r1i1p1f1 r1i1p1f1
GISS-E2-1-G	NASA GISS (USA)	historical (1900-2014) ssp2-4.5 (2015-2021)	r1i1p1f1 r1i1p1f1
HadGEM3-GC31-LL	MOHC (UK)	historical (1900-2014) ssp2-4.5 (2015-2021)	r1i1p1f1 r1i1p1f1
INM-CM4-8	INM (Russia)	historical (1900-2014) ssp2-4.5 (2015-2021)	r1i1p1f1 r1i1p1f1
INM-CM5-0	INM (Russia)	historical (1900-2014) ssp2-4.5 (2015-2022)	r1i1p1f1 r1i1p1f1
IPSL-CM6A-LR	IPSL (France)	historical (1900-2014) ssp2-4.5 (2015-2022)	r1i1p1f1 r1i1p1f1
MIROC-ES2L	JAMSTEC (Japan)	historical (1900-2014) ssp2-4.5 (2015-2021)	r1i1p1f1 r1i1p1f1
MIROC6	JAMSTEC (Japan)	historical (1900-2014) ssp2-4.5 (2015-2021)	r1i1p1f1 r1i1p1f1
MPI-ESM1-2-HR	MPI (Germany)	historical (1900-2014) ssp2-4.5 (2015-2021)	r1i1p1f1 r1i1p1f1
MPI-ESM1-2-LR	MPI (Germany)	historical (1900-2014) ssp2-4.5 (2015-2021)	r1i1p1f1 r1i1p1f1
MRI-ESM2-0	MRI (Japan)	historical (1900-2014)	r1i1p1f1

		ssp2-4.5 (2015-2021)	r1i1p1f1
UKESM-1-0	MOHC (UK)	historical (1900-2014) ssp2-4.5 (2015-2021)	r1i1p1f1 r1i1p1f1

Supplementary Table 4: List of 29 CMIP6 Earth System Model historical simulations (single members) for the ESM evaluation and estimated forced Atlantic Meridional Overturning Circulation component from Fig. 5.

Model name	Modelling center (country)	Experiment (period)	Member
ACCESS-CM2	CSIRO-ARCCSS (Australia)	piControl (1150-1449)	r1i1p1f1
CAS-ESM2-0	CAS (China)	piControl (200-499)	r1i1p1f1
CMCC-CM2-SR5	CMCC (Italy	piControl (2050-2349)	r1i1p1f1
CESM2	NCAR (USA)	piControl (700-999)	r1i1p1f1
CanESM5	CCCma (Canada)	piControl (5901-6200)	r1i1p1f1
E3SM-1-0	DOE (USA)	piControl (201-500)	r1i1p1f1
FGOALS-g3	CAS (China)	piControl (400-699)	r1i1p1f1
HadGEM3-GC31-MM	MOHC (UK)	piControl (2050-2349)	r1i1p1f1
INM-CM5-0	INM (Russia)	piControl (2897-3196)	r1i1p1f1
MIROC6	JAMSTEC (Japan)	piControl (1900-2014)	r1i1p1f1
MPI-ESM1-2-HR	MPI (Germany)	piControl (2050-2349)	r1i1p1f1
UKESM1-0-LL	MOHC (UK)	piControl (2650-2949)	r1i1p1f2

Supplementary Table 5: List of 12 CMIP6 Earth System Model preindustrial control simulations (single members) for the EWS analysis presented in Fig. 6b.

Supplementary Note 1: Other Machine learning (ML) methods compared with Convolutional Neural Network (CNN).

1) Generalities

As for the CNN, each machine learning (ML) method must be tuned (Methods). Compared to the CNN, the other ML methods require a pre-screening of the explainable data field (sea surface temperature, SST, here), here denoted by F and described as a time-varying two-dimensional field (*i.e,* a three-dimensional data): $F = \left(f_{ijt}\right), i \in \varphi, j \in \Theta, t \in T$. Here, $\varphi, \Theta, \text{and } T$ are longitude, latitude, and time spaces, with sizes denoted p, q, and r, respectively. For each ML method, the longitude/latitude space is described by the regular 1°x1° grid over the study area: [20°N-70°N, 80°W-0°]. T depends on the different data considered in the study. The prescreening of F consists in arranging all its time series of the two-dimensional spatial field as the columns of a same initial matrix, denoted $X^{(i)} \in \mathbb{R}^{n\times r}$, where, $r = p \times q$ is the number of grid points in the SST field.

Because of the large computational efforts required to produce the results, we made the decision to apply the machine learning techniques on Principal Components (PCs) of $X^{(i)}$ to realize a finer tuning of the hyperparameters. PCs are computed as the projection of $X^{(i)}$ onto an optimal (in terms of explained variance) orthogonal basis of the correlation matrix of $X^{(i)}$. Each PC from $X^{(i)}$ with eigenvalues (*i.e.*, fraction of explained variance) higher than $1/r$ are kept since $1/r$ is the theoretical weight all PCs would have under the hypothesis that all columns of $X^{(i)}$ are independent in the case of a standardized principal components. In the following, the matrix of PCs has dimension $\mathbb{R}^{n\times s}$. The target AMOC$_{26}$ to model by ML methods is denoted $Y \in \mathbb{R}^{n}$. A given AMOC reconstruction for k time steps is denoted $\hat{Y} \in \mathbb{R}^{k}$, and is obtained using the first a PCs from a new SST field projected on the eigenvectors of $X^{(i)}$, denoted $X' \in \mathbb{R}^{k\times s}$

The above principal component analysis prescreening was not necessary for the CNN, as it deals with 3-dimensional data directly.

Methods are summarized, readers may refer to the associated references for more details[51-59].

2) Machine learning methods

We here present the seven machine learning methods and identify the different control parameters that we tuned (Supplementary Fig. 2) to produce the comparisons with CNN performances from Fig. 2.

a) (Principal Component) regression (PCR)

Since we work on PCs for ML methods (see 1), what is called PC regression[51] in the main text, is a linear regression of X that is the matrix of the first s PCs from $X^{(i)}$ (see 1). The one difference is that $X_1, \ldots, X_s$ are sorted by their explained variance from the initial $X^{(i)}$ data, so there is a number $a \leq s$ of first PCs with highest explained variances that can be tuned using 10-fold cross-validation[48] (Methods). Once this parameter is tuned (Methods), the PCR model is given by:

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_a X_a + \varepsilon$$

The ordinary least squares estimator of beta is given by:

$$\hat{\beta} = \arg\min_{\beta \in \mathbb{R}^{p+1}} \varepsilon = (\mathbb{X}^T \mathbb{X})^{-1} \mathbb{X}^T Y$$

With $\mathbb{X} = (\mathbb{I}_n, X) \in \mathbb{R}^{n \times (a+1)}$, with $\mathbb{I}_n$ a unique vector (only composed of ones) of size n. $\beta = (\beta_0, \beta_1, \ldots, \beta_a) \in \mathbb{R}^{a+1}$ the vector of regression coefficients to estimated, $\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_a) \in \mathbb{R}^{a+1}$.

The obtained AMOC reconstruction is given by:

$$\hat{Y} = (\mathbb{I}_n, X'_1, \dots, X'_a)\hat{\beta}$$

b) Support Vector Machine

Support Vector Machine (SVM) is a classical ML technique used in general for classification tasks[55]. However, a variant of it may be used for regression. It is called ε-insensitive SVM regression (ε-SVR). We have inputs $X \in \mathbb{R}^{n\times s}$, the timeseries of length n of the s first PCs of the original SST fields (see 1) ; and target AMOC values $Y \in \mathbb{R}^n$. The objective of ε-SVR is to find a function f that best maps the inputs to the outputs, an error ε being allowed. The test function f is defined as[55,56]:

$$f(X) = \sum_{i=1}^{n}(\alpha_i^* - \alpha_i)G(X_i, X) + b, \forall X \in \mathbb{R}^s$$

Where $G(u, v) =< \varphi(u), \varphi(v) >$ is the kernel function and α_i^*, α_i, b are real parameters to be optimized. We will use a chose kernel function so that we don't have to know the exact transformation φ. Here, we use the radial basis functions kernel:

$$G(u, v) = e^{-s\|u-v\|^2}, \forall u, v \in \mathbb{R}^s$$

Optimizing the function f amounts to minimizing the following Lagrangian in dual space:

$$L(\alpha, \alpha^*) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)G(X_i, X_j) + \varepsilon\sum_{i=1}^{n}(\alpha_i - \alpha_i^*)\sum_{i=1}^{n}Y_i(\alpha_i - \alpha_i^*)$$

Subject to the constraints:

$$\begin{cases} \forall i = \{1, \dots, n\}, 0 \leq \alpha_i \leq C \\ \forall i = \{1, \dots, n\}, 0 \leq \alpha_i^* \leq C \\ \sum_{i=1}^{n}\alpha_i = \sum_{i=1}^{n}\alpha_i^* \end{cases}$$

Where we have set $C = 1$ and ε is the parameter on which the cross validation is applied.

Once the tuned parameters have been obtained, we can apply the optimized function $\hat{f}$ to obtain the reconstructed AMOC values. Given, $X' \in \mathbb{R}^{n \times s}$, the ε-SVR reconstruction is obtained via:

$$\forall j \in \{1, \dots, k\}, \hat{Y}_j = \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) G\big(X_i, X'_j\big) + b$$

c) Lasso, Ridge, and Elastic Net (ENet)

The linear problem from 2) the ordinary least squares estimator is obtained solving the convex optimization of the L cost function:

$$L(\beta) = \left\| Y - \sum_{j=1}^{p} \beta_j X^j \right\|^2$$

The usual linear model, as well as PCR (a), are based on a heteroscedasticity assumption (*i.e.*, normality of residuals), that is not always verified. This why so-calleed. regularized regression models, such as Lasso[52], Ridge[53], and Enet[54] regressions have been developed. The Enet method combines Lasso and Ridge and may result as equivalent to one of these two regression models after parameter tuning[54].

In Lasso and Ridge, the regularization term acts as a threshold constraint, based on the β l_k norm: $||\beta||_k^k = \sum_{j=1}^{p} |\beta_j|^k$, with $k = 1$ for Lasso and $k = 2$ for Ridge. Cost functions for the three approaches are given by:

$$L^{Ridge}(\beta) = \left\| Y - \sum_{j=1}^{p} \beta_j X^j \right\|^2 + \lambda_2 \sum_{j=1}^{p} \beta_j^2$$

$$L^{Lasso}(\beta) = \left\| Y - \sum_{j=1}^{p} \beta_j X^j \right\|^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j|$$

$$L^{Enet}(\beta) = \left\| Y - \sum_{j=1}^{p} \beta_j X^j \right\|^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=1}^{p} \beta_j^2$$

With $\lambda_1, \lambda_2 > 0$.

Now let $\omega = (\omega_j)_{1 \leq j \leq p} = (sgn(\beta_j))_{1 \leq j \leq p}$, where sgn is the sign function. Costs functions can then be written:

$$L^{Ridge}(\beta) = \left\| Y - \sum_{j=1}^{p} \beta_j X^j \right\|^2 + \lambda_2 \beta^T \beta$$

$$L^{Lasso}(\beta) = \left\| Y - \sum_{j=1}^{p} \beta_j X^j \right\|^2 + \lambda_1 \omega^T \beta$$

$$L^{Enet}(\beta) = \left\| Y - \sum_{j=1}^{p} \beta_j X^j \right\|^2 + \lambda_1 \omega^T \beta + \lambda_2 \beta^T \beta$$

Estimates obtained from resolving the above convex problems[54] are given by:

$$\hat{\beta}^{Lasso} = (X^T X)^{-1}(X^T Y - \frac{\lambda_1}{2}\omega)$$

$$\hat{\beta}^{Ridge} = (X^T X + \lambda_2 I)^{-1} X^T Y$$

$$\hat{\beta}^{Enet} = (X^T X + \lambda_2 I)^{-1} X^T Y$$

The $\hat{\beta}^{Enet}$ estimator can also be written[54]:

$$\hat{\beta}^{Enet} = (X^T X + (1-\alpha) I)^{-1}(X^T Y - \frac{\alpha\lambda}{2}\omega)$$

Where $\alpha \in [0,1]$. Thus if $\alpha = 1$, $\hat{\beta}^{Enet}$ is effectively a Ridge regression estimator, whereas if $\alpha = 0$, it is a Lasso regression estimator. For Enet, α and $\lambda > 0$ are both tuned (Methods). For Lasso (resp. Ridge) the tuning (Methods) is only made for λ_1 (resp. λ_2).

The AMOC predictions for the three methods are respectively given by:

$$\hat{Y}^{Enet} = ({X'}_1, \dots, {X'}_a)\hat{\beta}^{Enet}$$

$$\hat{Y}^{Lasso} = ({X'}_1, \dots, {X'}_a)\hat{\beta}^{Lasso}$$

$$\hat{Y}^{Ridge} = ({X'}_1, \dots, {X'}_a)\hat{\beta}^{Ridge}$$

d) Random Forest (RF) and Extremely randomized trees (EXT)

Both RF[57] and EXT[58] are bootstrap aggregating methods that consist in aggregating reconstructions obtained from several Regression Trees[57] (RTs), where RTs differ for some aspects for RF and EXT (see below).

i) Original RT

We not the learning sample composed of pairs of AMOC index values (Y) and first PCs of SST values (X), organized year by year, and denoted $\{(Y_i, X_i)_{1 \le i \le n}\}$. Hence $(X_i^j)_{1 \le j \le s}$ is the i^{th} values of the j^{th} from $X^{(i)}$, modeled with the associated with the AMOC values for the same i^{th} value.

RT is a recursive algorithm consisting of several steps each depending on the former one. The initial node of a given tree $\mathcal{T}$ to which the learning sample $\{(Y_i, X_i)_{1 \le i \le n}\}$ is assigned, is called the root of the RT and is denoted η. In the following notation, we consider that each pair of direct branches from a given node goes to two child nodes (left and right). A node without subsequent child node is named a leaf[57].

The first step of the algorithm consists in finding the optimal cut of η for which the sum of variances of Y in child nodes is minimized. This procedure where a not is separated in two son-nodes is called a cut[55]. Any cut of the j^{th} variable (here, PC, see 1), $1 \leq j \leq s$, is denoted:

$$\{X^j < d\} \cup \{X^j > d\}$$

Where d is the threshold value for which each value from X^j that are lower (resp. higher) are set in the left (resp. right) child node.

Finding the optimal cut of the root is equivalent to solving the following bivariate convex problem:

$$(j, d) = \arg \min_{\substack{d \in \mathbb{R} \\ 1 \leq j \leq p}} \sum_{i:X_i^j < d} (Y_i - \frac{1}{\#(\{i: X_i^j < d\})} \sum_{i:X_i^j < d} Y_i)^2$$
$$+ \sum_{i:X_i^j > d} (Y_i - \frac{1}{\#(\{i: X_i^j > d\})} \sum_{i:X_i^j > d} Y_i)^2$$

Where # is the cardinal operator.

After this first step, we obtain two child nodes η_1, η_2 of the root. In the next step, the best cut among all existing child nodes is selected, meaning that a third variable is optimized:

$$(\eta, j, d) = \arg \min_{\substack{d \in \mathbb{R} \\ 1 \leq j \leq p \\ \eta \in L(\mathcal{T})}} \sum_{i:X_i^j < d} (Y_i - \frac{1}{\#(\{i: X_i^j < d\})} \sum_{i:X_i^j < d} Y_i)^2$$
$$+ \sum_{i:X_i^j > d} (Y_i - \frac{1}{\#(\{i: X_i^j > d\})} \sum_{i:X_i^j > d} Y_i)^2$$

Where $L(\mathcal{T})$ is the number of leaves in $\mathcal{T}$ at the current step of the algorithm (*i.e.*, $\mathcal{T} = \{\eta_1, \eta_2\}$ at this step of the algorithm. $L(\mathcal{T})$ updated every time a leaf is created or turned to a node only (*i.e.*, when it gets two child nodes) with the new set of leaves.

The algorithm stops when one newly built child note contains c values or less (see below).

ii) RF

The accuracy of RT is limited in that if overfits strongly the training data[57]. Indeed, if the first selected variables by the algorithm to cut the root of tree is removed, the model turns out to be completely different. This highlights very poor robustness of the algorithm. It is based on this observation that RF was developed[55].

RF consists in generating a large number of B RTs and aggregating their reconstructions. A randomness aspect is introduced by randomly selecting $d \leq s$ variables from X (thus from the s first PCs of $X^{(i)}$ here). Once B RTs have been constructed, they are browed using their established cuts using data from X'. For a given RT, values from X'_i ending up in a leaf denoted l, the AMOC reconstruction for the corresponding i^{th} value of the sample is given by the average of training values that endud-up in that same leaf:

$$\frac{1}{\#\{l\}} \sum_{\substack{i:Y_i \in l \\ i \in \{1,\ldots,n\}}} Y_i$$

The reconstructed value thus corresponds to the average of training AMOC values (Y) that ended up in leaf l after training.

One can identify 3 parameters for RF tuning: The number of trees B, the threshold c at which RTs computations stop, and the number of d randomly drawn variables from X (here s first PCs of $X^{(i)}$, see 1). However, it was shown that the gain in accuracy from tuning c was not worth the computation time for most datasets tested, and a value of $s = 5$ is often used by default[57]. Similarly, for values of $B \geq 256$ RTs, the computation time for the global RF tuning is not worth the computation time[59]. We thus set $s = 5$[57] and $B = 300$[59] for these two parameters. The only parameter we tune here (Methods) is thus d, *i.e.*, the number of randomly variables in X (here s first PCs of $X^{(i)}$, see 1).

iii) EXT

EXT is a close variant of RF[58]. The way trees are constructed and aggregated is the same, so as the reconstruction. The difference is in the way the RT algorithm is computed. After the first step (cut of η, see ii) that is the same as RF's RTs, the best cut is found on a randomly drawn leaf rather than optimized over all leaves. This reduces the individual accuracies but increases their RTs' computing speed dramatically[58]. EXT is thus generally faster than RF with similar levels of accuracy, even when computing thousands of trees[58].

The only parameter we tune here (Methods) is also d, *i.e.*, the number of randomly variables in X (here s first PCs of $X^{(i)}$, see 1). We set $s = 5$[57] and $B = 3000$[58] for the other two parameters (see ii).

51. Joliffe, I. T. Chapter 8: Principal Components in Regression Analysis. In: *Principal Component Analysis*, pp. 129-155 (1986).

52. Tibshirani, R. Regression shrinkage and selection via Lasso. *Journ. Roy. Stat. Soc.* **58**, 267–288 (1996).

53. Hoerl, A. E. and Kennard, R. W. Ridge regression: Biased estimation of nonorthogonal problems. *Technometrics* **12**, 55–67 (1970).

54. Zou, H. and Hastie, T. Regularization and variable selection via the elastic net. *Journ. Roy. Stat. Soc.*, **67** :301–320 (2005).

55. Vapnik, V. N. The Nature of Statistical Learning Theory. *Statistics for Engineering and Information Science* (2000).

56. Drucker, H., Burges, C. J. C., Kaufman, L. et al. Support Vector Regression Machines. In: Proceedings of the 9th International Conference on Neural Information Processing Systems, *MIT Press*, 155-161.

57. Breiman, L. (2001). Random forests. *Mach. Learn.* **45**, 5-32.

58. Geurts, P., Ernst, D. & Wehenkel, L. Extremely randomized trees. *Mach. Learn.* **63**, 3–42 (2006).

59. Oshiro, T. M., Perez, P. S., and Baranauskas, J. A. How many trees in a random forest ? *Lect. Note Comp. Sci.*, **7376**, 154–168 (2012).