

Title

## How to Interact with Climate Models through Machine Learning-based Emulators

Creating Interactive Climate Emulators with Physics-informed Machine Learning  
How to Emulate our Climate with Machine Learning  
How to Emulate our Climate with Physics-informed Machine Learning  
How to use Machine Learning to Play with Climate Models  
Exploring Climate Policies with Machine Learning Emulators  
Interacting with our Climate Through Machine Learning Emulators  
How to interact with Climate Models through Machine Learning

### Abstract (2000 chars)

Decision-makers critically rely on climate models for understanding climate change and setting climate policies. These models however require inaccessible supercomputers. As a result, any task that requires multiple model runs, such as risk or sensitivity analyses, is expensive. In particular, interacting with climate models through educational "what-if" scenarios, such as "what-if we plant 1 trillion trees", is too expensive.

The crux of the computational complexity is that the underlying differential equations have to be solved for every scenario or parameter update from scratch. Machine learning (ML) promises a paradigm shift: the idea behind ML-based ,emulators' is to train ML models on large databases of simulated data. After training, the model has memorized how to solve the differential equations and can be quickly run for new parameter updates.

ML-based emulators have attracted attention in the Earth sciences by predicting weather in seconds rather than hours and in machine learning due to challenges in physical structure, high-dimensionality, and long-term stability. While there exists tutorials for emulating the weather (Pathak et al., 2022), emulating the climate poses different challenges. Specifically, physical-consistency and long-term stability are fundamental in developing emulators that have to extrapolate multiple years beyond the training data.

This tutorial will implement an interactive climate emulator that can be used to explore „what-if“ climate policies. Our goal is to enable Earth scientists to develop emulators for their research and ML scientists to take on the challenges in climate data. Because every Earth science emulator will require different choices, our tutorial will focus on explaining those choices, e.g., when to use non-ML vs. ML and how to incorporate physical knowledge. The final model can be used to illustrate and explore climate policies. Building on [En-ROADS], this will be the first interactive and spatially-resolved climate policy emulator.

Co-authors:

- Lea Hadzic
- Dava Newman
- Mark Veillette

## 1. Tutorial Outline \* (8000 chars)

Provide an outline of the structure of your proposed tutorial, including relevant sections and subsections.

- Why need emulators?
  - o Computationally expensive of models prevents parameter exploration, uncertainty quantification, data assimilation, downscaling, and real-time forecasting. Here, we focus on parameter exploration, in the context of educating policymakers through what-if scenarios.
- Why is climate emulation different from weather? (averages rather than predictions)
  - o Need long-term stability and physical consistency to create model that is trusted for prediction of future climate.
- How to source climate data:
  - o When to use simulated vs. reanalysis vs. observational data. What are the go-to data source.
- o We will implement a data loader for CMI6 CESM data, similar but more targeted than the ClimateBench data.
- When to use non-ML vs. ML emulators:
  - o Reduced-order models, such as POD, Projection-based ROMs, DMD, Koopman. We will explain in which cases climatology or PCA-based reduced-order models are sufficient.
  - o Hierarchical physical models, such as simplifications in modeled dimensions or terms. We will explain when energy-balance models are sufficient.
  - o Data-fit surrogates, such as GPs, Sparse Grids, Polynomials, Regression, Generative Models. We will explain in which cases deep neural network-based approaches are promising.
  - o We will implement a baseline deep learning models, e.g., UNet, FNO, or FourCastNet
- Which ML model to use:
  - o This section will very briefly differentiate myopic vs. temporal and local vs. global models. A full review of deep learning methods is out-of-scope.
- Why need physical consistency in data-fit surrogates?
  - o Climate emulation requires trust and data-efficiency
- What are accessible methods to incorporate physical consistency:
  - o There are various methods to incorporate equi- or invariances in the model architecture, loss, or data.
  - o We will implement a hybrid physics – ML architecture.
- How to evaluate?
  - o Evaluating on simulated vs. observational data.
  - o How to evaluate physical-consistency and long-term stability.
  - o We will implement evaluation with RMSE, ACC, and a use-case-centric metrics.
- Final emulator:
  - o The tutorial will result in a usable climate emulator. Participants will be able to use and share the emulator to rapidly explore the impact of various climate policies, e.g., via mapping selected CO2 emission pathways to temperature extremes. Other ML tasks in Earth Science, such as downscaling or data assimilation will be briefly mentioned during the tutorial, but the code section of the tutorial will focus on emulators for parameter exploration. The focus on a useable climate emulator for parameter exploration will differentiate our tutorial from [Bansal et al., 22], on top of us providing a novel review of physical consistency and emulator-choices.

## 2. Key Learning Objectives \* (8000 chars)

Please provide a few bullet points on the pedagogical goals of the tutorial, emphasizing the key concepts that users will learn from the tutorial.

- Our tutorial will focus on explaining modeling choices, such that Earth scientists are empowered to take different choices when adapting our tutorial to their own research.
- Similarly our tutorial will motivate ML scientists to tackle challenges that arise in climate data, specifically long-term stability and physical consistency
- Lastly, we want to convey the impact emulators can have in educating climate policy choices.

### 3. Pathway to Impact \* (8000 chars)

Please explain how the methods in the proposed tutorial are relevant to a climate change problem, and how you aim to showcase this aspect in the tutorial (e.g. through general discussion, using sample real-world datasets).

We kindly ask authors to emphasize the real-world impact of the methods/models in the tutorial by answering questions such as: Who will be using the models/outputs and how will they be used? What decisions will be made based on these models? How will this impact existing systems/the environment/affected communities on the ground?

Simplified climate models are already used, most prominently En-ROADS is an hierarchical physics-based emulator that has educated >200.000 policy- and decision-makers on climate choices. While En-ROADS is a global emulator, our implemented climate emulator will be the first to offer interactive exploration of climate policy impacts (temperature) at the local (100km) level.

More broadly, our tutorial will convey the toolkit to build emulators in Earth science. Earth scientists will be empowered to adapt the toolkit for risk or sensitivity analyses, parameter exploration, downscaling, or data assimilation in their climate-related fields. The progress of emulators in Earth science is hindered by challenges in physical-consistency and long-term stability – hopefully this tutorial can motivate ML scientistst to address those challenges.

#### Target Audience (5000 chars)

Kindly specify the intended audience for the tutorial. Please be as specific as possible, and feel free to elaborate on their expected background in ML and the climate-relevant domain.

The tutorial will target two groups. The first are domain scientists, including but not limited to Earth system modelers and Earth scientists, that intend to experiment with machine learning emulators for, e.g., sensitivity analyses, parameter exploration, or real-time inference. They are expected to have familiarity with Python, numerical modeling, and basic surrogates that are taught in Science curricula, such as EMD/PCA-based method. The scientists are expected to walk away with a clarification on where ML can be useful, data expectations, and model choices.

The second group are students in machine learning that are looking for time-series, video, or real-world datasets that introduce challenges in long-term stability and physical structure. They are expected to have familiarity with Python, pytorch, and implementing ML models. The students will walk away with a notebook on how to access a train a baseline pure and hybrid ML model on a common weather/climate dataset.

Lastly, all students will gain an appreciation for the educational value and insight that fast and accessible emulators can generate.

## 5. ML Difficulty Level

Beginner (requires little to no background in ML)

**Intermediate** (requires knowledge of basic ML concepts)

Advanced (requires knowledge of advanced ML/DL concepts)

## 6. Climate Difficulty Level \*

**Beginner** (requires little to no background in climate change)

Intermediate (requires some knowledge of climate-relevant domain-specific concepts)

Advanced (requires knowledge of advanced climate-relevant domain-specific concepts)

## 7. Prerequisites (5000 chars)

Specify the prerequisites and basic knowledge required for the users to follow the tutorial.

Experience in Python. The tutorial will assume basic familiarity with pytorch and having vague familiarity with numerical modeling. The tutorial will use domain specific processing libraries, such as xarray, but does not assume familiarity with them.

## 8. Tutorial Format \*

In what format will you be submitting the tutorial?

[Colab notebook](#)

## 10. Programming Language \*

Python

## 12. Datasets \*

Kindly provide a description of the dataset(s) you plan to use for your tutorial, including link(s) when possible. If you are introducing a novel dataset, kindly specify information such as the data type or file format (e.g. text, image, video, tabular), size, spatial resolution, temporal resolution, labels or categories, etc.

- ClimateBench (<https://github.com/duncanwp/ClimateBench>)
- Open-source CMIP6 CESM data from Pangeo archive (link redacted for anonymity)

## 13. Highlighted Tools (5000 chars)

Kindly specify the tools, such as libraries, packages, services, datasets, or frameworks, that you would like to highlight in the tutorial, e.g. Pytorch, NetworkX, Earth Engine. Feel free to include link(s), if necessary.

pytorch, xarray, pytorch.models, numpy, pandas, matplotlib. Possibly leaflet and streamlit.

## 14. Supplementary Materials (8000 chars)

Include links to anonymized source code, slide decks, video presentations, and other resources that add context to your proposal. If you already have an initial draft of your proposed tutorial, please provide a link to your tutorial notebook here. Kindly redact author names and affiliations.

We have already published a tutorial on Nowcasting for a NeurIPS dataset publication and a university class, but are redacting the link to maintain anonymity.

#### 15.Statement of Commitment \*

Please note that we will post your tutorial to the CCAI website for public consumption after the ICLR'23 workshop event. We expect that you will be available to make adjustments as needed to this tutorial in the future (e.g. updating dependencies). If you are unable to meet this obligation after the workshop event, we will remove your tutorial from the website once the code becomes inoperable.

If accepted, I understand that I will be expected to spend around 5 hours/week on tutorial development over the period January 28 - April 7.

I agree

#### 16.Additional Comments

Please provide any additional comments here.