Supplemental Information: *GEN^2: A Generative Prediction-Correction Framework for Long-time Emulations of Spatially-Resolved Climate Extremes*

Mengze Wang[1†], Benedikt Barthel Sorensen[1†], Themistoklis P. Sapsis[1*]

[1*]Department of Mechanical Engineering, Massachusetts Institute of Technology, Street, Cambridge, 02139, MA, USA.

*Corresponding author(s). E-mail(s): sapsis@mit.edu;
†These authors contributed equally to this work.

Supplementary Methods

Supplementary Notes

Supplementary Figures

Supplementary Tables

Supplementary Methods

1 Stochastic Emulator

Here we describe in detail the first part of our climate modeling framework, the linear stochastic emulation. In summary, the emulator takes as input a time series of the global mean temperature $T_g(t)$ and outputs a time series of the local state of the climate $\boldsymbol{u}(\boldsymbol{x},t) = [u_1, u_2, u_3, u_4]^\top = [U(\boldsymbol{x},t), V(\boldsymbol{x},t), T(\boldsymbol{x},t), Q(\boldsymbol{x},t)]^\top$ where U, V, T, Q are the zonal and meridional wind speeds, temperature and humidity respectively and the spatial dimensions $\boldsymbol{x} = (\theta, \varphi)$ are the longitude and latitude, $\theta \in [-\pi/2, \pi, 2]$ and $\varphi \in [0, 2\pi)$. The time step size of t is three hours for ERA5 dataset and one day for the CMIP6 MPI model. Consistent with the formulation of modern climate models – and to reduce the data to a manageable size – our model operates at a fixed altitude, and thus the spatial dimension is 2D. We focus here exclusively on the near-surface climate, but our model could be directly applied to any altitude.

Stated succinctly, our approach consider a principal component analysis (PCA) of the climate data $\boldsymbol{u}(\boldsymbol{x},t) = \sum_j a_j(t)\boldsymbol{\phi}_j(\boldsymbol{x})$ and attempts to model the temporal coefficients $a_j(t)$ for a given spatial basis $\boldsymbol{\phi}_j(\boldsymbol{x})$. Our emulator is therefore built on three fundamental assumptions:

1. The PCA basis $\boldsymbol{\phi}_j(\boldsymbol{x})$ computed from the climate during a sufficiently long time period (e.g. historical and SSP5-8.5 scenario) remains an efficient basis for describing other future climate change scenarios;
2. The seasonal mean and variance of the coefficients $a_j(t)$ vary linearly with global mean temperature.
3. The statistics of daily fluctuations, given the season, are independent of the year and the climate change scenarios.

The construction of the emulator can be divided into two distinct steps: dimensionality reduction and stochastic modeling of PCA time series. The emulator is then nudged towards the observation data to facilitate machine-learning-based debiasing. We now describe each of these steps in detail.

1.1 Dimensionality Reduction

First we describe how the spatial PCA basis $\boldsymbol{\phi}_j(\boldsymbol{x})$, which provides the structure for our emulator, is computed. Given a dataset consisting of N years, we extract the climatological mean $\bar{\boldsymbol{u}}(\boldsymbol{x},t)$, defined as phase-average of $\boldsymbol{u}$ on the same calendar day (e.g. Jan 1st),

$$\bar{\boldsymbol{u}}(\boldsymbol{x},t) = \frac{1}{N}\sum_{n=0}^{N-1} \boldsymbol{u}(\boldsymbol{x}, t+nT), \quad 1 \le t \le T. \tag{1}$$

, where the period T is one year. When the emulator is trained on the daily maximum data from the MPI model, the climatological mean (1) only quantifies the seasonal cycle. For the three-hourly ERA5 data, $\bar{\boldsymbol{u}}$ accounts for not only the seasonal variation but also the diurnal cycles. To obtain the scaling of each state variable, we compute

the global-and-time-averaged standard deviation,

$$\sigma_{g,k} = \left[\frac{1}{\mathcal{T}S} \int_0^{\mathcal{T}} \int_S \left(u_k(\boldsymbol{x},t) - \bar{u}_k(\boldsymbol{x},t) \right)^2 \cos\theta d\theta d\varphi dt \right]^{1/2}. \tag{2}$$

The notations $\mathcal{T}$ and S are the duration of training window and the Earth's surface, respectively. The data are then centered to have zero climatological mean and scaled by the global standard deviation,

$$u'_k(\boldsymbol{x},t) = \left(u_k(\boldsymbol{x},t) - \bar{u}_k(\boldsymbol{x},t) \right) / \sigma_{g,k}. \tag{3}$$

Now that each component of q'_k has the same order of magnitude, we construct its spatial covariance function,

$$\mathcal{R}_{jk}(\boldsymbol{x},\boldsymbol{x}^*) = \frac{1}{\mathcal{T}} \int_0^T u'_j(\boldsymbol{x},t) u'_k(\boldsymbol{x}^*,t) dt, \quad j,k = 1,2,3,4. \tag{4}$$

The PCA modes are acquired by solving the eigenvalue problem,

$$\int_S \sum_k \mathcal{R}_{jk}(\boldsymbol{x},\boldsymbol{x}^*) \phi_k(\boldsymbol{x^*}) \cos\theta d\theta d\varphi = \lambda \phi_j(\boldsymbol{x}), \quad j = 1,2,3,4, \tag{5}$$

This set of equations has multiple solutions $(\lambda^{(i)}, \boldsymbol{\phi}^{(i)})$, $i = 1,2,3,\ldots$, which are the PCA eigenvalues and mode shapes, respectively. Without loss of generality we rank the eigenpairs such that the eigenvalues, which represent variance, satisfy $\lambda_1 > \lambda_2 > \ldots > \lambda_I$. The temporal PCA coefficients which govern the time dependence of the spatial PCA modes are found by projecting the normalized fluctuation field onto $\boldsymbol{\phi}^{(i)}$,

$$a_i(t) = \int_S \sum_k u'_k(\boldsymbol{x},t) \phi_k^{(i)}(\boldsymbol{x}) \cos\theta d\theta d\varphi. \tag{6}$$

The state of the climate can then be expressed as superposition of PCA modes,

$$u_k(\boldsymbol{x},t) = \bar{u}_k(\boldsymbol{x},t) + \sigma_{g,k} \sum_{i=1}^{I} a_i(t) \phi_k^{(i)}(\boldsymbol{x}). \tag{7}$$

When the number of PCA modes I is equal to the number of grid points or the number of snapshots, whichever is smaller, we recover the full field, and any smaller value of I represents a truncation. In this work, we always retain 500 PCA modes, which represent 79.6% of the total variance of 1979-2018 ERA5 data and 78.2% of 1950-2100 MPI data. To reiterate, we assume that the climatological mean $\bar{u}_k(\boldsymbol{x},t)$, global standard deviation $\sigma_{g,k}$, and PCA mode shapes $\phi_k^{(i)}$ are unchanged with time

or future scenarios. As a result, we focus purely on modeling $a_i(t)$, and the emulated state is written as,

$$\hat{q}_k(\boldsymbol{x}, t) = \bar{u}_k(\boldsymbol{x}, t) + \sigma_{g,k} \sum_{i=1}^{500} \hat{a}_i(t) \phi_k^{(i)}(\boldsymbol{x}). \tag{8}$$

Here the notations with $\hat{\cdot}$ are emulated quantities.

1.2 Stochastic emulator of PCA time series

1.2.1 Seasonal Decomposition

Our goal is to construct a time series of $\hat{a}_i(t)$ that *statistically* resembles the reference data $a_i(t)$. Although we have removed the climatological mean, the statistics of $a_i(t)$ still exhibit seasonal variation that is important to take into account. Therefore, we divide $a_i(t)$ into four seasons $a_{s,i}(t)$ – of approximately equal length – and model them separately where the additional subscript $s = 1, 2, 3, 4$ represents winter (Dec-Feb), spring (Mar-May), summer (Jun-Aug) and autumn (Sep-Nov). The number of days in each season is 90, 92, 92, and 91 respectively.

1.2.2 Formulation and Estimation of Model Parameters

We postulate a decomposition of the time series of PCA coefficients,

$$\hat{a}_{s,i}(t) = \hat{\mu}_{s,i}\left(T_{s,g}\right) + \hat{\sigma}_{s,i}\left(T_{s,g}\right) \hat{\eta}_{s,i}(t), \tag{9}$$

which is a superposition of the seasonal mean $\hat{\mu}_{s,i}$ and fluctuations parameterized through an envelope of the seasonal variance $\hat{\sigma}^2_{s,i}$. The seasonal mean and variance are assumed to be functions of the global mean temperature $T_{s,g}$, defined as the seasonal average of the daily T_g. The time-dependent daily fluctuations in each season are modelled as autoregressive Gaussian processes $\hat{\eta}_{s,i}(t)$. We will now discuss the formulation and computation of each of these terms in detail.

Linear Regression of Seasonal Mean and Variance. For each season s and each mode i, in the nth year, we compute the $T_{s,g}$ as well as the seasonal mean $\mu_{s,i}$ and variance $\sigma^2_{s,i}$ of the PCA coefficients of the reference data $a_{s,i}(t)$. Note that for each s, i, and n, the mean $\mu_{s,i}$ and variance $\sigma^2_{s,i}$ are constants – we generally omit explicit notation of the year n to avoid notational clutter. Grouping these values by season s and mode i allows us to perform a linear regression using $\{\mu_{s,i}(n), T_{s,g}(n)\}$ and $\{\sigma^2_{s,i}(n), T_{s,g}(n)\}$

$$\begin{aligned} \hat{\mu}_{s,i}(T_{s,g}) &= \hat{p}_{s,i,0} + \hat{p}_{s,i,1} T_{s,g} \\ \hat{\sigma}^2_{s,i}(T_{s,g}) &= \hat{q}_{s,i,0} + \hat{q}_{s,i,1} T_{s,g}, \end{aligned} \tag{10}$$

an assumption which is justified by the linear trends which have been observed in data by a number of sources [1, 2] and illustrated in figure 1.

Time Lagged Cross-Mode Covariance. After extracting the linear trends of the seasonal mean and standard deviation in response to the global mean temperature, we remove these trends from the true PCA coefficients, resulting in the residuals $\eta_{s,i} = (a_{s,i} - \hat{\mu}_{s,i})/\hat{\sigma}_{s,i}$. To accurately capture the spatio-temporal dynamics, our model must reflect not only the contemporaneous correlations between different modes, e.g. $\eta_{s,1}(t)$ and $\eta_{s,2}(t)$, but also their correlations across time. To this end, we define the *time-lagged cross-mode covariance*,

$$\mathbf{\Sigma}_s(m) = \frac{1}{\mathcal{T}_s} \int_{\mathcal{T}_s} \boldsymbol{\eta}_s(t) \boldsymbol{\eta}_s^\top (t + m\Delta t) dt, \quad m = 0, 1, \dots M, \tag{11}$$

where $\boldsymbol{\eta}_s = [\eta_{s,1}, \eta_{s,2}, \dots, \eta_{s,m}]^\top$ is the vector of fluctuations of each PCA mode, $M\Delta t$ is maximum time lag considered, and $\mathcal{T}_s$ represents the set of time indices corresponding to season s across all training years.

Now we want to model the observed fluctuations $\eta_{s,i}(t)$ as a multivariate Gaussian process $\hat{\eta}_{s,i}(t)$, which has the same covariance matrix $\mathbf{\Sigma}_s(m)$ as $\eta_{s,i}(t)$. To further simplify our notation, the subscript s will be omitted. Mathematically, we seek to construct an autoregressive model of order M,

$$\hat{\boldsymbol{\eta}}(t) = \hat{\mathbf{\Psi}}_1 \hat{\boldsymbol{\eta}}(t - \Delta t) + \hat{\mathbf{\Psi}}_2 \hat{\boldsymbol{\eta}}(t - 2\Delta t) + \cdots + \hat{\mathbf{\Psi}}_M \hat{\boldsymbol{\eta}}(t - M\Delta t) + \boldsymbol{\epsilon}(t) \tag{12}$$

where the noise term is a multivariate Gaussian random vector $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \hat{\mathbf{R}})$. The unknown matrices $\hat{\mathbf{\Psi}}_1, \hat{\mathbf{\Psi}}_2, \dots, \hat{\mathbf{\Psi}}_M, \hat{\mathbf{R}}$ are solved such that the simulated process (12) satisfy the given covariance matrices with different time lags $\mathbf{\Sigma}(0), \mathbf{\Sigma}(1), \dots \mathbf{\Sigma}(M\Delta t)$. By multiplying both sides of (12) by $\hat{\boldsymbol{\eta}}(t - i\Delta t)$ and averaging in time, we can derive a set of equations, the so-called Yule-Walker equations,

$$\begin{bmatrix} \mathbf{\Sigma}(0) & \mathbf{\Sigma}^\top(1) & \cdots & \mathbf{\Sigma}^\top(M-1) \\ \mathbf{\Sigma}(1) & \mathbf{\Sigma}(0) & \cdots & \mathbf{\Sigma}^\top(M-2) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{\Sigma}(M-1) & \mathbf{\Sigma}(M-2) & \cdots & \mathbf{\Sigma}(0) \end{bmatrix} \begin{bmatrix} \hat{\mathbf{\Psi}}_1^\top \\ \hat{\mathbf{\Psi}}_2^\top \\ \vdots \\ \hat{\mathbf{\Psi}}_M^\top \end{bmatrix} = \begin{bmatrix} \mathbf{\Sigma}(1) \\ \mathbf{\Sigma}(2) \\ \vdots \\ \mathbf{\Sigma}(M) \end{bmatrix}. \tag{13}$$

which may be readily solved for the $\hat{\mathbf{\Psi}}_j$ [3]. The corresponding noise covariance is then given by

$$\hat{\mathbf{R}} = \mathbf{\Sigma}(0) - \sum_{m=1}^{M} \hat{\mathbf{\Psi}}_m \mathbf{\Sigma}(m). \tag{14}$$

After solving equations (13,14), the matrices $\hat{\mathbf{\Psi}}_1, \hat{\mathbf{\Psi}}_2, \dots, \hat{\mathbf{\Psi}}_M, \hat{\mathbf{R}}$ are substituted into the autoregressive model (12) to simulate the daily fluctuations. The complete procedures for running the emulator are summarized in Algorithm 1.

1.3 Nudging the Stochastic Emulator

The stochastic emulator introduced previously was designed to capture the second-order statistics of the leading PCA modes. While this emulator has demonstrated

Algorithm 1 Stochastic emulator of global climate.

Input: Temporal evolution of global mean temperature $T_g(t)$
Output: Emulated statistics of climate variables

Step 1: Emulate seasonal mean and variance;
- Compute seasonal global mean temperature $T_{s,g}$ for each season s;
- For each mode i, predict the seasonal mean $\hat{\mu}_{s,i}(T_{s,g})$ and variance $\hat{\sigma}^2_{s,i}(T_{s,g})$.

Step 2: Generate stochastic daily fluctuations;
- At every time step t, sample a Gaussian random vector $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \hat{\mathbf{R}})$;
- Compute the vector autoregressive process $\hat{\boldsymbol{\eta}}(t)$ according to equation (12).

Step 3: Construct time series of spatial fields;
- Combine seasonal mean and variance with daily fluctuations to obtain $\hat{a}_i(t)$;
- Multiply PCA coefficients by their mode shapes and superpose all the modes;
- Denormalize by $\boldsymbol{\sigma}_g$ and $\bar{u}$ to compute U, V, T, Q in physical space (8).

Step 4: Estimate statistics of interest;
- Average the spatial fields of U, V, T, Q over window to calculate statistics;
- If needed, input $T_g(t)$ from a different ensemble member, repeat steps 1-3 and average over multiple members.

effectiveness in representing the conditional Gaussian distribution of certain variables, such as temperature [4], it inherently struggles to reproduce the non-Gaussian characteristics of climate data, including extreme events associated with higher-order PCA modes. A common approach to addressing this limitation involves using machine-learning models to debias the emulator. However, due to the stochastic nature of the emulated spatiotemporal data, instantaneous matches with reference data are not achievable. For instance, an emulated wind speed field on January 1st, 2025, would significantly differ from the corresponding ground truth dataset, whether sourced from ERA5 or CMIP6. Ideally, an infinite ensemble of realizations could be produced by the emulator, enabling selection of instances closest to reference observations for training a debiasing model. However, this method is impractical. A more realistic alternative is the nudging approach [5–8], where the emulator is forced by the deviation from the reference data to produce a time series of fields that approximately maintain the emulator's statistical characteristics while closely aligning with the observed ground truth. Herein we interpret how to nudge the stochastic emulator. In the following, we detail how to implement nudging within the stochastic emulator framework.

Recall the formulation of the emulator (8,9),

$$\hat{a}_{s,i}(t) = \hat{\mu}_{s,i}\left(T_{s,g}\right) + \hat{\sigma}_{s,i}\left(T_{s,g}\right)\hat{\eta}_{s,i}(t) \tag{15}$$

$$\hat{q}_k(\boldsymbol{x}, t) = \bar{u}_k(\boldsymbol{x}, t) + \sigma_{g,k} \sum_{i=1}^{500} \hat{a}_i(t)\phi_k^{(i)}(\boldsymbol{x}). \tag{16}$$

The only stochastic component is the time series of daily fluctuations $\hat{\eta}_{s,i}$. All other parts are deterministic and constructed to align with the reference data. Therefore we focus on nudging $\hat{\eta}_{s,i}$, given the true fluctuations $\eta_{s,i}$. Hereafter we omit any subscripts to simplify the notation.

The nudged emulator, denoted as $\boldsymbol{\nu}$, is designed to follow the dynamics of the free-running emulator $\hat{\boldsymbol{\eta}}$, while driven by the deviation from the reference data,

$$\dot{\boldsymbol{\nu}} = \dot{\hat{\boldsymbol{\eta}}} - \frac{1}{\tau}\left(\boldsymbol{\nu} - \boldsymbol{\eta}\right). \tag{17}$$

The relaxation time scale τ is a constant that is independent from the season or the PCA mode. Equation (17) has a closed-form solution,

$$\boldsymbol{\nu}(t) = \boldsymbol{\nu}(0)e^{-t/\tau} + \int_0^t e^{-(t-s)/\tau}\left(\dot{\hat{\boldsymbol{\eta}}}(s) + \frac{1}{\tau}\boldsymbol{\eta}(s)\right) ds. \tag{18}$$

The time derivative term $\dot{\hat{\boldsymbol{\eta}}}$ is approximated using the first-order Euler scheme and computed from the free-running emulator data. Combining the nudged time series of daily fluctuations $\boldsymbol{\nu}$ with seasonal mean and variance, we obtain the complete nudged PCA time series and the spatiotemporal fields,

$$\hat{a}^{\nu}_{s,i}(t) = \hat{\mu}_{s,i}\left(T_{s,g}\right) + \hat{\sigma}_{s,i}\left(T_{s,g}\right)\hat{\nu}_{s,i}(t) \tag{19}$$

$$\hat{q}^{\nu}_{k}(\boldsymbol{x}, t) = \bar{u}_k(\boldsymbol{x}, t) + \sigma_{g,k}\sum_{i=1}^{500}\hat{a}^{\nu}_{i}(t)\phi^{(i)}_{k}(\boldsymbol{x}). \tag{20}$$

The interpretation of nudging and the selection of τ have been thoroughly discussed in [9]. Briefly speaking, the relaxation timescale τ serves to separate the time scales between slow and fast dynamics. The feedback term in equation (17) drives the slow dynamics of $\boldsymbol{\nu}$ towards the reference trajectory $\boldsymbol{\eta}$ in the state space, while allowing the fast dynamics of $\boldsymbol{\nu}$ to freely evolve. Thus, when pairs of the nudged and reference data are used for training a machine-learning model, we are essentially learning a map that corrects the fast features of the imperfect emulator and improve the performance on extreme events. In our case, the relaxation timescale is set as $\tau = 6$hrs, consistent with previous work [10]. Minor adjustments of τ, such as to 3 or 12 hours, do not significantly alter the results.

The feedback term in (17), although driving the nudged emulator towards the reference, introduces artificial dissipation not present in the free-running emulator. Such an effect leads to a distribution of $\boldsymbol{nu}$ and $\hat{\boldsymbol{q}}^{\nu}$ that is slightly different from the free-running emulator. In order for a neural network trained on the nudged dataset to generalize to unseen free-running emulator data, this discrepancy must be remedied. To this end, we rescale the nudged solution $\hat{\boldsymbol{q}}^{\nu}$ in each season so that its mean and variance match those of the free-running emulator $\hat{\boldsymbol{q}}$ at each grid point.

2 Machine Learned Debiasing

2.1 Conditional Score-based Diffusion model

Here we describe the training strategy and network architecture used in the ML correction step of our model. Our model relies on the framework introduced by

Barthel Sorensen et al. [10], which aims to learn a deterministic map from the nudged trajectory to the reference trajectory,

$$\boldsymbol{u} = \mathcal{F}\left(\hat{\boldsymbol{q}}^{\nu}\right). \tag{21}$$

In practice, such a mapping is not necessarily deterministic. There could exist multiple reference state $\boldsymbol{u}$ that are close to the same nudged state $\hat{\boldsymbol{q}}^{\nu}$. Therefore, we generalize this framework by learning a conditional probability distribution function,

$$p(\boldsymbol{u} \mid \hat{\boldsymbol{q}}^{\nu}). \tag{22}$$

If the mapping is actually deterministic, the conditional probability distribution will collapse to a Dirac delta function $\delta(\boldsymbol{u} - \mathcal{F}\left(\hat{\boldsymbol{q}}^{\nu}\right))$. Once the conditional PDF (22) is learned, we can provide the free-running emulation $\hat{\boldsymbol{q}}$ as the conditional information to generate debiased estimations of the state variables $\hat{\boldsymbol{u}}$,

$$\hat{\boldsymbol{u}}(\boldsymbol{x}, t) \sim \mathcal{G}_{\theta,2}\left[\hat{\boldsymbol{q}}(\boldsymbol{x}, t)\right] \tag{23}$$

Although learning and sampling high-dimensional PDFs were long considered intractable, these tasks have recently become practical thanks to advances in deep generative models. In this study, we adopt conditional score-based diffusion model [11, 12] that has been demonstrated effective for geophysical datasets [13]. Other frameworks, such as flow matching [14] and stochastic interpolant [15], could likewise address the debiasing problem considered here. The choice of the generative model is beyond the scope of this work and will be investigated in the future.

Our implementation of score-based diffusion model follows that of Bischoff and Deck [13]. To simplify the notation, we will use $\boldsymbol{q}$ to represent the nudged emulation $\hat{\boldsymbol{q}}^{\nu}$. The diffusion model consists of a forward diffusion process, which maps the data distribution to normal distribution, and a reverse denoising process that transforms Gaussian noise to a sample or image of the climate state. Specifically, given an initial condition $\mathbf{u}(\mathfrak{t} = 0) \sim p_{\text{data}}(\boldsymbol{u}|\boldsymbol{q})$ drawn from the training data, the forward diffusion process is defined by the stochastic differential equation (SDE),

$$\mathrm{d}\mathbf{u} = g(\mathfrak{t})\mathrm{d}\mathbf{W}, \tag{24}$$

where the diffusion coefficient $g(\mathfrak{t})$ is a non-negative prescribed function and $\mathbf{W}$ is a Wienner process. Note that the diffusion time $\mathfrak{t}$ is within $[0, 1]$ and should be distinguished from the physical time t. At any time $\mathfrak{t}$, the solution to the SDE (24) is a "noised" image $\mathbf{u}(\mathfrak{t})$, which follows a normal distribution conditioned on $\mathbf{u}(0)$,

$$\mathbf{u}(t) \sim \mathcal{N}\left(\mathbf{u}(0), \sigma^2(\mathfrak{t})\right) = p(\mathbf{u}(\mathfrak{t})|\mathbf{u}(0)), \tag{25}$$

where the variance $\sigma^2(\mathfrak{t})$ depends on $g(\mathfrak{t})$,

$$\sigma^2(\mathfrak{t}) = \int_0^{\mathfrak{t}} g^2(\mathfrak{t}')\mathrm{d}\mathfrak{t}'. \tag{26}$$

The marginal distribution of $\mathbf{u}(\mathrm{t})$ after integrating out $\mathbf{u}(0)$ is defined as $p_{\mathrm{t}}(\mathbf{u}(\mathrm{t})|\mathbf{q})$, which is generally non-Gaussian. The diffusion coefficient $g(\mathrm{t})$ is chosen such that at $\mathrm{t}=1$, the variance $\sigma^2(\mathrm{t}=1)$ has much larger magnitude than the original $\mathbf{u}(\mathrm{t}=0)$. Therefore, $\mathbf{u}(\mathrm{t}=0)$ has lost all memory of initial condition, and

$$p(\mathbf{u}(1)|\mathbf{u}(0)) = p(\mathbf{u}(1)) = \mathcal{N}(\mathbf{0}, \sigma^2(1)). \tag{27}$$

According to Anderson's theorem [16], the reverse of equation (24) is also a diffusion process, running backward in time and governed by the following SDE,

$$\mathrm{d}\mathbf{u} = -g^2(\mathrm{t})\mathbf{s}(\mathbf{u}, \mathbf{q}, \mathrm{t})\mathrm{d}\mathrm{t} + g(\mathrm{t})\mathrm{d}\overline{\mathbf{W}}, \tag{28}$$

where $\overline{\mathbf{W}}$ is a reverse-time Wienner process, and $\mathbf{s}(\mathbf{u}, \mathrm{t})$ is the conditional score,

$$\mathbf{s}(\mathbf{u}, \mathbf{q}, \mathrm{t}) = \nabla_{\mathbf{u}} \log p_{\mathrm{t}}(\mathbf{u}(\mathrm{t})|\mathbf{q}). \tag{29}$$

If we have access to the score for all t, we can derive the reverse diffusion process, simulate it from $\mathrm{t}=1$ to $\mathrm{t}=0$, and generate samples that follow the data distribution. To this end, we approximate the score by a neural network, $\mathbf{s}_\theta(\mathbf{u}, \mathbf{q}, \mathrm{t})$, which is obtained by minimizing the score-matching loss or Fisher's divergence,

$$\mathcal{L}_{SM}(\theta) := \frac{1}{2}\mathbb{E}_{\substack{t\sim U(0,1) \\ \mathbf{u}(\mathrm{t}),\mathbf{q}\sim p_{\mathrm{t}}(\mathbf{u}(\mathrm{t})|\mathbf{q})}} \left[\sigma^2(\mathrm{t}) \left\|\nabla_{\mathbf{u}} \log p_{\mathrm{t}}\left(\mathbf{u}|\mathbf{q}\right) - \mathbf{s}_\theta(\mathbf{u}, \mathbf{q}, \mathrm{t})\right\|_2^2\right], \tag{30}$$

where $U(0,1)$ stands for a uniform distribution from 0 to 1. However, the loss function (30) cannot be directly optimized, since the true conditional score $\nabla_{\mathbf{u}} \log p_{\mathrm{t}}\left(\mathbf{u}|\mathbf{q}\right)$ is unknown. Taking advantage of the Gaussian property of the forward diffusion process (25,27), [11] showed that $\mathcal{L}_{SM}$ is equal to the following loss up to an additive term,

$$\mathcal{L}(\theta) := \frac{1}{2}\mathbb{E}_{\substack{t\sim U(0,1) \\ \mathbf{u}(0),\mathbf{q}\sim p(\mathbf{u}_0|\mathbf{q}) \\ \mathbf{u}(\mathrm{t})\sim p(\mathbf{u}(\mathrm{t})|\mathbf{u}(0))}} \left[\sigma^2(\mathrm{t}) \left\|\nabla_{\mathbf{u}} \log p_{\mathrm{t}}\left(\mathbf{u}(\mathrm{t})|\mathbf{u}(0)\right) - \mathbf{s}_\theta(\mathbf{u}, \mathbf{q}, \mathrm{t})\right\|_2^2\right]. \tag{31}$$

This expression only involves $\nabla_{\mathbf{u}} \log p_{\mathrm{t}}\left(\mathbf{u}(\mathrm{t})|\mathbf{u}(0)\right)$ which can be computed analytically from the forward diffusion process.

Once the conditional score is learned through training, it can be substituted into the backward SDE (equation 28) to generate a debiased sample. The backward SDE is simulated using Euler-Maruyama scheme. The complete procedures of training and sampling are summarized in Algorithm 2.

2.2 Network Architecture and Training Parameters

Before feeding the emulation and reference data into the diffusion model, we remove the true climatological mean and scale each variable (U, V, T, Q) by twice its own globally-averaged standard deviation. In other words, we focus on correcting the fluctuation fields provided by the emulator, and each variable is scaled to the same order of

Algorithm 2 Conditional score-based diffusion model.

Training

Input: Reference data and nudged emulation $\{\mathbf{u}_i, \mathbf{q}_i\} \sim p(\mathbf{u}|\mathbf{q})$

repeat

$\mathbf{u}(0), \mathbf{q} \sim p(\mathbf{u}|\mathbf{q})$

$\mathrm{t} \sim U(0, 1)$

$\mathbf{u}(\mathrm{t}) \sim p(\mathbf{u}(\mathrm{t})|\mathbf{u}(0))$

Take gradient descent step on $\nabla_\theta \left[\sigma^2(\mathrm{t}) \left\|\nabla_{\mathbf{u}} \log p_{\mathrm{t}} \left(\mathbf{u}(\mathrm{t})|\mathbf{u}(0)\right) - \mathbf{s}_\theta(\mathbf{u}, \mathbf{q}, \mathrm{t})\right\|_2^2\right]$

until converged

Output: Trained neural network $\mathbf{s}_\theta(\mathbf{u}, \mathbf{q}, \mathrm{t})$.

End

Sampling

Input: Snapshot of emulated state $\mathbf{q}$

$\mathbf{u}(1) \sim \mathcal{N}(\mathbf{0}, \sigma^2(1))$

for $\mathrm{t} = 1$ **to** 0 **do**

Evaluate $\mathbf{s}_\theta(\mathbf{u}(\mathrm{t}), \mathbf{q}, \mathrm{t})$

$\mathbf{u}(\mathrm{t} - \Delta\mathrm{t}) = \mathbf{u}(\mathrm{t}) + g^2(\mathrm{t})\mathbf{s}_\theta(\mathbf{u}, \mathbf{q}, \mathrm{t})\Delta\mathrm{t} - g(\mathrm{t}) \left(\overline{\mathbf{W}}(\mathrm{t}) - \overline{\mathbf{W}}(\mathrm{t} - \Delta\mathrm{t})\right)$

$\mathrm{t} \leftarrow \mathrm{t} - \Delta\mathrm{t}$

end for

Output: Debiased snapshot $\hat{\mathbf{u}} = \mathbf{u}(\mathrm{t} = 0)$

End

magnitude. Regarding the diffusion coefficient $g(\mathrm{t})$ in equation (24), we adopt the "variance-exploding" schedule,

$$g(\mathrm{t}) = \sigma_{\min} \left(\frac{\sigma_{\max}}{\sigma_{\min}}\right)^{\mathrm{t}} \sqrt{2 \log \left(\frac{\sigma_{\max}}{\sigma_{\min}}\right)}, \tag{32}$$

where $\sigma_{\min} = 0.01$ and $\sigma_{\max}$ is chosen as the maximum 2-norm distance between any two snapshots of $\mathbf{u}$.

The neural network architecture we adopted is a U-Net [13, 17], denoted as

$$\mathbf{s}_\theta(\mathbf{u}, \mathbf{q}, \mathrm{t}) = \mathcal{U}(\mathbf{X}, \mathrm{t}; \boldsymbol{\theta}). \tag{33}$$

The first input X is a tensor of size (N_1, N_2, C_{in}), where (N_1, N_2) are longitude and latitude dimensions, and C_{in} is the number of channels. In our case, $C_{in} = 8$, including (U, V, T, Q) from the nudged emulation and reference data, respectively. The output of our U-Net is another tensor of size (N_1, N_2, C_{out}). The number of output channels is $C_{out} = 4$. The U-Net architecture consists of

1. A lifting layer which increases the number of channels form C_{in} to 32;
2. Three downsampling convolutional layers, each of which reduces the spatial dimension and increase the number of channels by a factor of 2;
3. Eight residual blocks [18] to promote continuity in the latent space;

4. Three nearest neighbor up sampling layer and convolution layers which mirror the downsampling operations;
5. A Final projection layer that decreases the number of channels to C_{out}.

3 Data Post-processing and Evaluation Metrics

This section provides detailed definitions and calculation methods for all statistics and metrics presented in the main figures. Given that the climatological mean $\bar{\boldsymbol{u}}(\boldsymbol{x}, t)$ (equation 1) is assumed known, our analysis focuses on evaluating the statistics of the fluctuation fields, $\boldsymbol{u}-\bar{\boldsymbol{u}}$. This approach enables a clearer comparison between reference statistics and GEN^2 prediction. In the following subsections, the fluctuations from the climatological mean, $\boldsymbol{u} - \bar{\boldsymbol{u}}$, will be simply written as $\boldsymbol{u}$ for notational convenience.

3.1 Single-point and two-point statistics

Without loss of generality, we use the zonal wind speed $U(\boldsymbol{x}, t)$ as an example. To evaluate the statistics of U at location $\boldsymbol{x}$, we perform a time average (e.g. from 1979 to 2018 for ERA5 dataset). If we have N_t time steps available, the mean and standard deviation are computed as,

$$\mu_U(\boldsymbol{x}, t_j) = \frac{1}{N_t}\sum_{j=1}^{N_t} U(\boldsymbol{x}, t_j) \quad \text{and} \quad \sigma_U(\boldsymbol{x}) = \sqrt{\frac{1}{N_t - 1}\sum_{j=1}^{N_t} \left(U(\boldsymbol{x}, t_j) - \mu_U(\boldsymbol{x}, t_j)\right)^2}. \tag{34}$$

To obtain the unbiased skewness, we first compute

$$s_U(\boldsymbol{x}) = \frac{\frac{1}{N_t}\sum_{j=1}^{N_t} \left(U(\boldsymbol{x}, t_j) - \mu_U(\boldsymbol{x}, t_j)\right)^3}{\left(\frac{1}{N_t}\sum_{j=1}^{N_t} \left(U(\boldsymbol{x}, t_j) - \mu_U(\boldsymbol{x}, t_j)\right)^2\right)^{3/2}}, \tag{35}$$

which is then substituted into

$$s_U^0(\boldsymbol{x}) = \frac{\sqrt{N_t(N_t-1)}}{N_t - 2} s_U(\boldsymbol{x}). \tag{36}$$

The calculation of unbiased kurtosis also consists of two steps:

$$k_U(\boldsymbol{x}) = \frac{\frac{1}{N_t}\sum_{j=1}^{N_t} \left(U(\boldsymbol{x}, t_j) - \mu_U(\boldsymbol{x}, t_j)\right)^4}{\left(\frac{1}{N_t}\sum_{j=1}^{N_t} \left(U(\boldsymbol{x}, t_j) - \mu_U(\boldsymbol{x}, t_j)\right)^2\right)^2}, \tag{37}$$

$$k_U^0(\boldsymbol{x}) = \frac{N_t - 1}{(N_t - 2)(N_t - 3)}\left((N_t + 1)k_U(\boldsymbol{x}) - 3(N_t - 1) + 3\right). \tag{38}$$

At the same location $\boldsymbol{x}$, the correlation coefficient between two variables U and V are defined as,

$$\rho(U,V) = \frac{\mathrm{cov}\left(U(\boldsymbol{x},t), V(\boldsymbol{x},t)\right)}{\mathrm{cov}\left(U(\boldsymbol{x},t), U(\boldsymbol{x},t)\right)\mathrm{cov}\left(V(\boldsymbol{x},t), V(\boldsymbol{x},t)\right)}, \tag{39}$$

where $\mathrm{cov}\left(U(\boldsymbol{x},t), V(\boldsymbol{x},t)\right)$ is the time-averaged covariance between U and V.

The two-point correlation coefficient of U is defined as,

$$\rho(U(\boldsymbol{x}_0), U(\boldsymbol{x})) = \frac{\mathrm{cov}\left(U(\boldsymbol{x}_0,t), U(\boldsymbol{x},t)\right)}{\mathrm{cov}\left(U(\boldsymbol{x}_0,t), U(\boldsymbol{x}_0,t)\right)\mathrm{cov}\left(U(\boldsymbol{x},t), U(\boldsymbol{x},t)\right)}. \tag{40}$$

The anchor point $\boldsymbol{x}_0$ is selected as major cities (e.g. Boston, Hong Kong) in figure 3, 6 and in section **??**.

The global root-mean-square error (RMSE) of an arbitrary statistic $\mathcal{Q}$ (e.g. in figure 5 and table 3,4) is defined as,

$$\mathrm{RMSE}(\mathcal{Q}) = \left[\frac{1}{S}\int_S \left(\hat{\mathcal{Q}}(\theta,\varphi,t) - \mathcal{Q}(\theta,\varphi,t)\right)^2 \cos\theta d\theta d\varphi\right]^{1/2}, \tag{41}$$

where $\hat{\mathcal{Q}}$ is the GEN^2 prediction and $\mathcal{Q}$ is the reference statistics.

3.2 Wheeler-Kiladis spectrum

The Wheeler-Kiladis spectrum is computed following the procedure described in Wheeler and Kiladis [19], Kiladis et al. [20]. Given data as a function of longitude, latitude, and time $[(\theta,\phi,t)$ the latitude range is first truncated to $\phi \in [-15°, 15°]$. Then for each latitude ϕ_j, and time t_j the Fourier spectrum is computed in the azimuthal direction θ giving rise to a azimuthal wavenumber m. Then for each ϕ_j and m_j the time series of data is split into a series of overlapping segments. Following [19, 20] we set the length of each segment to 96 days and the overlap to 65 days. Each segment is then detrended using a linear fit and Fourier transformed in time - giving a temporal frequency f. After averaging over all latitudes and all temporal segments, the raw Wheeler-Kiladis spectra are shown in figure 2.

Due to the strong "redness" of the spectra, detailed features corresponding to the equatorial waves are obscured. To better identify the ridges of the spectra, we first apply a 1-2-1 filter ten times to obtain a much smoother "background" spectra. Then the raw spectra in figure 2 are divided by the background [19]. The results, as shown in figure 2(c) of the main manuscript, more clearly show the spectral peaks that correspond to different types of equatorial waves. Note that the spectra in our results should not be directly compared against the plots in [19]. The reason is that their analysis was based on the long-wave radiation data, which are proxy for cloudiness, whereas our analysis focuses on near-surface zonal wind speed.

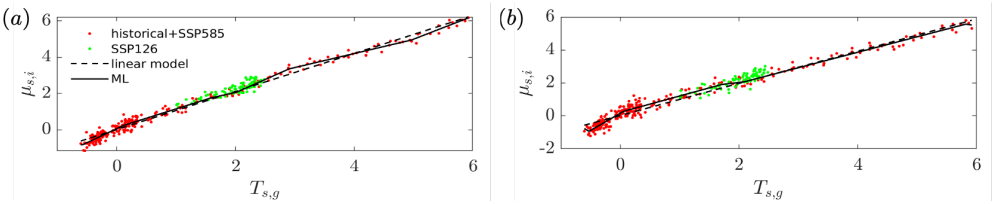
Supplementary Notes

To illustrate the debiasing capabilities of the ML correction step, we show in figure 3 the bias in the 97.5% quantile, predicted with or without ML correction. As explained at the beginning of Appendix 3, the statistics are evaluated for the fluctuation fields. Before applying ML correction (middle column), the error of the conditional Gaussian emulator is already moderately accurate. For example, the error of T at most locations is within $3K$, and the highest error is within $3K$. The ML model (right column) consistently reduces the error of all the state variables at almost all the locations. A more quantitative comparison is provided in table 1,2. The bias in standard deviation, quantile, skewness, and kurtosis are all significantly reduced by ML correction.

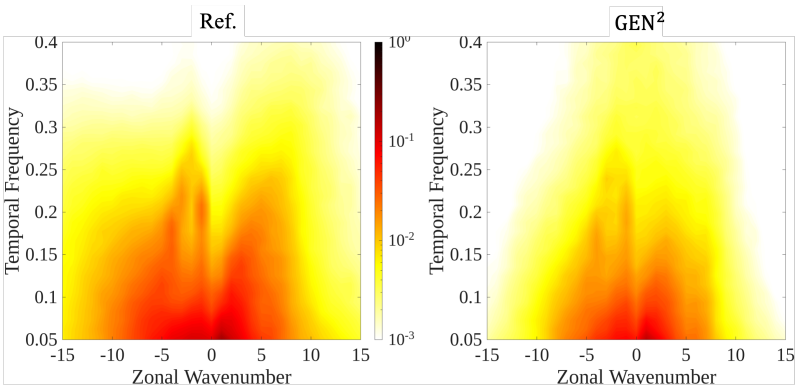
The bias reduction in two-point correlations are shown in figure 4. We select Lagos and Tehran for visualization, because the bias of the conditional Gaussian emulator is more pronounced at these two locations. Top panels in figure 4 are the bias of the conditional Gaussian emulator, without ML correction. Bottom panels are the bias of GEN^2 prediction.

Table 3 and 4 summarize the error of two-point correlations at more locations, selected from different regions and climate over the world. At most locations, the conditional Gaussian emulator already achieves an accurate prediction, with RMSE lower than 0.03. After applying ML correction, the errors are significantly reduced at all the locations considered. These results demonstrate the robustness of the ML correction.

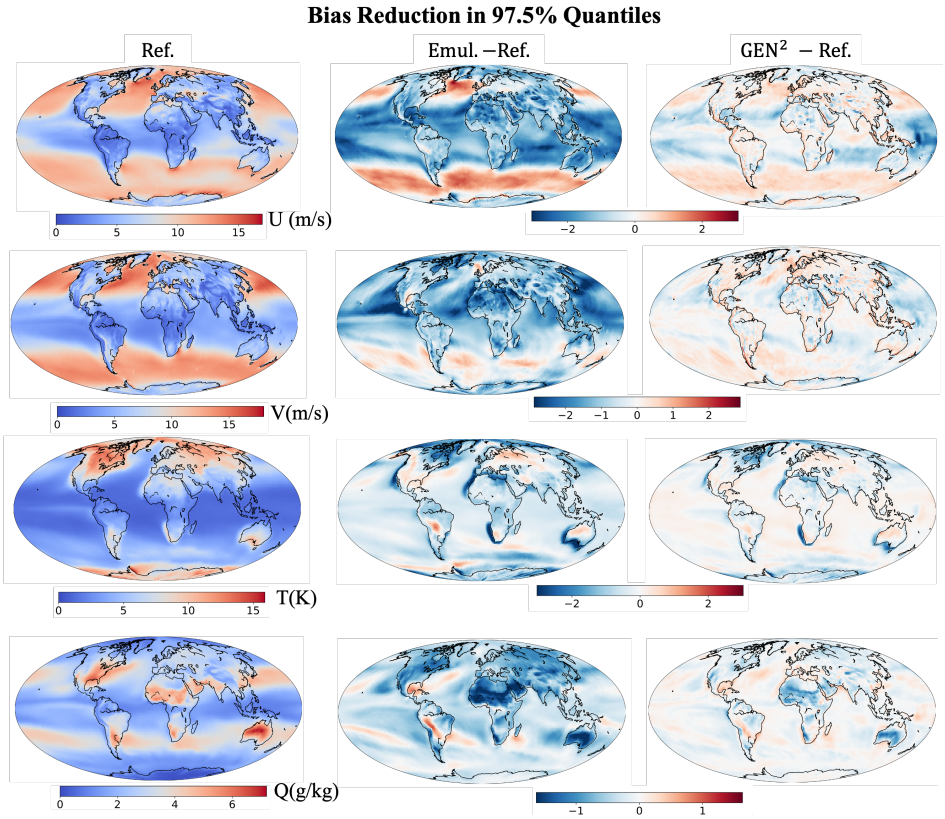
Supplementary Figures



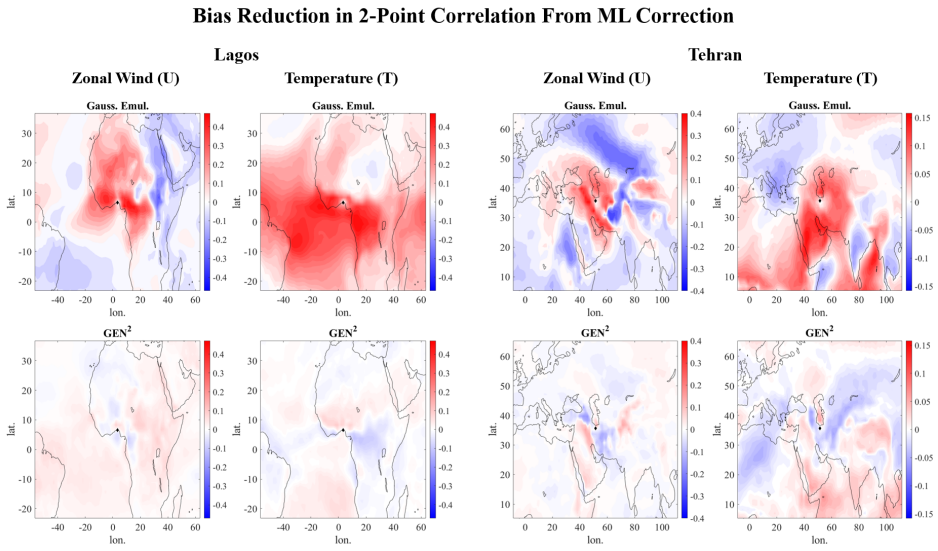
Supplementary Fig 1 Linearity of PCA coefficients as functions of the global mean temperature. Jun-Aug mean of (a) the first and (b) second PCA coefficients in each year of MPI dataset, from 1950 to 2100, plotted versus the global mean temperature. Red dots: true seasonal mean obtained from the historical and SSP5-8.5 scenario. Green dots: SSP1-2.6 scenario. Black dashed line: linear regression; Solid line: machine-learned function.



Supplementary Fig 2 Raw Wheeler-Kiladis spectrum of zonal wind. Spectra are computed using 1979-2018 ERA5 reference data and GEN^2 prediction. These spectra are normalized by the "background power" to obtain the spectra in figure 2(c) of the main manuscript.



Supplementary Fig 3 Machine-learning correction of the bias in quantiles. Left column: 97.5% quantile of zonal wind, meridional wind, temperature, computed from reference data. Middle column: bias of conditional Gaussian emulation. Right column: bias of "GEN^2" approach.



Supplementary Fig 4 Machine-learning correction of the bias in two-point correlations. Bias in two-point correlations of zonal wind and temperature, centered at Lagos and Tehran. Contour plots represent bias relative to reference ERA5 data. Panels labeled "Gauss. Emul." correspond to predictions of the conditional Gaussian emulator *only* (no ML correction) and "GEN^2" represents the full model prediction (with ML correction) .

Supplementary Tables

Statistics	RMSE of U stats			RMSE of V stats		
	Em.	GEN^2	Change	Em.	GEN^2	Change
Std	0.43	0.15	-65%	0.47	0.13	-73%
97.5% quantile	1.14	0.46	-60%	0.99	0.33	-66%
Skewness	0.41	0.19	-53%	0.27	0.14	-48%
Kurtosis	0.84	0.57	-33%	0.68	0.46	-33%

Supplementary Table 1 RMSE of single-point statistics of U, V. RMSE is defined as equation (41). Columns labeled as "Em." are the RMSE of the prediction of conditional Gaussian emulator, and "GEN^2" is the full-model prediction. "Change" columns are the relative error change from conditional Gassuain emulator to GEN^2, more precisely, (RMSE(GEN^2) − RMSE(Em.))/RMSE(Em.).

Statistics	RMSE of T stats			RMSE of Q stats		
	Em.	GEN^2	Change	Em.	GEN^2	Change
Std	0.23	0.10	-56%	0.19	0.05	-72%
97.5% quantile	0.67	0.35	-48%	0.46	0.17	-63%
Skewness	0.34	0.20	-42%	0.50	0.27	-47%
Kurtosis	0.89	0.68	-24%	1.78	1.38	-23%

Supplementary Table 2 Same as table 1, but for temperature T and Q.

Anchor point $\boldsymbol{x}_0$			RMSE of $\rho(U(\boldsymbol{x}_0), U(\boldsymbol{x}))$			RMSE of $\rho(V(\boldsymbol{x}_0), V(\boldsymbol{x}))$		
City	Lon(E)	Lat(N)	Em.	GEN^2	Change	Em.	GEN^2	Change
Boston	-71.1	42.4	0.016	0.009	-43%	0.018	0.008	-53%
Los Angeles	-118.2	34.1	0.028	0.011	-62%	0.026	0.009	-67%
Chicago	-87.6	41.9	0.020	0.008	-59%	0.017	0.008	-54%
Houston	-95.4	29.8	0.020	0.009	-57%	0.017	0.009	-48%
Kansas City	-94.6	39.1	0.022	0.008	-64%	0.017	0.008	-51%
London	-0.1	51.5	0.018	0.009	-50%	0.019	0.008	-60%
Anchorage	-149.9	61.2	0.019	0.012	-36%	0.021	0.009	-58%
Paris	2.4	48.9	0.019	0.009	-53%	0.019	0.008	-57%
Athens	23.7	38.0	0.024	0.010	-57%	0.022	0.009	-60%
Moscow	37.6	55.8	0.024	0.010	-58%	0.022	0.008	-66%
Stockholm	18.1	59.3	0.022	0.010	-57%	0.020	0.008	-61%
Tokyo	139.7	35.7	0.017	0.008	-51%	0.017	0.009	-48%
Hong Kong	114.2	22.3	0.026	0.009	-64%	0.025	0.009	-65%
New Delhi	77.1	28.6	0.028	0.010	-64%	0.028	0.010	-65%
Tehran	51.4	35.7	0.030	0.011	-63%	0.034	0.010	-70%
Astana	71.5	51.2	0.022	0.009	-60%	0.022	0.009	-60%
Cairo	31.2	30.0	0.029	0.009	-70%	0.027	0.008	-69%
Cape Town	18.4	-33.9	0.018	0.009	-50%	0.022	0.009	-60%
Lagos	3.4	6.5	0.043	0.015	-65%	0.040	0.009	-78%
Kisangani	25.2	0.1	0.047	0.015	-69%	0.036	0.009	-76%
Mombasa	39.7	-4.0	0.041	0.018	-55%	0.038	0.010	-74%
Sydney	151.2	-33.9	0.020	0.009	-56%	0.020	0.008	-58%
Brasília	-47.9	-15.8	0.029	0.012	-58%	0.044	0.010	-78%
Bogota	-74.1	4.7	0.054	0.023	-57%	0.034	0.017	-49%
Buenos Aires	-58.4	-34.6	0.020	0.009	-55%	0.019	0.009	-54%

Supplementary Table 3 RMSE of two-point correlation of U, V. Columns labeled as "Em." are the RMSE of the prediction of conditional Gaussian emulator, and "GEN^2" is the full-model prediction. "Change" columns are the relative error change from conditional Gassuain emulator to GEN^2, more precisely, (RMSE(GEN^2) − RMSE(Em.))/RMSE(Em.).

Anchor point $\boldsymbol{x}_0$			RMSE of $\rho(T(\boldsymbol{x}_0), T(\boldsymbol{x}))$			RMSE of $\rho(Q(\boldsymbol{x}_0), Q(\boldsymbol{x}))$		
City	Lon(E)	Lat(N)	Em.	GEN^2	Change	Em.	GEN^2	Change
Boston	-71.1	42.4	0.017	0.012	-30%	0.016	0.009	-43%
Los Angeles	-118.2	34.1	0.021	0.011	-48%	0.022	0.010	-54%
Chicago	-87.6	41.9	0.014	0.010	-29%	0.017	0.009	-46%
Houston	-95.4	29.8	0.019	0.014	-28%	0.015	0.008	-47%
Kansas City	-94.6	39.1	0.014	0.010	-30%	0.016	0.009	-45%
London	-0.1	51.5	0.023	0.010	-55%	0.022	0.011	-49%
Anchorage	-149.9	61.2	0.026	0.013	-48%	0.026	0.014	-48%
Paris	2.4	48.9	0.023	0.010	-54%	0.023	0.012	-47%
Athens	23.7	38.0	0.018	0.012	-35%	0.023	0.009	-60%
Moscow	37.6	55.8	0.017	0.011	-36%	0.021	0.010	-51%
Stockholm	18.1	59.3	0.021	0.010	-53%	0.021	0.011	-49%
Tokyo	139.7	35.7	0.019	0.011	-40%	0.015	0.009	-39%
Hong Kong	114.2	22.3	0.023	0.014	-41%	0.020	0.011	-46%
New Delhi	77.1	28.6	0.031	0.018	-43%	0.022	0.011	-47%
Tehran	51.4	35.7	0.030	0.018	-42%	0.033	0.010	-68%
Astana	71.5	51.2	0.017	0.011	-32%	0.026	0.011	-59%
Cairo	31.2	30.0	0.031	0.014	-55%	0.027	0.008	-69%
Cape Town	18.4	-33.9	0.022	0.012	-45%	0.022	0.010	-55%
Lagos	3.4	6.5	0.094	0.018	-81%	0.034	0.013	-63%
Kisangani	25.2	0.1	0.072	0.015	-79%	0.046	0.011	-77%
Mombasa	39.7	-4.0	0.100	0.023	-77%	0.059	0.015	-75%
Sydney	151.2	-33.9	0.025	0.012	-52%	0.021	0.010	-53%
Brasília	-47.9	-15.8	0.042	0.013	-68%	0.030	0.012	-62%
Bogota	-74.1	4.7	0.092	0.026	-71%	0.054	0.025	-53%
Buenos Aires	-58.4	-34.6	0.019	0.012	-38%	0.017	0.010	-42%

Supplementary Table 4 Same as table 3, but for temperature T and humidity Q.

References

[1] Tebaldi, C., Arblaster, J.M.: Pattern scaling: Its strengths and limitations, and an update on the latest model simulations. Climatic Change **122**(3), 459–471 (2014) https://doi.org/10.1007/s10584-013-1032-9

[2] Osborn, T.J., Wallace, C.J., Lowe, J.A., Bernie, D.: Performance of Pattern-Scaled Climate Projections under High-End Warming. Part I: Surface Air Temperature over Land (2018) https://doi.org/10.1175/JCLI-D-17-0780.1

[3] Box, G.E.P., Jenkins, G.M., Reinsel, G.C.: Time Series Analysis: Forecasting and Control, Third edition edn. Prentice Hall, Englewood Cliffs, N.J. (1994). http://www.gbv.de/dms/bowker/toc/9780130607744.pdf

[4] Wang, M., Souza, A., Ferrari, R., Sapsis, T.: Spatially-resolved emulation of climate extremes via machine learning stochastic models (2023)

[5] Storch, H.v., Langenberg, H., Feser, F.: A Spectral Nudging Technique for Dynamical Downscaling Purposes. Monthly Weather Review **128**(10), 3664–3673 (2000) https://doi.org/10.1175/1520-0493(2000)128⟨3664:ASNTFD⟩2.0.CO;2

[6] Miguez-Macho, G., Stenchikov, G.L., Robock, A.: Regional Climate Simulations over North America: Interaction of Local Processes with Improved Large-Scale Flow. Journal of Climate **18**(8), 1227–1246 (2005) https://doi.org/10.1175/JCLI3369.1

[7] Sun, J., Zhang, K., Wan, H., Ma, P.-L., Tang, Q., ZHANG, S.: Impact of Nudging Strategy on the Climate Representativeness and Hindcast Skill of Constrained EAMv1 Simulations. Journal of Advances in Modeling Earth Systems **11** (2019) https://doi.org/10.1029/2019MS001831

[8] Huang, Z., Zhong, L., Ma, Y., Fu, Y.: Development and evaluation of spectral nudging strategy for the simulation of summer precipitation over the Tibetan Plateau using WRF (v4.0). Geoscientific Model Development **14**(5), 2827–2841 (2021) https://doi.org/10.5194/gmd-14-2827-2021

[9] Barthel Sorensen, B., Zepeda-Núñez, L., Lopez-Gomez, I., Wan, Z.Y., Carver, R., Sha, F., Sapsis, T.: A probabilistic framework for learning non-intrusive corrections to long-time climate simulations from short-time training data. arXiv (2024). https://doi.org/10.48550/arXiv.2408.02688 . http://arxiv.org/abs/2408.02688

[10] Barthel Sorensen, B., Charalampopoulos, A., Zhang, S., Harrop, B.E., Leung, L.R., Sapsis, T.P.: A Non-Intrusive Machine Learning Framework for Debiasing Long-Time Coarse Resolution Climate Simulations and Quantifying Rare Events Statistics. Journal of Advances in Modeling Earth Systems **16**(3), 2023–004122 (2024) https://doi.org/10.1029/2023MS004122

[11] Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-Based Generative Modeling through Stochastic Differential Equations. arXiv (2021). https://doi.org/10.48550/arXiv.2011.13456 . http://arxiv.org/abs/2011.13456

[12] Batzolis, G., Stanczuk, J., Schönlieb, C.-B., Etmann, C.: Conditional image generation with score-based diffusion models. arXiv preprint arXiv:2111.13606 (2021)

[13] Bischoff, T., Deck, K.: Unpaired Downscaling of Fluid Flows with Diffusion Bridges. arXiv (2023). https://doi.org/10.48550/arXiv.2305.01822 . http://arxiv.org/abs/2305.01822

[14] Lipman, Y., Chen, R.T., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. arXiv preprint arXiv:2210.02747 (2022)

[15] Albergo, M.S., Boffi, N.M., Vanden-Eijnden, E.: Stochastic interpolants: A unifying framework for flows and diffusions. arXiv preprint arXiv:2303.08797 (2023)

[16] Anderson, B.D.: Reverse-time diffusion equation models. Stochastic Processes and their Applications **12**(3), 313–326 (1982)

[17] Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation. arXiv (2015). https://doi.org/10.48550/arXiv.1505.04597 . http://arxiv.org/abs/1505.04597

[18] He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. arXiv (2015). https://doi.org/10.48550/arXiv.1512.03385 . http://arxiv.org/abs/1512.03385

[19] Wheeler, M., Kiladis, G.N.: Convectively Coupled Equatorial Waves: Analysis of Clouds and Temperature in the Wavenumber–Frequency Domain (1999)

[20] Kiladis, G.N., Wheeler, M.C., Haertel, P.T., Straub, K.H., Roundy, P.E.: Convectively coupled equatorial waves. Reviews of Geophysics **47**(2) (2009) https://doi.org/10.1029/2008RG000266