DiffESM: Conditional Emulation of Temperature and Precipitation in Earth System Models with 3D Diffusion Models

Seth Bassetti\textsuperscript{1}, Brian Hutchinson\textsuperscript{2}, Claudia Tebaldi\textsuperscript{3}, and Ben Kravitz\textsuperscript{4}

\textsuperscript{1}Utah State University
\textsuperscript{2}Western Washington University
\textsuperscript{3}Pacific Northwest National Laboratory
\textsuperscript{4}Indiana University

January 24, 2024

Abstract

Earth System Models (ESMs) are essential tools for understanding the interaction of the human and Earth systems. One key application of these models is studying extreme weather events, such as heat waves or high intensity precipitation events, which have significant socioeconomic consequences. However, the computational demands of running a sufficient number of simulations to robustly characterize expected changes in these hazards, and therefore provide a strong basis to analyze the ensuing risks, are often prohibitive. In this paper we demonstrate that diffusion models – a class of generative deep learning models – can effectively emulate the spatio-temporal trends of ESM daily output. Trained on a handful of runs, reflecting a wide range of radiative forcings, our DiffESM model takes monthly mean precipitation or temperature as input and is capable of producing daily values of temperature and precipitation that have statistical characteristics close to the ESM output. This approach requires only a small fraction of the computational resources that would be needed to run a large ensemble under any scenario of interest. We evaluate model behavior over a range of scenarios, time horizons and two ESMs, using a number of extreme metrics, including ones that have been long established in the climate modeling and analysis community. Our results show that the samples produced by DiffESM closely matches the spatio-temporal behavior of the ESM output it emulates in terms of the frequency and spatial characteristics of phenomena such as heat waves, dry spells, or rainfall intensity.
DiffESM: Conditional Emulation of Temperature and Precipitation in Earth System Models with 3D Diffusion Models

Seth Bassetti\textsuperscript{1*}, Brian Hutchinson\textsuperscript{2,3}, Claudia Tebaldi\textsuperscript{4}, Ben Kravitz\textsuperscript{5,6}

\textsuperscript{1}Computer Science Department, Utah State University, Logan, UT
\textsuperscript{2}Computer Science Department, Western Washington University, Bellingham, WA
\textsuperscript{3}Foundational Data Science, Pacific Northwest National Laboratory, Seattle, WA
\textsuperscript{4}Joint Global Change Research Institute, Pacific Northwest National Laboratory, College Park, MD
\textsuperscript{5}Earth and Atmospheric Sciences Department, Indiana University, Bloomington, IN
\textsuperscript{6}Atmospheric Sciences and Global Change Division, Pacific Northwest National Laboratory, Richland, WA

Key Points:

\begin{itemize}
\item Earth system models (ESMs) are key devices for understanding how human actions will affect the future global climate.
\item Generating enough ESM samples for analyses like the characterization of extreme events is computationally demanding (and often intractable).
\item We present DiffESM as a data-driven emulator of ESMs that closely matches the spatial and temporal distributions of their output variables.
\end{itemize}

*Work performed while at Western Washington University.

Corresponding author: Brian Hutchinson, Brian.Hutchinson@wwu.edu
Abstract
Earth System Models (ESMs) are essential tools for understanding the interaction of the human and Earth systems. One key application of these models is studying extreme weather events, such as heat waves or high intensity precipitation events, which have significant socioeconomic consequences. However, the computational demands of running a sufficient number of simulations to robustly characterize expected changes in these hazards, and therefore provide a strong basis to analyze the ensuing risks, are often prohibitive. In this paper we demonstrate that diffusion models – a class of generative deep learning models – can effectively emulate the spatio-temporal trends of ESM daily output. Trained on a handful of runs, reflecting a wide range of radiative forcings, our DiffESM model takes monthly mean precipitation or temperature as input and is capable of producing daily values of temperature and precipitation that have statistical characteristics close to the ESM output. This approach requires only a small fraction of the computational resources that would be needed to run a large ensemble under any scenario of interest. We evaluate model behavior over a range of scenarios, time horizons and two ESMs, using a number of extreme metrics, including ones that have been long established in the climate modeling and analysis community. Our results show that the samples produced by DiffESM closely matches the spatio-temporal behavior of the ESM output it emulates in terms of the frequency and spatial characteristics of phenomena such as heat waves, dry spells, or rainfall intensity.

Plain Language Summary
Ideally, to study how damaging phenomena like heatwaves, droughts and downpours will change in the future under global warming, we would want a large number of climate model runs producing many realizations of climate futures that we can analyze and from which the new characteristics of climate extremes can be quantified. Running climate models is however extremely expensive and we show how training a machine learning model over the available climate model output can fill in the gaps, thus enriching our data and potentially enabling more precise estimates of how these extremes will change in the future.

1 Introduction
Extreme weather events, such as heat waves, droughts, and floods have become more frequent and intense in recent years (on Climate Change (IPCC), 2023). These events have significant impacts on human societies and ecosystems, highlighting the urgent need to understand how they may change in the future under different emission scenarios. One important tool for investigating future climate change and its impacts on extreme weather events is the use of Earth System Models (ESMs) run under plausible future scenarios of greenhouse gas emissions.

ESMs are complex computer models that simulate the interactions between Earth’s atmosphere, oceans, land surface, biosphere, cryosphere and more. They are used to simulate a wide range of climate variables under different emissions scenarios. However, the computational demands of ESMs limit the number of simulations that can be performed, especially when climate modeling centers need to allocate experiments to meet demands from a range of scientific and practical uses. This is especially problematic when investigating rare extreme weather events, as it is necessary to aggregate data over numerous runs to obtain reliable statistics. To address this issue, emulators can be used to generate thousands of realizations of global climate data in the scale of minutes or hours, rather than weeks or months (Kasim et al., 2021). Emulators learn the statistical characteristics of ESM output from existing data, and can then generate new data under the same scenario used for training but also, importantly, under different emissions scenarios, only utilizing some type of simplified, coarser-scale “conditioning” from the target.
scenario. Machine learning approaches, especially generative deep learning methods, are well-suited to building such emulators, as they are capable of approximating complicated, high-dimensional distributions such as high resolution natural images (Rombach et al., 2022; Saharia et al., 2022; Ho et al., 2020), thus having the potential to substitute costly initial condition large ensembles.

In this paper, we present a denoising diffusion probabilistic model which learns to closely model the spatio-temporal behavior of an ESM, producing month-long samples of either daily mean temperature or precipitation. Diffusion models have shown great success in the realm of generative modeling, extending to the domains of image, audio, video, and recently even climate. Our emulator, DiffESM, can be steered to generate samples under existing or novel climate scenarios by conditioning generation on a monthly mean map of the climate variable. When not provided by an existing scenario simulation, such monthly averages can be produced by existing emulators such as fldgen (Link et al., 2019), MESMER (Beusch et al., 2020; Nath et al., 2022; Quilcaille et al., 2022) or STITCHES (Tebaldi et al., 2022). Once trained, the emulator offers a dramatic improvement over traditional ESMs in terms of speed, allowing for rapid investigation of the effect of climate scenario on the distribution of extreme weather events. This makes it a valuable tool for climate researchers and policy-makers who need to make informed decisions about the future of our planet. By enabling the generation of large ensembles of climate simulations, DiffESM can provide valuable insights into the range and magnitude of potential climate impacts, and help inform adaptation and mitigation strategies.

2 Related Work

In recent years, machine learning has gained increased attention as a tool to support research in the earth sciences (Reichstein et al., 2019). One promising area of application for data-driven algorithms is in forecasting. Traditional weather forecasting relies on physically-constrained models, but these models can be highly sensitive to initial conditions. In contrast, neural networks have shown greater robustness to uncertainties and variations in initial conditions (Wang et al., 2019; Scher & Messori, 2018; Rasel et al., 2018; Narvekar & Fargose, 2015). As a result, machine learning algorithms have been proposed as a complement or alternative to traditional methods for weather forecasting. Additionally, generative models, which can be used to create spatially coherent data have also been used as forecasting tools (Gagne II et al., 2020; Bihlo, 2021). Outside of forecasting, machine learning-based solutions have interacted directly with ESMs and Regional Climate Models (RCMs). Specifically, the spatial resolution of ESM and RCM outputs can be increased for more local-scale predictions as shown by (Hobeichi et al., 2023; Jebeile et al., 2021; Babaousmail et al., 2021). In addition to increasing spatial resolutions in certain regions, approaches have used neural networks to model fine-scale processes such as cloud formation that are too fine-grained for many climate models (Rasp et al., 2018; Beucler et al., 2019).

The computational complexity of climate models has long been an issue hampering their use in impact research and there have been many other approaches to produce emulators. Specifically, there has been much work using statistical approaches towards climate model emulation (Holden et al., 2015; Castruccio et al., 2014). Recently, generative modelling as a means to emulate the spatio-temporal behavior of climate models has become a popular choice. Previously, much of this has revolved around the use of Generative Adversarial Networks (GANs) (Puchko et al., 2020; Hutchinson et al., 2022; Ayala et al., 2021; Kashinath et al., 2021). However, as diffusion models have begun to exceed the performance of GANs (Dhariwal & Nichol, 2021), many have begun to explore their use for emulation (Bassetti et al., 2023b; Addison et al., 2022).
3 Background

3.1 Discrete Time Diffusion

Deep generative models are a type of generative model that use deep learning techniques such as neural networks to generate data that is similar to real-world data. Unlike traditional generative models, deep generative models are able to generate data with a much higher level of detail, accuracy, and complexity. In essence, deep generative models are capable of learning the underlying patterns and distributions of the data, allowing them to generate highly realistic data samples.

Denoising Diffusion Probabilistic Models (DDPM), or simply diffusion models, are a class of generative models that are both flexible and tractable, which learns to transform a sample from a known distribution, such as a Gaussian, into a sample that could be drawn from an unknown distribution (Sohl-Dickstein et al., 2015; Ho et al., 2020; Nichol & Dhariwal, 2021).

Diffusion models are trained by systematically destroying information in a sample and learning the distribution from where that sample was taken by reconstructing the destroyed sample. This sample is destroyed with an iterative forward diffusion process over a series of time-steps, the inspiration for which was drawn from non-equilibrium statistical physics. The noised sample at a given time-step is only dependent on the noised sample at the previous time-step. The conditional probability is defined as follows:

$$q(x_t| x_{t-1}) = N(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I)$$ (1)

where $\beta_t$ is a hyperparameter responsible for dictating the amount of noise added.

However, because the addition of normal distributions results in a normal distribution, the amount of noise at a given time-step from time-step zero can be calculated without needing to calculate each time-step in between. The updated conditional probability is defined below:

$$\alpha_t = \beta_t - 1$$ (2)

$$\bar{\alpha}_t = \prod_{i=0}^{T} \alpha_i$$ (3)

$$q(x_t| x_0) = N(x_t; \sqrt{\bar{\alpha}x_0}, \sqrt{1-\bar{\alpha}})$$ (4)

The reverse process is a Markov chain which converts the sample from a known distribution (Gaussian) into a sample from the unknown distribution. The training regime involves only making small denoising steps, which is far more tractable than having to model the sample distribution directly. The loss for each denoising step is simply the mean-squared error between the true noise added to a sample and the predicted noise added to the sample. For further information on discrete time diffusion, we refer readers to (Ho et al., 2020; Nichol & Dhariwal, 2021; Dhariwal & Nichol, 2021).

3.2 Continuous Time Diffusion

Our approach closely follows similar work in video diffusion modelling, specifically (Ho, Salimans, et al., 2022; Ho, Chan, et al., 2022), with certain architectural decisions from (Saharia et al., 2022). We use continuously spaced timesteps for our diffusion processes (Kingma et al., 2021). Our timesteps exist within $[0,1]$ and we construct the signal-noise-ratio based on the timesteps as follows:

$$\text{log}_{\text{snr}} = -\ln(e^{1e^{-4+10t^2}} - 1)$$ (5)
Using the signal-noise ratio obtained from a timestep, $t$, we can perform a forward pass on a sequence of climate data using the following equations, where $x_0$ is the original sequence of data, $x_t$ is a noisy sequence, and $s(\cdot)$ denotes the logistic sigmoid function.

$$\alpha = \sqrt{s(\log_{\text{snr}})} \tag{6}$$

$$\sigma = \sqrt{s(-\log_{\text{snr}})} \tag{7}$$

$$x_t = \alpha x_0 + \sigma N(0, I) \tag{8}$$

We choose to learn $\nu$, a reparameterization of $\epsilon$, as our learning objective after witnessing improved results, following (Ho, Chan, et al., 2022). Training is performed with the following gradient update equations.

$$\nu = \alpha N(0, I) - \sigma x_0 \tag{9}$$

$$\nabla_\theta ||\nu - \nu_\theta(x_t, \log_{\text{snr}_t})||^2 \tag{10}$$

4 Methods

4.1 Data

Our work focuses on two CMIP5-era (Taylor et al., 2012) datasets made up of initial condition ensemble members, which we refer to as “realizations” from the Community Earth System Model (CESM1-CAM5) (Kay et al., 2015) and the Institut Pierre-Simon Laplace Earth System Model (IPSL-CM5A-LR) (Dufresne et al., 2013). Specifically, we use 10 realizations from CESM and 6 realizations from IPSL. We reserve two realizations from each ESM for our test and validation set, and utilize the rest to train our model. All realizations consist of daily average temperature (Celsius) or total precipitation (mm) output from 1850 - 2100 for IPSL and 1920 - 2100 for CESM. Refer to Figure 1 for an example of one day of data. We train our model using both historical data as well as output from the highest-emission scenario available (RCP8.5 (Moss et al., 2010)) in order to expose the emulator to a wide range of forcing levels. We then evaluate our model on unseen scenarios to test its ability to generalize to different scenarios of anthropogenic forcing. We include Table 4.1 to show how the two models' available realizations are allocated between training and evaluation sets, and the different scenarios and time periods used. Since we are interested in generating a “month” of data with our model, we train the diffusion model using datasets that have been segmented into 28-day chunks. This has the advantage of representing regular four-week periods.
4.2 Data Preprocessing

Like many machine learning methods, diffusion models operate best when working with data normalized to a certain range. Towards this, we transform the units of temperature to degrees Celsius and normalize the data by dividing all values by twenty; although this exact value is arbitrary, it succeeds in the goal of normalizing values into a range suitable for our neural network models, reducing numerical instability. For the precipitation values, we perform log(1+x) normalization. This helps to shrink the long tail of extreme precipitation values and ensure that days with zero or very small amounts of precipitation (which happen frequently in many areas of the world in a climate model) fall within a regime that is more suitable for training. Note that neither normalization strategy is data-driven, and thus neither depends on the training or target scenarios.

4.3 Model Architecture

Diffusion based-approaches have seen much recent success in the realm of video generation. In this work, we construct our own model architecture that is highly inspired by the Video Diffusion (Ho, Salimans, et al., 2022) architectures. Specifically, our denoising model is a 3D U-Net (Ronneberger et al., 2015); however, we strip out all attention layers for computational efficiency. Each downsampling and upsampling stage in our U-Net is composed of multiple spatial-only convolutions, followed by a single temporal-only convolutional block. This allows our model to separate learning spatial structures and temporal structures, both of which are vital to creating realistic climate sequences.

4.4 Model Training

Our training algorithm is described in Algorithm 1. For our diffusion processes, we use a linearly spaced noise schedule, and 250 sampling steps for all observed samples. Additionally, we implement an exponential moving average (EMA) strategy for weight updates during training (Song & Ermon, 2020). This stabilizes the training process, leading to more robust weights for inference. To facilitate this, we maintain two separate versions of our U-Net model. The first version is the standard model, which we update using traditional gradient descent after each batch of inputs. The second version is the EMA model. In this model, rather than updating the weights based on the latest batch, we blend the weights using a historical average of past weights.

### Table 1. Data Split for each of our models. $r_i$ indicates a realization

<table>
<thead>
<tr>
<th></th>
<th>IPSL</th>
<th>CESM</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Training Realizations</strong></td>
<td>r3, r4, r5, r6</td>
<td>r3, r4, r5, r6, r7, r8, r9, r10</td>
</tr>
<tr>
<td><strong>Validation Realizations</strong></td>
<td>r2</td>
<td>r2</td>
</tr>
<tr>
<td><strong>Test Realizations</strong></td>
<td>r1</td>
<td>r1</td>
</tr>
<tr>
<td><strong>Training Scenarios</strong></td>
<td>rcp85</td>
<td>rcp85</td>
</tr>
<tr>
<td><strong>Training Years</strong></td>
<td>1850-2100</td>
<td>1920-2100</td>
</tr>
<tr>
<td><strong>Evaluation Scenarios</strong></td>
<td>rcp45, rcp85</td>
<td>rcp60, rcp85</td>
</tr>
<tr>
<td><strong>Evaluation Years</strong></td>
<td>2080-2100</td>
<td>2080-2100</td>
</tr>
</tbody>
</table>
Algorithm 1 Training Denoising Diffusion Probabilistic Models (DDPM)

1: Initialize model parameters $\theta$
2: Sample random month from the dataset, $x_0$
3: Take the average of $x_0$ over time to create a conditioning map, $c$
4: Sample random noise $\epsilon$ from $\mathcal{N}(0, I)$ and a random timestep $t$ from $[0, 1]$
5: Use $\epsilon$ to apply $t$ steps of noise to $x_0$, obtaining $x_t$, a noisy version of $x_0$
6: Concatenate $[x_0, c]$
7: Obtain $\nu$, a reparameterization of $\epsilon$
8: Use the denoising model $\theta([x_0, c])$ to obtain $\nu_\theta$, a prediction of $\nu$
9: Apply mean squared error to $\nu$ and $\nu_\theta$
10: Take a gradient descent step

Table 2. Training Hyperparameters

<table>
<thead>
<tr>
<th>Adam HyperParameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning Rate</td>
<td>0.0001</td>
</tr>
<tr>
<td>$\beta_1$</td>
<td>0.9</td>
</tr>
<tr>
<td>$\beta_2$</td>
<td>0.99</td>
</tr>
<tr>
<td>$\epsilon$</td>
<td>$1e^{-8}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Diffusion Hyperparameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sampling Steps</td>
</tr>
<tr>
<td>Noise Schedule</td>
</tr>
<tr>
<td>Loss type</td>
</tr>
</tbody>
</table>

4.5 Model Tuning

For training, we use the Adam optimizer, with hyperparameters described in Table 2. Due to time and computational constraints, we manually explored the hyperparameter space. Resources permitting, a guided hyperparameter search would likely yield even better results.

5 Results and Analyses

5.1 Evaluating DiffESM

We present here a number of evaluations of the ability of DiffESM to generate month-long sequences of daily temperature or precipitation whose statistical characteristics match those of the target ESM output to be emulated. For this set of analyses, we use data from the IPSL ESM, under RCP8.5, from the years 2080-2100. Specifically, after training the model, we generate new daily sequences by conditioning on each month within a 20-year range from realizations (runs) from our validation set – data that the emulator has not seen during training. In the following, we call the emulated daily time series DiffESM produces the “generated” set. We will compare these generated sequences to the independent (and also unseen during training) “test” set of the same length and characteristics (daily sequences for each month over the 2080-2100 period under RCP8.5). (Refer to Table 4.1 for the details of the split into training, validation and test data.) We compare not only the generated set to the test set, but also the validation set itself to the test set. The latter comparison gives us an oracle baseline against which to gauge the performance of our emulator: the validation-test discrepancies provide a lower bound on the discrepancies expected between the generated data and test. Ideally we would want to have many test sets to build a distribution of such differences, but the same constraint that we are addressing by building an emulator limits how many independent realiza-
tions we can get from an ESM. In our analysis of precipitation data, we set values below 1mm/day (i.e., dry days) to exactly 0mm/day, a common threshold choice to circumvent the tendency of ESMs to “drizzle” too frequently and therefore overestimate the number of wet days.

Figure 2 displays time series of daily values of temperature and precipitation the generated, test and validation sets for three example locations, chosen to provide diversity of climates. We extract the output of the ESM and the diffusion model at the three grid points closest to the big island of Hawaii, and the cities of Melbourne (Australia) and Novosibirsk (Russia). Here and in the following we call attention to the different ranges that the values of temperature and precipitation cover, and the different characteristics of their seasonal cycles. From visual inspection our generated, test and validation data have indistinguishable behavior, but we analyze and document this in greater detail in the coming sections.

5.1.1 Spatial and Temporal Distributions

Figure 3 displays the result of the two-sample Kolgomorov-Smirnov (KS) test comparing the cumulative distribution functions of two sets of daily output at every grid-point (or pixel location in diffusion model terminology). To create this figure, we start with the three 20-year long daily time series of generated, test and validation data. For each pixel location within the validation and test sets, we perform a KS test comparing the two empirical CDFs of daily temperature (or precipitation) estimated using the entire length of the time series (about 7000 days). The value of the test statistic represents the maximum distance between the two empirical CDFs. The non-zero KS values reflect natural variability of the model output across realization from the ESM with different initial conditions. We also perform the same KS calculation between generated and test data and display the values of the test statistic at each grid-point. Lastly, to distill this information down to a single scalar value per comparison, we average all the pixel values over the two maps to compute a mean “KS value.” We perform the analyses described above separately for our temperature and our precipitation data.

Looking at Figure 3 for precipitation, we notice that globally, the two maps share many of the same features, indicating that our generated data is capturing a very similar distribution across the globe as our validation data (whose monthly means were used to condition the generated set). Notably, the average KS score is only 0.005 (25%) higher on our generated map, indicating that the CDF of our generated precipitation data is hardly any more discrepant from the test CDF than the CDF from the validation set, produced by the same ESM, is. We notice similar behavior in our temperature data.

We now consider the temporal behavior of emulated versus ESM data, concerned with the memory characteristics of each time series set. Figure 4 and 5 contain autocorrelation (ACF) and partial autocorrelation (PACF) function plots for temperature and precipitation at our three locations. They represent the correlation between data points of the time series separated by an increasing number of lags (days in our case). Our evaluation is performed after eliminating the seasonal cycle from the time series, thus considering each month as an independent realization, and therefore borrowing strength along the length of the sets for the estimation of the correlations. The ACF behavior is affected by the lag-1 correlation dying off slowly and affecting subsequent lags. The PACF computes only the residual correlation at lag n after accounting for correlations at lag 1 through n−1. ACF and PACF behavior is evaluated by comparing the overall shape of the former, and the number of significant “spikes” in the latter. Together these characteristics indicate the order of the Auto-regressive/Moving Average (ARMA) process (Box et al., 2015) that generated the time series.
Figure 2. Time series of generated, test and validation sets of daily temperature (T) and precipitation (PR) at the three locations, covering the period 2080-2100. The x-axis shows integer values indexing the days that span the period 2080-2100 (4 weeks per month).
Figure 3. Global Kolmogorov-Smirnov Tests for Precipitation and Temperature

Table 3. Description of climate metric calculations

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temperature</td>
<td></td>
</tr>
<tr>
<td>Average Monthly Temperature</td>
<td>The average of the daily temperature values within the month</td>
</tr>
<tr>
<td>Average Monthly Hot Streak</td>
<td>The longest consecutive number of days with daily temperature values</td>
</tr>
<tr>
<td></td>
<td>above a precomputed 90th quantile threshold value (threshold computed from</td>
</tr>
<tr>
<td></td>
<td>a reference period in 1960-1990)</td>
</tr>
<tr>
<td>Average Monthly Hot Days</td>
<td>The total number of days within a month with daily temperature values above</td>
</tr>
<tr>
<td></td>
<td>the precomputed 90th quantile threshold value</td>
</tr>
<tr>
<td>Average 90th Quantile</td>
<td>The average temperature on days that exceed the precomputed 90th quantile.</td>
</tr>
<tr>
<td>Precipitation</td>
<td></td>
</tr>
<tr>
<td>Average Monthly Precipitation</td>
<td>The average of the daily rainfall values within the month (mm/day)</td>
</tr>
<tr>
<td>Average SDII</td>
<td>The sum of rainfall on days exceeding 1mm/day divided by the total number of</td>
</tr>
<tr>
<td></td>
<td>days exceeding 1 mm/day</td>
</tr>
<tr>
<td>Average Rainy Streak</td>
<td>The longest consecutive number of days within a month exceeding 1 mm/day of</td>
</tr>
<tr>
<td></td>
<td>rainfall</td>
</tr>
<tr>
<td>Average Rainy Days</td>
<td>The total number of days within a month with rainfall exceeding 1 mm/day.</td>
</tr>
</tbody>
</table>

The plots confirm the consistency of the temporal structure of the generated, test and validation data. It appears that the day-to-day memory of temperature and precipitation do not differ significantly between emulated and ESM data.

5.1.2 Climate Metrics

Often, daily data from ESMs is used to derive metrics that are representative of extreme behavior, such as hot streaks, rainy or dry streaks, intensity of hot days and rainy days, etc. We therefore choose a set of such metrics, each one summarizing the daily behavior over a month. Table 3 describes each metric computation, among which the simple daily intensity index (SDII) has been borrowed from the standard set of ETCCDI metrics (Zhang et al., 2011). Figure 6 shows our performance on a range of such metrics. Separately for the generated, validation, and test sets, at each grid-point, the metric of interest is computed for each month in the datasets (over 2081-2100). We then take the average over all months for the test set to produce a test set map, and average over all months in the validation set to produce a validation set map, and then subtract the test set map from the validation set map. This again is our baseline, showing the level of internal variability between two realizations from the same ESM. We then compute the same difference map between the generated and test sets, which we compare to this baseline.

Referring back to Figure 6, we can see that our temperature data shows remarkably similar difference plots for generated and test as for validation and test. We tend to match the validation set’s performance globally (when averaging the values over all the grid-points), and capture many of the same spatial patterns when considering the sign and magnitude of the differences over both land and oceans (some level of tempo-
Figure 4. Autocorrelation and partial autocorrelation functions of daily time series of temperature (T) at the three locations. Generated data ACFs and PCFs along the left column can be compared to those of the test and validation sets, along the middle and right column respectively. Each pair of rows corresponds to one of the three locations, with Novosibirsk at the top, Hawaii in the middle, and Melbourne at the bottom.
Figure 5. Like Fig. 4, for daily precipitation time series.
Figure 6. Relevant chosen metrics between generated set and validation set conditioned on IPSL RCP8.5 runs

Figure 7. Five generated sequences (red/blue) overlaid on top of a validation sequence whose monthly temperature and precipitation average values were used as conditioning

5.1.3 Variability

Previously, we have explored the use of GANs to emulate Earth System Models. However, a prevalent issue observed with GANs in the realm of image generation is “mode collapse” (Lala et al., 2018), a tendency to generate only a small subset of the data distribution, leading to samples that have little internal variability. We have observed the same phenomenon in the context of ESM emulation with GANs, so we assess here the variability of the samples produced by our diffusion model. To produce Figure 7, we sampled a random month (28-day sequence) from the validation set, took its average to serve as the conditioning monthly average map, and generated five samples by DiffESM. For three different locations, we plot the resulting samples as time series (with the time series of the original validation monthly sequence in black). The figure shows that DiffESM generates sequences with a wide range of behaviors, with peaks and troughs at different times during the month. In past attempts with GAN models the generated sequences tended to move in synchronicity.
5.2 Analysis: Performance Across RCPs

In this section and the next few, we present analyses of how our model performs on a wide range of forcing scenarios and years, and then replicate the evaluation on a distinct Earth system model. For brevity, we only include a subset of the analyses described above that characterize the performance of our model both spatially and temporally.

Figure 8 displays our IPSL trained model generating years for different RCP forcing scenarios. Specifically, using our diffusion model, which was trained on data from RCP8.5, the highest emissions scenario available thus covering the largest range of radiative forcings along the 21st century, we aim to compare if our model can emulate the distribution of never before seen (to the model) forcing scenarios. Aside from varying the RCP, the experimental setup is the same as above: we take the years 2080-2100, use one realization for the test set, one for the validation set, and generate 20 years of data with our model, conditioned on the validation set, for each scenario. Our analyses show that our model displays similar characteristics in each of the chosen metrics across previously unseen scenarios. Additionally, the KS test shows that we closely match the underlying distribution of these scenarios, despite the model having never seen these emissions scenarios during training. We consider this a reflection of the fact that the output we are emulating (daily temperature and precipitation) does not show a path-dependent behavior, thus the conditioning to a map of average temperature of precipitation is sufficient to recreate the correct behavior as long as the emulator has been trained on output that reflected those kinds of mean maps, independently of the scenario along which they were reached.

5.3 Analysis: Performance Across Time Periods

In this section, we analyze how our model performs across different time periods. Specifically, we analyze the performance of our model across 4, 20-year windows: 2020-2040, 2040-2060, 2060-2080, and 2080-2100. Our results in Figure 9 show that overall our results stay mostly consistent between multiple time periods. Since our model has been trained on all time periods, it makes sense that its performance would not degrade on any of them.

5.4 Analysis: Generalization to a New ESM

Although we focused our resources on the IPSL ESM, we demonstrate that the emulation process can be replicated on another ESM; specifically, CESM. Figure 10 shows our performance on IPSL compared to a model trained on CESM data. Specifically, we train an entirely new DiffESM model on CESM data and analyze its performance. We see that again DiffESM closely matches the spatial and temporal distributions of ESM. According to the Mean KS Statistic, compared to IPSL, we see better emulation (and a smaller performance gap) for precipitation, but worse performance (and a larger gap) for temperature. The larger KS value for validation-vs-test indicates greater variability between the validation and test set realizations, and the larger gap for temperature suggests that DiffESM found it more challenging to model the spatiotemporal temperature distributions conditioned on monthly means. One challenge training DiffESM on CESM is the increased size of the CESM outputs compared to IPSL outputs. The spatial resolution used for CESM is 96 × 144, about 1.5 times larger than that of our IPSL data. There are techniques for scaling diffusion model training to higher resolutions (e.g., latent diffusion (Rombach et al., 2022)) – since we see our work here as an initial exploration of diffusion models for this area of application in general, we leave for future work investigations of how to best tune them more specifically to a given model’s emulation.
Figure 8. IPSL Model, evaluated under multiple forcing scenarios

[Refer to RCPComparison.pdf, submitted separately per journal request]
Figure 9. Performance of DiffESM across multiple timespans
Conclusion and Future Work

In this paper, we have demonstrated the capability of DiffESM, a conditional video diffusion model, to emulate ESM output of daily temperature and precipitation conditioned on monthly means from a climate scenario unseen during training. We observe that the samples produced by DiffESM are comparable to those of ESMs in some fundamental characteristics, such as temporal correlation and spatial behavior, and in several extreme-relevant metrics, such as frequency and spatial distribution of hot streaks or dry spells, and intensity of precipitation during extremely wet days. In fact, we have shown that for many performance metrics, the emulator errors (the differences from the ESM output it targeted to emulate) are similar to differences between different realization from the ESM itself, i.e., comparable to internal variability. The ability to generate such simulations in a timely manner could significantly enhance our ability to characterize the risks from extreme weather events under various future climate scenarios.

Another – more pragmatic – use of emulation of daily quantities from monthly means could be as a solution to decrease the cost of archiving and handling ESM daily output, which is becoming increasingly high due to ESMs’ higher and higher resolution.

There are numerous directions for future work. One promising area would be to integrate multiple variables into a single diffusion model, since modeling the correlation between, for example, temperature and precipitation would allow for investigation of co-occurring phenomena, such as the interaction and correlation between temperature and precipitation. This would also result in output that preserves the joint characteristics of the variables and allow to address more consistently those types of extremes that result from the combination of hot and dry, or cool and wet behavior of the climate system. Despite the speed advantages over ESMs, the diffusion models could themselves be further sped up using sampling techniques such as progressive distillation (Kingma
et al., 2021). Lastly, while the work reported in this manuscript emulates two ESMs and evaluates the emulated output on three scenarios (two unseen in training), we plan to replicate these findings over many more ESMs and scenarios to further evaluate the promise of these techniques.

7 Open Research

The code used for training and evaluating the models in the study are available at https://github.com/JGCRI/diffesm and https://doi.org/10.5281/zenodo.10420734 with open access via an MIT license (Bassetti et al., 2023a). The data used to train our models was obtained from the Earth System Grid, part of the CMIP5 archive.

Acknowledgments

This research was supported by the U.S. Department of Energy, Office of Science, as part of research in MultiSector Dynamics, Earth and Environmental System Modeling Program. The Pacific Northwest National Laboratory is operated for DOE by Battelle Memorial Institute under contract DE-AC05-76RL01830. Support for BK was provided in part by the National Science Foundation through agreement SES-1754740 and the Indiana University Environmental Resilience Institute. The views and opinions expressed in this paper are those of the authors alone.

Authors would also like to acknowledge the World Climate Research Programme’s Working Group on Coupled Modelling, which is responsible for CMIP, and we thank the climate modeling groups at the National Center for Atmospheric Research (USA) and Institut Pierre Simon Laplace (France) for producing and making available their model output. For CMIP the U.S. Department of Energy’s Program for Climate Model Diagnosis and Intercomparison provides coordinating support and led development of software infrastructure in partnership with the Global Organization for Earth System Science Portals.

References


Beusich, L., Gudmundsson, L., & Seneviratne, S. I. (2020). Emulating earth system


Kay, J. E., Deser, C., Phillips, A., Mai, A., Hannay, C., Strand, G., . . . others (2015). The community earth system model (cesm) large ensemble project: A community resource for studying climate change in the presence of internal


doi: 10.5194/gmd-12-1477-2019


Figure 6.
IPSL RCP 8.5

Average Monthly Temperature

Average Monthly Precip

Average Monthly Hot Streak

Average SDII

Average Monthly Hot Days

Average Rainy Streak

Average 90th Quantile

Average Rainy Days
Figure 9.
DiffESM: Conditional Emulation of Temperature and Precipitation in Earth System Models with 3D Diffusion Models

Seth Bassetti1∗, Brian Hutchinson2,3, Claudia Tebaldi4, Ben Kravitz5,6

1Computer Science Department, Utah State University, Logan, UT
2Computer Science Department, Western Washington University, Bellingham, WA
3Foundational Data Science, Pacific Northwest National Laboratory, Seattle, WA
4Joint Global Change Research Institute, Pacific Northwest National Laboratory, College Park, MD
5Earth and Atmospheric Sciences Department, Indiana University, Bloomington, IN
6Atmospheric Sciences and Global Change Division, Pacific Northwest National Laboratory, Richland, WA

Key Points:

• Earth system models (ESMs) are key devices for understanding how human actions will affect the future global climate.
• Generating enough ESM samples for analyses like the characterization of extreme events is computationally demanding (and often intractable).
• We present DiffESM as a data-driven emulator of ESMs that closely matches the spatial and temporal distributions of their output variables.

*Work performed while at Western Washington University.

Corresponding author: Brian Hutchinson, Brian.Hutchinson@wwu.edu
Abstract
Earth System Models (ESMs) are essential tools for understanding the interaction of the human and Earth systems. One key application of these models is studying extreme weather events, such as heat waves or high intensity precipitation events, which have significant socioeconomic consequences. However, the computational demands of running a sufficient number of simulations to robustly characterize expected changes in these hazards, and therefore provide a strong basis to analyze the ensuing risks, are often prohibitive. In this paper we demonstrate that diffusion models – a class of generative deep learning models – can effectively emulate the spatio-temporal trends of ESM daily output. Trained on a handful of runs, reflecting a wide range of radiative forcings, our DiffESM model takes monthly mean precipitation or temperature as input and is capable of producing daily values of temperature and precipitation that have statistical characteristics close to the ESM output. This approach requires only a small fraction of the computational resources that would be needed to run a large ensemble under any scenario of interest. We evaluate model behavior over a range of scenarios, time horizons and two ESMs, using a number of extreme metrics, including ones that have been long established in the climate modeling and analysis community. Our results show that the samples produced by DiffESM closely matches the spatio-temporal behavior of the ESM output it emulates in terms of the frequency and spatial characteristics of phenomena such as heat waves, dry spells, or rainfall intensity.

Plain Language Summary
Ideally, to study how damaging phenomena like heatwaves, droughts and downpours will change in the future under global warming, we would want a large number of climate model runs producing many realizations of climate futures that we can analyze and from which the new characteristics of climate extremes can be quantified. Running climate models is however extremely expensive and we show how training a machine learning model over the available climate model output can fill in the gaps, thus enriching our data and potentially enabling more precise estimates of how these extremes will change in the future.

1 Introduction
Extreme weather events, such as heat waves, droughts, and floods have become more frequent and intense in recent years (on Climate Change (IPCC), 2023). These events have significant impacts on human societies and ecosystems, highlighting the urgent need to understand how they may change in the future under different emission scenarios. One important tool for investigating future climate change and its impacts on extreme weather events is the use of Earth System Models (ESMs) run under plausible future scenarios of greenhouse gas emissions.

ESMs are complex computer models that simulate the interactions between Earth’s atmosphere, oceans, land surface, biosphere, cryosphere and more. They are used to simulate a wide range of climate variables under different emissions scenarios. However, the computational demands of ESMs limit the number of simulations that can be performed, especially when climate modeling centers need to allocate experiments to meet demands from a range of scientific and practical uses. This is especially problematic when investigating rare extreme weather events, as it is necessary to aggregate data over numerous runs to obtain reliable statistics. To address this issue, emulators can be used to generate thousands of realizations of global climate data in the scale of minutes or hours, rather than weeks or months (Kasim et al., 2021). Emulators learn the statistical characteristics of ESM output from existing data, and can then generate new data under the same scenario used for training but also, importantly, under different emissions scenarios, only utilizing some type of simplified, coarser-scale “conditioning” from the target.
scenario. Machine learning approaches, especially generative deep learning methods, are well-suited to building such emulators, as they are capable of approximating complicated, high-dimensional distributions such as high resolution natural images (Rombach et al., 2022; Saharia et al., 2022; Ho et al., 2020), thus having the potential to substitute costly initial condition large ensembles.

In this paper, we present a denoising diffusion probabilistic model which learns to closely model the spatio-temporal behavior of an ESM, producing month-long samples of either daily mean temperature or precipitation. Diffusion models have shown great success in the realm of generative modeling, extending to the domains of image, audio, video, and recently even climate. Our emulator, DiffESM, can be steered to generate samples under existing or novel climate scenarios by conditioning generation on a monthly mean map of the climate variable. When not provided by an existing scenario simulation, such monthly averages can be produced by existing emulators such as fldgen (Link et al., 2019), MESMER (Beusch et al., 2020; Nath et al., 2022; Quilcaille et al., 2022) or STITCHES (Tebaldi et al., 2022). Once trained, the emulator offers a dramatic improvement over traditional ESMs in terms of speed, allowing for rapid investigation of the effect of climate scenario on the distribution of extreme weather events. This makes it a valuable tool for climate researchers and policy-makers who need to make informed decisions about the future of our planet. By enabling the generation of large ensembles of climate simulations, DiffESM can provide valuable insights into the range and magnitude of potential climate impacts, and help inform adaptation and mitigation strategies.

2 Related Work

In recent years, machine learning has gained increased attention as a tool to support research in the earth sciences (Reichstein et al., 2019). One promising area of application for data-driven algorithms is in forecasting. Traditional weather forecasting relies on physically-constrained models, but these models can be highly sensitive to initial conditions. In contrast, neural networks have shown greater robustness to uncertainties and variations in initial conditions (Wang et al., 2019; Scher & Messori, 2018; Rasel et al., 2018; Narvekar & Fargose, 2015). As a result, machine learning algorithms have been proposed as a complement or alternative to traditional methods for weather forecasting. Additionally, generative models, which can be used to create spatially coherent data have also been used as forecasting tools (Gagne II et al., 2020; Bihlo, 2021). Outside of forecasting, machine learning-based solutions have interacted directly with ESMs and Regional Climate Models (RCMs). Specifically, the spatial resolution of ESM and RCM outputs can be increased for more local-scale predictions as shown by (Hobeichi et al., 2023; Jebeile et al., 2021; Babaousmail et al., 2021). In addition to increasing spatial resolutions in certain regions, approaches have used neural networks to model fine-scale processes such as cloud formation that are too fine-grained for many climate models (Rasp et al., 2018; Beucler et al., 2019).

The computational complexity of climate models has long been an issue hampering their use in impact research and there have been many other approaches to produce emulators. Specifically, there has been much work using statistical approaches towards climate model emulation (Holden et al., 2015; Castruccio et al., 2014). Recently, generative modelling as a means to emulate the spatio-temporal behavior of climate models has become a popular choice. Previously, much of this has revolved around the use of Generative Adversarial Networks (GANs) (Puchko et al., 2020; Hutchinson et al., 2022; Ayala et al., 2021; Kashinath et al., 2021). However, as diffusion models have begun to exceed the performance of GANs (Dhariwal & Nichol, 2021), many have begun to explore their use for emulation (Bassetti et al., 2023b; Addison et al., 2022).
3 Background

3.1 Discrete Time Diffusion

Deep generative models are a type of generative model that use deep learning techniques such as neural networks to generate data that is similar to real-world data. Unlike traditional generative models, deep generative models are able to generate data with a much higher level of detail, accuracy, and complexity. In essence, deep generative models are capable of learning the underlying patterns and distributions of the data, allowing them to generate highly realistic data samples.

Denoising Diffusion Probabilistic Models (DDPM), or simply diffusion models, are a class of generative models that are both flexible and tractable, which learns to transform a sample from a known distribution, such as a Gaussian, into a sample that could be drawn from an unknown distribution (Sohl-Dickstein et al., 2015; Ho et al., 2020; Nichol & Dhariwal, 2021).

Diffusion models are trained by systematically destroying information in a sample and learning the distribution from where that sample was taken by reconstructing the destroyed sample. This sample is destroyed with an iterative forward diffusion process over a series of time-steps, the inspiration for which was drawn from non-equilibrium statistical physics. The noised sample at a given time-step is only dependent on the noised sample at the previous time-step. The conditional probability is defined as follows:

\[ q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I) \]  

where \( \beta_t \) is a hyperparameter responsible for dictating the amount of noise added.

However, because the addition of normal distributions results in a normal distribution, the amount of noise at a given time-step from time-step zero can be calculated without needing to calculate each time-step in between. The updated conditional probability is defined below:

\[ \alpha_i = \beta_i - 1 \]

\[ \bar{\alpha}_i = \prod_{i=0}^{T} \alpha_i \]

\[ q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\alpha x_0}, \sqrt{1-\bar{\alpha}}) \]

The reverse process is a Markov chain which converts the sample from a known distribution (Gaussian) into a sample from the unknown distribution. The training regime involves only making small denoising steps, which is far more tractable than having to model the sample distribution directly. The loss for each denoising step is simply the mean-squared error between the true noise added to a sample and the predicted noise added to the sample. For further information on discrete time diffusion, we refer readers to (Ho et al., 2020; Nichol & Dhariwal, 2021; Dhariwal & Nichol, 2021).

3.2 Continuous Time Diffusion

Our approach closely follows similar work in video diffusion modelling, specifically (Ho, Salimans, et al., 2022; Ho, Chan, et al., 2022), with certain architectural decisions from (Saharia et al., 2022). We use continuously spaced timesteps for our diffusion processes (Kingma et al., 2021). Our timesteps exist within \([0,1]\) and we construct the signal-noise-ratio based on the timesteps as follows:

\[ \log_{\text{snr}} = -\ln(e^{1e^{-4+10t^2}} - 1) \]
Figure 1. Example data from IPSL; a single day. Left: Total daily precipitation (mm). Right: daily average temperature (°C).

Using the signal-noise ratio obtained from a timestep, \( t \), we can perform a forward pass on a sequence of climate data using the following equations, where \( x_0 \) is the original sequence of data, \( x_t \) is a noisy sequence, and \( s(\cdot) \) denotes the logistic sigmoid function.

\[
\alpha = \sqrt{s(\log \text{snr})} \quad (6)
\]

\[
\sigma = \sqrt{s(-\log \text{snr})} \quad (7)
\]

\[
x_t = \alpha x_0 + \sigma N(0, I) \quad (8)
\]

We choose to learn \( \nu \), a reparameterization of \( \epsilon \), as our learning objective after witnessing improved results, following (Ho, Chan, et al., 2022). Training is performed with the following gradient update equations.

\[
\nu = \alpha N(0, I) - \sigma x_0 \quad (9)
\]

\[
\nabla_\theta \|\nu - \nu_0(x_t, \log \text{snr}_t)\|^2 \quad (10)
\]

4 Methods

4.1 Data

Our work focuses on two CMIP5-era (Taylor et al., 2012) datasets made up of initial condition ensemble members, which we refer to as “realizations” from the Community Earth System Model (CESM1-CAM5) (Kay et al., 2015) and the Institut Pierre-Simon Laplace Earth System Model (IPSL-CM5A-LR) (Dufresne et al., 2013). Specifically, we use 10 realizations from CESM and 6 realizations from IPSL. We reserve two realizations from each ESM for our test and validation set, and utilize the rest to train our model. All realizations consist of daily average temperature (Celsius) or total precipitation (mm) output from 1850 - 2100 for IPSL and 1920 - 2100 for CESM. Refer to Figure 1 for an example of one day of data. We train our model using both historical data as well as output from the highest-emission scenario available (RCP8.5 (Moss et al., 2010)) in order to expose the emulator to a wide range of forcing levels. We then evaluate our model on unseen scenarios to test its ability to generalize to different scenarios of anthropogenic forcing. We include Table 4.1 to show how the two models’ available realizations are allocated between training and evaluation sets, and the different scenarios and time periods used. Since we are interested in generating a “month” of data with our model, we train the diffusion model using datasets that have been segmented into 28-day chunks. This has the advantage of representing regular four-week periods.
Table 1. Data Split for each of our models. $r_i$ indicates a realization

<table>
<thead>
<tr>
<th></th>
<th>IPSL</th>
<th>CESM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training Realizations</td>
<td>r3,r4,r5,r6</td>
<td>r3,r4,r5,r6,r7,r8,r9,r10</td>
</tr>
<tr>
<td>Validation Realizations</td>
<td>r2</td>
<td>r2</td>
</tr>
<tr>
<td>Test Realizations</td>
<td>r1</td>
<td>r1</td>
</tr>
<tr>
<td>Training Scenarios</td>
<td>rcp85</td>
<td>rcp85</td>
</tr>
<tr>
<td>Training Years</td>
<td>1850-2100</td>
<td>1920-2100</td>
</tr>
<tr>
<td>Evaluation Scenarios</td>
<td>rcp45, rcp85</td>
<td>rcp60, rcp85</td>
</tr>
<tr>
<td>Evaluation Years</td>
<td>2080-2100</td>
<td>2080-2100</td>
</tr>
</tbody>
</table>

4.2 Data Preprocessing

Like many machine learning methods, diffusion models operate best when working with data normalized to a certain range. Towards this, we transform the units of temperature to degrees Celsius and normalize the data by dividing all values by twenty; although this exact value is arbitrary, it succeeds in the goal of normalizing values into a range suitable for our neural network models, reducing numerical instability. For the precipitation values, $x$, we perform log(1+$x$) normalization. This helps to shrink the long tail of extreme precipitation values and ensure that days with zero or very small amounts of precipitation (which happen frequently in many areas of the world in a climate model) fall within a regime that is more suitable for training. Note that neither normalization strategy is data-driven, and thus neither depends on the training or target scenarios.

4.3 Model Architecture

Diffusion based-approaches have seen much recent success in the realm of video generation. In this work, we construct our own model architecture that is highly inspired by the Video Diffusion (Ho, Salimans, et al., 2022) architectures. Specifically, our denoising model is a 3D U-Net (Ronneberger et al., 2015); however, we strip out all attention layers for computational efficiency. Each downsampling and upsampling stage in our U-Net is composed of multiple spatial-only convolutions, followed by a single temporal-only convolutional block. This allows our model to separate learning spatial structures and temporal structures, both of which are vital to creating realistic climate sequences.

4.4 Model Training

Our training algorithm is described in Algorithm 1. For our diffusion processes, we use a linearly spaced noise schedule, and 250 sampling steps for all observed samples. Additionally, we implement an exponential moving average (EMA) strategy for weight updates during training (Song & Ermon, 2020). This stabilizes the training process, leading to more robust weights for inference. To facilitate this, we maintain two separate versions of our U-Net model. The first version is the standard model, which we update using traditional gradient descent after each batch of inputs. The second version is the EMA model. In this model, rather than updating the weights based on the latest batch, we blend the weights using a historical average of past weights.
Algorithm 1 Training Denoising Diffusion Probabilistic Models (DDPM)

1: Initialize model parameters $\theta$
2: Sample random month from the dataset, $x_0$
3: Take the average of $x_0$ over time to create a conditioning map, $c$
4: Sample random noise $\epsilon$ from $\mathcal{N}(0, I)$ and a random timestep $t$ from $[0, 1]$
5: Use $\epsilon$ to apply $t$ steps of noise to $x_0$, obtaining $x_t$, a noisy version of $x_0$
6: Concatenate $[x_0, c]$
7: Obtain $\nu$, a reparameterization of $\epsilon$
8: Use the denoising model $\theta([x_0, c])$ to obtain $\nu_{\theta}$, a prediction of $\nu$
9: Apply mean squared error to $\nu$ and $\nu_{\theta}$
10: Take a gradient descent step

Table 2. Training Hyperparameters

<table>
<thead>
<tr>
<th>Adam HyperParameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning Rate</td>
<td>0.0001</td>
</tr>
<tr>
<td>$\beta_1$</td>
<td>0.9</td>
</tr>
<tr>
<td>$\beta_2$</td>
<td>0.99</td>
</tr>
<tr>
<td>$\epsilon$</td>
<td>1e-8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Diffusion Hyperparameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sampling Steps</td>
</tr>
<tr>
<td>Noise Schedule</td>
</tr>
<tr>
<td>Loss type</td>
</tr>
</tbody>
</table>

4.5 Model Tuning

For training, we use the Adam optimizer, with hyperparameters described in Table 2. Due to time and computational constraints, we manually explored the hyperparameter space. Resources permitting, a guided hyperparameter search would likely yield even better results.

5 Results and Analyses

5.1 Evaluating DiffESM

We present here a number of evaluations of the ability of DiffESM to generate month-long sequences of daily temperature or precipitation whose statistical characteristics match those of the target ESM output to be emulated. For this set of analyses, we use data from the IPSL ESM, under RCP8.5, from the years 2080-2100. Specifically, after training the model, we generate new daily sequences by conditioning on each month within a 20-year range from realizations (runs) from our validation set – data that the emulator has not seen during training. In the following, we call the emulated daily time series DiffESM produces the “generated” set. We will compare these generated sequences to the independent (and also unseen during training) “test” set of the same length and characteristics (daily sequences for each month over the 2080-2100 period under RCP8.5). (Refer to Table 4.1 for the details of the split into training, validation and test data.) We compare not only the generated set to the test set, but also the validation set itself to the test set. The latter comparison gives us an oracle baseline against which to gauge the performance of our emulator: the validation-test discrepancies provide a lower bound on the discrepancies expected between the generated data and test. Ideally we would want to have many test sets to build a distribution of such differences, but the same constraint that we are addressing by building an emulator limits how many independent realiza-
tions we can get from an ESM. In our analysis of precipitation data, we set values below 1mm/day (i.e., dry days) to exactly 0mm/day, a common threshold choice to circumvent the tendency of ESMs to “drizzle” too frequently and therefore overestimate the number of wet days.

Figure 2 displays time series of daily values of temperature and precipitation the generated, test and validation sets for three example locations, chosen to provide diversity of climates. We extract the output of the ESM and the diffusion model at the three grid points closest to the big island of Hawaii, and the cities of Melbourne (Australia) and Novosibirsk (Russia). Here and in the following we call attention to the different ranges that the values of temperature and precipitation cover, and the different characteristics of their seasonal cycles. From visual inspection our generated, test and validation data have indistinguishable behavior, but we analyze and document this in greater detail in the coming sections.

5.1.1 Spatial and Temporal Distributions

Figure 3 displays the result of the two-sample Kolgomorov-Smirnov (KS) test comparing the cumulative distribution functions of two sets of daily output at every grid-point (or pixel location in diffusion model terminology). To create this figure, we start with the three 20-year long daily time series of generated, test and validation data. For each pixel location within the validation and test sets, we perform a KS test comparing the two empirical CDFs of daily temperature (or precipitation) estimated using the entire length of the time series (about 7000 days). The value of the test statistic represents the maximum distance between the two empirical CDFs. The non-zero KS values reflect natural variability of the model output across realization from the ESM with different initial conditions. We also perform the same KS calculation between generated and test data and display the values of the test statistic at each grid-point. Lastly, to distill this information down to a single scalar value per comparison, we average all the pixel values over the two maps to compute a mean “KS value.” We perform the analyses described above separately for our temperature and our precipitation data.

Looking at Figure 3 for precipitation, we notice that globally, the two maps share many of the same features, indicating that our generated data is capturing a very similar distribution across the globe as our validation data (whose monthly means were used to condition the generated set). Notably, the average KS score is only 0.005 (25%) higher on our generated map, indicating that the CDF of our generated precipitation data is hardly any more discrepant from the test CDF than the CDF from the validation set, produced by the same ESM, is. We notice similar behavior in our temperature data.

We now consider the temporal behavior of emulated versus ESM data, concerned with the memory characteristics of each time series set. Figure 4 and 5 contain autocorrelation (ACF) and partial autocorrelation (PACF) function plots for temperature and precipitation at our three locations. They represent the correlation between data points of the time series separated by an increasing number of lags (days in our case). Our evaluation is performed after eliminating the seasonal cycle from the time series, thus considering each month as an independent realization, and therefore borrowing strength along the length of the sets for the estimation of the correlations. The ACF behavior is affected by the lag-1 correlation dying off slowly and affecting subsequent lags. The PACF computes only the residual correlation at lag n after accounting for correlations at lag 1 through n−1. ACF and PACF behavior is evaluated by comparing the overall shape of the former, and the number of significant “spikes” in the latter. Together these characteristics indicate the order of the Auto-regressive/Moving Average (ARMA) process (Box et al., 2015) that generated the time series.
Figure 2. Time series of generated, test and validation sets of daily temperature (T) and precipitation (PR) at the three locations, covering the period 2080-2100. The x-axis shows integer values indexing the days that span the period 2080-2100 (4 weeks per month).
The plots confirm the consistency of the temporal structure of the generated, test and validation data. It appears that the day-to-day memory of temperature and precipitation do not differ significantly between emulated and ESM data.

### 5.1.2 Climate Metrics

Often, daily data from ESMs is used to derive metrics that are representative of extreme behavior, such as hot streaks, rainy or dry streaks, intensity of hot days and rainy days, etc. We therefore choose a set of such metrics, each one summarizing the daily behavior over a month. Table 3 describes each metric computation, among which the simple daily intensity index (SDII) has been borrowed from the standard set of ETCCDI metrics (Zhang et al., 2011). Figure 6 shows our performance on a range of such metrics. Separately for the generated, validation, and test sets, at each grid-point, the metric of interest is computed for each month in the datasets (over 2081-2100). We then take the average over all months for the test set to produce a test set map, and average over all months in the validation set to produce a validation set map, and then subtract the test set map from the validation set map. This again is our baseline, showing the level of internal variability between two realizations from the same ESM. We then compute the same difference map between the generated and test sets, which we compare to this baseline.

Referring back to Figure 6, we can see that our temperature data shows remarkably similar difference plots for generated and test as for validation and test. We tend to match the validation set’s performance globally (when averaging the values over all the grid-points), and capture many of the same spatial patterns when considering the sign and magnitude of the differences over both land and oceans (some level of tempo-

---

**Table 3.** Description of climate metric calculations

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Temperature</strong></td>
<td></td>
</tr>
<tr>
<td>Average Monthly Temperature</td>
<td>The average of the daily temperature values within the month</td>
</tr>
<tr>
<td>Average Monthly Hot Streak</td>
<td>The longest consecutive number of days with daily temperature values above a precomputed 90th quantile threshold value (threshold computed from a reference period in 1960-1990)</td>
</tr>
<tr>
<td>Average Monthly Hot Days</td>
<td>The total number of days within a month with daily temperature values above the precomputed 90th quantile threshold value</td>
</tr>
<tr>
<td>Average 90th Quantile</td>
<td>The average temperature on days that exceed the precomputed 90th quantile.</td>
</tr>
<tr>
<td><strong>Precipitation</strong></td>
<td></td>
</tr>
<tr>
<td>Average Monthly Precipitation</td>
<td>The average of the daily rainfall values within the month (mm/day)</td>
</tr>
<tr>
<td>Average SDII</td>
<td>The sum of rainfall on days exceeding 1 mm/day divided by the total number of days exceeding 1 mm/day</td>
</tr>
<tr>
<td>Average Rainy Streak</td>
<td>The longest consecutive number of days within a month exceeding 1 mm/day of rainfall</td>
</tr>
<tr>
<td>Average Rainy Days</td>
<td>The total number of days within a month with rainfall exceeding 1 mm/day.</td>
</tr>
</tbody>
</table>

---

**Figure 3.** Global Kolmogorov-Smirnov Tests for Precipitation and Temperature
Figure 4. Autocorrelation and partial autocorrelation functions of daily time series of temperature (T) at the three locations. Generated data ACFs and PCFs along the left column can be compared to those of the test and validation sets, along the middle and right column respectively. Each pair of rows corresponds to one of the three locations, with Novosibirsk at the top, Hawaii in the middle, and Melbourne at the bottom.
Figure 5. Like Fig. 4, for daily precipitation time series.
Figure 6. Relevant chosen metrics between generated set and validation set conditioned on IPSL RCP8.5 runs

![Figure 6](image)

Figure 7. Five generated sequences (red/blue) overlaid on top of a validation sequence whose monthly temperature and precipitation average values were used as conditioning

![Figure 7](image)

5.1.3 Variability

Previously, we have explored the use of GANs to emulate Earth System Models. However, a prevalent issue observed with GANs in the realm of image generation is “mode collapse” (Lala et al., 2018), a tendency to generate only a small subset of the data distribution, leading to samples that have little internal variability. We have observed the same phenomenon in the context of ESM emulation with GANs, so we assess here the variability of the samples produced by our diffusion model. To produce Figure 7, we sampled a random month (28-day sequence) from the validation set, took its average to serve as the conditioning monthly average map, and generated five samples by DiffESM. For three different locations, we plot the resulting samples as time series (with the time series of the original validation monthly sequence in black). The figure shows that DiffESM generates sequences with a wide range of behaviors, with peaks and troughs at different times during the month. In past attempts with GAN models the generated sequences tended to move in synchronicity.
5.2 Analysis: Performance Across RCPs

In this section and the next few, we present analyses of how our model performs on a wide range of forcing scenarios and years, and then replicate the evaluation on a distinct Earth system model. For brevity, we only include a subset of the analyses described above that characterize the performance of our model both spatially and temporally.

Figure 8 displays our IPSL trained model generating years for different RCP forcing scenarios. Specifically, using our diffusion model, which was trained on data from RCP8.5, the highest emissions scenario available thus covering the largest range of radiative forcings along the 21st century, we aim to compare if our model can emulate the distribution of never before seen (to the model) forcing scenarios. Aside from varying the RCP, the experimental setup is the same as above: we take the years 2080-2100, use one realization for the test set, one for the validation set, and generate 20 years of data with our model, conditioned on the validation set, for each scenario. Our analyses show that our model displays similar characteristics in each of the chosen metrics across previously unseen scenarios. Additionally, the KS test shows that we closely match the underlying distribution of these scenarios, despite the model having never seen these emissions scenarios during training. We consider this a reflection of the fact that the output we are emulating (daily temperature and precipitation) does not show a path-dependent behavior, thus the conditioning to a map of average temperature of precipitation is sufficient to recreate the correct behavior as long as the emulator has been trained on output that reflected those kinds of mean maps, independently of the scenario along which they were reached.

5.3 Analysis: Performance Across Time Periods

In this section, we analyze how our model performs across different time periods. Specifically, we analyze the performance of our model across 4, 20-year windows: 2020-2040, 2040-2060, 2060-2080, and 2080-2100. Our results in Figure 9 show that overall our results stay mostly consistent between multiple time periods. Since our model has been trained on all time periods, it makes sense that its performance would not degrade on any of them.

5.4 Analysis: Generalization to a New ESM

Although we focused our resources on the IPSL ESM, we demonstrate that the emulation process can be replicated on another ESM; specifically, CESM. Figure 10 shows our performance on IPSL compared to a model trained on CESM data. Specifically, we train an entirely new DiffESM model on CESM data and analyze its performance. We see that again DiffESM closely matches the spatial and temporal distributions of ESM. According to the Mean KS Statistic, compared to IPSL, we see better emulation (and a smaller performance gap) for precipitation, but worse performance (and a larger gap) for temperature. The larger KS value for validation-vs-test indicates greater variability between the validation and test set realizations, and the larger gap for temperature suggests that DiffESM found it more challenging to model the spatiotemporal temperature distributions conditioned on monthly means. One challenge training DiffESM on CESM is the increased size of the CESM outputs compared to IPSL outputs. The spatial resolution used for CESM is 96 × 144, about 1.5 times larger than that of our IPSL data. There are techniques for scaling diffusion model training to higher resolutions (e.g., latent diffusion (Rombach et al., 2022)) – since we see our work here as an initial exploration of diffusion models for this area of application in general, we leave for future work investigations of how to best tune them more specifically to a given model’s emulation.
Figure 8. IPSL Model, evaluated under multiple forcing scenarios.
Figure 9. Performance of DiffESM across multiple timespans
6 Conclusion and Future Work

In this paper, we have demonstrated the capability of DiffESM, a conditional video diffusion model, to emulate ESM output of daily temperature and precipitation conditioned on monthly means from a climate scenario unseen during training. We observe that the samples produced by DiffESM are comparable to those of ESMs in some fundamental characteristics, such as temporal correlation and spatial behavior, and in several extreme-relevant metrics, such as frequency and spatial distribution of hot streaks or dry spells, and intensity of precipitation during extremely wet days. In fact, we have shown that for many performance metrics, the emulator errors (the differences from the ESM output it targeted to emulate) are similar to differences between different realization from the ESM itself, i.e., comparable to internal variability. The ability to generate such simulations in a timely manner could significantly enhance our ability to characterize the risks from extreme weather events under various future climate scenarios.

Another — more pragmatic — use of emulation of daily quantities from monthly means could be as a solution to decrease the cost of archiving and handling ESM daily output, which is becoming increasingly high due to ESMs’ higher and higher resolution.

There are numerous directions for future work. One promising area would be to integrate multiple variables into a single diffusion model, since modeling the correlation between, for example, temperature and precipitation would allow for investigation of co-occurring phenomena, such as the interaction and correlation between temperature and precipitation. This would also result in output that preserves the joint characteristics of the variables and allow to address more consistently those types of extremes that result from the combination of hot and dry, or cool and wet behavior of the climate system. Despite the speed advantages over ESMs, the diffusion models could themselves be further sped up using sampling techniques such as progressive distillation (Kingma
et al., 2021). Lastly, while the work reported in this manuscript emulates two ESMs and evaluates the emulated output on three scenarios (two unseen in training), we plan to replicate these findings over many more ESMs and scenarios to further evaluate the promise of these techniques.

7 Open Research

The code used for training and evaluating the models in the study are available at https://github.com/JGCRI/diffesm and https://doi.org/10.5281/zenodo.10420734 with open access via an MIT license (Bassetti et al., 2023a). The data used to train our models was obtained from the Earth System Grid, part of the CMIP5 archive.

Acknowledgments

This research was supported by the U.S. Department of Energy, Office of Science, as part of research in MultiSector Dynamics, Earth and Environmental System Modeling Program. The Pacific Northwest National Laboratory is operated for DOE by Battelle Memorial Institute under contract DE-AC05-76RL01830. Support for BK was provided in part by the National Science Foundation through agreement SES-1754740 and the Indiana University Environmental Resilience Institute. The views and opinions expressed in this paper are those of the authors alone.

Authors would also like to acknowledge the World Climate Research Programme’s Working Group on Coupled Modelling, which is responsible for CMIP, and we thank the climate modeling groups at the National Center for Atmospheric Research (USA) and Institut Pierre Simon Laplace (France) for producing and making available their model output. For CMIP the U.S. Department of Energy’s Program for Climate Model Diagnosis and Intercomparison provides coordinating support and led development of software infrastructure in partnership with the Global Organization for Earth System Science Portals.

References


Beusch, L., Gudmundsson, L., & Seneviratne, S. I. (2020). Emulating earth system
model temperatures with mesmer: from global mean temperature trajectories
doi: 10.5194/esd-11-139-2020
www.sciencedirect.com/science/article/pii/S0893608021000459
doi: https://doi.org/10.1016/j.neunet.2021.02.003
ing and control*. Hoboken, New Jersey: John Wiley and Sons Inc.
Castruccio, S., McNerney, D. J., Stein, M. L., Liu Crouch, F., Jacob, R. L., &
others (2013). Climate change projections using the ipsl-cm5 earth system
model: from cmip3 to cmip5. *Climate dynamics, 40*, 2123–2165.
Gagne II, D. J., Christensen, H. M., Subramanian, A. C., & Monahan, A. H.
(2020). Machine learning for stochastic parameterization: Generative ad-
versarial networks in the lorenz 96 model. *Journal of Advances in Mod-
agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2019MS001896
doi: https://doi.org/10.1029/2019MS001896
Hobeichi, S., Nishant, N., Shao, Y., Abramowitz, G., Pitman, A., Sherwood, S.,...
Green, S. (2023). Using machine learning to cut the cost of dynamical
Hutchinson, B., Ayala, A., Drazic, C., Bassetti, S., Slyman, E., Nieva, B.,...
Kashinath, K., Mustafa, M., Albert, A., Wu, J., Jiang, C., Esmaeilzadeh, S.,...
others (2021). Physics-informed machine learning: case studies for weather
Kasim, M. F., Watson-Parris, D., Deaconu, L., Oliver, S., Hatfield, P., Froula, D. H.,
Kay, J. E., Deser, C., Phillips, A., Mai, A., Hannay, C., Strand, G.,... others (2015). The community earth system model (cesm) large ensemble project:
A community resource for studying climate change in the presence of internal


Supporting Information for “DiffESM: Conditional Emulation of Temperature and Precipitation in Earth System Models with 3D Diffusion Models”

Seth Bassetti\textsuperscript{1,*}, Brian Hutchinson\textsuperscript{2,3}, Claudia Tebaldi\textsuperscript{4}, Ben Kravitz\textsuperscript{5,6}

\textsuperscript{1}Computer Science Department, Utah State University, Logan, UT

\textsuperscript{2}Computer Science Department, Western Washington University, Bellingham, WA

\textsuperscript{3}Foundational Data Science, Pacific Northwest National Laboratory, Seattle, WA

\textsuperscript{4}Joint Global Change Research Institute, Pacific Northwest National Laboratory, College Park, MD

\textsuperscript{5}Earth and Atmospheric Sciences Department, Indiana University, Bloomington, IN

\textsuperscript{6}Atmospheric Sciences and Global Change Division, Pacific Northwest National Laboratory, Richland, WA

\textbf{Supplementary Figures Overview}

1. \textit{Extended Data Figures} - This section includes additional figures that complement and extend the findings presented in the main paper. These figures offer further insights and metrics, which could not be included in the primary paper due to space.

*Work performed while at Western Washington University.
- Figure S1: This figure displays the metrics over multiple time-intervals not shown in the main paper. It includes comprehensive metrics from an IPSL-trained model for the RCP85 scenario over the specified time periods.

- Figure S2: Illustrates the performance of DiffESM across different scenarios for a constant ESM and timeframe, specifically from 2080 to 2100 using an IPSL-trained model. It encompasses all metrics that were not part of the main paper.

- Figure S3: Shows the performance of DiffESM trained and evaluated on a CESM dataset, highlighting all metrics that were not included in the main paper.

Figures
Figure S1: Performance of DiffESM across multiple timespans
Figure S2: Performance of DiffESM across multiple Scenarios
Figure S3: Performance of DiffESM on the CESM dataset