Data Science for Polar Ice Core Climate Reconstructions

Capstone Project Final Report

June 28, 2023

Authors

Daniel Cairns, Jakob Thoms, and Shirley Zhang

Master of Data Science Program, University of British Columbia

Mentor

G. Alexi Rodríguez-Arelis, PhD

Department of Statistics, University of British Columbia

Partner

Anais Orsi, PhD

Department of Earth, Ocean and Atmospheric Sciences, University of British Columbia

Contents

1. Executive Summary

The isotopic composition of ice cores (i.e. $\delta^{18}O$) is a proxy for understanding historical climate and its variability in Antarctica. Our project aims to model the relationship between $\delta^{18}O$ and three climate variables: temperature, precipitation, and geopotential height. We preprocess data simulated from a climate model, build Gaussian Process (GP) and Neural Network (NN) models to predict outcomes across space and time, and analyze the accuracy and precision of our models using residuals. Our GP and NN models outperformed baseline models, but warrant further improvement. Our project delivers a reproducible and well-documented workflow (in a GitHub repository) for future researchers to build upon in terms of parameter and architecture exploration, and refinement of preprocessing methods.

2. Introduction

2.1. Background

The earliest climate observations in Antarctica date back to 1958, when the first weather stations were set up (Bromwich et al. 2013). To characterize the climate before this time, scientists study water-stable isotopic records in ice cores dating back thousands of years (Stenni et al. 2017). One such measure is the isotopic composition of Oxygen in precipitation (expressed as $\delta^{18}O$, or delta Oxygen-18). $\delta^{18}O$ can act as a proxy to estimate key climate variables such as temperature, precipitation, and geopotential height (see Table 1).

$\delta^{18}O$ reflects a ratio of the heavy oxygen isotope ^{18}O to the light isotope ^{16}O in a sample of water (see Appendix for equation). Broadly speaking, warmer temperatures result in more ^{18}O in the ice cores, since the heavier ^{18}O isotopes require more energy than ^{16}O to evaporate (Mulvaney 2004). Precipitation processes also affect $\delta^{18}O$, as the heavier isotope preferentially precipitates before the lighter one. Finally, geopotential height is a measure of the height above the Earth's surface of a specific level in the atmosphere (Holton and Hakim 2013). Patterns in this climate variable affect the pathways of air masses, guides the travel paths of temperature and moisture (Noone and Simmonds 2002). This impacts the movement of precipitation, and hence $\delta^{18}O$ is affected (Holton and Hakim 2013).

Real ice core data is scarce, but $\delta^{18}O$ and other climate variables can be estimated uniformly, across Antarctica, over decades, and to high accuracy using climate models (Stevens et al. 2013). These models simulate natural processes with computer codes implementing complex mathematical equations. They can be extremely computationally intensive, requiring significant computing time and resources to run (Bastos and O'Hagan 2009). As a consequence, building surrogate models, or simplified models which use data from climate models to approximate them, becomes valuable. Surrogate models are faster to run and flexible, allowing researchers to model a wider range of scenarios (Bastos and O'Hagan 2009).

Previous research has build surrogate models using linear regression with ordinary least squares (OLS) to model the relationship between $\delta^{18}O$ and temperature (Stenni et al. 2017). Building upon this research, our project will use data science techniques on data obtained from the *IsoGSM* climate model (Yoshimura et al. 2008) to model the relationships between the isotopic composition of Oxygen in precipitation and other climate variables (outlined in Table 1 below).

Table 1. Definitions of key climate variables for this project.

Variable	Definition
$\delta^{18}O$	Delta Oxygen-18 in precipitation (‰)'
Temperature	Air temperature 2 metres above the surface (K)
Precipitation Rate	Rate of precipitation reaching the Earth's surface (mm/s)
Geopotential Height	A vertical coordinate relative to Earth's mean sea level at 500 milibars (1/2 the atmosphere) (m)

2.2. Research Question

The Data Science related-question guiding our project is as follows:

> *"Can we build surrogate models using simulated climate data which yield accurate and precise predictions of **temperature**, **geopotential height**, and **precipitation** across Antarctica?"*

2.3. Objectives

To address our research question, our project aimed to achieve the following three objectives:

1. Implement machine learning (ML) models:

   Using data from the IsoGSM climate model, implement Gaussian Process (GP) and neural network (NN) models to predict temperature, precipitation, and geopotential height (Y) from values of $\delta^{18}O$ (X) *(see Figure 1).*

2. Examine the predictions of the ML models in terms of accuracy and precision:

   Use heatmap visualizations to examine the models' predictive performance for different regions of Antarctica and for predicting different climate variables in terms of accuracy (using residuals) and precision (using the standard deviations of residuals).

3. Develop reproducible data products implementing our models for future researchers to build upon:

   Create a comprehensive and readable workflow notebook and GitHub repository with reproducible code, allowing others to examine our results and easily pick-up the project *where we left it.*

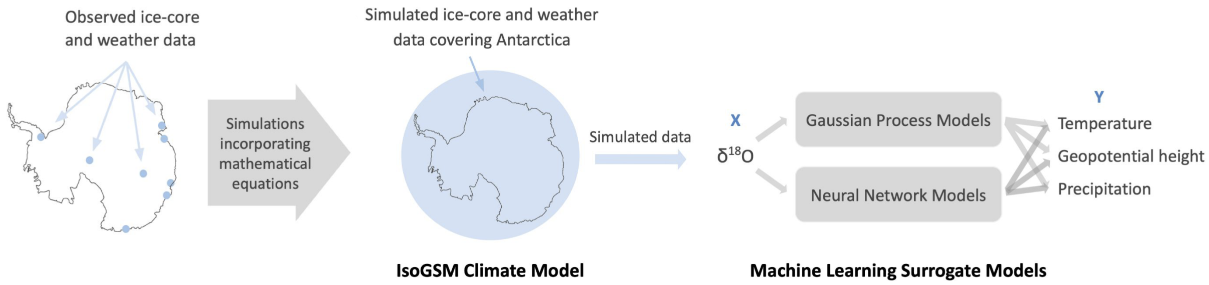

Figure 1: Observed data, climate models, and surrogate models. Real ice-core and weather data is collected sparsely across Antarctica. This data is combined with complex mathematical equations to create climate models, such as IsoGSM. Simulated data from these models are used as inputs and outputs to machine learning 'surrogate' models to find a relationship between $\delta^{18}O$ and climate variables.

3. Data Science Methods

The workflow of our project is as follows:

1. Data Preprocessing
2. Building Baseline Models
3. Building Gaussian Process Models
4. Building Neural Network Models
5. Results Post-processing

3.1. Data Preprocessing

Training, Validation, and Test Splits The full IsoGSM dataset (42 years) was firstly split into 3 smaller datasets:

1. Training: data from 1995 - 2020 inclusive (26 years). Used to train the models.
2. Validation: data from 1987 - 1994 inclusive (8 years). Used to continuously evaluate the different ML models.
3. Test: data from 1979 - 1986 inclusive (8 years). Used at the end of the project to get performance scores of the final ML models chosen.

Spatial Features Due to the variable geographic features of Antarctica, we decided to add the following non-temporal features to our datasets:

1. **Easting and Northing Coordinates**: Two features representing a projection of the latitude and longitude values to universal polar stereographic coordinates (UPS) with the true latitude scale set to 80°S. Distances calculated with these coordinates will be more accurate.
2. **Distance to Coast**: A measure of the cartesian distance to the nearest coast measured on the scale of the UPS coordinates. Values are positive if the point is over land, and negative if the point is over sea.
3. **Surface Orography**: The surface height above sea level of each point, in meters.
4. **Land / Sea Boolean Mask**: A boolean mask where 1 represents a point over land and 0 represents a point over sea.

Temporal Features $\delta^{18}O$, temperature, geopotential height, and precipitation are known to have consistent seasonal patterns. Removing the seasonality is important to help our models learn relationships different from these temporal patterns. To do this, we computed the monthly "anomalies", which are values of the variables after subtracting the monthly mean of the variable over all years in the training data (for a specific latitude and longitude point):

$$v_{lat,lon,month}^{anomaly} = v_{lat,lon,month} - \bar{v}_{lat,lon,month} \quad (1)$$

Where

- $v_{lat,lon,month}$: one of the spatial-temporal variables $\delta^{18}O$, temperature, geopotential height, and precipication for a given latitude, longitude, and month
- $\bar{v}_{lat,lon,month}$: the mean of the variable for a specific latitude, longitude, and month over all years

Note that the spatial features added do not vary with time, and thus did not need to be deseasonalized.

Scaling Finally, as our variables differ in units and scale, it was necessary to standardize all values (except the land/sea boolean mask) to ensure our models treated features equally during training:

$$v_{lat,lon,month}^{scaled} = \frac{v_{lat,lon,month} - \bar{v}}{\hat{\sigma}} \quad (2)$$

Where

- $v_{lat,lon,month}$: one of the variables $\delta^{18}O$, temperature, geopotential height, precipication, easting/northing, distance to coast, and surface orography
- $\bar{v}$: overall mean of one of the variables to be scaled (regardless of latitude, longitude, or month)
- $\hat{\sigma}$: overall standard deviation of one of the variables to be scaled (regardless of latitude, longitude, or month)

3.2. Baseline Models

As a benchmark, we took inspiration from previous research (Stenni et al. 2017) to build a simple OLS regression baseline model. We related the $\delta^{18}O$ scaled anomalies to the scaled anomalies of our three target variables linearly as follows:

$$\hat{y}_{anomaly,scaled_i} = \beta_{0_i} + \beta_{1_i} x_{anomaly,scaled} \quad (3)$$

Where - i: one of the target variables temperature, geopotential height, and precipication

- $x_{anomaly,scaled}$: the variable $\delta^{18}O$
- $\hat{y}_{anomaly,scaled}$: the predicted target variable
- β_{0_i}: intercept for the corresponding i target variable
- β_{1_i}: slope for the corresponding i target variable

Our goal moving forwards was to build more complex GP and NN models which would outperform these simple linear models.

3.3. Gaussian Process Models

Mathematical Background of Gaussian Processes

Gaussian Process (GP) models were first introduced by Sacks et al. (1989). A GP model extends the ordinary least squares (OLS) regression model in a way that offers more model flexibility by accommodating non-linear relationships and capturing the correlation structure of the data.

Assuming that we have a training data set with n observations, let y_i denote the ith observed value of a specific climate variable (e.g. temperature or precipitation), and let

$$\mathbf{x}_i = \begin{pmatrix} x_{i,1} \; x_{i,2} \; x_{i,3} \; x_{i,4} \; x_{i,5} \end{pmatrix}^{\mathsf{T}} = \begin{pmatrix} \delta^{18}\mathrm{O}_i \; \mathrm{E}_{\mathrm{UPS},i} \; \mathrm{N}_{\mathrm{UPS,\,i}} \; \mathrm{oro}_i \; d_{\mathrm{coast},i} \end{pmatrix}^{\mathsf{T}} \qquad (4)$$

denote the corresponding observed values of the ML model's input variables (as prepared in the preprocessed dataset).

Recall that an OLS regression model relates the input variables in $\mathbf{x}$ to an output variable y via a linear equation:

$$y_i = \beta_0 + \beta^{\mathsf{T}}\mathbf{x}_i + \varepsilon_i \qquad (5)$$

In (5), β_0 is a scalar, $\beta = \begin{pmatrix} \beta_1 \; \beta_2 \; \beta_3 \; \beta_4 \; \beta_5 \end{pmatrix}^{\mathsf{T}}$ is a vector of scalar coefficients, and ε_i is a random variable. The parameter β_0 represents the linear model's intercept, and the parameters in β represent the linear model's slope with respect to each of the model's input variables. Further, the random variable ε_i represents the model's error-term. This term is assumed to be independently identically distributed as $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ for all $i = 1, \ldots, n$, where σ^2 is the common variance of the error-terms.

The assumption of a linear relationship between input and output variables is quite restrictive in the sense that it does not allow for modelling complex non-linear relationships. Further, the independence assumption of OLS does not hold for climate data due to the spatial-temporal nature of such data. As such, we believe that better model performance† can be achieved by utilizing models that can accommodate both non-linearity and non-independence (†performance as measured by various metrics; see the postprocessing section for more details).

One such model is a GP model, which relates the input variables in $\mathbf{x}$ to an output variable y via an equation quite similar to (5) from OLS:

$$y_i = \beta_0 + \beta^\intercal \mathbf{x}_i + Z(\mathbf{x}_i) \qquad (6)$$

In (6), β_0 and β are the same as they were in (5) (i.e. intercept and slopes). The random variable ϵ_i from (5) has been replaced by a different random variable $Z(\mathbf{x}_i)$, and, as before, this stochastic term represents the model's error-term. However, this error-term $Z(\mathbf{x}_i)$ is **not** assumed to be independent for all $i = 1, \ldots, n$. Instead, a GP model assumes that the error-terms follow an identical *marginal* distribution $Z(\mathbf{x}_i) \overset{\text{marginal}}{\sim} \mathcal{N}(0, \sigma^2)$ for all $i = 1, \ldots, n$, where σ^2 is the common variance of the error-terms (also known as the 'overall process variance').

The key distinction between OLS models and GP models lies in their correlation structure (Rasmussen and Williams 2006). While the OLS model assumes no correlation between error terms ($\text{Cor}(\varepsilon_i, \varepsilon_j) = 0$ for $i \neq j$), a GP model assumes that observations have a correlation structure which is determined by a correlation function $R(.\ ,\ .)$:

$$\text{Cor}(Z(\mathbf{x}_i), Z(\mathbf{x}_j)) = R(\mathbf{x}_i, \mathbf{x}_j) \in [0, 1] \qquad (7)$$

The correlation function (7) is often decomposed into a product of kernel functions $\mathcal{K}(.\ ,\ .)$. Recall that we have 5 input features, as specified in (4). Thus, we have:

$$\begin{aligned} R(\mathbf{x}_i, \mathbf{x}_j) &= \prod_{k=1}^{5} \mathcal{K}_k(x_{i,k},\ x_{j,k}) & (8) \\ &= \mathcal{K}_1(\delta^{18}\mathrm{O}_i,\ \delta^{18}\mathrm{O}_j) & \\ &\quad \times \mathcal{K}_2(\mathrm{E}_{\mathrm{UPS},i}\, \mathrm{E}_{\mathrm{UPS},j}) \times \mathcal{K}_3(\mathrm{N}_{\mathrm{UPS,\ i}}\, \mathrm{N}_{\mathrm{UPS,\ j}}) & \\ &\quad \times \mathcal{K}_4(\mathrm{oro}_i, \mathrm{oro}_j) \times \mathcal{K}_5(\ d_{\mathrm{coast},i},\ d_{\mathrm{coast},j}) & (9) \end{aligned}$$

A kernel function $\mathcal{K}$ models the correlation between observations as a function of their distance in the input feature space. In supervised machine learning the notion of similarity between data points is crucial; it is a basic assumption that points with input features x which are close are likely to have similar target values y, and thus training points that are near to a test point should be informative about the prediction at that point. In a GP model it is the kernel functions $\mathcal{K}$ that defines closeness or similarity (Rasmussen and Williams 2006).

In addition to allowing GP models to capture correlation between variables, the use of kernel functions also allows GP models to model non-linear relationships. The GP model equation (6) includes a linear regression component, but kernel functions can be (and usually are) non-linear.

There are many kernel functions available in the literature (Duvenaud 2014). For example, there are kernel functions which are able to directly model seasonal temporal correlation structures (Roberts et al. 2013). The kernel function is the crucial ingredient in a GP model since it encodes our assumptions about the correlation structures of the data (Rasmussen and Williams 2006). Thus, we dedicated a large amount of effort to selecting the most appropriate kernel functions.

For our project, we primarily used Radial Basis Functions (RBFs) as our kernel function:

$$\mathcal{K}_{\mathrm{RBF}}(x_i,\ x_j) = \exp\left(-\frac{1}{2}\frac{|x_i - x_j|}{\theta^2}\right) \tag{10}$$

This function assigns values near to 1 for variables with similar corresponding inputs and decreases the covariance exponentially as the distance between inputs increases. It includes a learned parameter θ which controls the rate of exponential decay for the covariance with respect to the input feature's distance. In the context of this project, we can combine equations (7) through (10) to see that our GP model's correlation function was given by:

$$\begin{aligned}
\mathrm{Cor}(Z(\mathbf{x}_i), Z(\mathbf{x}_j)) &= \prod_{k=1}^{5} \mathcal{K}_{\mathrm{RBF}}(x_{i,k},\ x_{j,k}) \qquad (11)\\
&= \exp\left(-\frac{1}{2}\frac{|\delta^{18}\mathrm{O}_i,\ \delta^{18}\mathrm{O}_j)|}{\theta_1^2}\right)\\
&\quad \times \exp\left(-\frac{1}{2}\frac{|\mathrm{E}_{\mathrm{UPS},i}\ \mathrm{E}_{\mathrm{UPS},j}|}{\theta_2^2}\right) \times \exp\left(-\frac{1}{2}\frac{|\mathrm{N}_{\mathrm{UPS},i}\ \mathrm{N}_{\mathrm{UPS},j}|}{\theta_3^2}\right)\\
&\quad \times \exp\left(-\frac{1}{2}\frac{|\mathrm{oro}_i, \mathrm{oro}_j|}{\theta_4^2}\right) \times \exp\left(-\frac{1}{2}\frac{|d_{\mathrm{coast},i},\ d_{\mathrm{coast},j}|}{\theta_5^2}\right) \qquad (12)
\end{aligned}$$

Importantly, notice that in (12) there is a distinct parameter θ_k for each of the 5 input features.

The GP models offer a powerful framework for capturing non-linear relationships and accounting for the correlation structure present in the data. By leveraging the flexibility of the kernel function, GP models have the potential to provide improved predictions of temperature, geopotential height, and precipitation compared to linear regression models like OLS.

Implementation of GP Models

Training GP models involve the computation of pairwise distances between all training data points in a matrix, which requires large memory and time resources. To overcome this challenge, we obtained computational resources from UBC ARC Sockeye, which allowed us to use computational nodes of up to 186 GB. We also reduced our training dataset of 26 years and ~780,000 examples into smaller splits of a few consecutive years each. Table 2 displays a summary of notable configurations run, varying the type of kernel, learning rate, number of splits, and validation RMSE scores on the three climate variables. It also displays the memory and runtime utilized on Sockeye when training the models.

The three kernels used were "RBFKernel" (kernel 1), "PiecewisePolynomialKernel" (kernel 2), and "RQKernel" (kernel 3). All three displayed similar performances when trained on ~60,000 examples. However, we noticed that kernel 1 required less memory resources (72 GB compared to >100 GB), which would allow us to include more examples during training. Decreasing the learning rate (from 0.0015 to 0.00075) also did not appear to significantly improve RMSE scores, but did increase runtime. Thus, we decided to move forwards with a GP model with kernel 1, a learning rate of 0.0015, and 9 splits (second last row).

Table 2. GP Model Details and Validation RMSEs on Scaled Anomalies. Different kernels, learning rates, number of epochs, and splits were experimented with when training GP models on Sockeye. RMSE scores correspond to the first split (of either 13 or 9) of the training data. The row with bolded cells corresponds to the final GP model parameters chosen.

Kernel	Learning Rate	Num. Epochs	Num. Splits	Num. Examples	Temp. RMSE	Geopt. RMSE	Precip. RMSE	Memory (GB)	Runtime (h)
1	0.15	10	13	~60,000	1.14	1.18	1.08	72	3-4
2	0.15	10	13	~60,000	1.11	1.15	1.14	155	2-3
3	0.15	10	13	~60,000	1.13	1.16	1.14	132	1-2
1	**0.0015**	**10**	**9**	**~87,000**	**0.99**	**1.03**	**1.00**	**150**	**2-3**

Kernel	Learning Rate	Num. Epochs	Num. Splits	Num. Examples	Temp. RMSE	Geopt. RMSE	Precip. RMSE	Memory (GB)	Runtime (h)
1	0.00075	10	9	~87,000	0.98	1.03	1.00	150	4-5

3.4. Neural Network Models

In parallel, we also developed deep neural network (NN) models to predict the target climate variables. We chose to pursue NN models because they can learn complex, non-linear relationships in flexible ways, should be capable of handling the the inherent space and time dependencies of our data, and could generate multiple outputs (in this case temperature, precipitation, and geopotential height) from a single trained model. These models are deep because they contain a series of layers, where the shape and style of each layer and their order affect model learning.

We experimented with several different architectures to see which was more effective at learning our data. These architectures are all combinations of convolutional and/or fully connected (labelled "linear") layers. Convolutional layers are designed for spatial pattern recognition in images, and model relationships using only the data nearest to each point along some dimensions (latitude and longitude in our case) (O'Shea and Nash 2015). "Linear", fully-connected layers use a "brute-force" approach where the model is not given any prior context about the data (such as which points are close to each other) before training. These layers can capture a wider range of relationships, but might struggle because they are not directed where to look.

Table 3 includes the validation RMSE results of a selected set of neural network architectures that we tried. It's important to note that NN training depends on the random seed used to initialize all the weights in each model. Repeated training of the same architecture with a different seed may yield different training behaviors (such as faster or slower convergence to a stable RMSE) and different final RMSE scores.

Table 3. Neural network architecture performance on the validation set. Overall RMSE (e.g., performance on all 3 outputs together), as well as RMSE scores specific to temperature, geopotential height, and precipitation are included. Each architecture was trained until there was no improvement in overall validation RMSE for `stall limit` consecutive epochs. Total runtime in seconds (on the same PC) and total number of epochs trained included. All architectures predicted the 3 targets using the 6 input variables specified in *Preprocessing*.

Architecture	Overall RMSE	Temp. RMSE	Geopt. RMSE	Precip. RMSE	Stall Lim.	Runtime (s)	Num. Epochs
`CNN-deep2`	0.94	0.93	0.90	0.99	15	2875	211
`CNN-deep`	0.97	0.93	0.93	1.04	10	1484	72

Architecture	Overall RMSE	Temp. RMSE	Geopt. RMSE	Precip. RMSE	Stall Lim.	Runtime (s)	Num. Epochs
`CNN-simple`	0.98	0.94	0.95	1.04	10	292	91
`Linear-narrow`	0.99	1.01	0.86	1.09	10	9	108
`Linear-deep`	1.00	1.02	0.84	1.11	10	64	65
`Hybrid`	1.05	1.05	0.99	1.11	10	58	55

The architecture `CNN-deep2` had the best overall performance on the validation data and so we chose it as our preferred neural network model. This architecture consists of 6 convolutional layers with shrinking kernel sizes. In each layer, the kernel, or "sliding window", passes across the 2-dimensional latitude, longitude axes, learning a fixed set of weights which map x input channels to y output channels (see Table 4). Between each layer, we use rectified linear unit (ReLU) activation functions, which have strong performance on deep NNs (Glorot, Bordes, and Bengio 2011).

Table 4. Layer specifications of the `CNN-deep2` neural network architecture. Each layer maps x input channels to y output channels using a fixed kernel of size (latitude x longitude).

Layer	Input Channels	Output Channels	Kernel Size
1	6	32	5 x 17
2	32	32	5 x 15
3	32	16	5 x 13
4	16	16	3 x 13
5	16	8	3 x 11
6	8	3	3 x 9

3.5. Postprocessing

We evaluated each model's prediction performance for each target variable using RMSE scores on the test dataset. To make fair comparisons between targets on different scales, we calculate RMSE on the *scaled anomaly* data using formula 4, where n is the total number of predictions, i is the *ith* data point, $y_i^{\text{(anomaly, scaled)}}$ is the true target value for i and $\hat{y}_i^{\text{(anomaly, scaled)}}$ is the model prediction for i.

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(y_i^{\text{(anomaly, scaled)}} - \hat{y}_i^{\text{(anomaly, scaled)}})^2}{n}} \tag{4}$$

Residual Analysis We also analyzed each model's residuals using a set of 4 plots. The benefit of this type of analysis is it helps identify whether there remains information in the data that a better model could capture, or whether the errors are truly random. It also helps identify whether our models are biased in a consistent way, and whether there are subsets of data that the model appears to perform better on that are worth exploring in isolation.

Any patterns in the residuals identify shortcomings of the model, and the best model should have minimal, consistent residuals which resemble random noise. Specifically, we looked for spatial patterns in the residuals (whether the model's errors varied over space), temporal patterns in the residuals (whether the model failed to capture some seasonality), and at the overall distribution of residuals (whether they are unbiased and normal looking). See *Results* for our residual plots and discussions.

4. Results

4.1. Selected RMSE Scores

Table 5 details our RMSE results on the test dataset, which was reserved to evaluate the results after model selection. Our best GP models (these are 3 separate models) outperformed the baseline OLS model for all three climate variables. Our best NN model performs better than both GP and baseline models for temperature and geopotential height, but not precipitation. Notably, all three models have very similar scores predicting precipitation; the advanced models failed to improve on the baseline.

Table 5. RMSE Results on Test Set The OLS baseline and best performing GP and NN models were evaluated on the test set to predict the scaled anomalies of three climate variables. Residual plots are included for values in bold.

Model	Temperature	Geopotential Height	Precipitation
Baseline	1.11	1.11	1.07
GP	**1.03**	1.04	**1.04**
NN	**0.96**	**0.96**	1.07

4.2. Selected Residual Diagnostics Plots

Figure 2 shows the residual plots for the NN model's performance predicting geopotential height. Figure 3 shows the same plots for GP model's performance predicting precipitation. Finally, Figure 4 and Figure 5 show the GP and the NN model performance predicting temperature, allowing us to examine their differences.

4.2.1. NN Geopotential Height

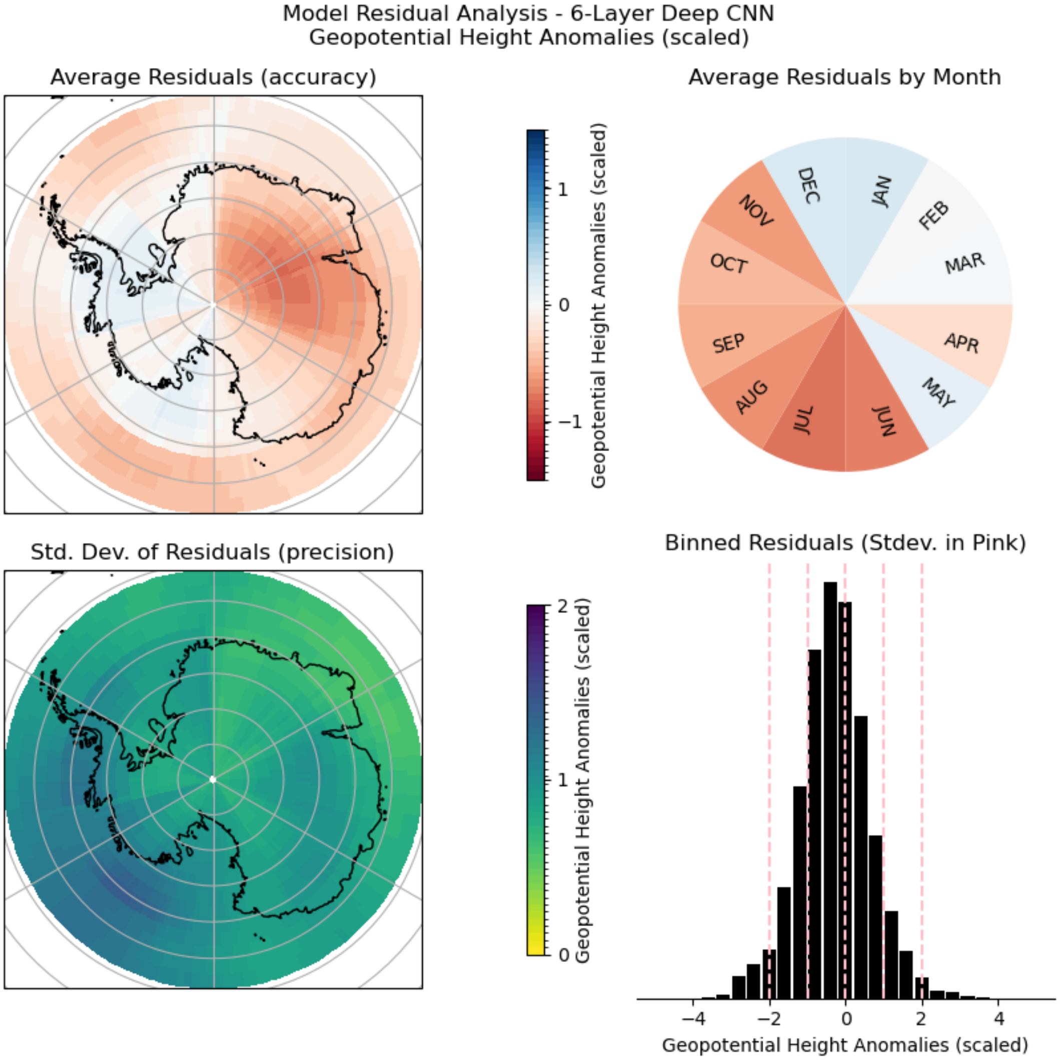

Figure 2: Residual plots of NN model predictions of geopotential height. The presence of majority red hues in the average residual plots and left-skew of the binned residuals reveal that the model tends to overpredict geopotential height. There are spatial and temporal patterns to this bias, both in the north-east areas of the continent on the map and in the distinct Antarctic spring period between June and November. The standard deviation of residuals appears fairly uniform across the map, but there are some darker blues in the south west quadrant, suggesting the model is less consistent there.

4.2.2. GP Precipitation

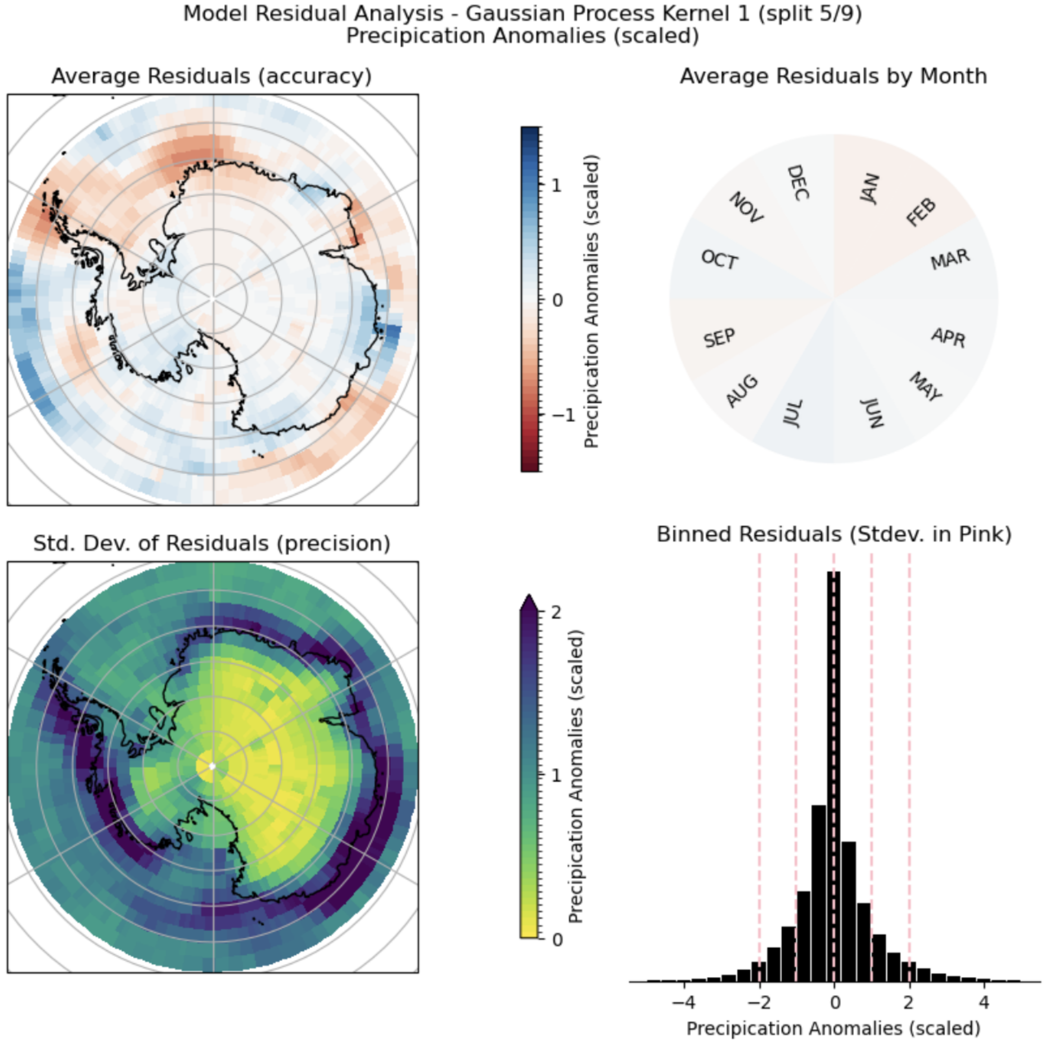

Figure 3: Residual plots of GP model predictions of precipitation. The top two plots show an unbiased model whose average residuals are near zero almost everywhere. There is also no seasonal pattern to the residuals, and the binned residuals confirm strong central tendency with a very large spike at 0. The standard deviation of residuals map shows the model's weakness: the predictions are erratic and very extreme in all coastal regions of Antarctica. This is where most Antarctic precipitation occurs (Souverijns 2019), so a lack of precision here is problematic.

4.2.3. GP Temperature

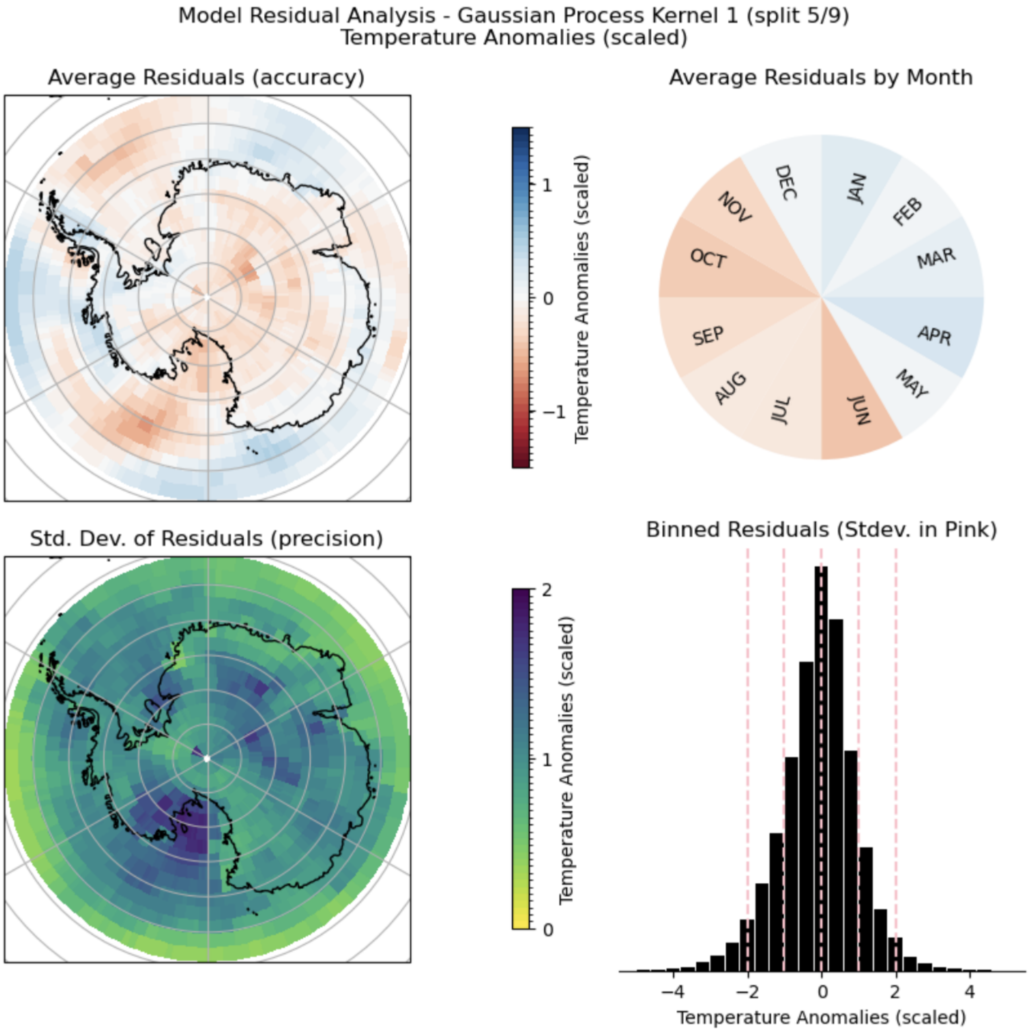

Figure 4: Residual plots of GP model predictions of temperature. Similar to precipitation, the top two plots appear to show relatively low average residuals across space and time. From the spatial plot, there appears to be a pattern of more overprediction (red hues) in the center of the continent, and underprediction (blue hues) around the South East and West coasts. Furthermore, there appears to be some seasonality in that June to November experiences more overprediction, whereas December to May experiences more underprediction. Finally, the binned residuals plot appears symmetrical, suggesting that there is roughly equal over and underprediction.

4.2.4. NN Temperature

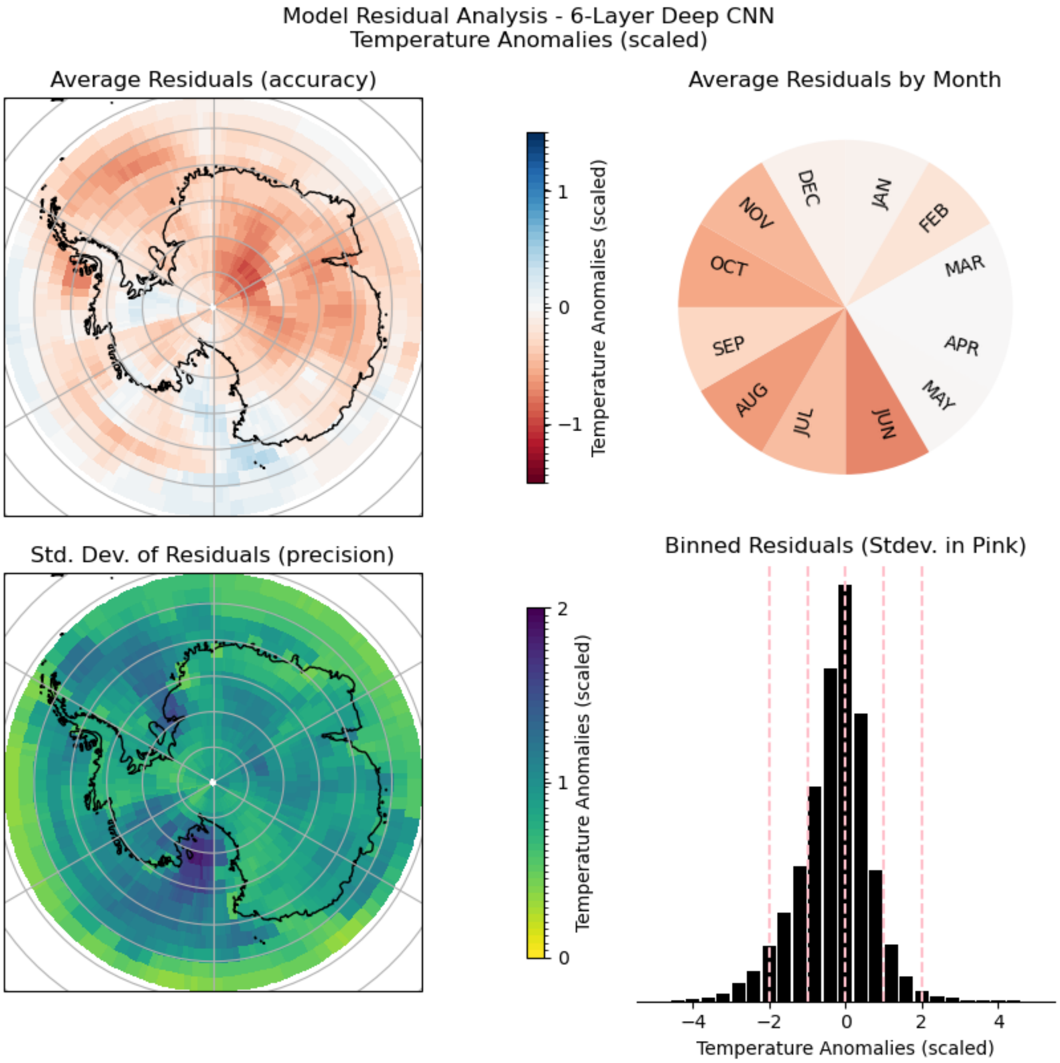

Figure 5: Residual plots of the neural network predictions of temperature. Similar to its predictions on geopotential height, the model is biased towards slightly overpredicting temperature, especially during the Antarctic spring period between June and November, and to a lesser extent in the north and west areas of the map. The standard deviation of residuals show more precision over land than over the ocean. Some sub-regions, such as the western peninsula, have average residuals close to zero (white) and low standard deviations (yellow-green), suggesting that the model has higher skill predicting temperature there.

5. Discussion

The NN model had the best overall performance of the models we trained, and both the NN and GP models beat the baseline on the test data (with the exception of precipitation). This is promising because the NN models are significantly less computationally intensive to train, and thus easier to use and more flexible to experiment and iterate with. The RMSE scores remain higher than we would hope, but are understandable in part because the models are predicting climate anomalies. Learning information on top of the seasonal cycle is a harder task because the bulk of the variation in temperature, for example, can be explained by time of year.

Both models struggled to explain precipitation. We can see from Figure 3 that even though our models show high accuracy, no seasonal patterns, and no bias, the model scores poorly because the standard deviations of the residuals are extremely high in the coastal regions of Antarctica. Here, the model produces both extreme over and under predictions with approximately equal frequency. Our explanation is that the temporal precipitation variance is not constant over space (Souverijns 2019). We must take this into account when scaling the data or else we will get uncharacteristically good-looking predictions of low variance areas (the interior) and also extreme prediction errors in high variance areas (the coast), neither of which we want.

Contrasting the GP and NN model residuals plots of temperature (Figure 4 and Figure 5) reveals how these models learn differently. The RMSE scores from both models were quite close (1.03 vs. 0.96), but the residual plots are extremely different. The NN model is noticeably red in the average residual plots and shows the bias towards over prediction that seems to be characteristic of this model. The GP model, by contrast, does not show much bias in the average residual plots and its residual histogram looks balanced. GP model scores worse because the predictions are less precise - the standard deviation of residuals plot is slightly darker in all regions. This is an interesting case of the bias-variance trade-off that we were not expecting to find (see Figure 4 and Figure 5).

Across all models, we noticed a drop in prediction accuracy between the validation and test sets. For example, the RMSE score of the NN model predicting temperature falls from 0.92 on validation to 0.96 on test. This means there are somewhat significant differences between these two data splits, which is a consequence of splitting the data by contiguous years. This raises concerns about how generalizable our models are, and whether we can be certain they are strong beyond the 1995 to 2020 period covered by the training set.

6. Data Product

We produced a well documented GitHub repository with reproducible code which does the following:

1. Preprocess the data
2. Train baseline, GP, and NN models
3. Evaluate trained models' performance

Inside the repository are the following detailed Jupyter Notebooks to guide future users:

1. Data preprocessing - demonstrating how the preprocessed training, validation, and test data was created
2. Building and training baseline models
3. Building and training GP models
4. Building and training NN models
5. Results post-processing - demonstrating how to generate RMSE scores and plots to evaluate the models' predictions

7. Conclusions and Recommendations

7.1. Conclusions

Both the GP and NN models beat the baseline OLS model. Thus, we consider that further work on this project is warranted.

7.2.Recommended Next Steps

1. Adjust the scaling procedure to incorporate local scaling to hopefully see better model performance on precipitation.
2. Develop sequential and/or ensemble models that use the models we developed as inputs so we can leverage the strenghts of both models.
3. Generalize the workflow so it can be applied to additional climate model data. Climate models worth investigating include:
    1. The ECHAM5-wiso climate model (Werner 2019)
    2. The LMDZ climate model (Hourdin et al. 2020)
4. Once generalized, compare the results for different climate models to explore:

1. Whether the models we trained on the IsoGSM data set generalize well when applied to other climate models, and
2. Whether models trained on data from different climate models have similar performance and learn similar patterns.

8. Appendix

This appendix provides additional equations which aim to help provide a better understanding of the background of our project.

8.1. Equations

Calculation of $\delta^{18}O$

$$\delta^{18}O = \left(\frac{(^{18}O/^{16}O)_{sample}}{(^{18}O/^{16}O)_{VSMOW}} - 1 \right) \times 1000\,‰ \qquad (3)$$

Where $(^{18}O/^{16}O)_{sample}$ is the ratio of the heavy to light isotope in a sample, and $(^{18}O/^{16}O)_{VSMOW}$ is the ratio in the Vienna Standard Mean Ocean Water (Wet, West, and Harris 2020; Stenni et al. 2017).

9. References

Bastos, Leonardo S., and Anthony O'Hagan. 2009. "Diagnostics for Gaussian Process Emulators." *Technometrics* 51 (4): 425–38. https://doi.org/10.1198/TECH.2009.08019.

Bromwich, David H, Julien P Nicolas, Andrew J Monaghan, Matthew A Lazzara, Linda M Keller, George A Weidner, and Aaron B Wilson. 2013. "Central West Antarctica Among the Most Rapidly Warming Regions on Earth." *Nature Geoscience* 6 (2): 139–45.

Duvenaud, David. 2014. "Automatic Model Construction with Gaussian Processes." PhD thesis.

Glorot, Xavier, Antoine Bordes, and Yoshua Bengio. 2011. "Deep Sparse Rectifier Neural Networks." In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, edited by Geoffrey Gordon, David Dunson, and Miroslav Dudík, 15:315–23. Proceedings of Machine Learning Research. Fort Lauderdale, FL, USA: PMLR. https://proceedings.mlr.press/v15/glorot11a.html.

Holton, James R., and Gregory J. Hakim. 2013. "Chapter 13 - Numerical Modeling and Prediction." In *An Introduction to Dynamic Meteorology (Fifth Edition)*, edited by James R. Holton and Gregory J. Hakim, Fifth Edition, 453–90. Boston: Academic Press. https://doi.org/https://doi.org/10.1016/B978-0-12-384866-6.00013-1.

Hourdin, Frédéric, Catherine Rio, Jean-Yves Grandpeix, Jean-Baptiste Madeleine, Frédérique Cheruy, Nicolas Rochetin, Arnaud Jam, et al. 2020. "LMDZ6A: The Atmospheric Component of the IPSL Climate Model with Improved and Better Tuned Physics." *Journal of Advances in Modeling Earth Systems* 12 (7): e2019MS001892.

Mulvaney, Robert. 2004. "How Are Past Temperatures Determined from an Ice Core?" *Scientific American.*

Noone, David, and I. Simmonds. 2002. "Associations Between Delta-18O of Water and Climate Parameters in a Simulation of Atmospheric Circulation for 1979–95." *Journal of Climate* 15 (22): 3150–69. https://doi.org/https://doi.org/10.1175/1520-0442(2002)015%3C3150:ABOOWA%3E2.0.CO;2.

O'Shea, Keiron, and Ryan Nash. 2015. "An Introduction to Convolutional Neural Networks." *CoRR* abs/1511.08458. http://arxiv.org/abs/1511.08458.

Rasmussen, Carl Edward, and Christopher K. I. Williams. 2006. *Gaussian Processes for Machine Learning.* Cambridge, MA: The MIT Press.

Roberts, S., M. Osborne, M. Ebden, S. Reece, N. Gibson, and S. Aigrain. 2013. "Gaussian Processes for Time-Series Modelling." *Philosophical Transactions of the Royal Society A: Mathematical,* *Physical and Engineering Sciences* 371 (1984): 20110550. https://doi.org/10.1098/rsta.2011.0550.

Sacks, Jerome, William J. Welch, Toby J. Mitchell, and Henry P. Wynn. 1989. "Design and Analysis of Computer Experiments." *Statistical Science* 4 (4): 409–23. https://doi.org/10.1214/ss/1177012413.

Souverijns, Niels. 2019. "Precipitation and Clouds over Antarctica from an Observational and Modelling Perspective." PhD thesis.

Stenni, Barbara, Mark AJ Curran, Nerilie J Abram, Anais Orsi, Sentia Goursaud, Valerie Masson-Delmotte, Raphael Neukom, et al. 2017. "Antarctic Climate Variability on Regional and Continental Scales over the Last 2000 Years." *Climate of the Past* 13 (11): 1609–34.

Stevens, Bjorn, Marco Giorgetta, Monika Esch, Thorsten Mauritsen, Traute Crueger, Sebastian Rast, Marc Salzmann, et al. 2013. "Atmospheric Component of the MPI-m Earth System Model: ECHAM6." *Journal of Advances in Modeling Earth Systems* 5 (2): 146–72.

Werner, M. 2019. "ECHAM5-Wiso Simulation Data–Present-Day, Mid-Holocene, and Last Glacial Maximum, PANGAEA."

Wet, Ruan F de, Adam G West, and Chris Harris. 2020. "Seasonal Variation in Tap Water δ2H and δ18O Isotopes Reveals Two Tap Water Worlds." *Scientific Reports* 10 (1): 13544.

Yoshimura, K, M Kanamitsu, D Noone, and T Oki. 2008. "Historical Isotope Simulation Using Reanalysis Atmospheric Data." *Journal of Geophysical Research: Atmospheres* 113 (D19).