

# DARPA ACTM: Milestone 2 Report

## AIBEDO: A hybrid AI framework to capture the effects of cloud properties on global circulation and regional climate patterns

*Prepared by Kalai Ramea\**

*with inputs from Hansi Alice Singh<sup>†</sup>, Soo Kyung Kim\*, Peetak Mitra\*, Subhashis Hazarika\*, Dipti Hingmire<sup>†</sup> and Haruki Hirasawa<sup>† ‡</sup>*

\*Palo Alto Research Center, Inc.

<sup>†</sup> University of Victoria

<sup>‡</sup> University of Toronto

February 13, 2022

This milestone objective is to identify the known and unknown parts of the modeling framework. It includes two major sections: (a) details on the 'alpha' model version of AIBEDO, specific architecture choices of sub-components, and lessons learned during the model development, and (b) data preprocessing steps, and compatibility checks.

### Overall Update

1. We have developed *alpha* versions of the model sub-components: spatial and temporal data-driven networks
2. We designed a geodesy-aware spherical sampling algorithm to convert from 2D grid data of climate variables into spherical coordinates
3. We are continuing to preprocess Earth System Model (ESM) output and reanalysis data which will be used for model training and validation
4. We have obtained HPC compute credits from AWS for preprocessing the climate datasets

## Hybrid Model

Our hybrid model development consists of two data-driven components: spherical-Unet for spatial network modeling and multi-timescale Long Short-Term Memory (LSTM) network for temporal modeling. Both components will be infused with physics-based constraints to ensure the generalizability of spatial and temporal scales. We have developed the alpha version or the "skeleton model architecture" of the data-driven components as the first step. This ensures that the model components work as intended and are compatible with the datasets.

### Spatial Data-Driven Component

The spatial data-driven component in AIBEDO consists of two major sub-modules

1. Sampling module
2. Model library module

## Sampling module

Typically, models use climate data in a uniform 2D rectangular gridded pattern. While this may suffice local/regional modeling attributes, they do not capture the physical/geodesy properties of the Earth, particularly as the focus moves away from the equator. For this reason, we developed a geodesy-aware sampling that converts 2D rectangular gridded coordinates to a geodesic grid type. There are several ways a geodesic grid can be manifested. Our method allows the modeler to choose the target gridding type between two such mesh types: (a) Icosahedral, (b) Healpix.

### Icosahedral Grids

Icosahedral grids consist of a certain number of equiangular triangles to form a convex polygon, called Icosahedron. The triangles are formed by equally spaced grid points to form a sphere. The number of grid points are defined by their *levels*. The equation to estimate the number of points in an icosahedron is shown below:

$$N = 10 * 2^{2g} + 2 \quad (1)$$

Here,  $g$  refers to the grid level we want to generate the mesh for, and  $N$  refers to the number of points in the grid that forms the icosahedron. Table 1 and Figure 1 show the total number of points for the respective grid levels.

Table 1: Number of points generated in an icosahedron for the chosen grid level

Grid level	Number of points in an icosahedron
0	12
1	42
2	162
3	642
4	2562
5	10242
6	40962

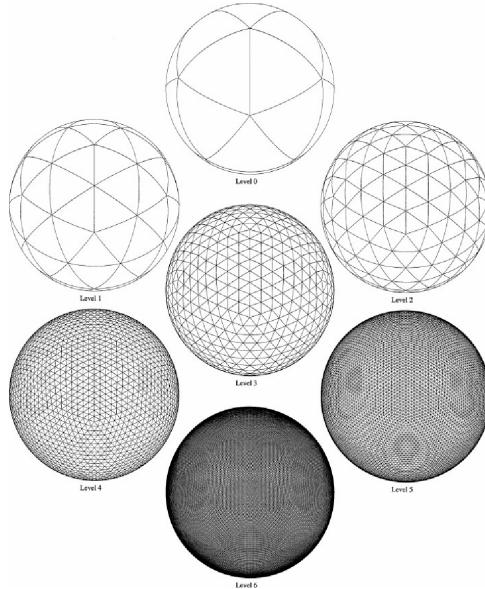


Figure 1: Icosahedral Mesh Types for different levels [1]

### Healpix grids

Healpix is a curvilinear partition grid type made up of equal area quadrilaterals. Unlike icosahedral grids, they are manifested as pixels, and the total number of pixels in a Healpix grid is calculated as follows:

$$P = 12 * S^2 \quad (2)$$

Here  $P$  is the number of pixels in the Healpix grid, and  $S$  is the grid sub-division or resolution parameter. Healpix is constructed in such a way that the areas of each pixel for a given sub-division is equal. Table 2 and Figure 2 show the number of pixels and the corresponding mesh manifestation of Healpix.

Table 2: Number of pixels generated in a Healpix for the given sub-division

Sub-division	Number of pixels in a healpix
1	12
2	48
4	192
8	768
16	3072
32	12288

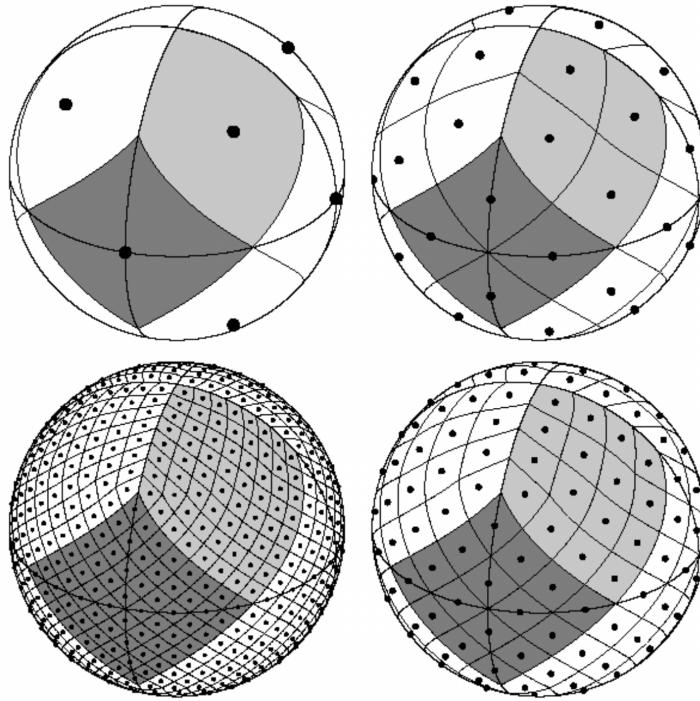


Figure 2: Healpix Mesh Types for different sub-divisions [2]

### Interpolation

We use the PyGSP library in Python to perform the grid transformation. It is commonly used for various graph operations to use in signal processing or social network analysis (e.g., Erdos-Reyni network). We first develop a 'backbone' structure of a spherical coordinate system (icosahedron, healpix, etc.). The properties of the spherical coordinates, such as levels or sub-divisions, are given as input. At this point, the coordinates are simply graph networks. In the next step, we assign latitude and longitude values to the graph network ( $x, y$ ) so that they can be manifested in a geographical coordinate system. Finally, we use the raw data from reanalysis or ESM output and perform bilinear interpolation to obtain the final spherically-sampled data. The data pipeline for the sampling module is shown in Figure 3.

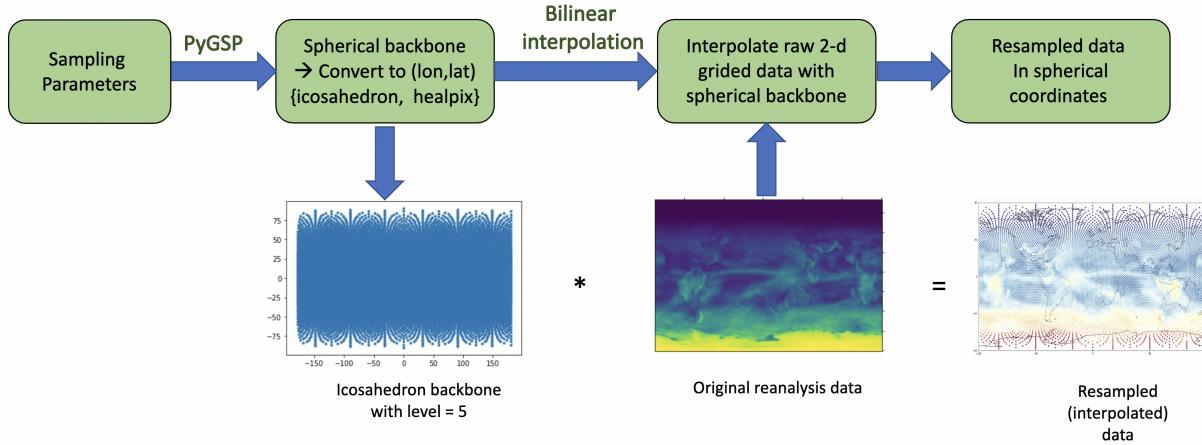


Figure 3: Sampling Module Pipeline

A fully interpolated dataset on a spherical grid is shown in Figure 4. The code that was used for generating the spherical sampling module can be found [here](#).

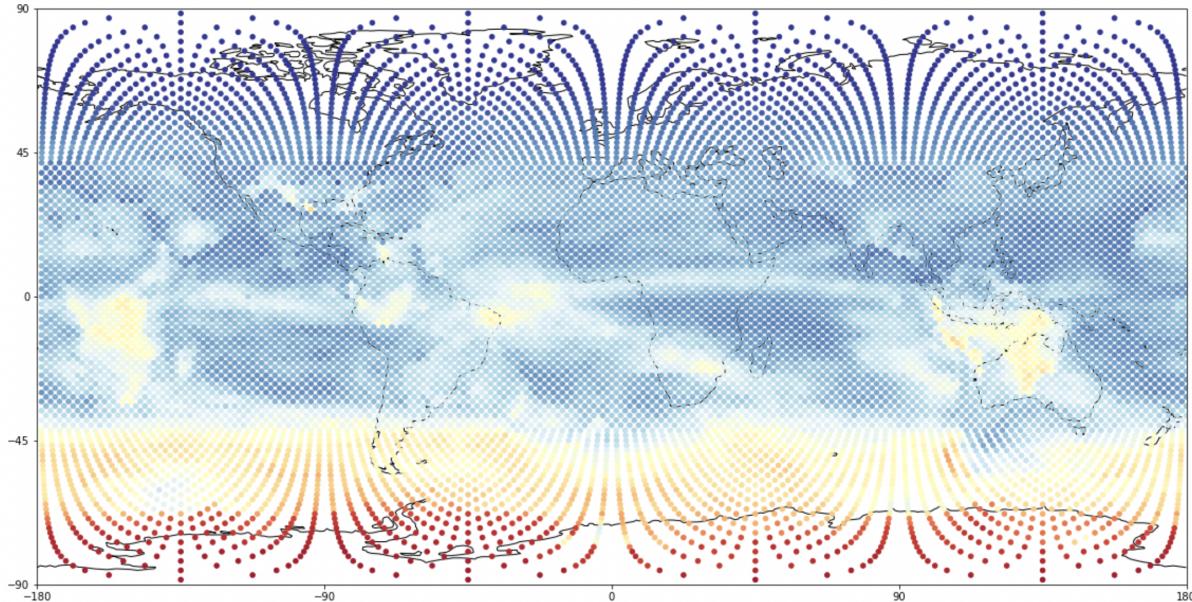


Figure 4: Interpolated Data on a Spherical Grid

### Model library module

We have created a library of models, ranging from regular Convolutional Neural Network (CNN), CNN-Autoencoder, Vanilla U-net to Spherical U-net architectures. This allows the modeler to explore the model performance and computational needs of different types of deep learning architectures. The sampling module is independent of the model library. There exists a function that combines the interpolated dataset obtained from the sampling module with the desired model architecture chosen by the modeler as shown in Figure 5. The ongoing development of this codebase can be found [here](#).

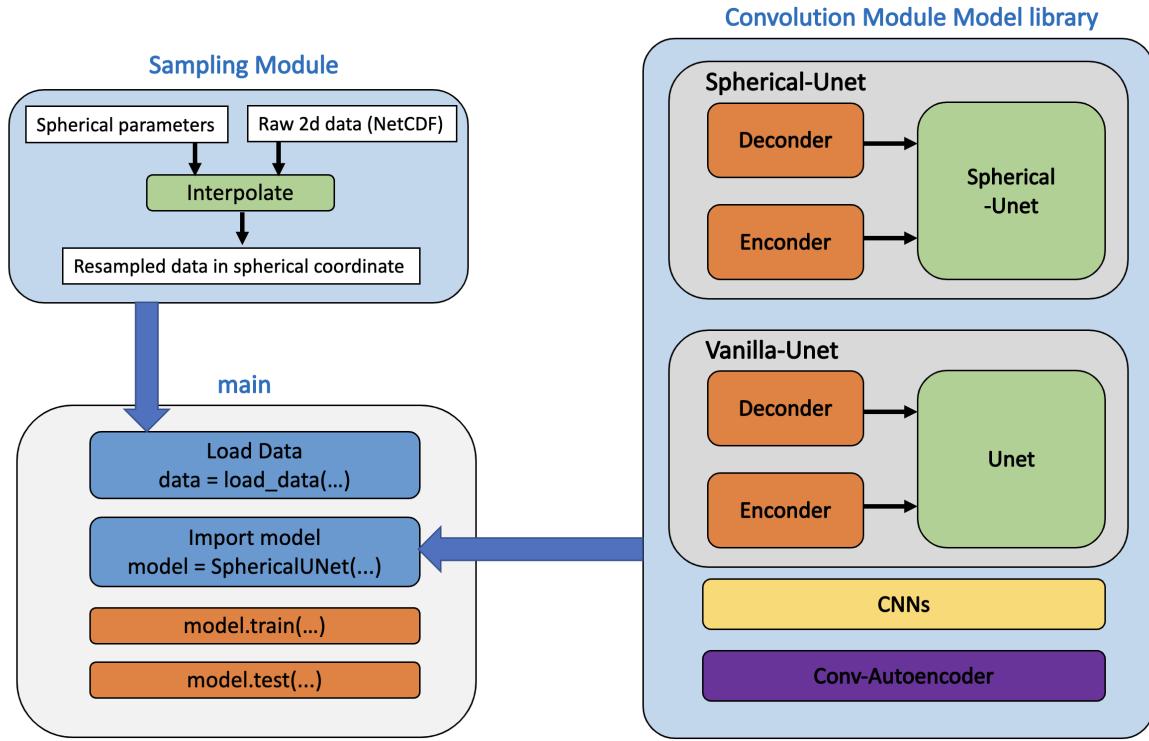


Figure 5: Spatial Data-Driven Component in AIBEDO

## Temporal Data-Driven Component

Our temporal data-driven component is a multi-timescale LSTM network consisting of two levels: yearly and monthly as shown in Figure 6.

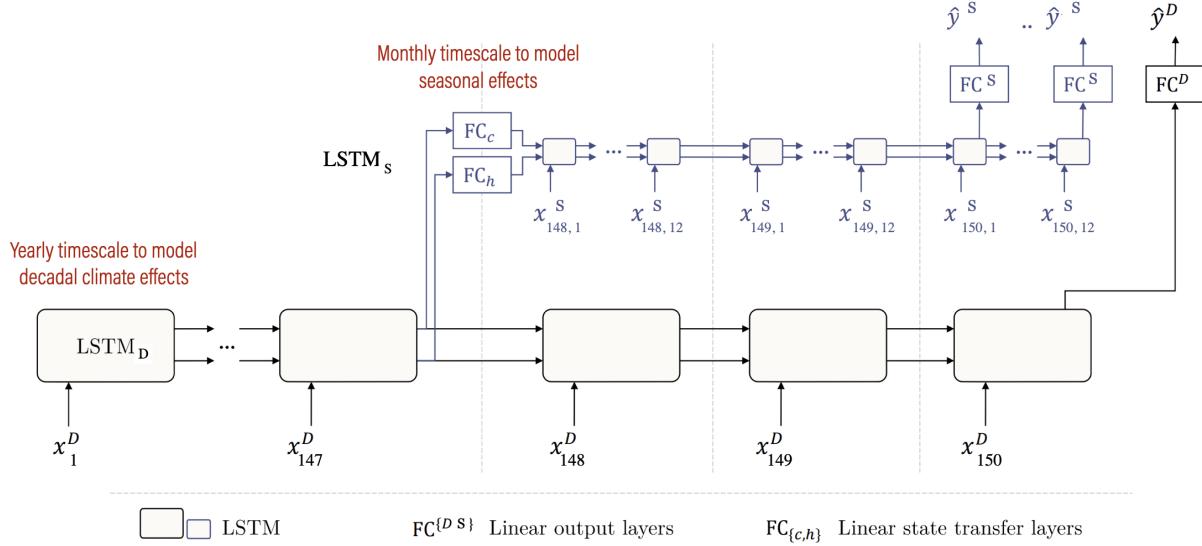


Figure 6: Multi-LSTM Architecture

We have developed two-stage LSTM model architecture to capture the multi-temporal nature of the model.

### Ongoing Work:

- We are exploring the nature of transfer layers between the two stages of LSTM models.
- We are investigating approaches to incorporate distributed computing such as dask to accelerate model training for both spatial and temporal networks.
- We are exploring the best approach to combine the spatial and temporal networks. We are investigating whether to train them separately or together, and what effect each approach would have on model prediction, computational efficiency, etc.

### Physics Constraints

The proposed physics constraints detailed in the Milestone 1 report are briefly reintroduced in this section. We proposed to constrain the energetics-based radiative and precipitation forcing budgets using the formulation provided by Roe et al. [3] and O’Gorman et al. [4], respectively shown in equations below:

$$R_{cl} = \lambda\Delta T - \nabla \cdot (\Delta F_{atm}) - \nabla \cdot (\Delta F_{ocn}) - \Delta OHU \quad (3)$$

$$L\delta P = \delta Q + \delta H \quad (4)$$

The model output from the spatial model (spherical U-net) will be used as an input to the temporal Multi-timescale LSTM model. The schematic in Figure 7 represents the loss function implementation overview. Since the energetics-based constraints are valid for longer timescales (larger than our shorter timescale LSTM model), we propose to apply these constraints on the longer timescale model (shown in blue blocks). Whereas for the shorter timescale model (shown in yellow blocks), we will only use standard loss functions such as pixel wise MSE. The difference between the shorter and longer timescales are shown in the following schematic by  $\delta$  (shorter) and  $\Delta$  (longer).

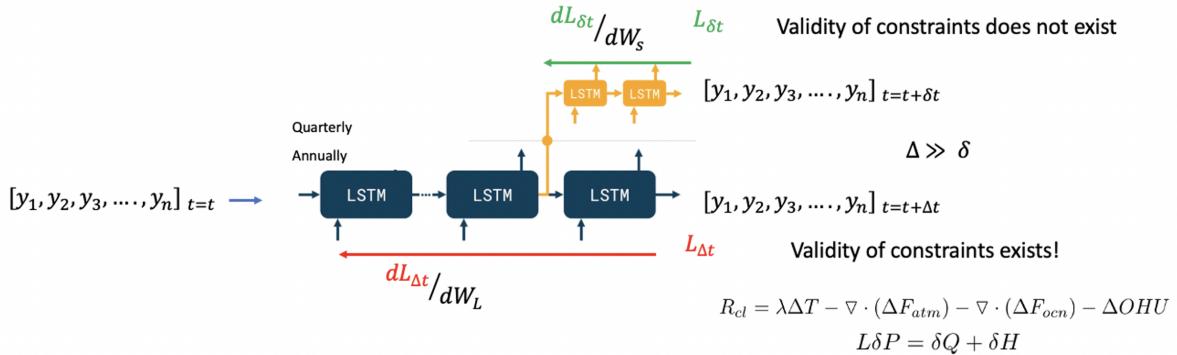


Figure 7: Physics Constraints Schematic

The other challenge in implementing this loss function is in taking spatial gradients for the variables such as  $F_{atm}$  and  $F_{ocn}$ . We propose using non-differentiable CNN layers (cite) to that effect. The 2D grid spatial information is available for a given data, and using that knowledge, we propose building a framework to calculate the various spatial derivatives needed for the calculation of the constraint. The schematic for the formulation is shown in Figure 8.

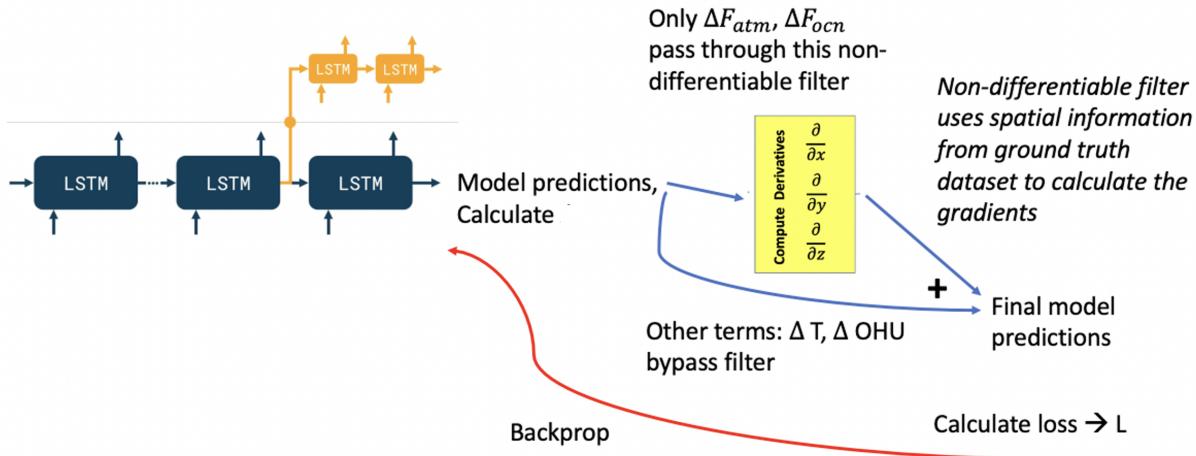


Figure 8: Spatial Gradient Physics Constraints

**Ongoing Work:** PARC team is closely working with the University of Victoria team to finalize the constraints for spatial and temporal networks that are valid for different temporal and spatial scales.

## Datasets

### Data access and compute

We have obtained an AWS HPC6a EC2 instance<sup>1</sup> that is custom-built to perform high-performance computing workloads. Our raw data consists of state-of-the-art Earth System Model (ESM) monthly output from the CMIP6 archive, and span the historical period (1850 to 2014) and SSP345 and SSP585 future projections (2015 to 2100). They are available on AWS<sup>2</sup>, and the datasets are transferred to the working HPC system on an as-needed basis through the pre-processing pipeline workflow.

### Data preprocessing

We follow three major preprocessing steps to prepare the data ready for model training.

1. **Remove seasonal cycle or “Deseasonalizing”:** We perform this process to remove any trends in the season to prepare a seasonal stationary time series data.
2. **Remove linear trend or “Detrend”:** We fit a cubic spline to remove any linear trend in data over time. This removes any systematic increase (such as temperature increase over time) and allows the modeler to focus on the natural variability.
3. **Normalized anomalies:** The anomaly at each grid point and is computed relative to a running mean, computed over a centered 30-year window for that grid point and month. Anomalies are normalized by dividing by the standard deviation of the anomaly over the same 30-year window for that grid point and month.

The following subsections list the input, output and physical constraint related data hyper-cube that will be used in model training.

<sup>1</sup><https://aws.amazon.com/ec2/instance-types/hpc6/>

<sup>2</sup><https://registry.opendata.aws/cmip6/>

### Input Data Hyper-cube

- Shortwave Cloud Radiative Effect (CRE; variables rsdt, rsut, rsutcs)
- Albedo (variables rsdt and rsut)
- Cloud liquid water path (variable clwvi)
- Cloud ice water path (variable clivi)

### Output Data Hyper-cube

- Surface temperature (variable tas)
- Precipitation (variable pr)
- Sea Level Pressure (variable psl)

### Data Hyper-cube for enforcement of Physical Constraints

- Net top-of-atmosphere fluxes (variables rsdt, rsut, rlut)
- Net surface fluxes and net dry surface fluxes (variables hfss, hfls, rsds, rsus, rlds, rlus)
- Ocean heat storage anomaly (variable thetao)
- Latent Heat transport (variables prt and evspbsl)
- Top-of-atmosphere impacts of radiative feedbacks (variable tas; radiative feedbacks computed using radiative kernels)

All processed data hyper-cubes (input, output, and constraints) are saved in netcdf format, a self-describing binary data format readily convertible to a Python numpy array. For a single ensemble member for a single CMIP6 ESM, these three data hyper-cubes are between 100 GB and 500 GB, depending on the resolution of the respective ESM.

#### Ongoing Work:

We are continuing to preprocess the CMIP6 and Reanalysis datasets. We are a little behind on our dataset preprocessing timeline due to unforeseen delays in obtaining AWS compute access. However, we are now catching up to work towards our goal of producing the entire prepared dataset deliverable in Milestone 3.

## References

- [1] G. R. Stuhne and W. R. Peltier, “New icosahedral grid-point discretization of the shallow water equation on the sphere,” 1999.
- [2] K. M. G’orski, B. D. Wandelt, E. Hivon, F. K. Hansen, and A. J. Banday, “The healpix primer,” *NASA JPL*, 2010.
- [3] G. H. Roe, N. Feldl, K. C. Armour, Y.-T. Hwang, and D. M. Frierson, “The remote impacts of climate feedbacks on regional climate predictability,” *Nature Geoscience*, vol. 8, no. 2, pp. 135–139, 2015.
- [4] P. A. O’Gorman, R. P. Allan, M. P. Byrne, and M. Previdi, “Energetic constraints on precipitation under climate change,” *Surveys in geophysics*, vol. 33, no. 3, pp. 585–608, 2012.