Fourier Reservoir Computing for data-driven prediction of multi-scale coupled quasi-geostrophic dynamics

Hsin-Yi Lin[1,2] and Stephen G. Penny[1,2]

[1] *The Cooperative Institute for Research in Environmental Sciences (CIRES), University of Colorado, Boulder, CO, 80309, USA*
[2] *NOAA Physical Sciences Laboratory, Boulder, CO, 80305, USA*

Abstract

Reservoir Computing (RC), a simplified form of recurrent neural network, is composed with the Fourier Transform to produce data-driven prediction of multi-scale coupled quasi-geostrophic flows. Experiments are conducted using the Modular Arbitrary-Order Ocean-Atmosphere Model (MAOOAM) [10], a coupled quasi-geostrophic model that includes a 2-layer atmosphere (fast dynamics) and 1-layer ocean (slow dynamics). The Fourier Reservoir Computing (FRC) approach produces forecasts that extend the skillful forecast horizon beyond a comparable RC model trained on data in physical grid space. The FRC approach can be enhanced by applying localization in physical space, which extends the skillful forecast horizon further and facilitates practical application to high-dimensional geophysical problems.

Plain Language Summary

The science of predicting changes in weather and climate is typically enabled using a combination of idealized physical models and instrument measurements. Here, we offer a contribution as part of a growing community that is attempting to advance the use of machine learning for weather and climate prediction. We use a simplified coupled atmosphere-ocean model to generate synthetic 'observations', and show that reliable forecast models can be generating using machine learning applied to these observation data alone. We also provide an approach for scaling this method for use in more realistic operational weather prediction scenarios.

1 Introduction

Weather and climate science research relies on large volumes of historical observational data, computer models that integrate idealized equations constructed from fundamental principles of physics, and large-scale high performance computing [31]. Machine learning (ML) methods have been gaining increasing attention for their ability to process large volumes of data while taking advantage of new hardware architectures with relative ease. In general, ML techniques can be categorized as types of nonlinear optimization or regression methods that can extract useful information from large quantities of data. There have been groundbreaking successes with ML in a wide variety of fields, such as speech recognition [23] and autonomous driving [3], as well as the emulation and prediction of chaotic systems and fluid mechanics [4].

Our aim here is to make an incremental step toward the use of ML for real-time weather forecasts. We follow the history of numerical weather prediction (NWP) by considering the quasi-geostrophic (QG) equations, which are approximations of the shallow water equations with small Rossby number [5]. This is of particular relevance to atmospheric and oceanic dynamics, which in the mid-latitudes are largely determined by the balance of pressure forces and Coriolis forces. The QG equations were one of the first simplifications of the primitive equations of atmospheric motion to be used successfully for NWP [6]. Thus, we posit that the accurate emulation of QG dynamics would provide an indication that ML methods are capable of extracting the essential dynamical features needed for synoptic scale NWP.

As a practical objective, we ultimately seek to produce accurate forecasts directly from the historical record of atmospheric or oceanic data. Example sources of such data include direct observations, retrieval products, or reanalysis products. Thus, we generate a simulated historical record using an idealized 2-layer atmosphere / 1-layer ocean coupled QG model. Although the spectral information is sometimes accessible for idealized models, in many cases the source data are only available in gridded form. To ensure our method is applicable for most circumstances, our goal is to provide an effective data-driven prediction method based on gridded data of atmosphere and ocean.

In this work, we employ reservoir computing (RC), a simplified form of recurrent neural network (RNN), as our ML method for prediction. The RNN is a class of artificial neural networks where connected neurons form a directed graph along a time series. This structure has a natural dynamical character, and allows RNNs use their hidden states to capture the target dynamical behavior. RNNs first appeared in the 1980's [17; 26], and were further advanced in the early 1990's [9; 16]. Owing to its simplified structure, RC has the advantage of reduced training times [18; 22], which is a useful characteristic for transitioning to large scale applications. Moreover, RC has been shown to be effective for predicting chaotic dynamics such as Lorenz systems [29] and the Kuramoto-Sivashinsky equation [24], and positive results in short-term forecast horizons (e.g. less than 12 hours) have been obtained by applying RC to a more realistic gridded atmospheric dataset from the European Centre for Medium-Range Weather

Forecasts (ECMWF) [2]. In these previous examples, ML methods were applied directly to data represented in a physical grid space. We will show that that our RC approach using Fourier decomposition yields improved prediction quality for multi-scale coupled QG dynamics, and extends skillful forecasts to longer forecast horizons (in some cases over 72 hours).

Due to numerical advantages, the spectral-transform approach is frequently utilized in the modeling of earth systems, from QG models to more realistic model systems such as the Integrated Forecasting System (IFS) used by ECMWF. The spectral decomposition has also been useful in practical applications such as signal processing, especially in speech understanding [25; 14]. For example, the Fourier transform has been applied with architectures such as fully connected neural networks [32], Long-Short Term Memory [8], and Transformer [20]. In this work, we explore the RC approach with the Fourier decomposition for the prediction of QG models. However, we should point out that since the QG equations retain only the balanced components and eliminate interactions between the balanced and higher order modes, they potentially reduce predictability, which was observed in [11] for a simple case of the shallow water equation.

While we focus on RC, particularly for its potential for scaling up to higher-dimensional systems, we note that there are other ML architectures that may be effective for problems in dynamical prediction and modeling. For example, a Long-Short Term Memory (LSTM) RNN [16] was used for forecasting dynamical systems with extreme events [30] and hydrologic prediction [21; 13]. A generative adversarial network (GAN) [15] was explored for simulation of turbulent dynamics [19]. We save investigation of these more complex RNN architectures for future work.

2 Data and Methodology

2.1 Data: Numerical solutions of coupled QG model dynamics

We use the Modular Arbitrary-Order Ocean-Atmosphere Model (MAOOAM) [10] for our experiments. It describes a two-layer QG atmosphere, coupled thermally and mechanically to a QG shallow-water ocean layer by a system of spectral equations. The target observables include the atmosphere barotropic stream function ψ_a, the ocean stream function ψ_o, and the change of temperature δT_a in atmosphere and δT_o in ocean with respect to climatological reference temperatures. MAOOAM allows arbitrary truncation of the spectral modes of atmosphere and ocean variables. In this work we fix the spectral resolution at $4x-4y$ [10], which results in a total of 104 spectral modes for the whole system.

There are many cases where only physical space gridded datasets are available. To simulate this situation with QG dynamics, the spectral modes are transformed to physical space by the multiplications with its corresponding basis functions, and then sampled uniformly to produce a 20×20 gridded dataset.

The largest Lyapunov exponent λ_1 of our MAOOAM configuration is estimated to be 0.3 using the numerical procedure based on QR decomposition [1]. The Lyapunov time, defined as λ_1^{-1}, therefore corresponds to approximately 3.33 days.

2.2 MAOOAM

The coupled QG dynamics represented by MAOOAM [10] are described by the evolution of atmospheric barotropic stream function ψ_a, ocean stream function ψ_o, atmospheric temperature T_a, and ocean temperature T_o. The simulated data are governed by the QG vortex and thermal equations, considering both mechanical and thermal mechanisms. The atmospheric barotropic stream function ψ_a is set to be the average of the stream function fields of the lower atmospheric level ψ_1 at 250 hPa and upper atmospheric level ψ_2 at 750 hPa. That is, $\psi_a = (\psi_1 + \psi_2)/2$. The equations of motion of ψ_1 and ψ_2 are,

$$\frac{\partial}{\partial t}\nabla^2\psi_1 + J\left(\psi_1, \nabla^2\psi_1\right) + \beta\frac{\partial}{\partial x}\psi_1 = -k_d'\nabla^2(\psi_1 - \psi_2) + \frac{f_0}{\Delta p}\frac{dp}{dt}, \tag{1}$$

$$\frac{\partial}{\partial t}\nabla^2\psi_2 + J\left(\psi_2, \nabla^2\psi_2\right) + \beta\frac{\partial}{\partial x}\psi_2 = k_d'\nabla^2(\psi_1 - \psi_2) - \frac{f_0}{\Delta p}\frac{dp}{dt} - k_d\nabla^2(\psi_2 - \psi_o). \tag{2}$$

The parameters f_0 and β provide representation of the Coriolis effect. The parameter k_d' corresponds to the friction between two atmosphere layers, and k_d corresponds to the friction between the ocean and lower atmosphere layers. The term p represents pressure, and the pressure difference Δp is 500 hPa. Similarly, the equation of motion for the ocean stream function ψ_o is,

$$\frac{\partial}{\partial t}\left(\nabla^2\psi_o - \psi_o/L^2\right) + J\left(\psi_o, \nabla^2\psi_o\right) + \beta\frac{\partial}{\partial x}\psi_o = r\nabla^2\psi_o + C_1\nabla^2(\psi_2 - \psi_o), \tag{3}$$

where L is the baroclinic Rossby radius of deformation. The parameter r is the friction at the bottom of the active ocean layer. The last term represents the impact of the wind stress, with the drag coefficient of the mechanical ocean-atmosphere coupling C_1.

The time evolution of the atmospheric temperature T_a and ocean temperatures T_o is governed by,

$$\gamma_a\left(\frac{\partial}{\partial t}T_a + J\left(\psi_a, T_a\right) - \frac{dp}{dt}\sigma\rho_a T_a\right) = -\lambda(T_a - T_o) + \epsilon_a\sigma_B T_o^4 - 2\epsilon_a\sigma_B T_a^4 + R_a, \tag{4}$$

$$\gamma_o\left(\frac{\partial}{\partial t}T_o + J\left(\psi_o, T_o\right)\right) = -\lambda(T_o - T_a) - \sigma_B T_o^4 + \epsilon_a\sigma_B T_a^4 + R_o. \tag{5}$$

γ_a and γ_o are the heat capacities of the atmosphere and ocean layer. λ is the heat transfer coefficient at the ocean–atmosphere interface. σ is the static stability of the atmosphere. ϵ_a is the emissivity of the grey-body atmosphere and σ_B the Stefan–Boltzmann constant. ρ_a is the density of atmosphere.

The QG equations of motion constitute a complex coupled system of partial differential equations (PDEs). Spectral models, such as MAOOAM [10], consider the Fourier expansions of stream functions and temperature. Specifically, the temperatures in MAOOAM are linearized around spatially uniform temperatures T_a^0 and T_o^0, and decomposed as $T_a = T_a^0 + \delta T_a$ and $T_o = T_o^0 + \delta T_o$. Note that by the hydrostatic relation and the ideal gas relation, one can derive the spatially dependent atmospheric temperature $\delta T_a = f_0(\psi_1 - \psi_2)/R$, where R is the ideal gas constant. With the linearization, the system can be reduced to the streamfunction fields ψ_a, ψ_o, and the spatially dependent temperatures δT_a and δT_o. The Fourier expansions of these four variables are considered, and this transforms the PDE system in physical space into an ordinary differential equation (ODE) system in Fourier space. More precisely, the Fourier coefficients η_i of all variables satisfy a coupled ODE system, which can be formulated in the following form:

$$\frac{d\eta_i}{dt} = \sum_j \sum_k \mathcal{T}_{i,j,k}\, \eta_j\, \eta_k, \tag{6}$$

for some tensor $\mathcal{T}_{i,j,k}$. We refer to [28] for the detailed derivation. Roughly speaking, the derivation from the original system to the system of spectral equations includes several approximations, including the finite difference method, linearization in temperature, eigenfunction expansion, and spectral modes truncation [28]. Many earlier relevant works, for example [7], also employed these numerical techniques for the QG system.

2.3 Reservoir Computing (RC)

2.3.1 Network structure

Traditional artificial neural networks can be viewed as function approximators, taking an input and giving a corresponding predicted output. RNNs incorporate an element of temporal consistency by maintaining a hidden state vector that effectively stores information about past states. Through a training process, the RNNs define dynamics for this hidden state by training a set of connection weights within the RNN architecture. The intent is for the RNN dynamics to emulate the dynamics of the training data.

RC simplifies the RNN by assigning the recurrent weights within most of the network randomly. Only the final 'readout' layer of the network is trained. For RC, the high-dimensional hidden state is known as the reservoir state. During training, the RC receives two input signals: (1) the reservoir state, and (2) the observed data. The reservoir state is propagated using the random dynamics of the network, but includes time dependent forcing from the observed data signal. Over a typically short transient period, the reservoir state becomes synchronized with the input signal. The output mapping is then learned to determine a functional relationship between the evolution of the reservoir and the observed data. After the output mapping is learned, the input forcing can be replaced with the RC prediction of the observable, and the RC model becomes a self-contained dynamical system.

We use the following RC structure: **Given** two randomly chosen matrices $\mathbf{W} \in \mathbb{R}^{N_r \times N_r}$,$\mathbf{W}_{in} \in \mathbb{R}^{N_r \times N_u}$, the input $\mathbf{u}(t)$ is converted into the reservoir state $\mathbf{s}(t+1)$ by the following relation:

$$\mathbf{s}(t+1) = \ell \tanh(\mathbf{W} \cdot \mathbf{s}(t) + \mathbf{W}_{in} \cdot \mathbf{u}(t)) + (1-\ell)\,\mathbf{s}(t), \tag{7}$$

where ℓ is the 'leaky reservoir' constant having values between 0 and 1. The dimension N_r of reservoir states is normally much higher than inputs, so that complex temporal features can be represented. The reservoir state $\mathbf{s}(t+1)$ is then transformed to the forecast $\mathbf{v}(t+1)$ by

$$\mathbf{v}(t+1) = \mathbf{W}_{out} \cdot \mathbf{s}(t+1). \tag{8}$$

2.3.2 RC training and prediction

The RC is trained by utilizing the observational history and applying a least-squares regression for the 'readout' mapping W_{out} by solving

$$\mathbf{U} = \mathbf{W}_{out}(\mathbf{S} \cdot \mathbf{S}^T + \lambda \mathbf{I}), \tag{9}$$

where $\mathbf{S} \in \mathbb{R}^{N_r \times N_t}$ is a matrix whose columns are the reservoir state vectors, $\mathbf{I} \in \mathbb{R}^{N_r \times N_r}$ is the identity matrix, $\mathbf{U} \in \mathbb{R}^{N_u \times N_t}$ is a matrix whose columns are the target (truth) states in the original system space, and $\lambda \geq 0$ serves as the Tikhonov parameter for regularization [27]. More complicated readout mappings used in [24; 2] include squared-terms $\mathbf{s}(t)^2$ of reservoir states. However, we do not see improvement from the use of squared-terms for the emulation of QG dynamics, and so retain the classical structure (8).

After training process, the readout mapping is fixed and the prediction is made by the Reservoir Computing relations (7) and (8).

2.4 Fourier Reservoir Computing (FRC)

Assuming we are provided with data on a geospatial grid, the Fourier Reservoir Computing (FRC) applies a type of encoder/decoder operation so that the RC can be applied to the Fourier modes. For our experiments, we assume a long evolution of QG dynamics has been recorded on a uniform grid, providing a discrete representation of two-dimensional fields that contain the values of the observables measured at each grid point. The gridded data are first encoded into Fourier space using the two-dimensional Fast Fourier transform (2D-FFT) in space. Precisely, the data $\mathbf{x}^t_{n,m}$ at each fixed time t are transformed to the frequencies $\hat{\mathbf{x}}^t_{k_1,k_2}$. We then take the inputs $\mathbf{u}(t)$ of the RC to be the time-dependent coefficients $\hat{\mathbf{x}}^t_{k_1,k_2}$ of the Fourier modes, and produce a prediction of the evolution of the modes using RC as described by (7) and (8). The RC output $\mathbf{v}(t)$ is then decoded by the inverse 2D-FFT to produce our final prediction in the physical space (See Figure 1).

Unless otherwise stated, the configuration of RC networks are as follows: the entries are uniformly distributed on $[-0.5, 0.5]$, and the reservoir matrix

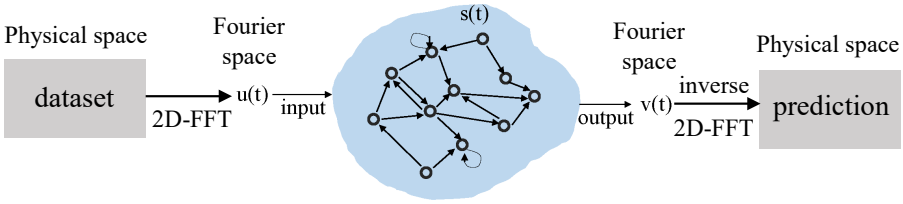

Figure 1: FRC approach for dynamical prediction. The dataset is transformed by 2D-FFT, and utilized in RC training. RC then produces a prediction for Fourier coefficients, which are later transformed back to physical space by inverse 2D-FFT.

$\mathbf{W}$ is further specified as a sparse matrix. The reservoir dimension is set at $N_r = 7000$. The values of the sparsity and spectral radius (i.e. the absolute value of the largest eigenvalue) of the reservoir adjacency matrix $\mathbf{W}$ are tuned between 0 and 1. We fix the leak parameter as $\ell = 0.95$ and Tikhonov parameter as $\lambda = 0.0001$.

2.4.1 Implementation of FRC

In this section, we use bold uppercase letters to denote matrices, bold lowercase for vectors, and regular lowercase for scalars. For convenience, we say a vector $\mathbf{v}$ has flattened entries $v_{n,m}$ if $\mathbf{v}$ is a vector of dimension $N \times M$ obtained from flattening a N by M matrix $\mathbf{V}$ with entries $v_{n,m}$. Precisely, the i-th entry of $\mathbf{v}$ is v_{n_i,m_i} where $(n_i, m_i) = (a, b)$, where $i = a \times M + b$, $a, b \in \mathbb{N} \cup \{0\}$. Given a two-dimensional historical gridded time sequence $\{x_{n,m}(t)\}$ from $t = 0$ to T, where $0 \leq n \leq N-1$ and $0 \leq m \leq M-1$. The FRC in our work is implemented to predict future dynamic for $t > T$ as follows:

1. Perform two-dimensional Fast Fourier transform (2D-FFT) on $\{x_{n,m}(t)\}$ to obtain a time sequence of gridded data $\{\hat{x}_{k_1,k_2}(t)\}$ in the Fourier space, where

$$\hat{x}_{k_1,k_2}(t) = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} x_{n,m}(t) \exp(-i2\pi n k_1/N) \exp(-i2\pi m k_2/M), \quad (10)$$

   for any $0 \leq k_1 \leq N-1$, $0 \leq k_2 \leq M-1$, and $t \leq T$. N and M are the total grid numbers in the x and y direction respectively.

2. Randomly initiate an $N_r \times NM$ input matrix $\mathbf{W}_{in}$ and an $N_r \times N_r$ reservoir matrix $\mathbf{W}$, where N_r is the reservoir dimension. All entries of both matrices are uniformly distributed in $[-0.5, 0.5]$. To tune the parameters sparsity and spectral radius, one can set a certain proportion of randomly chosen entries zero and perform a scaling for the whole matrix. Compute the reservoir (hidden) state $\mathbf{s}(t)$ by

$$\mathbf{s}(t) = \ell \tanh\left(\mathbf{W} \cdot \mathbf{s}(t-1) + \mathbf{W}_{in} \cdot \hat{\mathbf{x}}(t-1)\right) + (1-\ell)\,\mathbf{s}(t-1), \quad (11)$$

where $\hat{\mathbf{x}}(t)$ has flattened entries $\hat{x}_{k_1,k_2}(t)$ for all $t \leq T$ and $\tanh(\cdot)$ is performed element-wised. We note that the resulting reservoir state $\mathbf{s}(t)$ is a vector of dimension N_r for each fixed t. ℓ is commonly called the leak rate of RC, and is set at 0.95 for our experiments.

3. Compute the $NM \times N_r$ readout matrix $\mathbf{W}_{out}$:

$$\mathbf{W}_{out}^T = (\mathbf{S} \cdot \mathbf{S}^T + \lambda \mathbf{I})^{-1} \cdot \mathbf{S} \cdot \hat{\mathbf{X}}, \tag{12}$$

where $\hat{\mathbf{X}}$ is a $T \times NM$ matrix with entries $x_{t,i}$ set as the i-th entry of the flattened vector $\hat{x}_{k_1,k_2}(t)$, $\mathbf{S}$ is a $N_r \times T$ matrix with entries $s_{i,t}$ set as the i-th entry of $\mathbf{s}(t)$ computed in the last step. The matrix $\mathbf{I}$ is the identity matrix of dimension N_r, and λ is the Tikhonov regularization parameter, which is set at 0.0001 in our experiments.

4. Predict future dynamic through RC. We compute the future reservoir state $\mathbf{s}(t+1)$ and project it to the prediction $\hat{\mathbf{p}}$ of Fourier modes by the $\mathbf{W}_{out}$ matrix: For $t \geq T$,

$$\mathbf{s}(t+1) = \ell \tanh(\mathbf{W} \cdot \mathbf{s}(t) + \mathbf{W}_{in} \cdot \hat{\mathbf{p}}(t)) + (1-\ell)\,\mathbf{s}(t). \tag{13}$$

$$\hat{\mathbf{p}}(t) = \mathbf{W}_{out} \cdot \mathbf{s}(t), \tag{14}$$

where $\hat{\mathbf{p}}(t)$ has flattened entries $\hat{p}_{k_1,k_2}(t)$ and $\mathbf{s}(t)$ is the reservoir state vector of dimension N_r at each t.

5. Obtain the dynamical forecast $p_{n,m}(t)$ at the grid point (n,m) and time t by inverse 2D-FFT entrywisely:

$$p_{n,m}(t) = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} \hat{p}_{k_1,k_2}(t) \exp(i2\pi n k_1/N) \exp(i2\pi m k_2/M), \tag{15}$$

for all $n \leq N$, $m \leq M$ and $t \geq T$.

For the results shown in this work, the training dataset has in total 50000 time steps. It takes roughly 12 minutes to train a RC network with 7000 reservoir dimension for each trial on one CPU with 16 GB memory. We also note that all entries of the training dataset were scaled into values between 0 and 1, by first subtracting the minimum and then dividing the maximum over time. Precisely,

$$x_{n,m}^s(t) = \left(x_{n,m}(t) - \min_{t\in[0,T]} x_{n,m}(t) \right) / \max_{t\in[0,T]} x_{n,m}(t),$$

where $x_{n,m}^s(t)$ is the data at grid (n,m) and time t after $x_{n,m}(t)$ is scaled.

2.5 Scaling to higher-dimensions: FRC with spatial localization

The FRC method can be applied globally on the entire domain or locally on independent or partially overlapping patches. For more realistic high-dimensional

NWP model applications, it is necessary to decompose the model grid into multiple subdomains to enable parallel computation. To apply localization, we similarly divide the gridded data in physical space into several disjoint local patches and apply the FRC method to each subdomain. Each subdomain has its own reservoir computing training and prediction, which is performed independently of other subdomains. We present results using both the global and localized FRC approaches. Other approaches are possible for localization. For example, [24] and [2] used the same disjoint local patches for training, but applied interacting boundary points during prediction. It is also possible to do localization in the frequency space after applying the Fourier transform. We attempted both of these approaches, but found that the interacting boundary points did not improve prediction skill with the FRC, while localization in the frequency space degraded prediction skill.

Because we apply the Fourier transform on a spatial localization of the gridded field, we must be careful of artifacts such as the Gibbs phenomena or ringing, that may be introduced due to the non-periodicity of the original function. This issue can be avoided by extending the local spatial patch (in square-based grids) with a uniform-valued narrow neighborhood. With the same value on the boundary of the local patch, the function in each local patch can be considered as a periodic function when performing a discrete Fourier transform, because we can extend the local function periodically to the whole space with its own copies infinite times. In this way, the Fourier transform of the local function is equivalent to the usual Fourier transform of a continuous periodic function without the jump discontinuities that can cause artifacts.

Even though the Fourier basis is utilized with RC in this work, we emphasize that the localization is only performed in spatial space, not in Fourier space. Nevertheless, we should point out that a natural local Fourier basis exists due to our setting. In fact, after applying spatial localization extended with a uniform-valued boundary, the function in each local patch is considered as a periodic function when performing discrete Fourier transform. Owing to the periodicity, a natural finite frequency band is guaranteed even though we do not perform an additional cut-off in Fourier modes for experiments.

We utilize above FRC to obtain prediction and further investigate the effect of localization on our method. For localization, we divide the 20×20 gridded data into four disjoint 10×10 patches and our approach is applied independently on each subsystem. Our results suggest that RC has difficulty learning the QG dynamics directly from the gridded data set in physical space, and can be improved greatly by Fourier decomposition.

3 Results

We apply the FRC method to the gridded data recording the forecasts for the coupled QG dynamics, and compare the results to the RC in physical space as previously applied by [24; 2]. A total of 10 trials with different initial conditions are performed for each experiment, and the normalized root mean square error

(NRMSE) is computed as follows:

$$NRMSE(t) = (\langle (prediction(t) - truth(t))^2/\sigma^2 \rangle)^{1/2}, \tag{16}$$

where $\langle \cdot \rangle$ indicates the average over all entries, and σ^2 is the variance over time. The FRC results in lower NRMSE than the RC applied in physical space (Figure2). The NRMSE of the physical space RC grows to around order 1 after 1 day. On the other hand, the NRMSE produced by Fourier space RC is around 2×10^{-1} for atmosphere observables and 5×10^{-3} for ocean for over 1 Lyapunov time.

To our best knowledge, there is no previous work investigating RC prediction of QG dynamics and the effect of Fourier decomposition in the setting of gridded dataset originated from spectral models. We note that [2] also applied RC to data that were originally generated by a spectral transform model, but it was analyzed in physical space. While our experimental setting is unique without benchmarks, our RC implementation was tested on Lorenz 96 systems and capable of reasonable performance.

The target evolution and prediction error by the FRC approach are presented in Figure 3. From top to bottom, each column contains the two-dimensional target surface, and the error surfaces of FRC as well as Localized FRC on each day. We note that when applied in the physical space, the prediction errors for RC reach a magnitude over 0.15 for ψ_a, 0.08 for δT_a, 4×10^{-5} for ψ_o and 0.026 for δT_o, within 1 Lyapunov time. The forecast errors for both the global and local FRC are significantly smaller than the physical space RC, as indicated in Figure 2.

Although the values of sparsity and spectral radius in the construction of RC are commonly tuned for the best outcome, we note that our results above are fairly robust to these two hyperparameters. In Figure 4, the NRMSE curves by FRC are shown for different combinations of sparsity and spectral radius. The NRMSE curves behave similarly, with small standard deviation around 10^{-2} for atmosphere, and 10^{-3} for ocean around 1 Lyapunov time.

On the other hand, a sufficiently large reservoir dimension is particularly important for RC performance. In general, a higher reservoir dimension provides more capacity to capture dynamical complexity and hence leads to more accurate forecasts. Moreover, we also observe better robustness in prediction with larger reservoir dimensions. Our experiments test reservoir dimensions of 3000, 5000, and 7000, with the sparsity and spectral radius tuned for each case. As MAOOAM is a spectral model, we also perform RC directly on the native model space to provide a reference target performance skill. As a metric to compare the forecast skill of each method, we define the valid prediction time (VPT) by:

$$VPT = arg\max_t \{NRMSE(t) < \epsilon\}. \tag{17}$$

We take the maximal error by the RC applied in the physical grid space as a baseline, and consider VPT as the first time when NRMSE exceeds 5% of this maximal error. In Figure 5, the results show for all target observables, the prediction is improved by transforming the data from physical space to Fourier

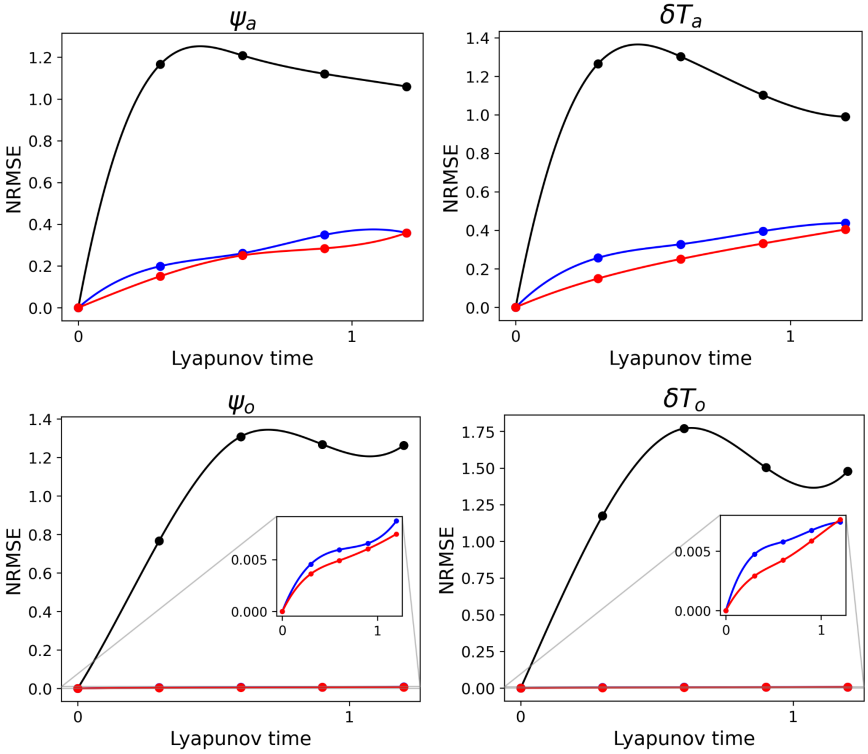

Figure 2: NRMSE comparison between methods. The black line represents the NRMSE by RC in physical space, blue line by RC in Fourier space, and red by Fourier RC with localization.

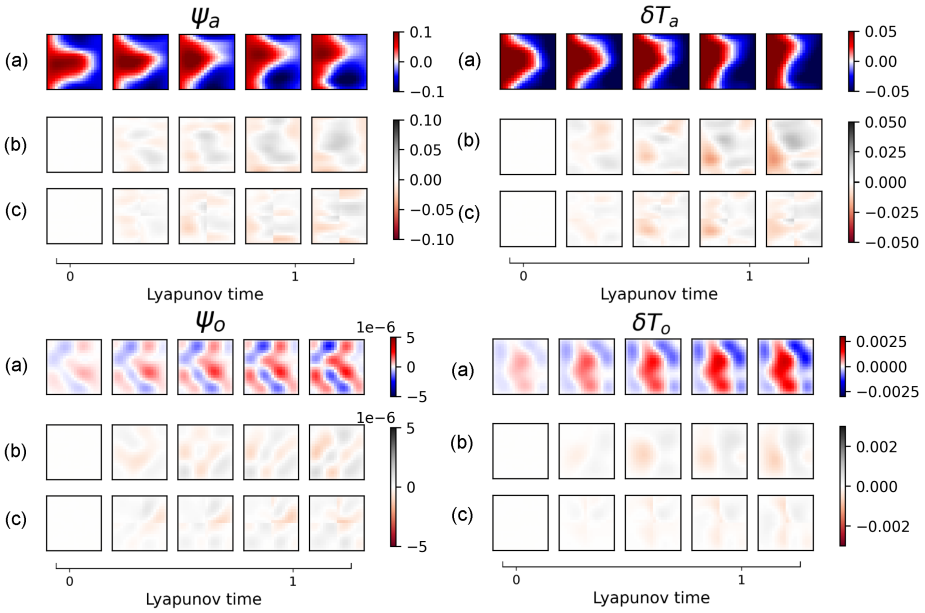

Figure 3: Prediction of stream function (left) and temperature (right) for atmosphere (top) and ocean (bottom). (a) target for each day. (b) target minus prediction by RC in Fourier space. (c) target minus prediction by Fourier RC with localization.

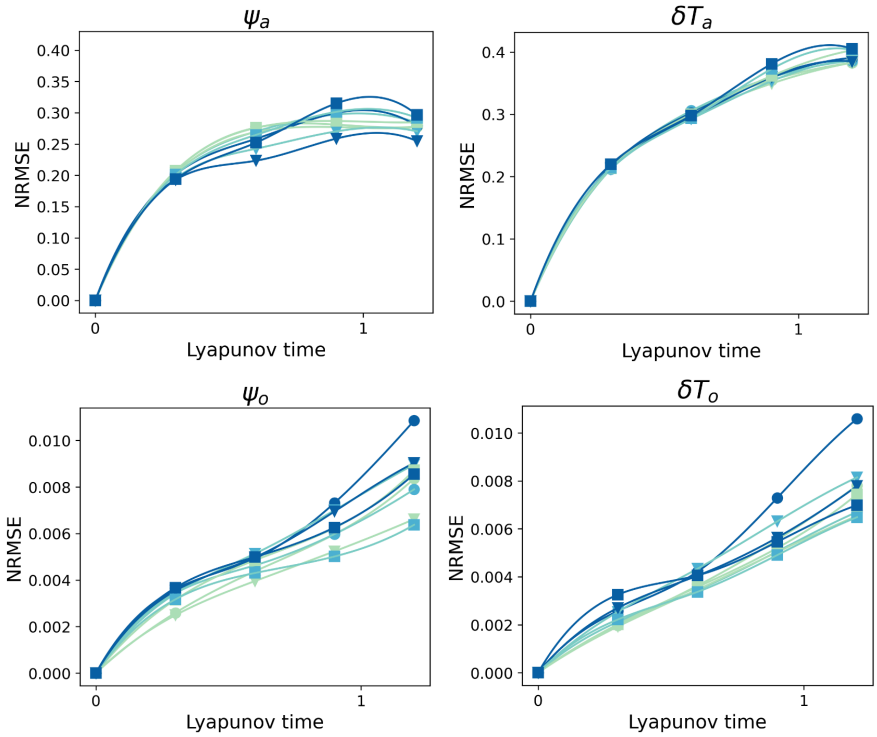

Figure 4: NRMSE results using FRC with 9 different combinations of sparsity and spectral radius chosen from $\{0.2, 0.5, 0.8\}$. The colors from light to dark correspond to the increase of radius from 0.2 to 0.8. We use different markers for different sparsity. Specifically, ▼ represents the case with sparsity 0.2, • with 0.5, and ■ with 0.8.

space, as well as by increasing the reservoir dimension. For example, using the same size reservoir the VPT for the atmosphere is doubled when applying FRC, and nearly tripled when applying the localized FRC. Alternatively, the FRC can attain nearly identical performance in the atmosphere and approximately 4x longer VPT in the ocean using a reservoir of less than half the size of the physical space RC. Furthermore, the superior performance of RC applied to the original spectral modes supports the assertion that the spectral information plays a key role in the successful prediction of QG dynamics.

From our observation, Fourier encoding leads to improved performance can be related to the data structure originated from the system of ordinary differential equations in spectral modes, which describes the wave-like type of dynamics in physical space derived from the diffusion effect. ML methods can be considered as nonlinear regression methods that approximates desired target dynamics with a certain function structure, which in our case here is the RC network. The result improvement obtained from moving data from physical to Fourier space, shows that plain RC networks encounter a noticeable degree of difficulty of capturing wave-like dynamics. This issue however can be nicely resolved by Fourier decomposition, which transform gridded physical positions into frequency channels.

We note that even though both spectral RC and FRC predict Fourier mode dynamics, the input datasets are of different accuracy. The input of the spectral RC is the time series of 'perfect' coefficients of the Fourier basis used by the original MAOOAM model. For the FRC, the input is a set of physical variables that have been translated to a discretized grid. In this process of discretization, there are errors introduced into the system and hence the input data provided to the FRC has degraded quality. The results of both the FRC and the idealized spectral RC experiments indicate that the RC-based ML approach is more effective for predicting QG dynamics when applied to data represented in the Fourier space than when applied to data represented in a physical grid space.

It is worth pointing out that the resolution of gridded data also affects the RC prediction. Although use of higher resolutions generally increases the complexity of target dynamics and thus requires a larger reservoir dimension for reasonable prediction, we notice the projection to lower resolutions (e.g. truncated to less modes) can also degrade RC performance as measured by the VPT. We found that for our idealized coupled QG application, RC and FRC are capable of better prediction when more information is provided. For example, increasing the resolution (of the physical grid representation) or increasing the number of Fourier modes (by using a higher order truncation) are both helpful in extending the VPT, assuming a sufficiently large reservoir dimension is provided.

4 Conclusions

We have applied a type of machine learning method called reservoir computing (RC) to a coupled atmosphere-ocean system to produce accurate short to medium-range forecasts. Further, we introduced the use of the fast Fourier

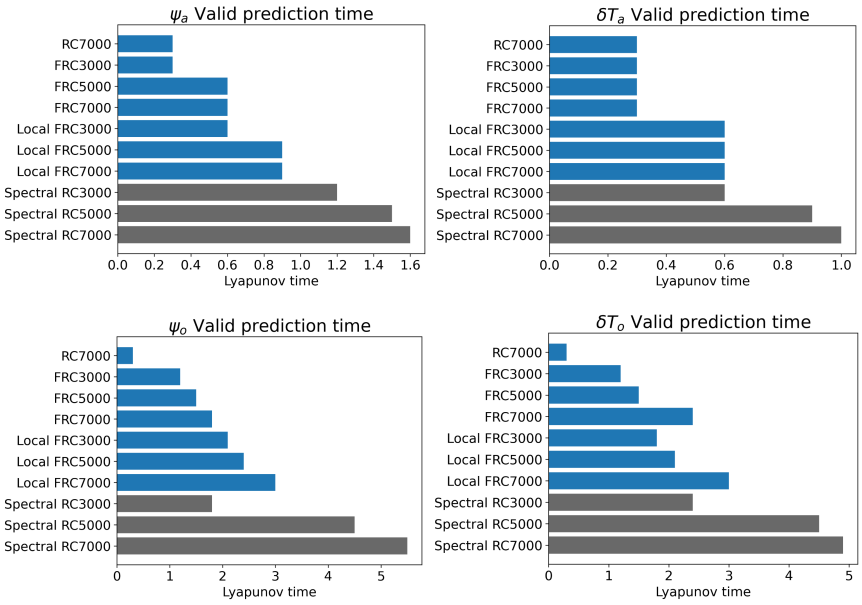

Figure 5: The Valid Prediction Time (VPT) obtained with different reservoir dimensions (3000, 5000, 7000) is compared between methods. The results are averaged over 10 different initial conditions, to ease initial condition dependence. We use 'RC' to refer to the control case in which the RC model takes as input the dataset in the physical grid space, 'FRC' for the case in which the dataset is transformed by the 2D-FFT, and 'Local FRC' for the case in which the FRC approach is applied with localization. 'Spectral RC' represents the case when the RC takes as input the original spectral mode coefficients generated by MAOOAM. The Spectral RC cases provide target performance levels for the RC models that are trained on data provided on a physical grid.

transform (FFT) and inverse FFT as an encoder/decoder step in the RC training and prediction operations as an effective tool for the emulation and prediction of quasi-geostrophic (QG) dynamics. We term this composition of Fourier transform and RC Fourier Reservoir Computing (FRC).

In general, ML methods can be considered as nonlinear regression methods that approximate desired target data with a specified function structure, which in our case is the RC network. A common approach for atmospheric models is to formulate the dynamics using a Fourier basis, as atmospheric motions can be largely described by wavelike dynamics. For this reason we developed and applied the FRC encoder/decoder. This transforms gridded datasets of physical states into frequency channels using the FFT. We found that this approach extends the skillful forecast horizon of RC model forecasts. The improvement obtained by transforming data from physical to Fourier space indicates that conventional RC networks may encounter a noticeable degree of difficulty in resolving wavelike dynamics.

Experimental results indicate that for identical reservoir sizes, the FRC is effective at doubling the skillful forecast horizon in the atmosphere while increasing the skillful forecast time in the ocean many times over. Applying localization to the FRC further extends the skillful forecast horizon for both the atmosphere and ocean. In addition, localization also provides a practical mechanism to scale the FRC to higher-dimensional geophysical problems, which will be a focus of future work.

Lin and Penny acknowledge support from ONR grants N00014-19-1-2522 and N00014-20-1-2580, and NOAA grant NA18NWS4680048. Penny further acknowledges support from NOAA grants NA20OAR4600277 and NA19NES4320002. We thank Brian R. Hunt and Henry D.I. Abarbanel for thoughtful discussions on machine learning applied to dynamical systems.

5 Data availability statement

For clarification and the reproducibility of this work, the software used to carry out ML experiments can be found at:
`https://github.com/hsinyilin19/Fourier-Reservoir-Computing`. The datasets used in this work were generated using the Modular Arbitrary-Order Ocean-Atmosphere Model (MAOOAM) [10], available at: `https://github.com/Climdyn/qgs`, with further details provided by [12]. The generated data used for the experiments in this work can be found at: `https://zenodo.org/record/4609483#.YFGLVUNKjEl`.

References

[1] H. Abarbanel. *Analysis of observed chaotic data*. Springer Science & Business Media, 2012.

[2] T. Arcomano, I. Szunyogh, J. Pathak, A. Wikner, B. R. Hunt, and E. Ott.

A machine learning-based global atmospheric forecast model. *Geophysical Research Letters*, 47(9):e2020GL087776, 2020.

[3] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

[4] S. L. Brunton, B. R. Noack, and P. Koumoutsakos. Machine learning for fluid mechanics. *Annual Review of Fluid Mechanics*, 52:477–508, 2020.

[5] J. G. Charney. On the scale of atmospheric motions. In *The Atmosphere—A Challenge*, pages 251–265. Springer, 1990.

[6] J. G. Charney and A. Eliassen. On the growth of the hurricane depression. *Journal of Atmospheric Sciences*, 21(1):68–75, 1964.

[7] J. G. Charney and N. Phillips. Numerical integration of the quasi-geostrophic equations for barotropic and simple baroclinic flows. *Journal of Meteorology*, 10(2):71–99, 1953.

[8] Z. Chen, S. Watanabe, H. Erdogan, and J. R. Hershey. Speech enhancement and recognition using multi-task learning of long short-term memory recurrent neural networks. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[9] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[10] L. De Cruz, J. Demaeyer, and S. Vannitsem. The modular arbitrary-order ocean-atmosphere model: MAOOAM v1.0. *Geoscientific Model Development*, 9(8):2793–2808, 2016. doi: 10.5194/gmd-9-2793-2016. URL `https://gmd.copernicus.org/articles/9/2793/2016/`.

[11] A. M. DeGennaro, N. M. Urban, B. T. Nadiga, and T. Haut. Model structural inference using local dynamic operators. *International Journal for Uncertainty Quantification*, 9(1), 2019.

[12] J. Demaeyer, L. De Cruz, and S. Vannitsem. qgs: A flexible python framework of reduced-order multiscale climate models. *Journal of Open Source Software*, 5(56):2597, 2020.

[13] J. Frame, G. Nearing, F. Kratzert, and M. Rahman. Post processing the us national water model with a long short-term memory network. 2020.

[14] S. K. Gaikwad, B. W. Gawali, and P. Yannawar. A review on speech recognition technique. *International Journal of Computer Applications*, 10 (3):16–24, 2010.

[15] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014.

[16] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[17] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.

[18] H. Jaeger and H. Haas. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *science*, 304(5667):78–80, 2004.

[19] B. Kim, V. C. Azevedo, N. Thuerey, T. Kim, M. Gross, and B. Solenthaler. Deep fluids: A generative network for parameterized fluid simulations. In *Computer Graphics Forum*, volume 38, pages 59–70. Wiley Online Library, 2019.

[20] Y. Koizumi, K. Yatabe, M. Delcroix, Y. Masuyama, and D. Takeuchi. Speech enhancement using self-adaptation and multi-head self-attention. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 181–185. IEEE, 2020.

[21] F. Kratzert, D. Klotz, M. Herrnegger, A. K. Sampson, S. Hochreiter, and G. S. Nearing. Toward improved predictions in ungauged basins: Exploiting the power of machine learning. *Water Resources Research*, 55(12):11344–11354, 2019.

[22] W. Maass, T. Natschläger, and H. Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural computation*, 14(11):2531–2560, 2002.

[23] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[24] J. Pathak, B. Hunt, M. Girvan, Z. Lu, and E. Ott. Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach. *Physical review letters*, 120(2):024102, 2018.

[25] L. R. Rabiner and B. Gold. Theory and application of digital signal processing. *Englewood Cliffs: Prentice-Hall*, 1975.

[26] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[27] A. N. Tikhonov and V. Y. Arsenin. Solutions of ill-posed problems. *New York*, pages 1–30, 1977.

[28] S. Vannitsem, J. Demaeyer, L. De Cruz, and M. Ghil. Low-frequency variability and heat transport in a low-order nonlinear coupled ocean–atmosphere model. *Physica D: Nonlinear Phenomena*, 309:71–85, 2015.

[29] P. R. Vlachas, J. Pathak, B. R. Hunt, T. P. Sapsis, M. Girvan, E. Ott, and P. Koumoutsakos. Backpropagation algorithms and reservoir computing in recurrent neural networks for the forecasting of complex spatiotemporal dynamics. *Neural Networks*, 2020.

[30] Z. Y. Wan, P. Vlachas, P. Koumoutsakos, and T. Sapsis. Data-assisted reduced-order modeling of extreme events in complex dynamical systems. *PloS one*, 13(5):e0197704, 2018.

[31] J. S. Wettlaufer. Climate science: An invitation for physicists. *Physical review letters*, 116(15):150002, 2016.

[32] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee. A regression approach to speech enhancement based on deep neural networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(1):7–19, 2014.