Highlights

Reconstruction of regional 3D ocean temperature fields from reanalysis data and real-time satellite and buoy surface measurements

Bianca Champenois, Themistoklis Sapsis

- We present a framework to model 3D ocean temperature fields and their uncertainty from real-time surface temperature sensor measurements.
- Our approach uses a convolutional neural network to capture structure from physics-based numerical models.
- The framework is validated with in-situ measurements of ocean temperature at various depths.

Reconstruction of regional 3D ocean temperature fields from reanalysis data and real-time satellite and buoy surface measurements

Bianca Champenois[a], Themistoklis Sapsis[a]

[a]*Massachusetts Institute of Technology Department of Mechanical Engineering, 77 Massachusetts Avenue, Cambridge, 02139, MA, USA*

Abstract

Despite advancements in computational science, nonlinear geophysical processes still present important modeling challenges. Physical sensors (e.g. satellites, AUVs, and buoys) can collect data at specific points, but are often sparse or inaccurate. We present a framework to build improved spatiotemporal models that combine dynamics inferred from high-fidelity models and sensor measurements. We are motivated by a temperature data set from sensors that are only able to make measurements at the ocean's surface. We first apply standard principal component analysis (PCA) at every ocean surface coordinate to a reanalysis data set of the time-evolving 3D temperature field. Next, a conditionally Gaussian model implemented through a temporal convolutional neural network (TCN) is built to predict the time coefficients of the PCA modes, and their variance, as a function of surface temperature. The 2D surface temperature field is estimated by a multi-fidelity Gaussian process regression scheme, for which buoy data have higher accuracy and precision (higher fidelity) compared to the satellite data. The surface temperature is then used as input to the neural network to probabilistically predict the PCA coefficients and reconstruct the full 3D temperature field. The results are compared to in-situ measurements at all depths, and the median absolute error is found to be 0.97°C. Overall, the proposed framework can build less expensive and more accurate conditionally Gaussian models in real time, and can be leveraged for optimal sampling and path planning.

Keywords:
ocean temperature, temporal convolutional network, uncertainty quantification

1. Introduction

Environmental and geophysical fluid systems can be modeled with nonlinear equations that typically require complex and computationally expensive numerical solvers. Even with highly accurate numerical methods, model errors still exist and intrinsic instabilities in the system make prediction difficult. Such challenges can be mitigated with physical sensors (e.g. buoys, drifters, and satellites) which can be used to collect additional information on quantities of interest. However, sensors only provide information about the system locally in space, with significant gaps, or with a high degree of sparsity. The field of ocean modeling is far-reaching, and significant work has been done to estimate surface properties, design numerical simulations, assimilate sensor measurements, and reconstruct 3D fields.

Work has been done on the development of methods to predict subsurface ocean temperature from satellite surface estimates. Historically, climatological averages have been used to estimate such properties [1]. Then, more advanced interpolation and regression schemes were employed to improve upon the climatological averages [2, 3, 4, 5]. In many of these studies, empirical orthogonal functions (EOF) are utilized [6, 7, 4, 8, 9]. In more recent studies, neural networks have also been used to predict subsurface properties [10, 11]. One study tested the use of recurrent neural networks (RNN) and long short-term memory (LSTM) neural networks to model vertical EOFs in the North Atlantic Ocean [12]. In contrast to making estimates from observations, it is also possible to make predictions from physics-based numerical simulations. Simulations that integrate real world measurements into the computation are referred to as reanalysis data. While there exist many data products and numerical simulations, the growing field of machine learning offers many opportunities for improvement. This paper identifies newly-developed methods to quickly and parsimoniously estimate the state and uncertainty of regional systems at a high resolution and in real time.

To provide a specific example, we consider the temperature of the Massachusetts and Cape Cod Bays, an area with great biodiversity (fish, shellfish, whales, etc.) and significant fishing and tourism industries. The ability to predict temperature is helpful in assessing general ocean health, tracking ecosystem functioning, and managing fisheries. Changes in temperature have greater implications in coastal waters where most fisheries are located

[13, 14, 15, 16]. Ocean temperature is governed by a set of high dimensional nonlinear equations. These are coupled with other physical quantities such as velocity over different depths. These nonlinear equations can, in principle, be solved to evaluate the temperature field using a numerical scheme, such as finite volumes or differences, resulting in an extremely high dimensional and chaotic dynamical system. In this context, ocean modeling equations are typically complemented by measurements from in-situ buoys, drifters and satellites. Buoys and drifters provide reliable measurements, but they are very scarce. On the other hand, a satellite can cover the whole domain, but there are gaps in the data due to cloud coverage, and the measurements are partial and less accurate. Most importantly, sensor measurements are only available at or near the surface of the ocean, leaving the bottom depths of the ocean unaccounted for [17, 18]. These challenges are combined with the inevitable uncertainty in the boundary, initial and excitation conditions, as well as modeling uncertainty [19], and eventually result in a very demanding problem that requires vast computational, mathematical, and modeling resources to be solved, especially in real time.

The goal of this work is different from traditional and direct ocean modeling efforts. Our aim is to utilize existing reanalysis data (obtained from physics-based and observation-driven ocean models offline) and leverage data science techniques to characterize the vertical structure of the ocean temperature field, and subsequently combine machine learning methods with real-time sensor measurements of surface temperature to reconstruct and hindcast the full 3D temperature field and its uncertainties, for a specific region of interest. We consider temperature data because it is readily available, but the techniques discussed can be applied to other quantities of interest such as salinity, dissolved inorganic carbon (DIC), aragonite, and pH. In addition, the model can estimate the uncertainty associated with both the system and the model. Specifically, we use a combination of data science techniques including principal component analysis (PCA), temporal convolutional neural networks (TCN) and Gaussian process regression (GPR). As a result, we develop a computationally inexpensive model for the Massachusetts and Cape Cod Bays that leverages data from physics-based numerical models, buoys, and satellites to predict the temperature and uncertainty in real time at all points in the domain of interest. The model is also useful to make decisions about where and how to sample future data [20, 21] and to evaluate the quality of new sensors. Secions 2 and 3 describe the different types of data that are used to train and evaluate the model. Section 4 explains the steps

of the framework. Section 5 shows the results of the model.

2. Reanalysis Data

Our starting point is reanalysis data consisting of a time-evolving 3D temperature field of the Northeast Coastal Ocean from the FVCOM (Finite Volume Community Ocean Model) simulation from Chen et al. [22, 23]. The model uses a fractional step method to solve the spatially and temporally evolving fields for velocity, density, temperature, and salinity, among other variables with a horizontal resolution of approximately 0.1-25 km over 45 sigma levels. Here, a sigma level refers to a layer of the sigma coordinate system. In the sigma coordinate system, horizontal layers follow the model terrain, so for a given (x, y) point, each horizontal layer has the same thickness [24]. This coordinate system is a convenient way to discretize the domain because it results in a continuous temperature field. In the data assimilation step, FVCOM uses observations from satellite SST and radiation, river discharge, NOAA C-MAN and NDBC buoys, and NERACOOS buoys. A study by Li et al. found that the model agreed well with in-situ measurements with a root mean squared error of 2.28 °C [18]. The entire hindcast ranges from the Delaware Shelf to the eastern end of the Scotian Shelf over several decades, but we only consider a truncated portion of the domain in the Massachusetts and Cape Cod Bays from January 2005 to December 2013 (9 years total). As an example, a snapshot of the data from September 13th, 2012 at sigma level -0.5 is plotted in Figure 1. In the spatial domain of interest, the maximum depth reaches 200 m, but most of the points are within 0 and 50 m.

3. Measurements and Observations

In addition to the data from the finite volume scheme, we have surface temperature data from physical sensors: satellites, in-situ stations, and buoys. Satellites measure sea surface temperature by quantifying the energy of wavelengths coming from the ocean. Different satellites operate at varying resolutions and levels of accuracy [25], but the main challenge associated with using satellite data is that there can be gaps due to cloud coverage. There exist many satellite sea surface temperature (SST) data products: e.g. Optimal Interpolation SST (OISST), Hadley Centre Global Sea Ice and SST (HadISST), Climate Change Initiative SST (CCI SST). Each of these is at a

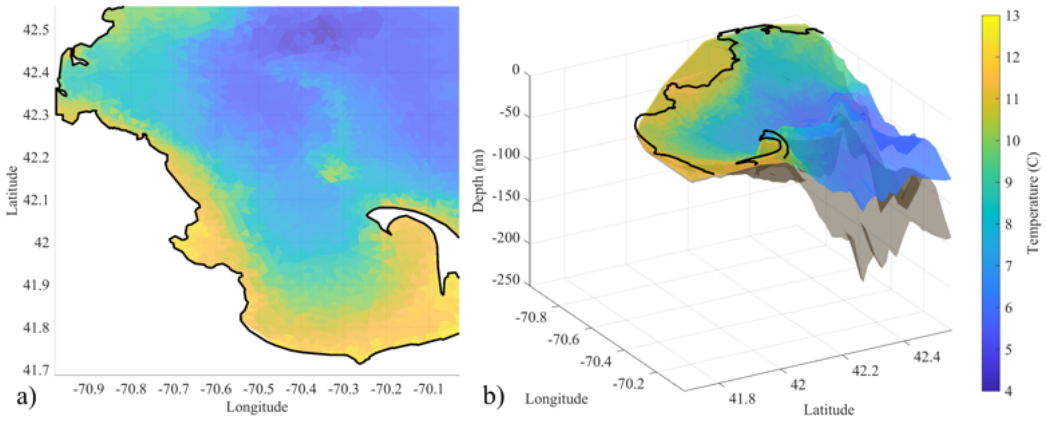

Figure 1: **Reanalysis Data of the Northeast Coastal Ocean from FVCOM**. The temperature field is plotted for September 13th, 2012 at sigma level -0.5. The bathymetry of the region of interest is shown in (b) in brown.

different spatial and temporal resolution, some of which are not sufficient for the region of interest. It is also critical for our analysis that the SST product we use comes with uncertainty estimates. For the region of interest, we have access to unprocessed daily satellite imagery from the MODerate-resolution Imaging Spectroradiometer (MODIS) Terra. In Figure 2 we observe that each day has a different amount of cloud coverage. Most importantly, many days during winter months have no available satellite measurements. In contrast to satellites, in-situ stations and buoys are not affected by cloud coverage. Measurements are available from the Massachusetts Water Resources Authority (MWRA) (Figure 2), but they are only collected on a monthly basis, and there are only 14 locations. The MWRA stations gather data by collecting samples of water at multiple depths and directly measuring the temperature. While this method is more accurate, it is also very costly. At the validation stage we employ below the surface measurements at multiple depths from the aforementioned sensors to assess the quality of our model.

4. Framework Description

The framework is organized into multiple steps as outlined in Figure 3. The first two steps are independent. First, in step 1, we use the reanalysis data to build a data-driven reduced order model and derive a functional relationship between 3D temperature and surface temperature; this connection is possible given the reduced-order vertical structure of the problem that we

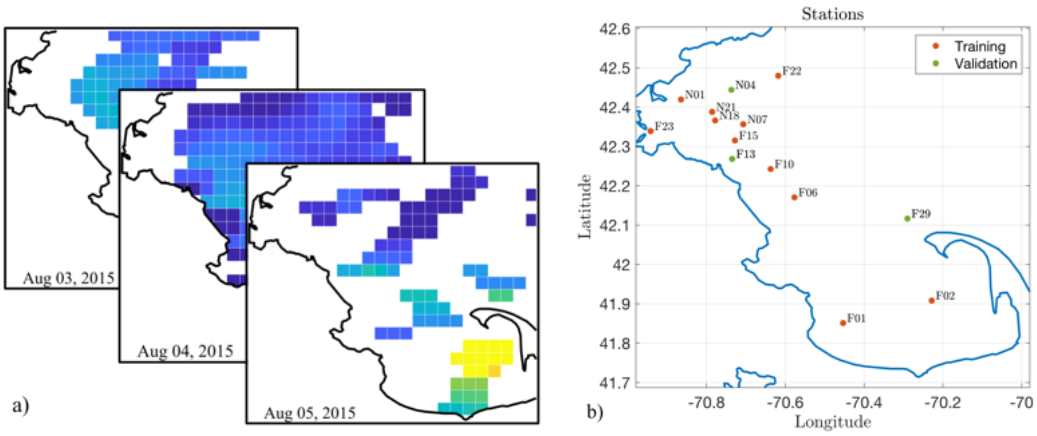

Figure 2: **Sensor Data.** The low fidelity data (satellite (a)) is only available on days with low cloud coverage. The high fidelity data (buoys (b)) is local in space and sparse.

obtain from principal component analysis (PCA). Next, in step 2, we use multi-fidelity Gaussian process regression (GPR) to estimate the ocean surface temperature by merging information from satellites and in-situ buoys as described in Babaee et al. [26]. We reuse most of the methodology from Babaee et al., but we modify the choice of inputs to speed up the process to cover a larger domain. Finally, in step 3, we input the real-time 2D surface temperature measurements into the reduced-order model to obtain a real-time estimate for the 3D temperature field and its uncertainty. The framework can be modified or rearranged based on the type and location of new data that become available.

4.1. Temperature Field Order-Reduction Using Vertical PCA

We first apply standard principal component analysis to the reanalysis data set to reduce the dimensionality while retaining patterns and information. Principal component analysis (PCA), also known as empirical orthogonal functions (EOF), proper orthogonal decomposition (POD) or Karhunen–Loève decomposition, among other names, has long been used in many fields. In the context of fluid mechanics, weather prediction [27, 28], and oceanography, PCA extracts features or trends from large empirical data sets to accurately reconstruct the dynamics of the system using a small number of EOFs and corresponding coefficients. Significant work has been done on the use of EOFs to reconstruct spatio-temporal SST for which empirical measurements from sensors are available [29, 30, 31, 32]. In some cases, the

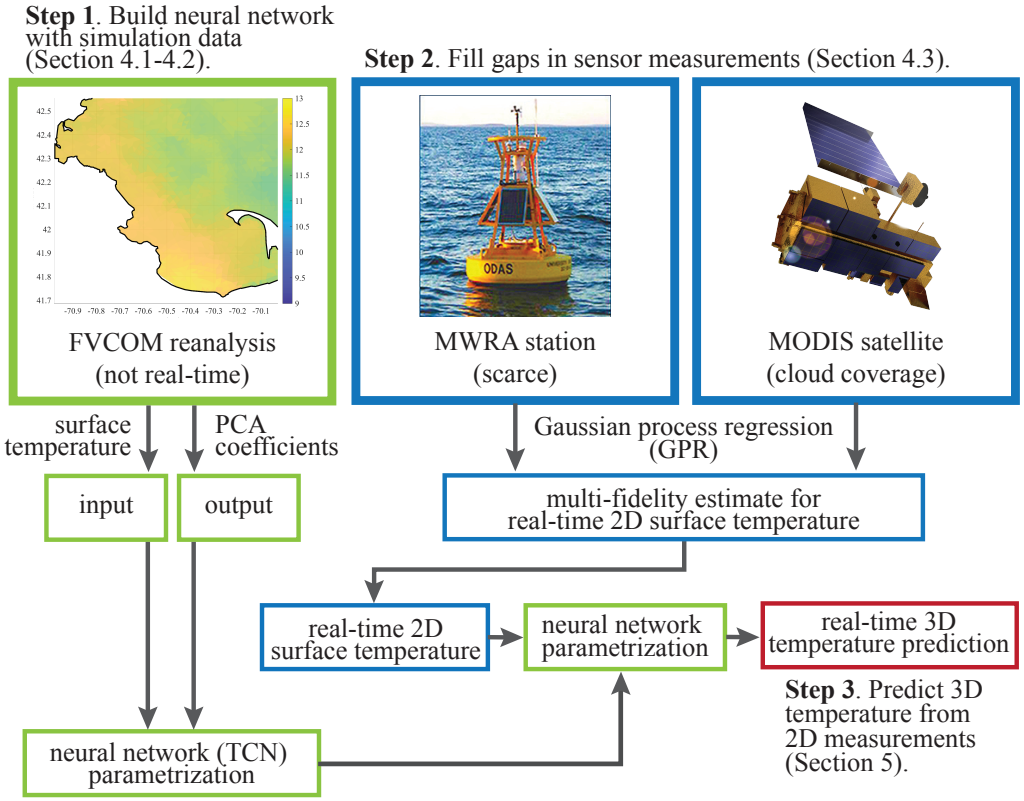

Figure 3: **Framework**. Flow chart describing the developed framework for real-time estimation of the 3D ocean temperature field. Reanalysis data are employed to estimate a reduced-order model. Ocean surface information, obtained from satellite and buoy measurements, are used as input.

basis is used to fill gappy data [33]. Here, we use PCA to represent the vertical structure of existing reanalysis data with just a few modes at each location of the ocean surface. We are interested in the vertical structure of the temperature field because most of the energy of the system is coming from solar radiative flux which is normal to the surface of the ocean, and the vertical modes capture vertical mixing and diffusion. Because we are only considering a regional coastal section of the ocean for which the dynamics are primarily driven by surface forcing, it is a reasonable assumption to only use a few modes. Furthermore, it can be proven that PCA results in an optimal

orthogonal transformation that captures maximum variance.

At each horizontal location i, (x_i, y_i), the temperature field is discretized into n depths and m time steps.

$$\mathbf{T}_i = \begin{bmatrix} T(z_1,t_1) & ... & T(z_1,t_m) \\ T(z_2,t_1) & ... & T(z_2,t_m) \\ & ... & \\ T(z_n,t_1) & ... & T(z_n,t_m) \end{bmatrix} \tag{1}$$

Using this data matrix, we evaluate the eigenvectors.

$$\mathbf{T}_i \mathbf{T}_i^T \boldsymbol{\phi}_{ij} = \lambda \boldsymbol{\phi}_{ij}, j = 1, ..., n \tag{2}$$

Finally, for each location i, the subsurface structure of the temperature is represented using 2 vertical modes and a mean temperature mode.

$$\mathbf{T}_{i,proj}(t) = \sum_{j=1}^{2} q_{ij}(t)\boldsymbol{\phi}_{ij} + \bar{\mathbf{T}}_i(t) \tag{3}$$

The eigenvalues obtained from the decomposition confirm that we have a low rank problem as the first two modes capture more than 85% of the data's energy and are sufficient for reconstructing the temperature field (Figure 4 (b)). The spatial modes $\boldsymbol{\phi}_{ij}$ represent the vertical structure of the field and vary with respect to the horizontal location. The first mode roughly corresponds to the thermocline (Figure 4 (d)). The coefficients $q_{ij}(t)$ and mean temperature $\bar{\mathbf{T}}_i(t)$ are functions of time and are extracted from the reanalysis data set via projection. The vertical temperature profiles of three (x_i, y_i) locations in Figure 5 suggest that there is good agreement between the original reanalysis and the reduced-order PCA projection. The error between the PCA projection and the original reanalysis field is also shown in Figure 6 for different sigma levels. For the case where there is no full 3D information, a functional relationship between surface information and these coefficients needs to be determined. This is the scope of the next section.

4.2. Machine Learning Functional Relationships Between PCA Coefficients and Surface Temperature

Next, we machine learn a functional relationship between the surface temperature and the temperature over depth at each horizontal location i, (x_i, y_i). We choose surface temperature as the input of the neural network

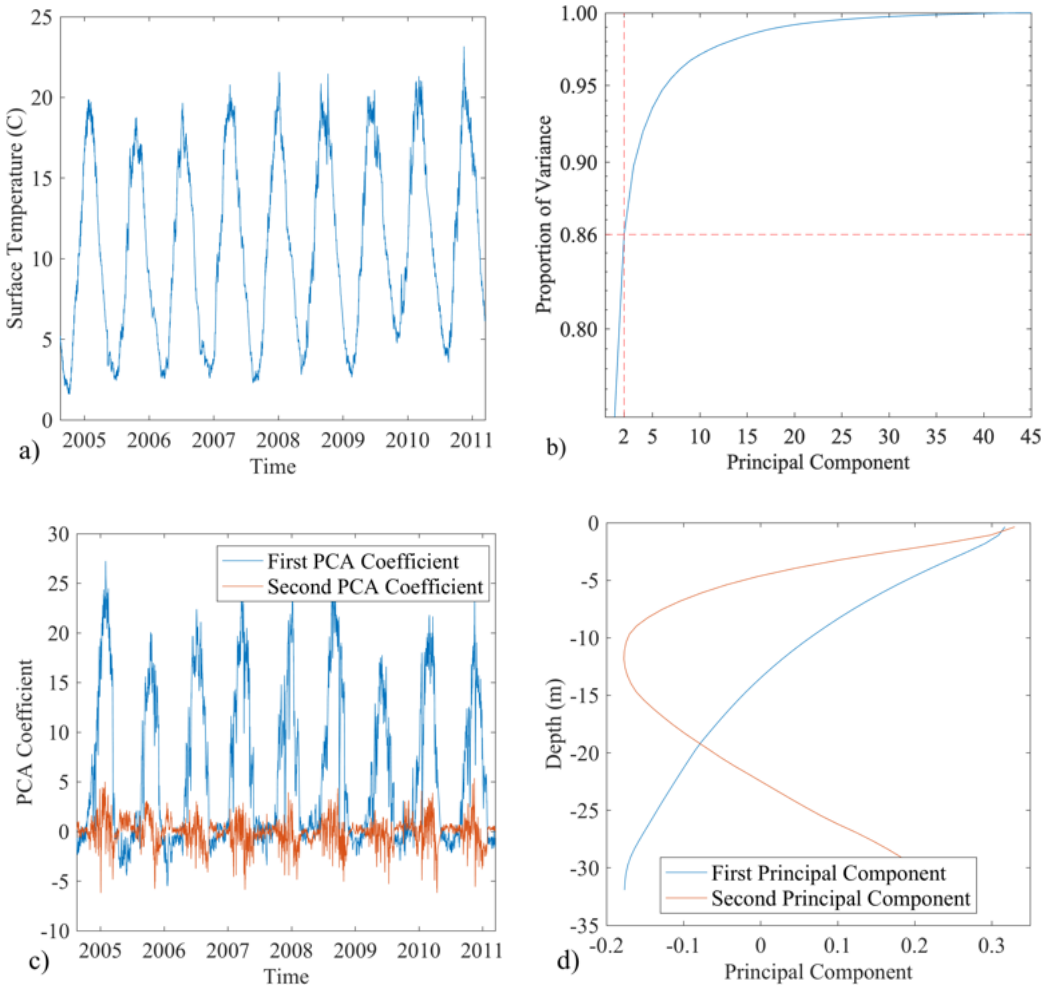

Figure 4: **Vertical Order-reduction at (42.41N, 70.86W).** Time series of surface temperature (a), energy distribution of the vertical modes (b), time series of PCA coefficients obtained by projection of the reanalysis temperature field (c), and first two vertical modes (d) at one (x_i, y_i) location.

because it is readily accessible from sensor measurements. We also build a second neural network to predict the associated standard deviation and estimate the uncertainty of our predictions. These uncertainties exclusively model the error made by the neural network in modeling the vertical PCA coefficients.

Recent developments in machine learning have increased the popularity of using neural networks to model geophysical processes [34, 10, 12, 11].

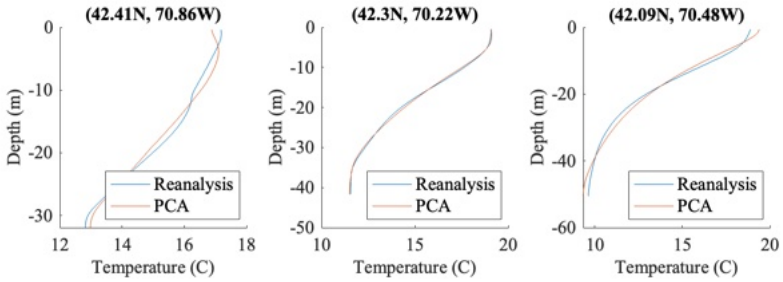

Figure 5: **PCA Vertical Profiles.** The vertical profiles of the reanalysis data and the PCA projection are shown for September 13th, 2012 at three different (x_i, y_i) locations. Corresponding locations are shown on the map in Figure 8.

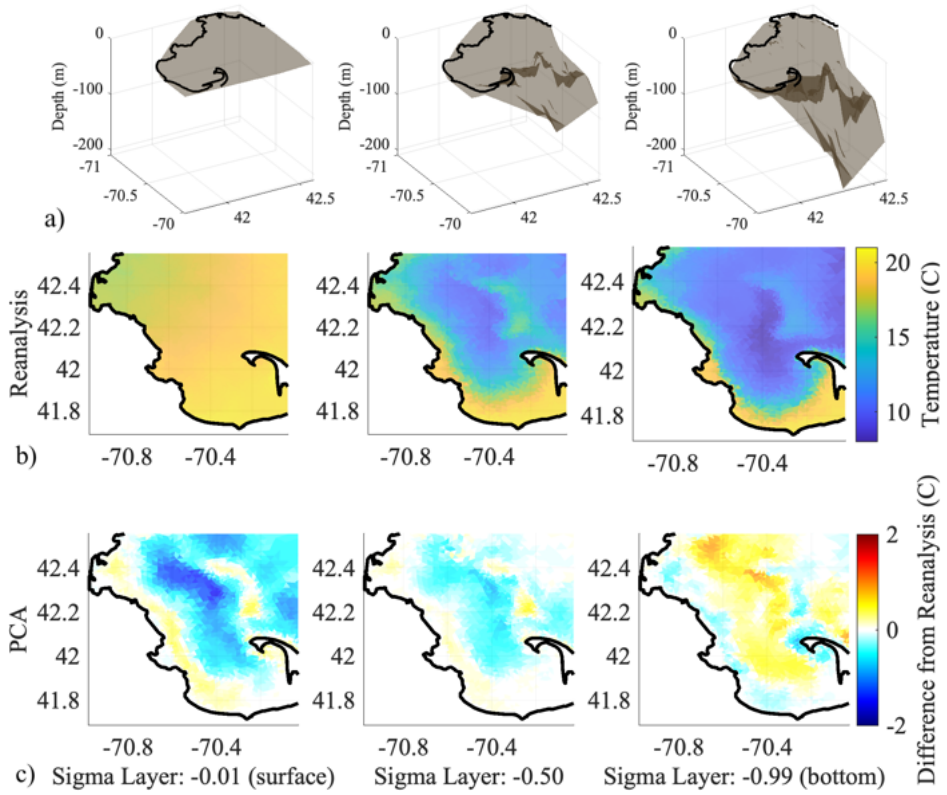

Figure 6: **PCA Projection Error.** The reanalysis data (b) and the difference between the PCA projection and the reanalysis (c) are plotted for September 13th, 2012 at three different sigma layers.

We specifically build a neural network that predicts the mean and standard deviation of the PCA coefficients $q_{ij}(t)$ and mean temperature $\bar{\mathbf{T}}_i(t)$ obtained in the previous section. Many studies have focused on the use of neural networks to predict such time-varying PCA coefficients [35, 36, 37, 38, 39].

In this project, we build a temporal convolutional network (TCN), a type of convolutional neural network (CNN) that performs convolutions on one dimensional time series data. Unlike a traditional CNN, a TCN is causal which is useful for modeling dynamic systems [40]. TCNs have also been shown to outperform other recurrent neural networks for sequence modeling [41, 42, 43]. As such, they are increasingly being used in geophysical applications [44, 45, 46]. We adapt the Stochastic Machine Learning (SMaL) code from Wan et al. and retain the same residual block architecture (Figure 7) [40]. The data are standardized before training for improved results. The

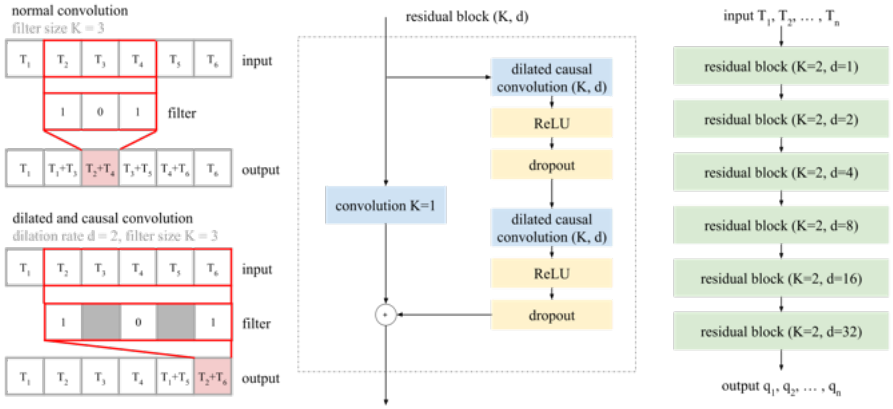

Figure 7: **Architecture of TCN.** The TCN is built with residual blocks that consist of a sequence of two convolutional layers with ReLU activation and a dropout. The dilation factor of each residual block is doubled at each depth.

batch size of the neural network, which is the number of samples that are used in a training set during one pass, is set to 5 because a smaller batch size is better for model generalizability and a larger number did not improve the results. The filter width is set to 2. In a standard CNN, a small filter width results in a small receptive field (receptive field refers to the amount of data that contributes to a feature of the neural network). In a TCN, the dilation factor is doubled at each depth to cover many different time scales, so the receptive field becomes larger. The small filter width thus reduces compu-

tational costs and improve generalizability. The dropout layer of the neural network is set to have a probability of 0.5 for regularization. This means that each weight has a probability of 0.5 of being ignored in the network, so the weights become decorrelated. The depth of the network determines how many nonlinear activations are performed. From the results of numerical experiments, a depth of 6 layers resulted in the lowest test error to adequately represent the underlying physical phenomena.

4.2.1. Loss functions for neural network training

Typically, the weights of a neural network are obtained by minimizing a loss function that quantifies the error between the true data and the model predictions.

$$J(\boldsymbol{\theta}) = \frac{1}{T} \sum L(\hat{\mathbf{y}}(\boldsymbol{\theta}) - \mathbf{y}). \tag{4}$$

Here, we build two neural networks at each location i, (x_i, y_i), one for the mean and one for standard deviation. We emphasize that each horizontal location is treated separately to account for spatial inhomogeneities. We train each network sequentially because we require the mean prediction to train the second neural network for the standard deviation. Furthermore, we optimize different loss functions for each network. To predict the mean of the PCA coefficients, we minimize the mean absolute error (MAE), a standard loss function for neural networks.

$$J_{MAE} = \frac{1}{m} \sum |\hat{y} - y|. \tag{5}$$

To predict the standard deviation of the PCA coefficients, we minimize the mean negative anomaly correlation coefficient (MNACC) [40]. It is a correlation-based loss function, so it does not scale with magnitude, therefore more effectively penalizing anomalies.

$$J_{MNACC} = \frac{1}{m} \sum \frac{\sum(\hat{z} - [\hat{z}])(z - [z])}{\sqrt{\sum(\hat{z} - [\hat{z}])^2}\sqrt{\sum(z - [z])^2}} \tag{6}$$

$$z = y - y^{ref} \tag{7}$$

Here, the reference y^{ref} is the cyclic mean, and for ocean temperature it corresponds to the annual variation due to seasons. Without a reference, this loss reduces to the Pearson correlation coefficient, another standard loss function in many machine learning applications.

4.2.2. Choice of number and location of input points

While the weights and biases can be found by optimizing a loss function, other parameters of the neural network need to be fine-tuned through discrete numerical experiments. For example, the choice of input points affects the output of the neural network. Many ocean models treat the ocean as being stratified, so these models do not include interactions in the horizontal direction. However, the governing physical equations of the system imply that the temperature gradients in the x and y direction have the potential to contribute to the vertical temperature profile. As such, we include neighboring points in the input of the neural network to produce a non-local parametrization. To select the neighboring points, we first choose a radius Δ and then compute the points $(x+\Delta, y)$, $(x-\Delta, y)$, $(x, y+\Delta)$, and $(x, y-\Delta)$. Then we find the points that are closest to these points and assign those to be the neighbors of the point of interest. We perform numerical experiments to find the number and location of input points that are best suited for generalizability. We first test the neural network with one, two, three, and five neighborhood input points. Then, we experiment with the distance between the input points and the point of interest. After completing the experiments, we choose the number of points and distance between points that result in the lowest testing error. We perform these experiments on three (x_i, y_i) pairs in the neighborhood domain, denoted A, B, and C in Figure 8, and we adopt the same parameters for the models of all other (x_i, y_i) pairs. From the results of the numerical experiments, we build the inputs of the TCN with the surface temperature of four additional nearby points for which the distance is between nine and ten kilometers.

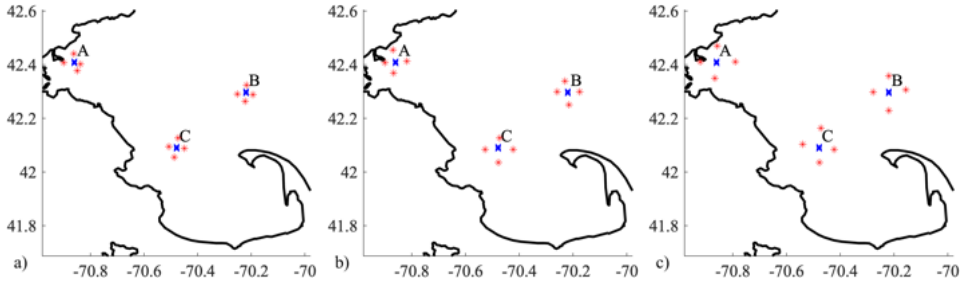

Figure 8: **Input Points**. The input of the neural network consists of the surface temperature at four nearby points in addition to the surface temperature at the corresponding point of interest. Different radii, shown here, are tested through numerical experiments.

4.2.3. Choice of memory for the neural network

The temporal convolutional network also has parameters associated with the dynamics in time, i.e. how much memory from the input should be retained in order to achieve the best prediction. Starting with time series arrays of surface temperature, T_S, PCA coefficients, q_1 and q_1, and mean temperature, $\bar{T}$,

$$\mathbf{x} = \begin{bmatrix} T_S(t_0) & T_S(t_1) & T_S(t_2) & ... & T_S(t_n) \end{bmatrix} \tag{8}$$

$$\mathbf{y} = \begin{bmatrix} q_1(t_0) & q_1(t_1) & q_1(t_2) & ... & q_1(t_n) \\ q_2(t_0) & q_2(t_1) & q_2(t_2) & ... & q_2(t_n) \\ \bar{T}(t_0) & \bar{T}(t_1) & \bar{T}(t_2) & ... & \bar{T}(t_n) \end{bmatrix} \tag{9}$$

we build matrices of smaller sequences on which we apply the convolutional filter.

$$\mathbf{x}_{TCN} = \begin{bmatrix} T_S(t_0) & T_S(t_1) & ... & T_S(t_m) \\ T_S(t_s) & T_S(t_{s+1}) & ... & T_S(t_{s+m}) \\ & ... & & \\ T_S(t_{n-m}) & ... & & T_S(t_n) \end{bmatrix} \tag{10}$$

$$\mathbf{y}_{TCN} = \begin{bmatrix} q_1(t_0) & q_1(t_1) & ... & q_1(t_m) \\ q_1(t_s) & q_2(t_{s+1}) & ... & q_1(t_{s+m}) \\ & ... & & \\ q_1(t_{n-m}) & ... & & q_1(t_n) \end{bmatrix} \tag{11}$$

When building these smaller sequences, we have the ability to choose how much data to use which affects the performance of the neural network. The sampling rate determines how many time steps to skip within an input time series, the stride, s, determines how many time steps to skip between each successive time series, and the memory length scale, m, determines how many points back in time to consider in one time series. Again, we perform numerical experiments to find the values for these parameters that result in the lowest testing error. The memory length scale is set to be 20 days, and the sampling rate and stride are both set to 1 day. In our final model, each PCA coefficient is predicted using the surface temperature from all of the data from the 20 previous days, a choice that is consistent with ocean time scales [47].

4.2.4. Surface temperature constraint

The output of the neural network is used to reconstruct the full 3D temperature field, but we want to ensure that the prediction at the surface of

the ocean matches exactly the input surface temperature:

$$q_1\boldsymbol{\phi}_1(z=0) + q_2\boldsymbol{\phi}_2(z=0) + \bar{T} = T(z=0) \tag{12}$$

This requirement can be written as a constraint function

$$f(\hat{\mathbf{y}}(\boldsymbol{\theta})) = q_1\boldsymbol{\phi}_1(z=0) + q_2\boldsymbol{\phi}_2(z=0) + \bar{T} - T(z=0) \tag{13}$$

We embed the soft constraint $\lambda|f(\hat{\mathbf{y}}(\boldsymbol{\theta}))|$ into the loss function

$$J(\boldsymbol{\theta}) = \frac{1}{T}\sum L(\hat{\mathbf{y}}(\boldsymbol{\theta}) - \mathbf{y}) + \lambda|f(\hat{\mathbf{y}}(\boldsymbol{\theta}))| \tag{14}$$

From numerical experiments, we find that the neural network is able to match the surface temperature without the soft constraint. Nevertheless the inclusion of the constraint guarantees that there will be no significant deviations.

4.2.5. Results of the neural network training

By using additional nearby points and previous time steps, we create a non-local parametrization in both space and time. To train a neural network, the inputs are typically split into training, validation, and testing sets. The training inputs are used to optimize the weights of the model, the validation inputs are used to select the best hyperparameters, and the test inputs are used to evaluate the model on unseen data. The neural network for this model is built using four years of data for training (mid 2005 - mid 2009), one and a half years for validation (mid 2009 - 2011), and two and a half years for testing (2011 until mid-2013) (Figure 9). The error associated with the neural network predictions are calculated relative to both the original reanalysis data and the PCA reconstruction (Table 1). The predicted time series for a representative horizontal location, as well as the predicted standard deviation, are shown in Figure 9. The raw outputs of the neural network are simply the PCA coefficients and mean temperature, as well as their standard deviations. However, these raw outputs can be combined with the PCA modes to reconstruct the full 3D temperature field. The vertical profile at three (x_i, y_i) locations is shown in Figure 10 and the spatial fields for three sigma layers are shown in Figure 11. The agreement between the original reanalysis data and the TCN prediction is worse at the surface, but the associated uncertainty is also higher. The model does not perform equally for all (x_i, y_i) locations, but the total root mean squared error for the test data is 0.55 °C.

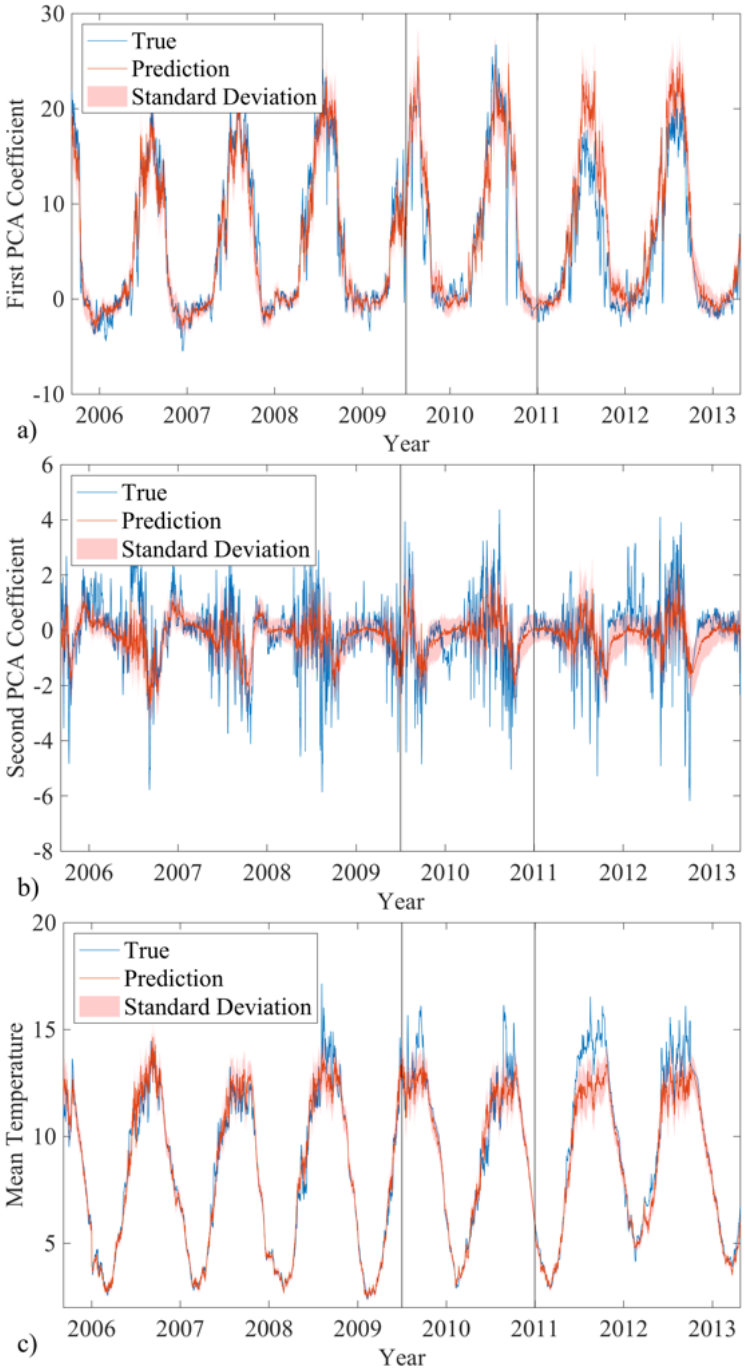

Figure 9: **TCN Predictions at (42.41N, 70.86W)**. The first (a) and second (b) PCA coefficients and the mean temperature (c), as well as their standard deviation, are predicted for the reanalysis data. The black lines delineate the training, validation, and test sets, respectively.

Table 1: Neural Network Model Evaluation

		MAE ($°C$)			RMSE ($°C$)		
y (target)	$\hat{y}$ (output)	train	val.	test	train	val.	test
FVCOM	TCN	0.2088	0.2999	0.3185	0.3552	0.4961	0.5470
PCA	TCN	0.1942	0.2846	0.3078	0.3359	0.4777	0.5364

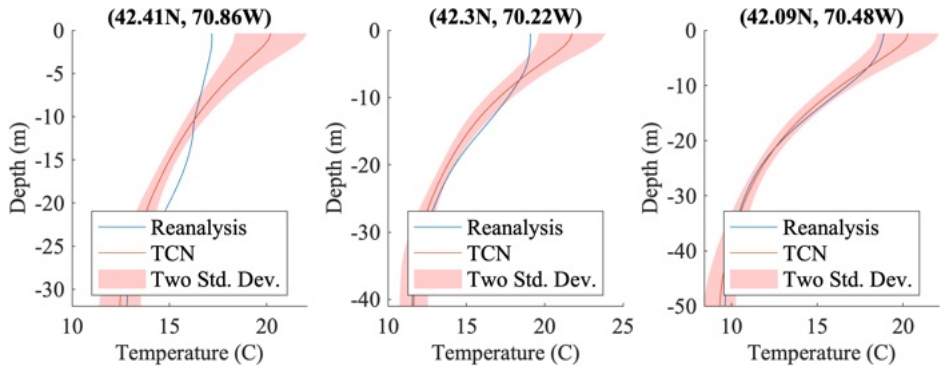

Figure 10: **TCN Vertical Profiles.** The vertical profiles of the reanalysis data and the TCN reconstruction are shown for September 13th, 2012 at three different (x_i, y_i) locations. The red shading corresponds two standard deviations predicted by the TCN.

For each (x_i, y_i) pair, it takes one minute to train a neural network on a standard CPU. Once the neural network is fully optimized, it only takes a few seconds to make a prediction.

4.3. Filling Gaps in the Surface Sensor Data

The next step in the framework is to estimate the full 2D surface temperature field. Satellites provide useful information about surface temperature, but they are significantly affected by cloud coverage. Work has been done to improve measurements from satellites and to blend data from multiple satellites [48, 49]. In many projects, in-situ buoy measurements are used to either validate or improve the accuracy of models [50, 51, 52, 53, 54, 55, 25]. One recent approach that has been shown to obtain quick, accurate, and useful results is Gaussian process regression (GPR) [26, 56]. GPR is a Bayesian approach which can estimate smooth nonlinear functions and provide an uncertainty measurement for a given prediction. Unlike optimal interpolation or objective mapping, GPR does not require background information to create the data correlation matrix. One downside of using GPR is that the matrix

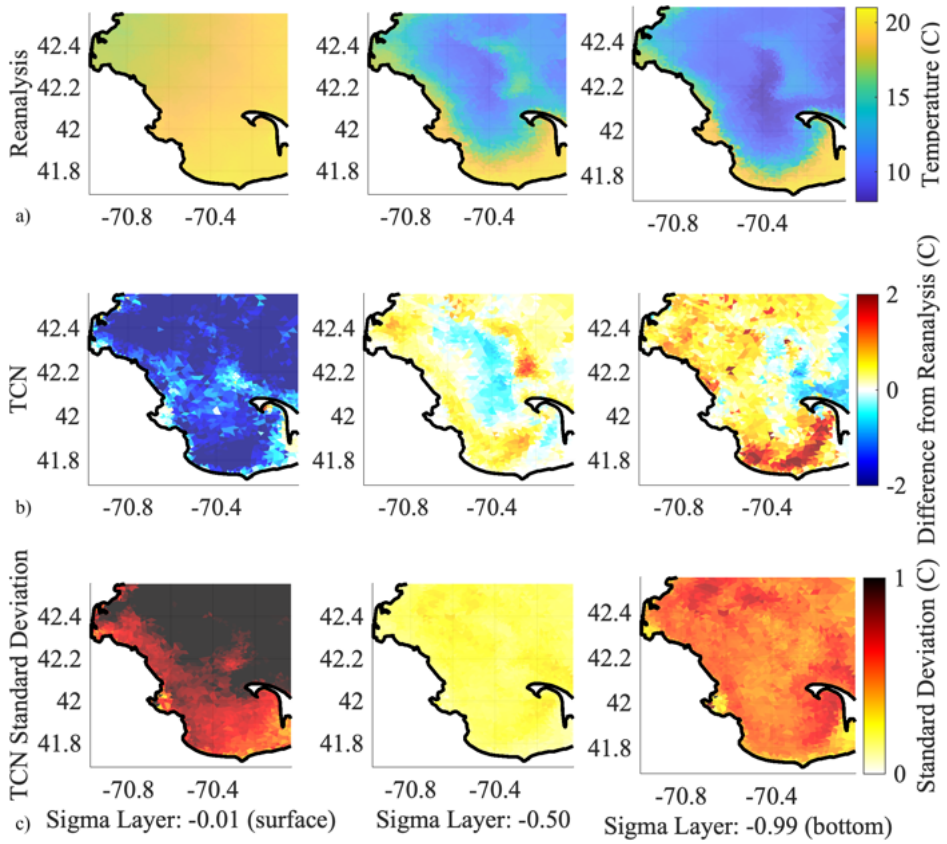

Figure 11: **TCN 3D Reconstruction of the Temperature Field.** From top to bottom, the reanalysis data (a), TCN prediction (b), and TCN standard deviation prediction (c) are plotted for September 13th, 2012 at three different sigma levels.

inversion can become slow for large numbers of input points. However, GPR is very successful for problems with a low number of input points. Furthermore, unlike with other machine learning techniques, the hyperparameters of the model, specifically those of the kernel, have an intuitive physical meaning and can be set according to properties of the system. Here, we use GPR to *extrapolate* the available surface data. Note that we use the term extrapolation (as opposed to interpolation) since in many cases the available surface data are so sparse that interpolation is not meaningful. The features (inputs) of the model are the longitude, latitude, and time, and the value that is being predicted is the surface temperature. For points at which sensor data are available, we keep the original data, but for points at which there are no

measurements, we predict the temperature using nearby points both in time and space.

4.3.1. Gaussian process regression

The mean and variance are predicted using the kernel, K, which relates all of the available data points [57]. Specifically, the mean prediction is

$$\bar{\mathbf{f}}_* = \mathbf{m}(X_*) + K(X_*, X)[K(X, X) + \sigma_n^2 I]^{-1}(\mathbf{y} - \mathbf{m}(X)) \tag{15}$$

and the variance is

$$cov(\mathbf{f}_*) = K(X_*, X_*) - K(X_*, X)[K(X, X) + \sigma_n^2 I]^{-1} K(X, X_*) \tag{16}$$

For our application, the mean function $\mathbf{m}(X_*)$ is explicitly set to be the spatial mean (Figure 12) of the available satellite data. To avoid inverting

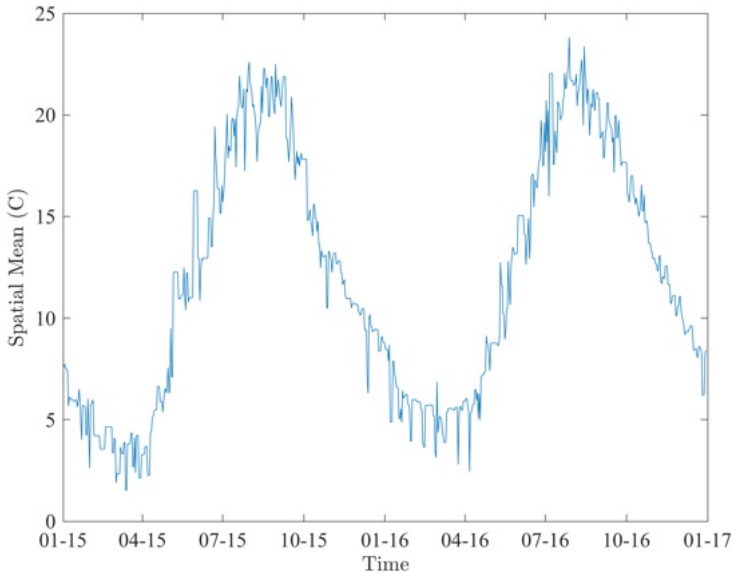

Figure 12: **Spatial Mean of the Satellite Data**. The mean over the whole spatial domain is plotted for each day.

large matrices, we keep the size of the kernel small by building a new GPR model for each time step. The input data consists of the available data on the day of interest, data from one day before and one day after. In other words, we only use data from three days to predict the surface temperature

for one day, and we repeat this process for all time steps. The features for each time step k are

$$\begin{bmatrix} x_i & y_i & t_{k-1} \\ x_i & y_i & t_k \\ x_i & y_i & t_{k+1} \end{bmatrix} = \begin{bmatrix} x_i & y_i & -1 \\ x_i & y_i & 0 \\ x_i & y_i & 1 \end{bmatrix} \tag{17}$$

where (x_i, y_i) are all of the available spatial points at each time step k.

4.3.2. Hyperparameter selection

For the kernel, we use the radial basis function (RBF) with automatic relevance determination as the covariance function.

$$cov(f(\mathbf{x_p}), f(\mathbf{x_q})) = k(\mathbf{x_p}, \mathbf{x_q}) = \sigma_f^2 * exp(-\frac{1}{2}(\mathbf{x_q} - \mathbf{x_p})^T \theta (\mathbf{x_q} - \mathbf{x_p})) \tag{18}$$

The signal variance σ_f and characteristic lengthscales $\mathbf{\Theta}$ are hyperparameters of the model. The characteristic lengthscale represents how far apart two points need to be for their function values to become uncorrelated. The inverse of the lengthscale represents how relevant a given feature is. The automatic relevance determination chooses different characteristic lengthscales for each input to determine the relevant inputs. As such, there are three characteristic lengthscales: one for the input longitude, one for the input latitude, and one for the input time. The noise variance, σ_n, is not a parameter of the kernel, but it can also be considered one of the hyperparameters of the whole system. This parameter assumes that we know the uncertainty of the sensors.

Typically, the hyperparameters are found by optimizing the following loss function.

$$\log p(\mathbf{y}|X) = -\frac{1}{2}\mathbf{y}^T(K + \sigma_n^2 I)^{-1}\mathbf{y} - \frac{1}{2}\log|K + \sigma_n^2 I| - \frac{n}{2}\log 2\pi \tag{19}$$

However, to avoid overfitting and to generalize the models, we manually set the same hyperparameters for all days, changing only the training data for each day. For days with no available training data, we take the average over 10 days (5 previous and 5 following days). For the spatial lengthscales, we choose a value of 0.25 degrees or 25 kilometers, which is equivalent to six "gridpoints" or "pixels," where one gridpoint is the spatial granularity. This choice assigns more weight to spatial points that are within 25 kilometers of the point of interest; it corresponds to the mesoscales of the ocean [47].

For the time lengthscale, we set the hyperparameter to one day. Finally, we choose to set the noise variance to $\sigma_n = 0.1$, and we set the signal variance to $\sigma_f = 0.3$ by taking the average of minimizing the objective function over all submodels.

4.3.3. Multi-fidelity Gaussian process regression

We improve the model by incorporating the buoy data, which has lower uncertainty than the satellite data, through a recursive multi-fidelity Gaussian process regression scheme described in Babaee et al [26]. Given s levels of fidelity, the model with the lowest fidelity is denoted with $\mathbf{x_1}$, y_1, $\bar{\mathbf{f}}_{*1}$, and the model with the highest fidelity is denoted with $\mathbf{x_s}$, y_s, $\bar{\mathbf{f}}_{*s}$ [58]. The prediction for the model with the lowest fidelity follows the Gaussian process regression steps from equations (15) and (16)

$$\bar{\mathbf{f}}_1(\mathbf{x}_*) = K(X_*, X_1)[K(X_1, X_1) + \sigma_{n1} I]^{-1}\mathbf{y}_1, \tag{20}$$

with covariance

$$cov(\bar{\mathbf{f}}_1) = K(X_*, X_*) - K(X_*, X_1)[K(X_1, X_1) + \sigma_{n1} I]^{-1}K(X_1, X_*). \tag{21}$$

Each following model has the form

$$\bar{\mathbf{f}}_t(\mathbf{x}_*) = \rho_{t-1}\bar{\mathbf{f}}_{t-1} + \delta_t \qquad t = 2, ..., s \tag{22}$$

In this project, there are only two levels of fidelity, so the prediction for the highest level of fidelity, $s = 2$, can be computed with the following equation

$$\bar{\mathbf{f}}_2(\mathbf{x}_*) = \rho\bar{\mathbf{f}}_1(\mathbf{x}_*) + \mu_d + K(X_*, X_2)[K(X_2, X_2) + \sigma_{n2} I]^{-1}(\mathbf{y} - \rho\bar{\mathbf{f}}_1(\mathbf{x}_2) - \mu_d). \tag{23}$$

Its corresponding covariance is

$$cov(\bar{\mathbf{f}}_2) = \rho^2 cov(\bar{\mathbf{f}}_1) + K(X_*, X_*) - K(X_*, X_2)[K(X_2, X_2) + \sigma_{n2} I]^{-1}K(X_2, X_*), \tag{24}$$

where, ρ and μ_d are hyperparameters that are different for each level of fidelity. Like σ_f and θ of the covariance function, ρ and μ_d can be chosen through maximum likelihood estimation or other optimization techniques. We use the Emukit [59] Python package, which builds on the GPy Python package, to build the multi-fidelity model. Such techniques have already been used to predict surface temperature, but our model differs with respect to the choice of input points [26]. Babaee et al. used all of the available data to

build a model while we only use spatial points from three time steps. Because we use less data at each time step, our model is faster at making predictions, and therefore more practical for real-time modeling. For consistency, we set ρ and μ_d to be the same as those from the optimized model in Babaee et al. [26].

4.3.4. Results of surface temperature extrapolation

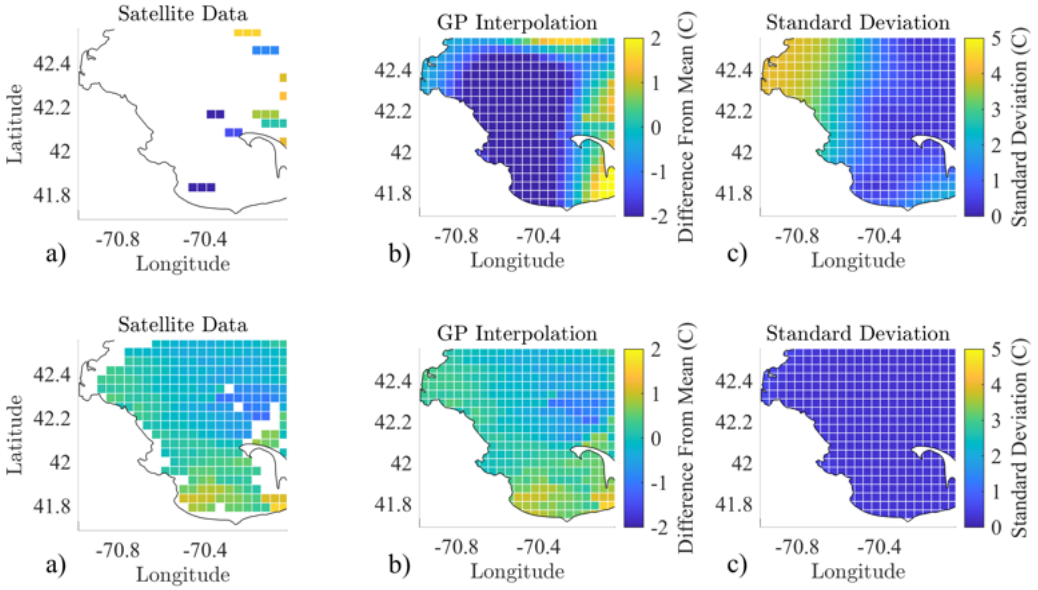

Figure 13: **Results of Extrapolation for Two Different Days**. The available satellite and buoy data are extrapolated to obtain a surface temperature field over the full domain. Each row represents a different day with high cloud coverage (March 8th 2016 (a)) and low cloud coverage (September 13th 2016 (b))

The results of the extrapolation are shown in Figure 13 both for a day with high cloud coverage (March 8th, 2016) and for a day with minimal cloud coverage (September 13th, 2016). As expected, the uncertainty of the extrapolation is higher in regions with significant cloud coverage. The root mean squared error between our new multi-fidelity model and the old multi-fidelity from Babaee et al. is 0.46° C across the three stations that are held out for validation. Overall, the results from using just three days compare favorably with those from Babaee et al., while the new model is significantly faster.

5. Results and Evaluation of the Full 3D Temperature Field

Finally, we utilize the real-time estimate for surface temperature obtained from GPR as input to the TCN to obtain the PCA coefficients and the mean temperature, as well as their uncertainty, at each horizontal location for the day of interest. To achieve this, we build a time series of surface temperature at each point for which satellite measurements are available, and we individually predict the PCA coefficients for each of these surface temperature time series. The spatial resolution of the model over longitude and latitude can be chosen during the GPR step, but here we choose to use the spatial resolution of the satellite data. The neural network predictions from the real-time sensor measurements are plotted in Figure 14 for 2015 and 2016. The neural network provides an estimate for the mean and standard deviation (red shading) of the quantities of interest.

The predicted PCA coefficients are then projected onto the deterministic PCA modes and summed with the predicted PCA mean to reconstruct the full 3D temperature and uncertainty fields. When estimating surface temperature, we left out measurements from three stations (N04, F13, F29 from Figure 2). We evaluate the results of our full model by comparing the predictions from the neural network to these withheld in-situ measurements that were never seen during the training phase. These stations also collect measurements for temperature over multiple depths, which we divide into shallow (0-25m), medium (25-45m), and deep (>45m).

The vertical profiles for stations N04 and F13 are shown in Figure 15, and the results of the full 3D reconstruction are plotted in Figure 18 for March 8th, 2016 and September 16th, 2016 at three sigma layers. We compare the accuracy of our model to the climatological mean, and we find that the model performs well for most points (e.g. station N04), but the predictions are worse for points that are near estuaries or other unusual geographic or human features (e.g. station F13). The mean absolute error of our predictions is 1.37°C, the median absolute error is 0.97°C, the root mean squared error is 1.73°C, and 79% of predictions fall withing two degrees of the truth. This root mean squared error is lower than the 2.28 °C found by Li et al. for FVCOM [18]. The model produces more outliers than the climatological mean, but eliminates the bias that is present in the estimates from the climatological mean (Figure 15 (a)).

We observe in Figure 16, which includes data from all MWRA stations (not just the withheld stations), that the model performs best for days with

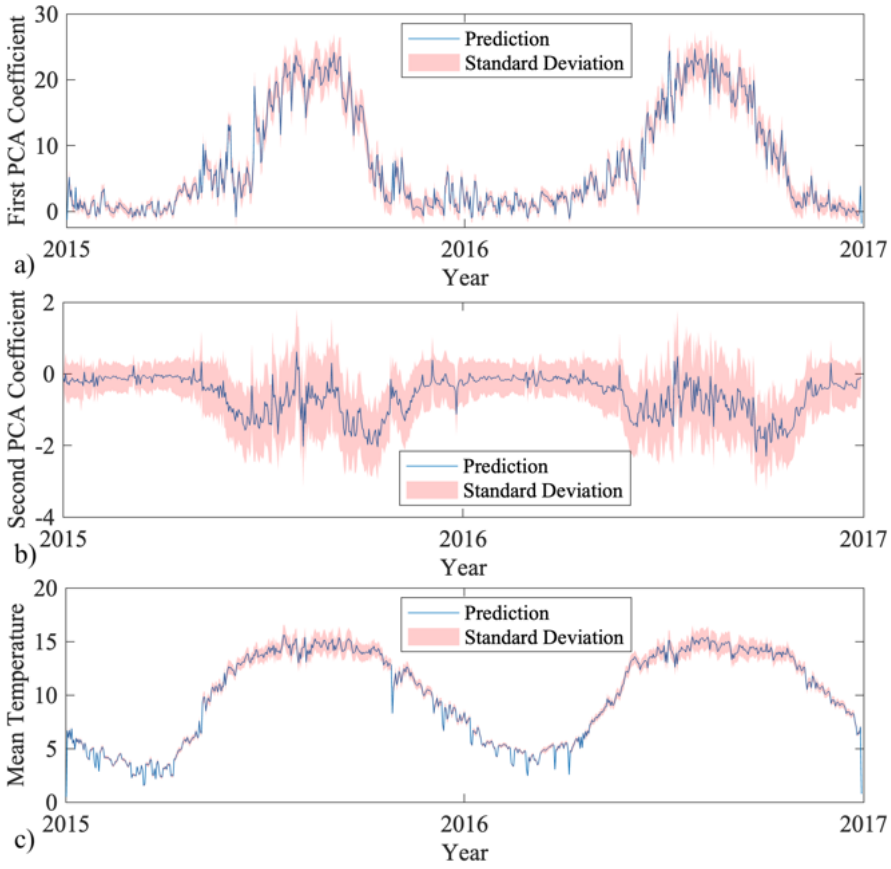

Figure 14: **TCN Predictions from Satellite Measurements at (42.41N, 70.86W)**. The first (a) and second (b) PCA coefficients, the mean temperature (c), as well as their uncertainties (red shading) are predicted for the available satellite surface temperature.

the most amount of available satellite data (80-100%). We also show the mean temperature and mean standard deviation (averaged over space), compared to the amount of available satellite data, as a function of time in Figure 17. It is possible that with additional and possibly targeted in-situ stations, the model could reduce the numbers of outliers. We also find that there are no significant improvements from including the buoy measurements when modeling the surface temperature. However, the framework allows us to seamlessly incorporate data from multiple sources which could be useful in applications where fewer measurements are available. Furthermore, the framework provides an estimate for uncertainty given the level of accuracy of each sensor.

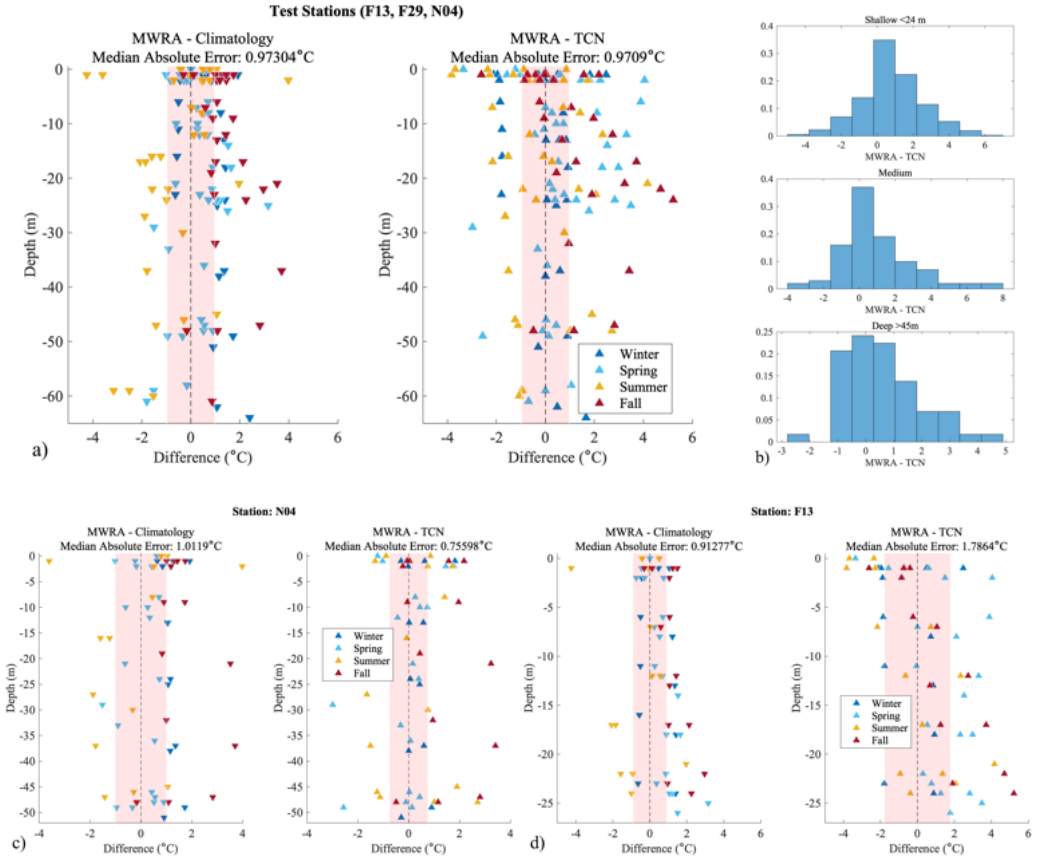

Figure 15: **Vertical Profiles of Error** Difference between MWRA buoy measurements and climatological mean compared to the difference between the MWRA buoy measurements and the neural network predictions for all stations (a) station N04 (c) and station F13 (d). The color of the markers indicates the season, and the histogram of the absolute error is shown in (b).

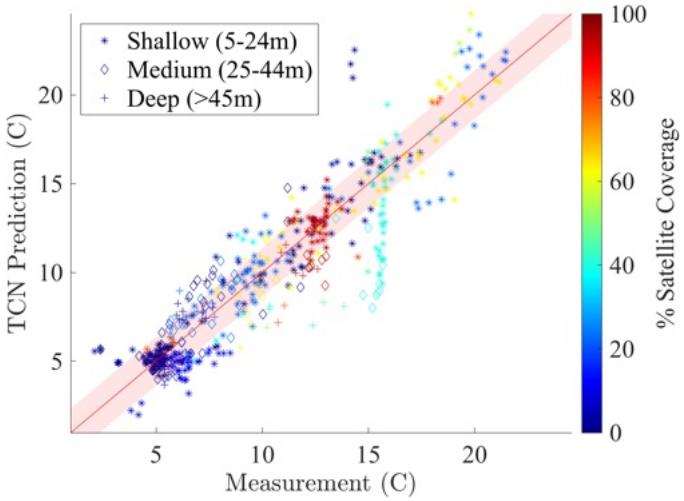

Figure 16: **Comparison Between MWRA Measurements and TCN Predictions** Comparison between buoy measurements and predictions from the neural network at different depths for days with different amounts of satellite coverage. Points along the line $x = y$ correspond to predictions with the lowest error. The red shading corresponds to the standard deviation of the absolute error.

6. Conclusions

We introduced a fast and accurate framework, based on recently developed machine learning techniques and reanalysis data obtained from comprehensive ocean models, to reconstruct 3D ocean temperature fields from real-time sensor measurements of surface temperature. We compared the results from our framework to in-situ measurements, and we found that the error associated with our predictions is comparable to that of other state of the art models that are significantly more expensive. In the future, we plan to use our model's estimates of uncertainty to make decisions about the system, a process often referred to as active sampling or optimal sampling. For example, we can define and optimize an acquisition function to decide where to place additional sensors or plan the trajectory of an ocean drifter. In some cases, properly formulated acquisition functions can be leveraged to identify extreme values [60]. Overall, the developed model is important for monitoring general ocean health, and the techniques described can be used for other geophysical systems.

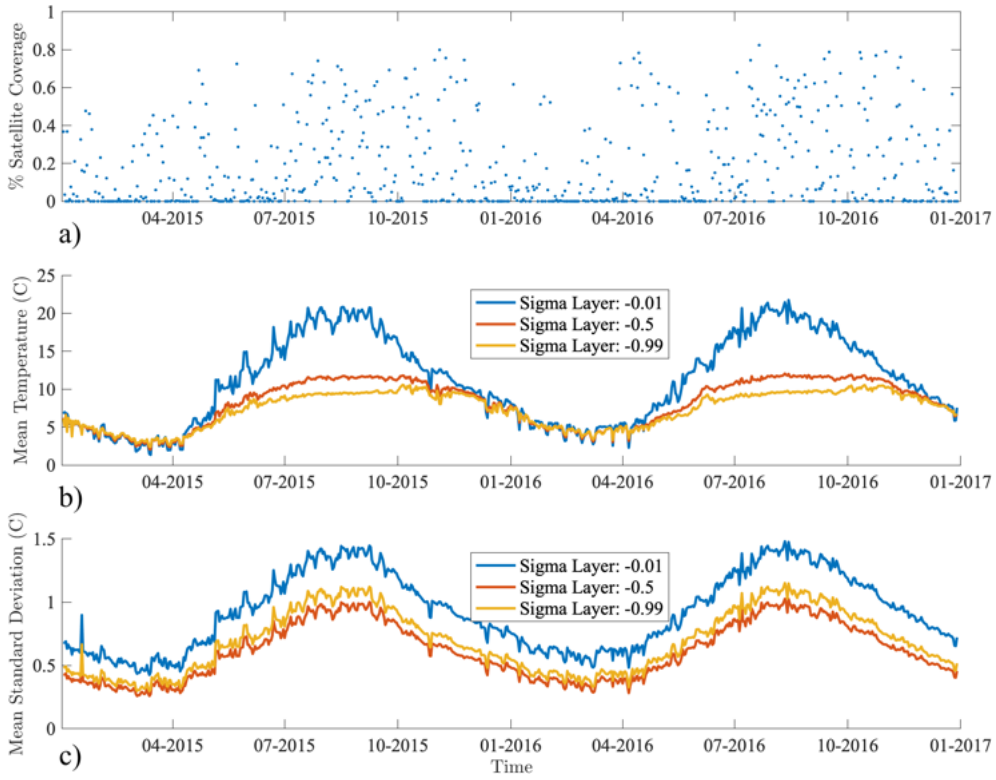

Figure 17: **Mean Temperature and Standard Deviation at Three Sigma Layers.** The mean temperature (b) and standard deviation (c) are plotted for sigma layers -0.01, -0.5, and -0.99, and the percent of satellite coverage (a) is shown for each corresponding day.

Appendix A. Open Research

The Finite Volume Community Ocean Model (FVCOM) data are available from the The Northeast Coastal Ocean Forecast System (NECOFS): http://fvcom.smast.umassd.edu/necofs/. The Moderate-resolution Imaging Spectroradiometer (MODIS) SST data come from the NASA EOSDIS Physical Oceanography Distributed Active Archive Center (PO.DAAC) at the Jet Propulsion Laboratory, in Pasadena, CA (https://doi.org/10.5067/MODST-1D4N4). The MWRA measurements are accessible at https://www.mwra.com/harbor/html/ wq_data.htm. The temporal convolutional network was built with Tensorflow, and the multi-fidelity Gaussian process regression was implemented with Emukit.

Appendix B. Acknowledgments

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. 1745302, MIT Sea Grant, and the Harrington Fellowship. We thank Dr. Carolina Bastidas and Michael Defilippo for providing and pre-processing the MWRA buoy data. We also thank Professor George Karniadakis and Professor Hessam Babaee for stimulating discussions.

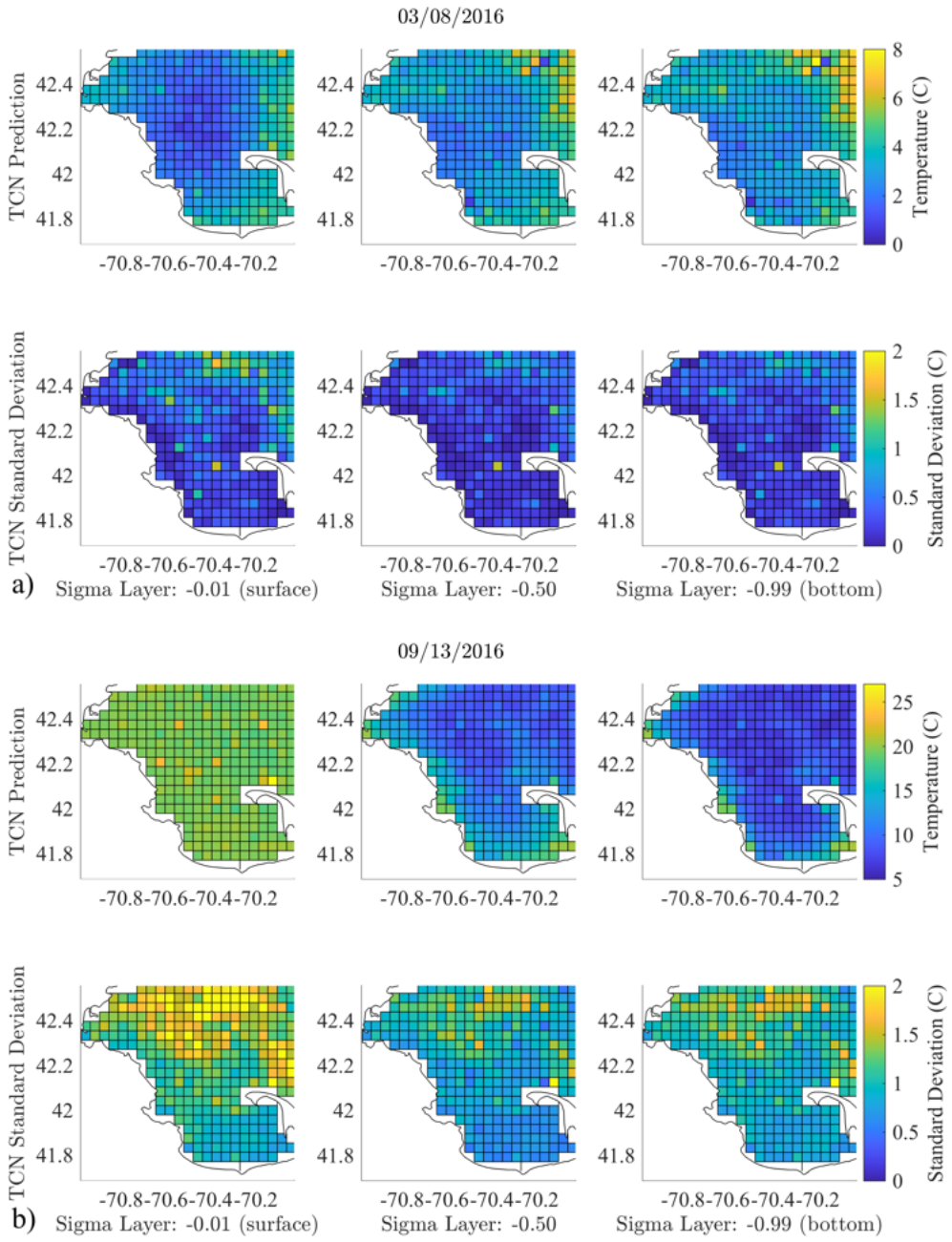

Figure 18: **3D Temperature Field Reconstruction From Real-time Measurements.** The predicted PCA coefficients are projected onto the corresponding modes and summed with the predicted mean temperature to reconstruct the full 3D temperature. The results and associated uncertainty are plotted for March 8th, 2016 (a) and September 13th, 2016 (b) at three sigma layers.

References

[1] M. R. Carnes, W. J. Teague, J. L. Mitchell, Inference of Subsurface Thermohaline Structure from Fields Measurable by Satellite, Journal of Atmospheric and Oceanic Technology 11 (1994) 551 – 566. Place: Boston MA, USA Publisher: American Meteorological Society.

[2] S. Guinehut, P. Y. Le Traon, G. Larnicol, S. Philipps, Combining Argo and remote-sensing data to estimate the ocean three-dimensional temperature fields—a first approach based on simulated observations, Journal of Marine Systems 46 (2004) 85–98.

[3] S. Guinehut, A.-L. Dhomps, G. Larnicol, P.-Y. Le Traon, High resolution 3-D temperature and salinity fields derived from in situ and satellite observations, Ocean Science 8 (2012) 845–857.

[4] H. Wang, G. Wang, D. Chen, R. Zhang, Reconstruction of Three-Dimensional Pacific Temperature with Argo and Satellite Observations, Atmosphere-Ocean 50 (2012) 116–128. Publisher: Taylor & Francis.

[5] J. Wang, G. R. Flierl, J. H. LaCasce, J. L. McClean, A. Mahadevan, Reconstructing the Ocean's Interior from Surface Data, Journal of Physical Oceanography 43 (2013) 1611 – 1626. Place: Boston MA, USA Publisher: American Meteorological Society.

[6] B. B. Nardelli, R. Santoleri, Methods for the Reconstruction of Vertical Profiles from Surface Data: Multivariate Analyses, Residual GEM, and Variable Temporal Signals in the North Pacific Ocean, Journal of Atmospheric and Oceanic Technology 22 (2005) 1762 – 1781. Place: Boston MA, USA Publisher: American Meteorological Society.

[7] A. Pascual, D. Gomis, Use of Surface Data to Estimate Geostrophic Transport, Journal of Atmospheric and Oceanic Technology 20 (2003) 912 – 926. Place: Boston MA, USA Publisher: American Meteorological Society.

[8] H. Yan, H. Wang, R. Zhang, J. Chen, S. Bao, G. Wang, A Dynamical-Statistical Approach to Retrieve the Ocean Interior Structure From Surface Data: SQG-mEOF-R, Journal of Geophysical Research: Oceans 125 (2020) e2019JC015840. _eprint: https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2019JC015840.

[9] H. Yan, R. Zhang, H. Wang, S. Bao, C. Bai, Practical Dynamical-Statistical Reconstruction of Ocean’s Interior from Satellite Observations, Remote Sensing 13 (2021).

[10] M. Sammartino, B. Buongiorno Nardelli, S. Marullo, R. Santoleri, An Artificial Neural Network to Infer the Mediterranean 3D Chlorophyll-a and Temperature Fields from Remote Sensing Observations, Remote Sensing 12 (2020).

[11] E. Pauthenet, L. Bachelot, K. Balem, G. Maze, A.-M. Tréguier, F. Roquet, R. Fablet, P. Tandeo, Four-dimensional temperature, salinity and mixed-layer depth in the Gulf Stream, reconstructed from remote-sensing and in situ observations with neural networks, Ocean Science 18 (2022) 1221–1244.

[12] B. Buongiorno Nardelli, A Deep Learning Network to Retrieve Ocean Hydrographic Profiles from Combined Satellite and In Situ Measurements, Remote Sensing 12 (2020).

[13] D. K. Gledhill, M. M. White, J. Salisbury, H. Thomas, B. Mook, J. Grear, A. C. Candelmo, R. C. Chambers, C. J. Gobler, C. W. H. A. L. King, N. N. Price, S. R. Signorini, E. Stancioff, C. Stymiest, R. A. Wahle, J. D. Waller, N. D. Rebuck, Z. A. Wang, T. L. Capson, J. R. Morrison, S. R. Cooley, S. C. Doney, Ocean and coastal acidification off New England and Nova Scotia, Oceanography (2015).

[14] R. B. Wallace, H. Baumann, J. S. Grear, R. C. Aller, C. J. Gobler, Coastal ocean acidification: the other eutrophication problem, Estuarine, Coastal and Shelf Science 148 (2014) 1–13.

[15] J. A. Ekstrom, L. Suatoni, S. R. Cooley, L. H. Pendleton, G. G. Waldbusser, J. E. Cinner, J. Ritter, C. Langdon, R. van Hooidonk, D. Gledhill, K. Wellman, M. W. Beck, L. M. Brander, D. Rittschof, C. Doherty, P. E. T. Edwards, R. Portela, Vulnerability and adaptation of us shellfisheries to ocean acidification, Nature Climate Change 5 (2015) 207–214.

[16] K. D. Friedland, R. E. Morse, J. P. Manning, D. C. Melrose, T. Miles, A. G. Goode, D. C. Brady, J. T. Kohut, E. N. Powell, Trends and change points in surface and bottom thermal environments of the US Northeast

Continental Shelf ecosystem, Fisheries Oceanography 29 (2020) 396–414.

[17] V. Klemas, X.-H. Yan, Subsurface and deeper ocean remote sensing from satellites: an overview and new results, Progress in Oceanography 122 (2014) 1–9.

[18] B. Li, K. R. Tanaka, Y. Chen, D. C. Brady, A. C. Thomas, Assessing the quality of bottom water temperatures from the finite-volume community ocean model (FVCOM) in the Northwest Atlantic Shelf region, Journal of Marine Systems 173 (2017) 21–30.

[19] P. F. Lermusiaux, P. Malanotte-Rizzoli, D. Stammer, J. Cummings, A. M. Moore, Progress and prospects of US data assimilation in ocean research, Oceanography issue volume (2006).

[20] T. P. Sapsis, Output-weighted optimal sampling for Bayesian regression and rare event statistics using few samples, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 476 (2020) 20190834.

[21] X. Yang, D. Venturi, C. Chen, C. Chryssostomidis, G. E. Karniadakis, EOF-based constrained sensor placement and field reconstruction from noisy ocean measurements: application to Nantucket Sound, Journal of Geophysical Research: Oceans 115 (2010).

[22] C. Chen, H. Liu, R. C. Beardsley, An unstructured grid, finite-volume, three-dimensional, primitive equations ocean model: application to coastal ocean and estuaries, Journal of Atmospheric and Oceanic Technology 20 (2003) 159 – 186.

[23] C. Chen, R. Beardsley, G. Cowles, An unstructured-grid finite-volume coastal ocean model (fvcom) system, Oceanography 19 (2006) 78–89.

[24] G. L. Mellor, S. M. Häkkinen, T. Ezer, R. C. Patchen, A Generalization of a Sigma Coordinate Ocean Model and an Intercomparison of Model Vertical Grids, Springer Berlin Heidelberg, Berlin, Heidelberg, 2002, pp. 55–72.

[25] Y. Chao, Z. Li, J. D. Farrara, P. Hung, Blending sea surface temperatures from multiple satellites and in situ observations for coastal oceans, Journal of Atmospheric and Oceanic Technology 26 (2009) 1415 – 1426.

[26] H. Babaee, C. Bastidas, M. Defilippo, C. Chryssostomidis, G. Karniadakis, A multi-fidelity framework and uncertainty quantification for sea surface temperature in the Massachusetts and Cape Cod Bays, Earth and Space Science 7 (2020).

[27] E. N. Lorenz, Empirical Orthogonal Functions and Statistical Weather Prediction, Scientific report, Massachusetts Institute of Technology, Department of Meteorology, 1956.

[28] A. Hannachi, I. T. Jolliffe, D. B. Stephenson, Empirical orthogonal functions and related techniques in atmospheric science: a review, International Journal of Climatology 27 (2007) 1119–1152.

[29] R. Everson, P. Cornillon, L. Sirovich, A. Webber, Empirical eigenfunction analysis of sea surface temperatures in the Western North Atlantic, AIP Conference Proceedings 375 (1996) 563–590.

[30] L. M. Berliner, C. K. Wikle, N. Cressie, Long-lead prediction of pacific SSTs via Bayesian dynamic modeling, Journal of Climate 13 (2000) 3953 – 3968.

[31] U. Ganzedo, A. Alvera-Azcárate, G. Esnaola, A. Ezcurra, J. Sáenz, Reconstruction of sea surface temperature by means of DINEOF: a case study during the fishing season in the Bay of Biscay, International Journal of Remote Sensing 32 (2011) 933–950.

[32] T. M. Smith, R. W. Reynolds, R. E. Livezey, D. C. Stokes, Reconstruction of historical sea surface temperatures using empirical orthogonal functions, Journal of Climate 9 (1996) 1403 – 1420.

[33] R. Everson, L. Sirovich, Karhunen–Loève procedure for gappy data, J. Opt. Soc. Am. A 12 (1995) 1657–1664.

[34] M. M. Ali, D. Swain, R. A. Weller, Estimation of ocean subsurface thermal structure from surface parameters: A neural network approach, Geophysical Research Letters 31 (2004).

[35] Q. Miao, B. Pan, H. Wang, K. Hsu, S. Sorooshian, Improving monsoon precipitation prediction using combined convolutional and long short term memory neural network, Water 11 (2019).

[36] R. Maulik, R. Egele, B. Lusch, P. Balaprakash, Recurrent neural network architecture search for geophysical emulation, in: SC20: International Conference for High Performance Computing, Networking, Storage and Analysis, 2020, pp. 1–14.

[37] X. Meng, G. E. Karniadakis, A composite neural network that learns from multi-fidelity data: application to function approximation and inverse PDE problems, Journal of Computational Physics 401 (2020) 109020.

[38] M. Raissi, P. Perdikaris, G. E. Karniadakis, Machine learning of linear differential equations using Gaussian processes, Journal of Computational Physics 348 (2017) 683–693.

[39] W. Yang, Q. Yao, K. Ye, C.-Z. Xu, Empirical mode decomposition and temporal convolutional networks for remaining useful life estimation, International Journal of Parallel Programming 48 (2020) 61–79.

[40] Z. Y. Wan, B. Dodov, C. Lessig, H. Dijkstra, T. P. Sapsis, A data-driven framework for the stochastic reconstruction of small-scale features with application to climate data sets, Journal of Computational Physics 442 (2021) 110484.

[41] S. Bai, J. Z. Kolter, V. Koltun, An empirical evaluation of generic convolutional and recurrent networks for sequence modeling, 2018.

[42] E. Aksan, O. Hilliges, STCN: Stochastic temporal convolutional networks, in: International Conference on Learning Representations, 2019.

[43] P. Lara-Benítez, M. Carranza-García, J. M. Luna-Romera, J. C. Riquelme, Temporal convolutional networks applied to energy-related time series forecasting, Applied Sciences 10 (2020).

[44] J. Yan, L. Mu, L. Wang, R. Ranjan, A. Y. Zomaya, Temporal convolutional networks for the advance prediction of ENSO, Scientific Reports (2020).

[45] J. Baño Medina, R. Manzanas, J. M. Gutiérrez, Configuration and intercomparison of deep learning neural models for statistical downscaling, Geoscientific Model Development 13 (2020) 2109–2124.

[46] Z. Gan, C. Li, J. Zhou, G. Tang, Temporal convolutional networks interval prediction model for wind speed forecasting, Electric Power Systems Research 191 (2021) 106865.

[47] H. M. Benway, L. Lorenzoni, A. E. White, B. Fiedler, N. M. Levine, D. P. Nicholson, M. D. DeGrandpre, H. M. Sosik, M. J. Church, T. D. O'Brien, M. Leinen, R. A. Weller, D. M. Karl, S. A. Henson, R. M. Letelier, Ocean time series observations of changing marine ecosystems: An era of integration, synthesis, and societal applications, Frontiers in Marine Science 6 (2019).

[48] T. M. Chin, J. Vazquez-Cuervo, E. M. Armstrong, A multi-scale high-resolution analysis of global sea surface temperature, Remote Sensing of Environment 200 (2017) 154–169.

[49] Y. Zhu, E. Kang, Y. Bo, J. Zhang, Y. Wang, Q. Tang, Hierarchical Bayesian model based on robust fixed rank filter for fusing MODIS SST and AMSR-E SST, Photogrammetric Engineering Remote Sensing 85 (2019) 119–131.

[50] Y. Zhu, Y. Bo, J. Zhang, Y. Wang, Fusion of multisensor SSTs based on the spatiotemporal hierarchical Bayesian model, Journal of Atmospheric and Oceanic Technology 35 (2018) 91 – 109.

[51] C. Donlon, I. Robinson, K. S. Casey, J. Vazquez-Cuervo, E. Armstrong, O. Arino, C. Gentemann, D. May, P. LeBorgne, J. Piollé, I. Barton, H. Beggs, D. J. S. Poulter, C. J. Merchant, A. Bingham, S. Heinz, A. Harris, G. Wick, B. Emery, P. Minnett, R. Evans, D. Llewellyn-Jones, C. Mutlow, R. W. Reynolds, H. Kawamura, N. Rayner, The global ocean data assimilation experiment high-resolution sea surface temperature pilot project, Bulletin of the American Meteorological Society 88 (2007) 1197 – 1214.

[52] A. Li, Y. Bo, Y. Zhu, P. Guo, J. Bi, Y. He, Blending multi-resolution satellite sea surface temperature (SST) products using Bayesian maximum entropy method, Remote Sensing of Environment 135 (2013) 52–63.

[53] R. W. Reynolds, A real-time global sea surface temperature analysis, Journal of Climate 1 (1988) 75 – 87.

[54] R. W. Reynolds, T. M. Smith, Improved global sea surface temperature analyses using optimum interpolation, Journal of Climate 7 (1994) 929 – 948.

[55] Y. Zhu, E. L. Kang, Y. Bo, Q. Tang, J. Cheng, Y. He, A robust fixed rank kriging method for improving the spatial completeness and accuracy of satellite SST products, IEEE Transactions on Geoscience and Remote Sensing 53 (2015) 5021–5035.

[56] M. Raissi, H. Babaee, G. E. Karniadakis, Parametric Gaussian process regression for big data, Computational Mechanics 64 (2019).

[57] C. E. Rasmussen, C. K. I. Williams, Gaussian Processes for Machine Learning, Adaptive Computation and Machine Learning, MIT Press, Cambridge, MA, USA, 2006.

[58] P. Perdikaris, D. Venturi, J. O. Royset, G. E. Karniadakis, Multi-fidelity modelling via recursive co-kriging and Gaussian-Markov random fields, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 471 (2015) 20150018.

[59] A. Paleyes, M. Pullin, M. Mahsereci, N. Lawrence, J. González, Emulation of physical processes with Emukit, in: Second Workshop on Machine Learning and the Physical Sciences, NeurIPS, 2019.

[60] Y. Yang, A. Blanchard, T. Sapsis, P. Perdikaris, Output-weighted sampling for multi-armed bandits with extreme payoffs, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 478 (2022) 20210781.