



Machine learning framework for the real-time reconstruction of regional 4D ocean temperature fields from historical reanalysis data and real-time satellite and buoy surface measurements

Bianca Champenois^{*}, Themistoklis Sapsis

Massachusetts Institute of Technology Department of Mechanical Engineering, 77 Massachusetts Avenue, Cambridge, 02139, MA, USA

ARTICLE INFO

Communicated by D. Faranda

Keywords:

Ocean temperature
Temporal convolutional network
Uncertainty quantification

ABSTRACT

The ability to model 4D properties of near-coast ocean regions in real time is valuable for both general environmental monitoring and for specialized industry applications. We present a computationally parsimonious framework to build improved spatiotemporal models that combine dynamics inferred from historical high-fidelity model outputs and real-time sensor measurements. We are motivated by a temperature data set of the Massachusetts Bay from sensors that can only obtain measurements at the ocean's surface. We first apply standard principal component analysis (PCA) to a historical reanalysis data set of the time-evolving 3D (x, y, z) temperature field. Next, we train a temporal convolutional neural network (TCN) to predict the time coefficients of the vertical PCA modes, and their variance, as a function of surface temperature. Simultaneously, we estimate the time-evolving 2D (x, y) surface temperature field from the satellite and buoy measurements with multi-fidelity Gaussian process regression (GPR). Finally, the surface temperature is used as input to the neural network to probabilistically predict the PCA coefficients and reconstruct the full 4D 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 is able to produce more accurate models in less time, and the uncertainty quantification can be leveraged for future decision-making.

1. Introduction

Access to fast and inexpensive regional near-coast ocean models is crucial: fisheries need these models to make decisions about aquaculture, engineers to design near-coast systems, policy makers to monitor changes in the ocean, and so on. Creating models from the high-dimensional nonlinear equations that describe the physics of the ocean requires complex and computationally expensive numerical solvers. Even with advanced numerical methods such as finite volumes or finite differences (first done for a full coastal region in [1]), uncertainty in the boundary, initial, and excitation conditions and intrinsic instabilities in the system can produce errors [2]. High-resolution, high-accuracy physics-based numerical simulations require vast computational resources, making forecasting and nowcasting, or even rapid hindcasting, out of reach. Alternatively, stakeholders can turn to real-world information on quantities of interest from physical sensors (e.g. buoys, drifters, conductivity–temperature–depth (CTD) instruments, and satellites). However, sensors only provide information about a system locally in space, with significant gaps, or with a high degree of sparsity, and the purchase, operation, and maintenance of sensors is financially

expensive. In a hybrid approach called data assimilation, measurements from sensors can be used to complement or improve numerical models, but such hybrid methods still suffer from insufficient instruments and costly computational needs. Overall, numerical methods, sensor measurements, and data assimilation form the actively-researched fields of ocean modeling and ocean monitoring.

One specific research problem of interest within ocean modeling involves the development of methods to predict subsurface ocean temperature from surface measurements [3]. Many methods have been developed to address this problem: both Carnes et al. and Nardelli et al. used empirical orthogonal functions from climatology data sets to infer subsurface temperature from satellite data [4,5]; Guinehut et al. (2004), Guinehut et al. (2012), and Wang et al. (2012) all led efforts to combine in-situ Argo measurements with satellite data [6–8]; Wang et al. (2013) integrated satellite data into a quasigeostrophic ocean model [9]; Li et al. assessed the accuracy of using a finite volume model to predict bottom water temperatures [10]; more broadly, Klemas and Yan discussed challenges related to incorporating surface sensor measurements into ocean models [11]. Significant research has

^{*} Corresponding author.

E-mail address: bchamp@mit.edu (B. Champenois).

been undertaken to predict sought-after subsurface properties. More recently, advancements in machine learning have allowed for new inference methods: researchers have explored using different neural network architectures to map surface measurements to historical vertical profiles from in-situ observations (ie. Argo drifters or CTDs) [12–14]. More generally, some of the aforementioned methods were applied to other ocean properties [15–18]. The motivation behind this open research question is that surface measurements are cheaper and more readily available than subsurface measurements: buoys and drifters are less expensive to operate at or near the surface of the ocean, and satellites can only collect sea surface temperature (SST) but cover large domains. However, even surface sensors present challenges: buoys and drifters are scarce, and satellites are unable to make measurements in the presence of cloud coverage. Temperature profiles from instruments such as the Argo drifters are plentiful and publicly available but do not exist for near-coast regions.

In this paper, we propose a new method that is able to quickly and parsimoniously predict regional temperature in 4D (x, y, z, t) with uncertainty estimates at a high resolution and in real time from nothing but surface measurements. The goal of this work is different from traditional and direct, comprehensive ocean modeling efforts. Our aim is to make use of existing reanalysis data obtained from previously-simulated physics-based and observation-driven ocean models to more rapidly and inexpensively make predictions at recent or current time steps. We take advantage of existing historical reanalysis data from the specific region of interest to characterize the vertical structure of the ocean temperature field in that region. Subsequently, we use machine learning methods to reconstruct and hindcast the full 4D temperature field and its uncertainties from real-time surface temperature sensor measurements. We intentionally choose to use probabilistic machine learning methods so that we can produce estimates for the uncertainty associated with both the system and the model. We consider temperature data because they are widely available, but the techniques discussed can be applied to other quantities of interest such as salinity, total alkalinity (TA), dissolved inorganic carbon (DIC), aragonite, and pH.

To demonstrate the effectiveness of our proposed method, we model temperature in the Massachusetts and Cape Cod Bays. These bays correspond to a region with great biodiversity (fish, shellfish, whales, etc.) and significant fishing and tourism industries. The ability to predict temperature in this type of coastal area is helpful in assessing general ocean health, tracking ecosystem functioning, and managing fisheries [19–21]. The final product is integrated into the online SEA-GLASS visualization tool, a web application for ocean data that is developed by MIT Sea Grant and is accessible to local stakeholders (e.g. fisheries, NGOs, students, etc.). The visualization tool receives surface measurements in real time from both internal databases and external APIs, which is why it is important that the model be able to quickly and seamlessly manage new unprocessed data sets as they become available. The model is also useful to make decisions about where and how to sample future data [22,23] and to evaluate the quality of new sensors. Sections 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 presents the real-world results of the model.

2. Reanalysis data

Our starting point is a reanalysis data set consisting of a time-evolving 3D temperature field of the Northeast Coastal Ocean from the Finite Volume Community Ocean Model (FVCOM) by Chen et al. [24, 25]. FVCOM uses a fractional step method to solve the spatially and temporally evolving fields for velocity, density, temperature, and salinity, among other variables over an unstructured grid with a horizontal resolution of approximately 0.1–25 km over 45 sigma levels (higher resolution closest to the coasts). 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 [26]. 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 sea surface temperature 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 [10]. 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 Fig. 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 and in-situ stations. Satellites measure sea surface temperature by measuring the amplitude of the infrared and microwave wavelengths from the electromagnetic radiation emitted by the ocean's surface. Different satellites operate at varying resolutions and levels of accuracy [27], but the main challenge associated with using satellite data is that there can be gaps due to cloud coverage. There exist many SST data products: e.g. Optimal Interpolation SST, Hadley Centre Global Sea Ice and SST, Climate Change Initiative SST. Each of these has a different spatial and temporal resolution, some of which are not sufficient for the region of interest in this study. Furthermore, for decision-making purposes, we are interested in using a SST product that includes uncertainty estimates. For the region of interest, we have access to unprocessed daily satellite imagery from the MODerate-resolution Imaging Spectroradiometer (MODIS) Terra. In Fig. 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) (Fig. 2), but they are only collected on a monthly basis, and there are only 14 stations with consistent data. 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. We only use surface measurements to train the model, but we use MWRA measurements at a variety of depths to assess the quality of our model during the evaluation stage.

4. Framework description

The framework is organized into multiple steps as outlined in Fig. 3. Steps 1 and 2 are independent while Step 3 relies on the completion of Steps 1 and 2. Furthermore, Step 1 only needs to be completed once while Steps 2 and 3 need to be repeated each time new measurements become available. 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 obtain from principal component analysis (PCA). Next, in Step 2, real-time information from satellites and in-situ buoys is pre-processed and interpolated to estimate the full ocean surface temperature field. We choose to interpolate with multi-fidelity Gaussian process regression (GPR) as described in Babaee et al. [28], but other SST products (introduced in Section 3) may be used. 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

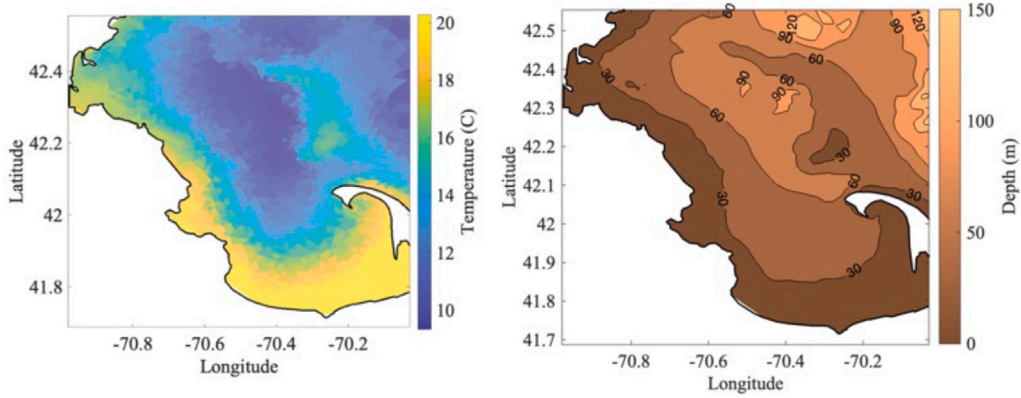


Fig. 1. Reanalysis Data of the Northeast Coastal Ocean from FVCOM. In the left panel, the high resolution of the FVCOM temperature field is demonstrated for September 13th, 2012 at sigma level -0.5 . The right panel shows the contour map of the bathymetry of the region of interest.

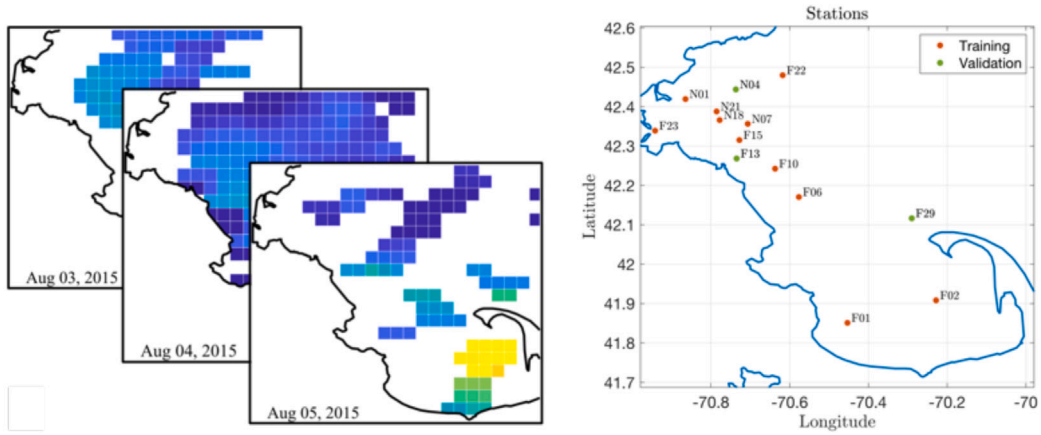


Fig. 2. Sensor Data. The low fidelity data (satellite (left)) are only available on days with low cloud coverage. The high fidelity data (buoys (right)) are local in space and sparse.

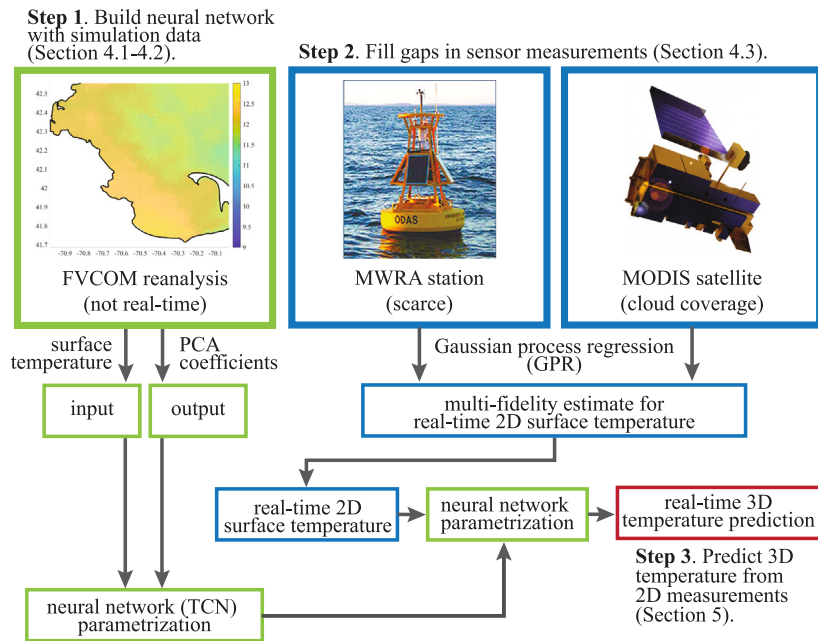


Fig. 3. Framework. Flow chart describing the developed framework for real-time estimation of the 4D ocean temperature field. Reanalysis data are used to estimate a reduced-order model. Ocean surface information, obtained from satellite and buoy measurements, are used as input.

the reduced-order model to obtain a real-time estimate for the time-evolving 3D temperature field and its uncertainty. The framework can be modified or rearranged based on the type and location of available data.

4.1. Temperature field order-reduction using vertical PCA

We first apply standard principal component analysis (PCA) to the reanalysis data set to reduce the dimensionality while retaining patterns and information. PCA, also known as empirical orthogonal functions (EOF), proper orthogonal decomposition (POD), or Karhunen–L ev e decomposition, among other names, has long been used in many fields. In the context of fluid mechanics, weather prediction [29,30], 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 [31–34]. In some cases, the basis is used to fill gaps in the data [35]. 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.

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

Using this data matrix, we evaluate the eigenvectors.

$$\mathbf{T}_i \mathbf{T}_i^T \boldsymbol{\phi}_{ij} = \lambda \boldsymbol{\phi}_{ij}, j = 1, \dots, n \quad (2)$$

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

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

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 (Fig. 4). 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 difference between the surface temperature and the temperature at depth while the second mode corresponds to the extent of the mixing layer (Fig. 4). 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 Fig. 5 suggest that there is good agreement between the original reanalysis and the reduced-order PCA projection. For time steps during which there is no full 3D information, a functional relationship between surface information and PCA 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

In the most important step of our framework (Step 1), we use machine learning to map surface temperature to subsurface temperature. Recent developments in machine learning have increased the popularity of using neural networks to model such geophysical processes. As discussed in the Introduction (Section 1), different neural networks (perceptrons [12,14], LSTM [13]) have been used to map surface observations to vertical temperature profiles. More generally, neural networks that are specifically designed for time series analysis are increasingly being used to predict the time-varying PCA coefficients obtained from geophysical data sets [36,37]. Many of these papers focus on large domains of the open ocean or on geophysical applications that are different from 4D ocean temperature. Here, we choose to focus on a near-coast regional ocean model, and we use high-resolution reanalysis data for training. We choose machine learning models with probabilistic modeling capabilities for uncertainty quantification and neural network architectures that are well-suited for time series analysis.

For each horizontal location i , (x_i, y_i) , we train a neural network to learn a functional relationship between surface temperature (for which sensor measurements are more readily available) and the subsurface vertical structure of ocean temperature (which we wish to predict at new time steps in the form of PCA time coefficients $q_{ij}(t)$ and mean temperature $\bar{\mathbf{T}}_i(t)$ obtained in the previous section). This step of the framework takes the most time but only needs to be performed once. 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. More specifically, 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 [38]. TCNs have also been shown to outperform recurrent neural networks for sequence modeling [39,40].

We adapt the Stochastic Machine Learning (SMaL) code from Wan et al. and retain the same residual block architecture (Fig. 6) [38]. The data are standardized before training for improved results. The 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. 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 contribute 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 computational costs and improves 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 corresponds to the number of nonlinear activations. 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 $J(\boldsymbol{\theta})$ that quantifies the error between the true data and the model predictions.

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

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

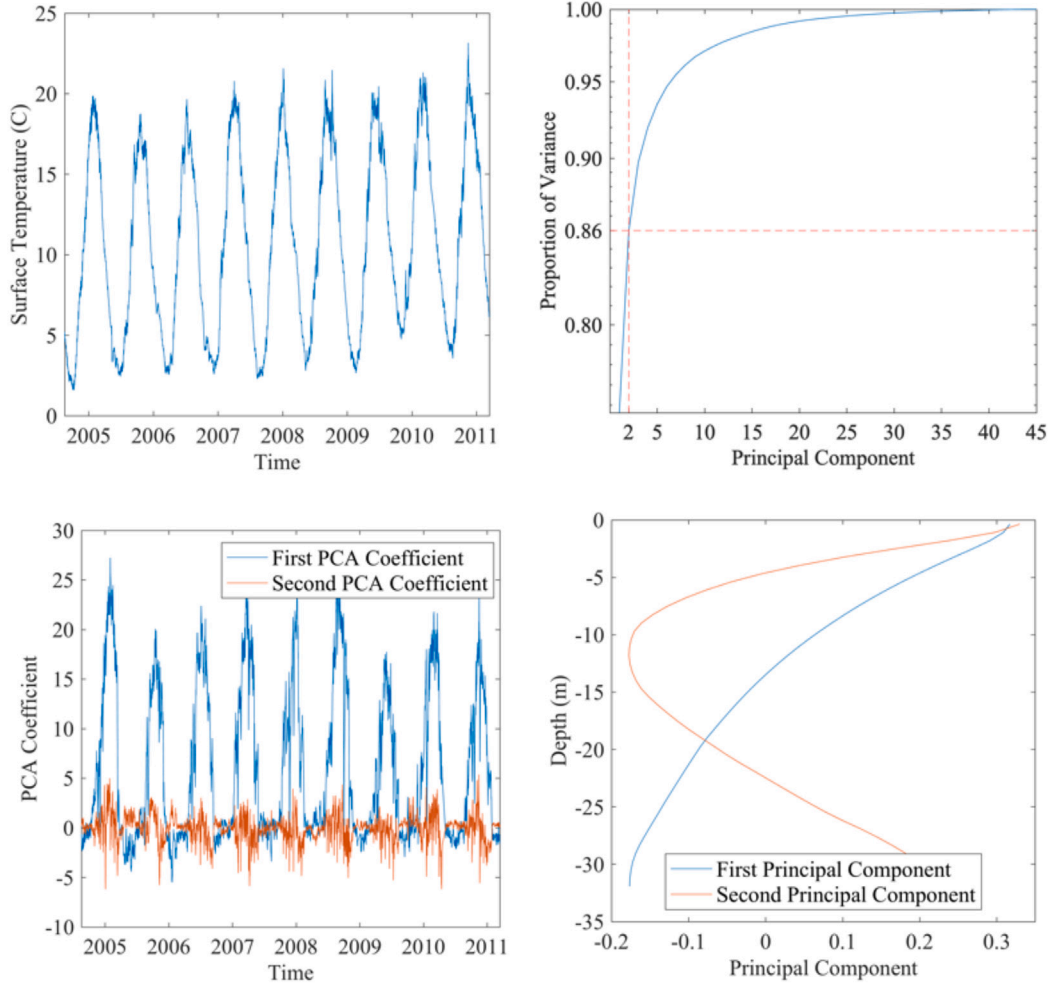


Fig. 4. Vertical Order Reduction at (42.41N, 70.86 W). Time series of surface temperature (top left), energy distribution of the vertical modes (top right), time series of PCA coefficients obtained by projection of the reanalysis temperature field (bottom left), and first two vertical modes (bottom right) at one (x_i, y_i) location.

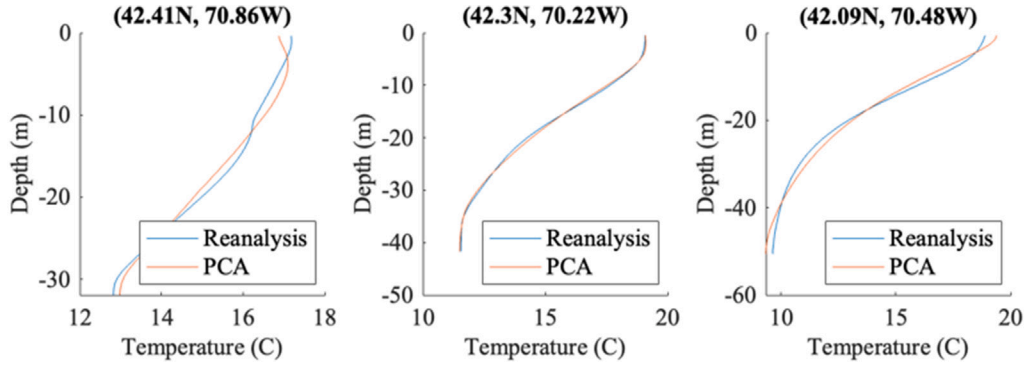


Fig. 5. PCA Vertical Profiles. The vertical profiles of the reanalysis data and the PCA projection are shown for September 13th, 2012 at three different (x_i, y_i) locations. Corresponding locations are shown on the map in Fig. 7.

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.

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

To predict the standard deviation of the PCA coefficients, we minimize the mean negative anomaly correlation coefficient (MNACC) [38]. It is a correlation-based loss function, so it does not scale with magnitude,

therefore more effectively penalizing anomalies.

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

$$z = y - y^{ref} \quad (7)$$

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.

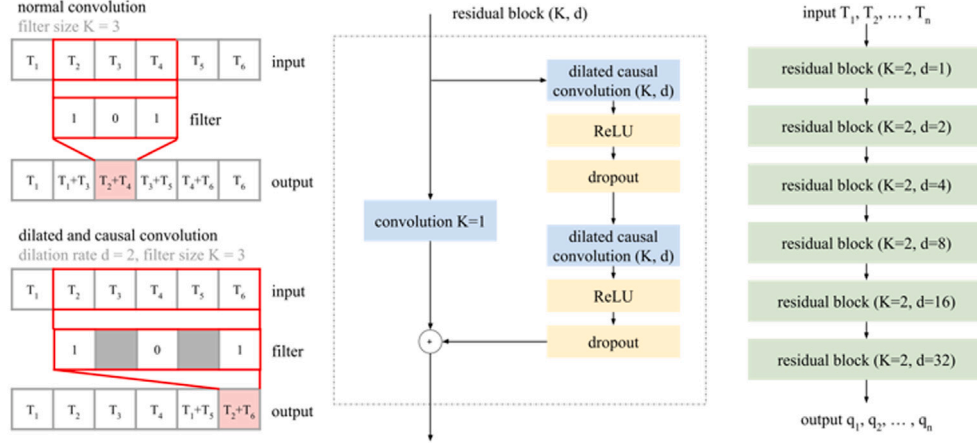


Fig. 6. Architecture of TCN. The TCN is built with residual blocks that consist of a sequence of two convolutional layers with ReLU activation and a dropout. The dilation factor of each residual block is doubled at each depth.

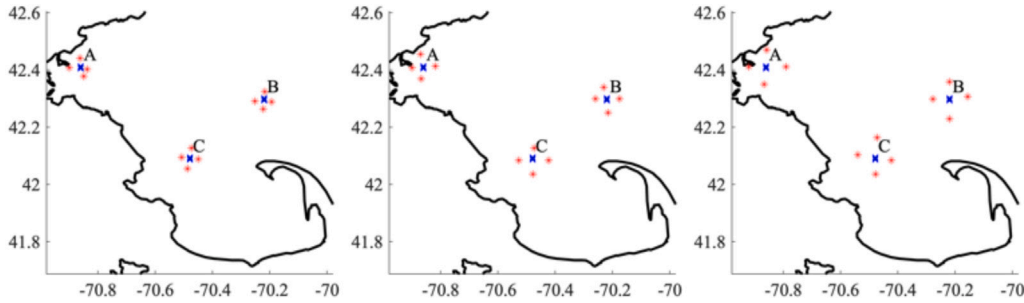


Fig. 7. Input Points. The input of the neural network consists of the surface temperature at four nearby points in addition to the surface temperature at the corresponding point of interest. Different radii, shown here, are tested through numerical experiments.

4.2.2. 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. 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. To more comprehensively include all dynamic interactions, we produce a non-local parametrization by including neighboring points in the input of the neural network. The FVCOM model uses an unstructured grid, so we select the neighboring points by choosing a radius Δ and computing the points $(x + \Delta, y)$, $(x - \Delta, y)$, $(x, y + \Delta)$, and $(x, y - \Delta)$. Then we find the points in the data set that are closest to the points computed in the previous step and assign those to be the neighbors of the point of interest. For points that are near the boundary, we use the same algorithm with the knowledge that the selected neighbors will also be along the boundary. While this is not the ideal solution, it is acceptable because the neural network will learn a new parametrization given the available neighboring points. To find the number and location of input points that are best suited for generalizability, we perform numerical experiments by minimizing the mean squared error using grid search. 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 Fig. 7, 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. These points all correspond to (x_i, y_i) pairs in the reanalysis data set.

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_2 , and mean temperature, \bar{T} ,

$$\mathbf{x} = [T_S(t_0) \quad T_S(t_1) \quad T_S(t_2) \quad \dots \quad T_S(t_n)] \quad (8)$$

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

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

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

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

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

Table 1
Evaluation of neural network model on reanalysis data set.

y (target)	\hat{y} (output)	MAE (°C)			RMSE (°C)		
		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

(Tables B.2, B.3). 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 [41].

4.2.4. Results of the neural network training

By using additional nearby points ((x_i, y_i) pairs in the reanalysis data set) 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 (number and location of inputs, memory, neural network depth), 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) (Fig. 8). The predicted time series for a representative horizontal location, as well as the predicted standard deviation, are shown in Fig. 8. Fig. 8, confirms that the first PCA coefficient (8a) captures the warming of the surface while the second PCA coefficient (8b) captures smaller scale effects which are most likely due to the depth of the mixing layer. Both PCA coefficients go to zero during the winter, and the overall uncertainty is also lower during the winter.

These raw outputs are combined with the PCA modes to reconstruct the full 4D temperature field. Table 1 lists the error associated with the neural network predictions relative to both the original reanalysis data and the PCA reconstruction.

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. Figs. 9, 10, and 11 compare the reconstruction to the original reanalysis data set for a time step in the test set (not seen during training). Fig. 10 shows the horizontal reconstruction at four depths and confirms that the neural network is able to learn and replicate the physical structures. Fig. 11 shows the vertical reconstruction at a fixed longitude to explain which depths are more difficult to model with PCA and a neural network. Figs. 9 and 11 show that the agreement between the original reanalysis data and the TCN prediction is worse at the surface, but the associated uncertainty is also higher. Table 1 also reveals that the test error is higher than the training and validation errors which is expected in many machine learning applications. In this problem especially, the testing data come from a future time period with more warming than any of the time periods seen in the training and validation sets.

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. While we did choose to dedicate some time to tune the hyperparameters of our model, we found that this neural network architecture is robust to variance in design choices (within a reasonable range). TCNs require less parameter tuning than standard computational fluid dynamics models and can be used by non-experts to achieve quick and high-resolution results.

4.3. Filling gaps in the surface sensor data

The next step in the framework is to estimate the full time-evolving 2D surface temperature field. Satellites measure SST, but they are frequently affected by cloud coverage. To overcome this problem, there exist methods that enhance measurements from satellites and blend

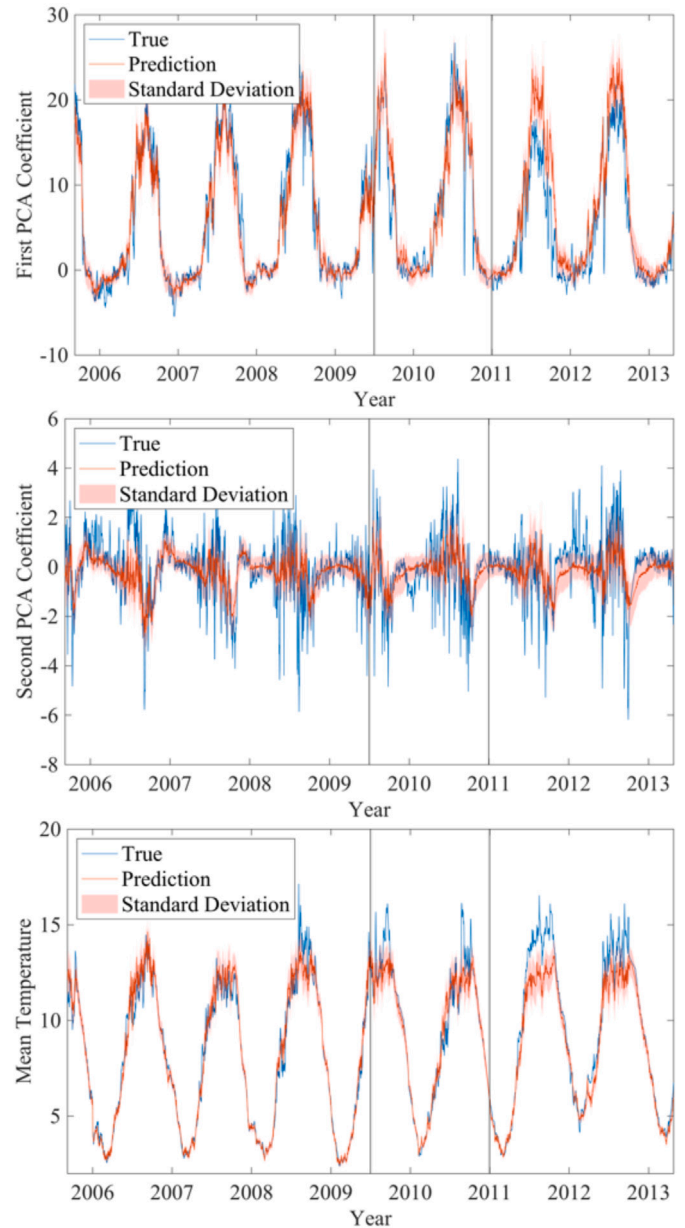


Fig. 8. TCN Predictions at (42.41N, 70.86 W). The first (top) and second (center) PCA coefficients and the mean temperature (bottom), as well as their standard deviations, are predicted for the reanalysis data. The black lines delineate the training, validation, and test sets, respectively.

data from multiple satellites [42,43]. In many projects, measurements from in-situ buoy are used to either validate or improve the accuracy of models [27,44–49]. Gaussian process regression (GPR), which is similar to optimal interpolation or objective analysis, is one approach that has been shown to produce quick, accurate, and useful results [28,50]. Babaee et al. use multi-fidelity GPR, a modified version of GPR, to combine measurements from satellites and buoys with the assumption that buoys have a higher fidelity than satellites [28]. We follow this

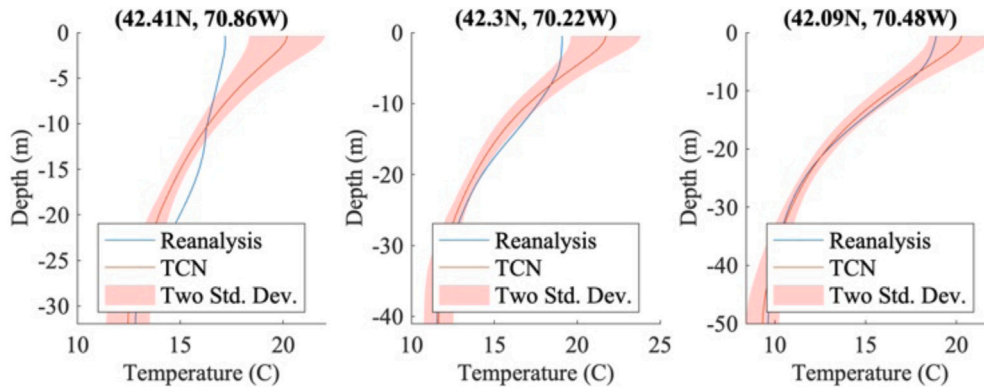


Fig. 9. TCN Vertical Profiles. The vertical profiles of the reanalysis data and the TCN reconstruction are plotted for September 13th, 2012 at three different (x_i, y_i) locations (shown in Fig. 7). The red shading corresponds to two standard deviations predicted by the TCN.

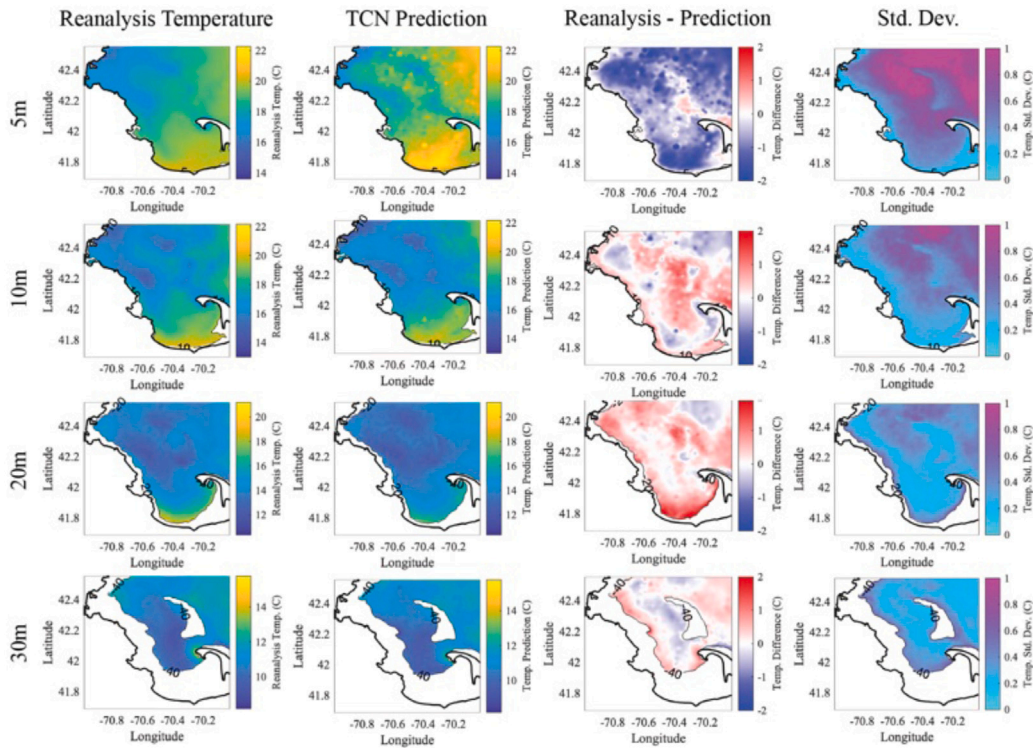


Fig. 10. Horizontal TCN Reconstruction. The reanalysis data (far left), the TCN reconstruction (center left), the difference between the two (center right), and the TCN standard deviation (far right) are shown for September 13th, 2012 (date in test set) at four depths (5 m (first row), 10 m (second row), 20 m (third row), 40 m (fourth row)). The neural network is able to replicate physically reasonable features.

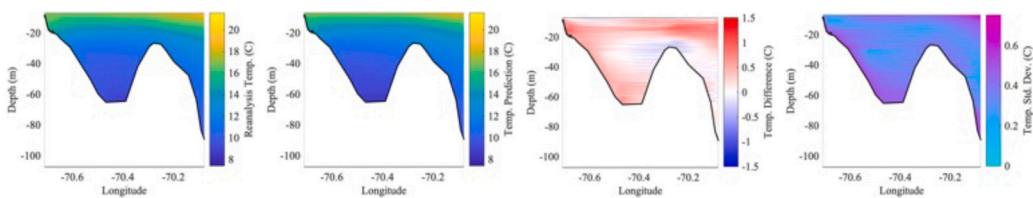


Fig. 11. Vertical TCN Reconstruction. The reanalysis data (far left), the TCN reconstruction (center left), the difference between the two (center right), and the TCN standard deviation (far right) are shown for September 13th, 2012 at a latitude of 42.2N.

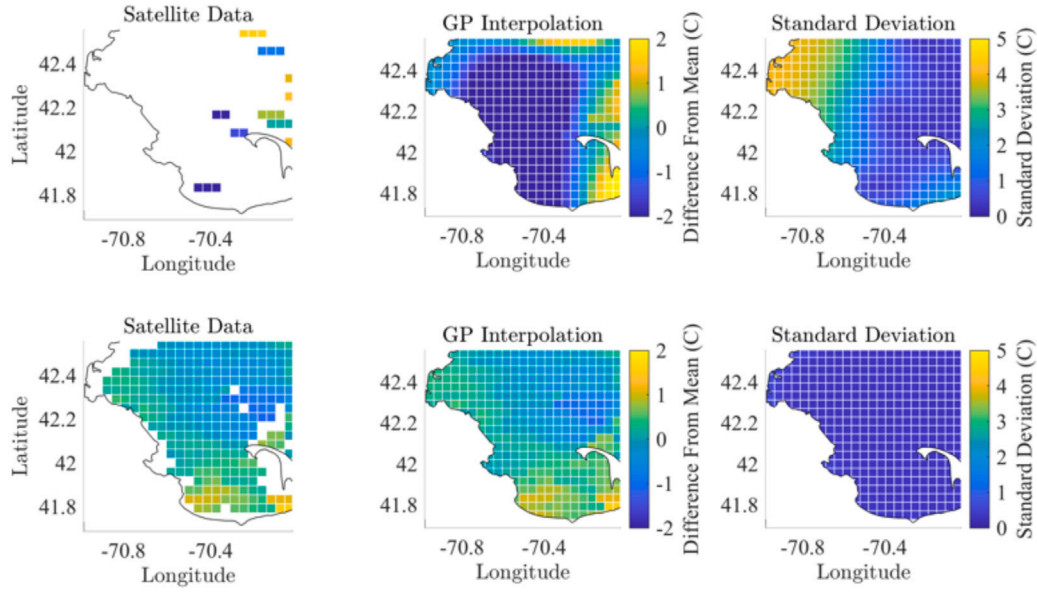


Fig. 12. Results of Extrapolation for Two Different Days. The available satellite and buoy data are extrapolated to obtain a surface temperature field over the full domain. The top row represents a day with high cloud coverage (March 8th 2016), and the bottom row represents a day with low cloud coverage (September 13th 2016).

implementation because we want to use real-time satellite data from a publicly available NASA API, and we want our model to be adaptable, even in the presence of cloud coverage. Details about the implementation of GPR and multi-fidelity GPR are outlined in Appendix A. One challenge of using GPR is that it requires matrix inversion, so it does not scale well with the size of the data set or the dimension of the input space. Because we are interested in predicting the temperature over the full spatial domain, we must reduce the size of the data set.

To address the challenge of scalability, we apply three modifications to the framework described in Babae et al. First, we set the prior of the mean function to be the daily spatial mean (Fig. A.15). Next, we keep the size of the data set small by building a new GPR model for each time step and setting the input data to be the available data on the day of interest, the day before, and the 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

$$\begin{bmatrix} x_i & y_i & t_{k-1} \\ x_i & y_i & t_k \\ x_i & y_i & t_{k+1} \end{bmatrix} = \begin{bmatrix} x_i & y_i & -1 \\ x_i & y_i & 0 \\ x_i & y_i & 1 \end{bmatrix} \quad (12)$$

where (x_i, y_i) are all of the available spatial points at each time step k . Finally, to prevent overfitting and to generalize the models, we manually set the hyperparameters to be the same for all days. For days with no available measurements, 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 roughly 25 kilometers, which is equivalent to six “gridpoints” or “pixels,” where one gridpoint is the spatial resolution. This choice assigns more weight to spatial points that are within 25 kilometers of the point of interest which corresponds to the mesoscales of the ocean [41]. For the time lengthscales, we choose a value of 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 the values found from minimizing the objective function over all submodels. For consistency, we set ρ and μ_d , the hyperparameters associated with the multi-fidelity method, to be the same as those from the optimized model in Babae et al. [28]. Because we use less data at each time step, our model is significantly faster at making predictions, robust to larger domains, and therefore more practical for real-time modeling. It is also different from the model in Babae et al. because it estimates the full surface temperature rather than just the temperature at select locations.

4.3.1. Results of surface temperature extrapolation

For points at which sensor data are available, we keep the original data, but for points at which there are no measurements, we use the method described in the previous section. The results of the extrapolation are shown in Fig. 12 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 more cloud coverage.

5. Results and evaluation of the full 4D temperature field

Finally, we use 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 of interest, 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 for simplicity, we choose to use the spatial resolution of the satellite data (4×4 km). We adopt the methodology described in 4.2.2 to select the neighboring points. The neural network predictions from the real-time sensor measurements are plotted in Fig. 13 for 2015 and 2016, and the standard deviation of the predictions is represented as red shading. Like in Fig. 8, the first PCA coefficient captures the general warming trend, the second PCA coefficient captures smaller scale effects, and the overall uncertainty is lower during the winter (Fig. 13).

The predicted PCA coefficients are then projected onto the deterministic PCA modes and summed with the predicted PCA mean to reconstruct the full 4D temperature and uncertainty fields. When estimating surface temperature, we left out measurements from three stations (N04, F13, F29 from Fig. 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.

The time series of the predicted temperature for stations N04, F13, and F29 are shown in Fig. 14 at multiple depths with associated uncertainty bounds. We compare the accuracy of our model to the climatological mean (the average daily mean calculated from the reanalysis data set over all 9 years), and we find that the model performs well for most points, but the predictions are worse for points

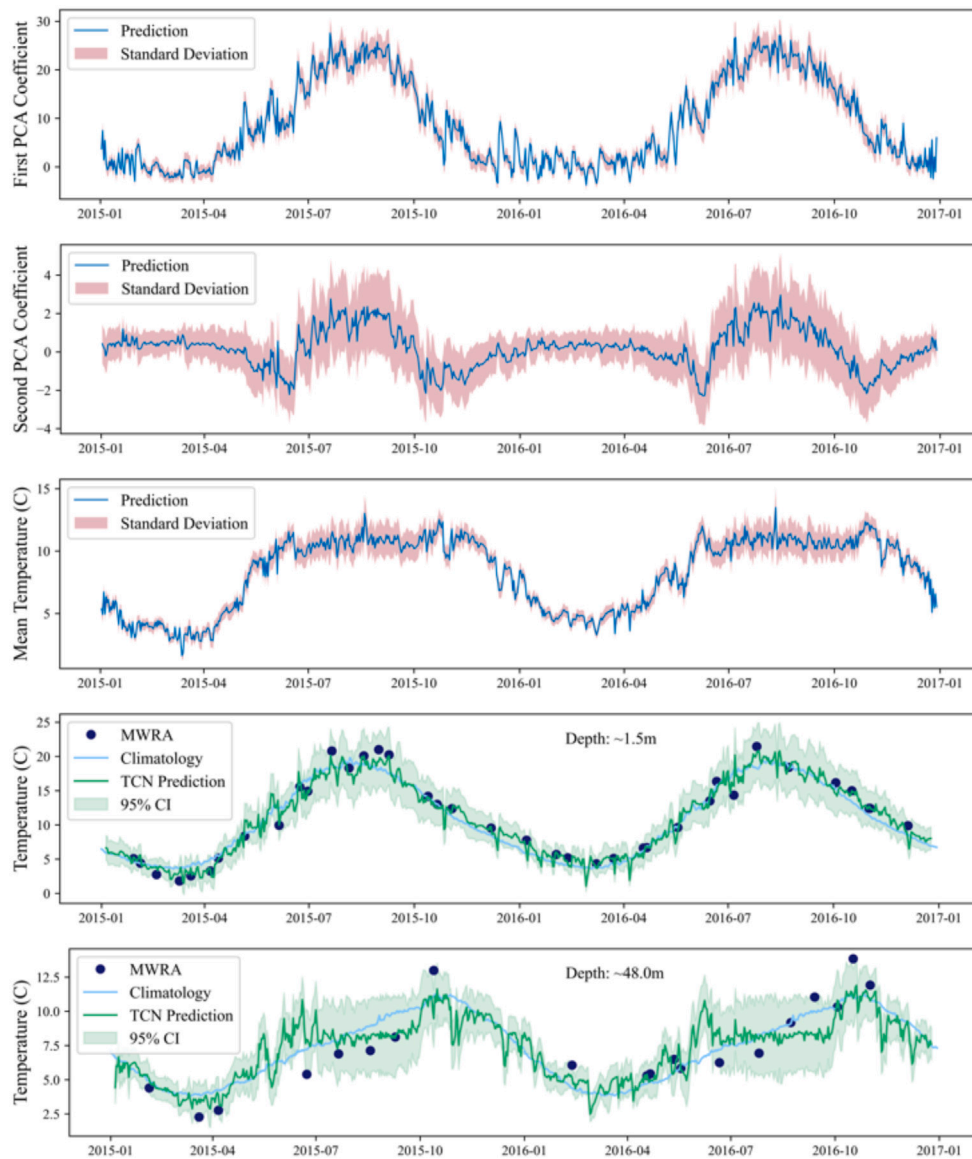


Fig. 13. TCN Predictions from Satellite Measurements at MWRA Station N04. In the first three panels, the first and second PCA coefficients, the mean temperature, as well as their standard deviations (red shading) are predicted for the available satellite surface temperature. These predictions are used to reconstruct the full 4D temperature field which is shown for shallow (~1.5 m) and deep (~48 m) water in the bottom two panels.

that are near estuaries or other unusual geographic or human features (e.g. station F13). Fig. 14 also includes the percentage of satellite coverage for each day (red dots), and we observe that the model performs best for days with the most amount of available satellite data (80%–100%). Specifically, 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 within two degrees of the truth. This root mean squared error is lower than the 2.28 °C found by Li et al. for FVCOM [10], but this comparison is incomplete given that the errors were calculated for different locations and time periods. We emphasize the favorable performance of the model with regards to bias elimination (Fig. 14). At the same time, it is important to note that the model has a more difficult time capturing the large temperature outliers (measurements that are significantly different from nearby-in-time measurements), but this observation is not a surprise as these differences may be caused by features that are not included in the current model (e.g. rivers, rainfall, and currents). The modeling of such extreme variations is a topic of high importance, and we leave it for future work. The accurate modeling of such features could be improved

not only by incorporating information about flow, rivers, and rain-fall, but also through targeted observations based on the uncertainty quantification skill of the presented approach.

6. Conclusions and future work

We introduced a fast and accurate framework, based on recently developed machine learning techniques and reanalysis data obtained from comprehensive ocean models, to reconstruct 4D 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. Overall, the developed model is important for monitoring general ocean health, and the techniques described can be used for other ocean properties.

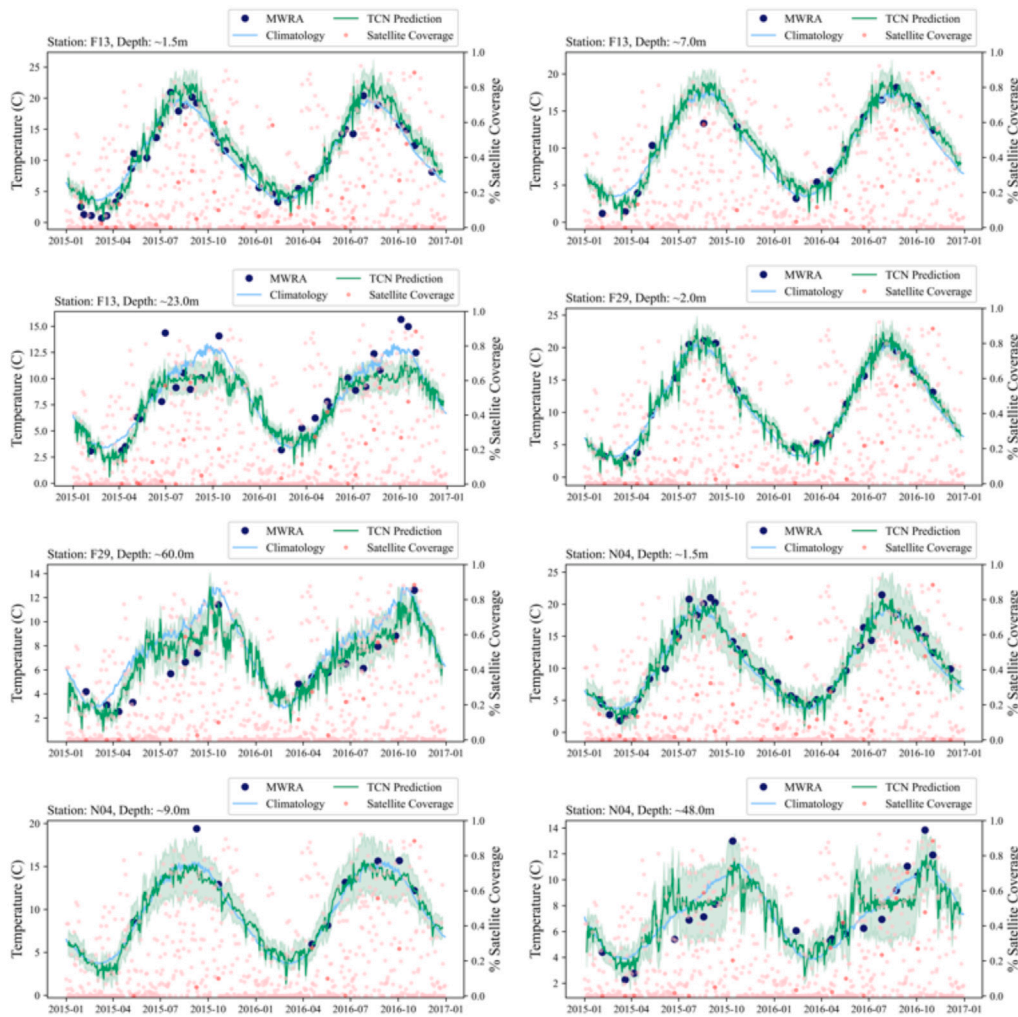


Fig. 14. Temperature Reconstruction Time Series. The temperature reconstruction is shown for the three validation stations (F13, F29, and N04) at varying depths over two years (2015–2016). The results (green line) are plotted along with the true MWRA buoy measurements (blue circles), the climatology (light blue line), and the satellite coverage (red dots).

CRedit authorship contribution statement

Bianca Champenois: Conceptualization, Methodology, Investigation, Software, Writing – original draft. **Themistoklis Sapsis:** Conceptualization, Methodology, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare no conflicts of interest.

Data availability

The data sources are listed at the end of the manuscript.

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 Babaei for stimulating discussions.

Appendix A. Gaussian process regression

Gaussian process regression (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 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.

A.1. Single fidelity

In ordinary GPR, the mean and variance are predicted using the kernel, K , which relates all of the available data points [51]. Specifically, the mean prediction is

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

and the variance is

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

For our application, the mean function $\mathbf{m}(X_*)$ is explicitly set to be the spatial mean (Fig. A.15) of the available satellite data.

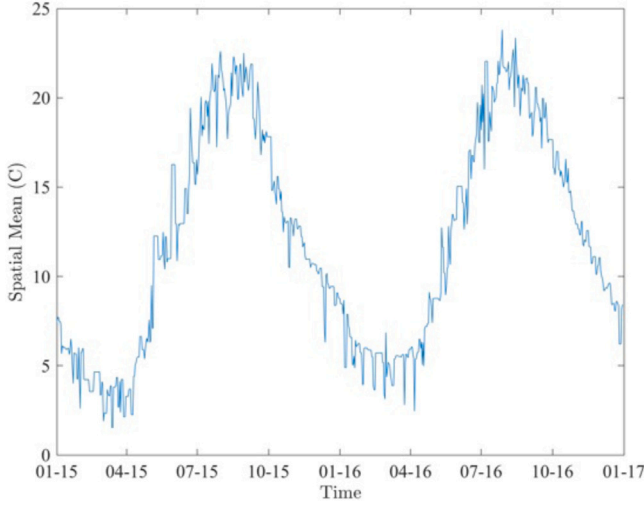


Fig. A.15. Spatial Mean of the Satellite Data. The mean over the whole spatial domain is plotted for each day.

A.2. Hyperparameter selection

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

$$\text{cov}(f(\mathbf{x}_p), f(\mathbf{x}_q)) = k(\mathbf{x}_p, \mathbf{x}_q) = \sigma_f^2 * \exp(-\frac{1}{2}(\mathbf{x}_q - \mathbf{x}_p)^T \boldsymbol{\theta}(\mathbf{x}_q - \mathbf{x}_p)) \quad (\text{A.3})$$

The signal variance σ_f and characteristic lengthscales $\boldsymbol{\theta}$ 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. The hyperparameters are typically found by optimizing the marginal log likelihood:

$$\log p(\mathbf{y}|X) = -\frac{1}{2}\mathbf{y}^T(K + \sigma_n^2 I)^{-1}\mathbf{y} - \frac{1}{2}\log |K + \sigma_n^2 I| - \frac{n}{2}\log 2\pi \quad (\text{A.4})$$

A.2.1. Multi-fidelity Gaussian process regression

GPR can be enhanced if higher-fidelity measurements become available. In the case of SST, buoy data have lower uncertainty than satellite data, and it can be incorporated through a recursive multi-fidelity Gaussian process regression scheme. 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 $\mathbf{x}_s, y_s, \bar{\mathbf{f}}_{*s}$ [52]. The prediction for the model with the lowest fidelity follows the Gaussian process regression steps from Eqs. (A.1) and (A.2)

$$\bar{\mathbf{f}}_1(\mathbf{x}_*) = K(X_*, X_1)[K(X_1, X_1) + \sigma_{n1}I]^{-1}\mathbf{y}_1, \quad (\text{A.5})$$

with covariance

$$\text{cov}(\bar{\mathbf{f}}_1) = K(X_*, X_*) - K(X_*, X_1)[K(X_1, X_1) + \sigma_{n1}I]^{-1}K(X_1, X_*). \quad (\text{A.6})$$

Each following model has the form

$$\bar{\mathbf{f}}_t(\mathbf{x}_*) = \rho_{t-1}\bar{\mathbf{f}}_{t-1} + \delta_t \quad t = 2, \dots, s \quad (\text{A.7})$$

In the case of satellite and buoy data, there are only two levels of fidelity, so the prediction for the highest level of fidelity, $s = 2$, can

Table B.2

Effect of memory length scale on first PCA coefficient.

Memory	Stride	Point A		
		MNACC	MSE	MAE
5 days	1	0.9082	0.3042	0.4301
10 days	1	0.9359	0.3201	0.4582
10 days	2	0.9204	0.3600	0.4728
14 days	2	0.9246	0.3623	0.4823
20 days	1	0.9563	0.2333	0.3796
30 days	3	0.9460	0.3134	0.4505
100 days	4	0.9558	0.2250	0.3624
Memory	Stride	Point B		
		MNACC	MSE	MAE
5 days	1	0.8975	0.3156	0.4393
10 days	1	0.9293	0.1809	0.3317
10 days	2	0.9061	0.2800	0.4161
14 days	2	0.9273	0.2166	0.3597
20 days	1	0.9471	0.1941	0.3538
30 days	3	0.9367	0.3094	0.4347
100 days	4	0.9501	0.1952	0.3538
Memory	Stride	Point C		
		MNACC	MSE	MAE
5 days	1	0.8848	0.4047	0.4926
10 days	1	0.9026	0.3235	0.4242
10 days	2	0.8950	0.3362	0.4513
14 days	2	0.9090	0.3684	0.4535
20 days	1	0.9421	0.1593	0.2893
30 days	3	0.9281	0.2360	0.3611
100 days	4	0.9448	0.1779	0.3124

be computed with the following equation

$$\bar{\mathbf{f}}_2(\mathbf{x}_*) = \rho\bar{\mathbf{f}}_1(\mathbf{x}_*) + \mu_d + K(X_*, X_2)[K(X_2, X_2) + \sigma_{n2}I]^{-1}(\mathbf{y} - \rho\bar{\mathbf{f}}_1(\mathbf{x}_2) - \mu_d). \quad (\text{A.8})$$

Its corresponding covariance is

$$\text{cov}(\bar{\mathbf{f}}_2) = \rho^2\text{cov}(\bar{\mathbf{f}}_1) + K(X_*, X_*) - K(X_*, X_2)[K(X_2, X_2) + \sigma_{n2}I]^{-1}K(X_2, X_*), \quad (\text{A.9})$$

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 [53] Python package, which builds on the GPy Python package [54], to build the multi-fidelity model.

Appendix B. Neural network hyperparameter tuning

To identify the optimal hyperparameters of the neural networks, a grid search is performed with the goal of minimizing different errors. The hyperparameters for which tuning was most important and most interesting were the ones associated with the memory of the neural network. Results from the numerical experiments are listed in Tables B.2 and B.3.

Appendix C. 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.

Table B.3
Effect of memory length scale on second PCA coefficient.

Memory	Stride	Point A		
		MNACC	MSE	MAE
5 days	1	0.4245	0.8236	0.6429
10 days	1	0.4595	0.8002	0.6196
10 days	2	0.1850	0.9724	0.6989
14 days	2	-0.0740	1.0125	0.7100
20 days	1	0.4566	0.7854	0.6441
30 days	3	0.3747	0.9471	0.6991
100 days	4	0.5764	0.6808	0.5928
Memory	Stride	Point B		
		MNACC	MSE	MAE
5 days	1	0.1812	0.7778	0.5944
10 days	1	0.0504	0.7956	0.5959
10 days	2	0.1769	0.7714	0.5810
14 days	2	0.1951	0.7725	0.5917
20 days	1	0.3595	0.7923	0.6144
30 days	3	0.2663	0.7673	0.5971
100 days	4	0.4509	0.6761	0.5656
Memory	Stride	Point C		
		MNACC	MSE	MAE
5 days	1	0.2313	0.7066	0.6090
10 days	1	0.4155	0.6843	0.6137
10 days	2	0.2484	0.6954	0.6106
14 days	2	0.4093	0.8342	0.6819
20 days	1	0.4550	0.6686	0.6102
30 days	3	0.4212	0.7193	0.6288
100 days	4	0.4508	0.6935	0.6105

References

- [1] A.F. Blumberg, G.L. Mellor, A description of a three-dimensional coastal ocean circulation model, in: *Three-Dimensional Coastal Ocean Models*, American Geophysical Union (AGU), 1987, pp. 1–16.
- [2] 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).
- [3] L. Meng, X.-H. Yan, Remote sensing for subsurface and deeper oceans: an overview and a future outlook, *IEEE Geosci. Remote Sens. Mag.* 10 (3) (2022) 72–92.
- [4] M.R. Carnes, W.J. Teague, J.L. Mitchell, Inference of subsurface thermohaline structure from fields measurable by satellite, *J. Atmos. Ocean. Technol.* 11 (2) (1994) 551–566, Place: Boston MA, USA Publisher: American Meteorological Society.
- [5] 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, *J. Atmos. Ocean. Technol.* 22 (11) (2005) 1762–1781, Place: Boston MA, USA Publisher: American Meteorological Society.
- [6] 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, *J. Mar. Syst.* 46 (1) (2004) 85–98.
- [7] S. Guinehut, A.-L. Dhomp, G. Larnicol, P.-Y. Le Traon, High resolution 3-D temperature and salinity fields derived from in situ and satellite observations, *Ocean Sci.* 8 (5) (2012) 845–857.
- [8] H. Wang, G. Wang, D. Chen, R. Zhang, Reconstruction of three-dimensional Pacific temperature with Argo and satellite observations, *Atmos.-Ocean* 50 (sup1) (2012) 116–128, Publisher: Taylor & Francis.
- [9] J. Wang, G.R. Flierl, J.H. LaCasce, J.L. McClean, A. Mahadevan, Reconstructing the ocean's interior from surface data, *J. Phys. Oceanogr.* 43 (8) (2013) 1611–1626, Place: Boston MA, USA Publisher: American Meteorological Society.
- [10] 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, *J. Mar. Syst.* 173 (2017) 21–30.
- [11] V. Klemas, X.-H. Yan, Subsurface and deeper ocean remote sensing from satellites: an overview and new results, *Prog. Oceanogr.* 122 (2014) 1–9.
- [12] M.M. Ali, D. Swain, R.A. Weller, Estimation of ocean subsurface thermal structure from surface parameters: a neural network approach, *Geophys. Res. Lett.* 31 (20) (2004).
- [13] B. Buongiorno Nardelli, A deep learning network to retrieve ocean hydrographic profiles from combined satellite and in situ measurements, *Remote Sens.* 12 (19) (2020).
- [14] 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 Sci.* 18 (4) (2022) 1221–1244.
- [15] A. Pascual, D. Gomis, Use of surface data to estimate geostrophic transport, *J. Atmos. Ocean. Technol.* 20 (6) (2003) 912–926, Place: Boston MA, USA Publisher: American Meteorological Society.
- [16] 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, *J. Geophys. Res.: Oceans* 125 (2) (2020) e2019JC015840, eprint: <https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2019JC015840>.
- [17] H. Yan, R. Zhang, H. Wang, S. Bao, C. Bai, Practical dynamical-statistical reconstruction of ocean's interior from satellite observations, *Remote Sens.* 13 (24) (2021).
- [18] 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 Sens.* 12 (24) (2020).
- [19] 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. Rebeck, 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).
- [20] J.A. Ekstrom, L. Suatoni, S.R. Cooley, L.H. Pendleton, G.G. Waldbusser, J.E. Cinner, J. Ritter, C. Langdon, R. van Hooijdonk, 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 Clim. Change* 5 (3) (2015) 207–214.
- [21] 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 (5) (2020) 396–414.
- [22] T.P. Sapsis, Output-weighted optimal sampling for Bayesian regression and rare event statistics using few samples, *Proc. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci.* 476 (2234) (2020) 20190834.
- [23] X. Yang, D. Venturi, C. Chen, C. Chrysostomidis, G.E. Karniadakis, EOF-based constrained sensor placement and field reconstruction from noisy ocean measurements: application to Nantucket Sound, *J. Geophys. Res.: Oceans* 115 (C12) (2010).
- [24] C. Chen, H. Liu, R.C. Beardsley, An unstructured grid, finite-volume, three-dimensional, primitive equations ocean model: application to coastal ocean and estuaries, *J. Atmos. Ocean. Technol.* 20 (1) (2003) 159–186.
- [25] C. Chen, R. Beardsley, G. Cowles, An unstructured-grid finite-volume coastal ocean model (FVCOM) system, *Oceanography* 19 (2006) 78–89.
- [26] 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, in: N. Pinardi, J. Woods (Eds.), *Ocean Forecasting: Conceptual Basis and Applications*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2002, pp. 55–72.
- [27] Y. Chao, Z. Li, J.D. Farrara, P. Hung, Blending sea surface temperatures from multiple satellites and in situ observations for coastal oceans, *J. Atmos. Ocean. Technol.* 26 (7) (2009) 1415–1426.
- [28] H. Babaee, C. Bastidas, M. Defilippo, C. Chrysostomidis, G. Karniadakis, A multi-fidelity framework and uncertainty quantification for sea surface temperature in the Massachusetts and Cape Cod Bays, *Earth Space Sci.* 7 (2020).
- [29] E.N. Lorenz, Empirical Orthogonal Functions and Statistical Weather Prediction, Scientific Report, Massachusetts Institute of Technology, Department of Meteorology, 1956.
- [30] A. Hannachi, I.T. Jolliffe, D.B. Stephenson, Empirical orthogonal functions and related techniques in atmospheric science: a review, *Int. J. Climatol.* 27 (9) (2007) 1119–1152.
- [31] R. Everson, P. Cornillon, L. Sirovich, A. Webber, Empirical eigenfunction analysis of sea surface temperatures in the Western North Atlantic, *AIP Conf. Proc.* 375 (1) (1996) 563–590.
- [32] L.M. Berliner, C.K. Wikle, N. Cressie, Long-lead prediction of Pacific SSTs via Bayesian dynamic modeling, *J. Clim.* 13 (22) (2000) 3953–3968.
- [33] 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, *Int. J. Remote Sens.* 32 (4) (2011) 933–950.
- [34] T.M. Smith, R.W. Reynolds, R.E. Livezey, D.C. Stokes, Reconstruction of historical sea surface temperatures using empirical orthogonal functions, *J. Clim.* 9 (6) (1996) 1403–1420.
- [35] R. Everson, L. Sirovich, Karhunen–Loève procedure for gappy data, *J. Opt. Soc. Amer. A* 12 (8) (1995) 1657–1664.
- [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] J. Yan, L. Mu, L. Wang, R. Ranjan, A.Y. Zomaya, Temporal convolutional networks for the advance prediction of ENSO, *Sci. Rep.* (2020).
- [38] 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, *J. Comput. Phys.* 442 (2021) 110484.

- [39] S. Bai, J.Z. Kolter, V. Koltun, An empirical evaluation of generic convolutional and recurrent networks for sequence modeling, 2018.
- [40] P. Lara-Benítez, M. Carranza-García, J.M. Luna-Romera, J.C. Riquelme, Temporal convolutional networks applied to energy-related time series forecasting, *Appl. Sci.* 10 (7) (2020).
- [41] 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, *Front. Mar. Sci.* 6 (2019).
- [42] T.M. Chin, J. Vazquez-Cuervo, E.M. Armstrong, A multi-scale high-resolution analysis of global sea surface temperature, *Remote Sens. Environ.* 200 (2017) 154–169.
- [43] 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, *Photogramm. Eng. Remote Sens.* 85 (2019) 119–131.
- [44] Y. Zhu, Y. Bo, J. Zhang, Y. Wang, Fusion of multisensor SSTs based on the spatiotemporal hierarchical Bayesian model, *J. Atmos. Ocean. Technol.* 35 (1) (2018) 91–109.
- [45] 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, *Bull. Am. Meteorol. Soc.* 88 (8) (2007) 1197–1214.
- [46] 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 Sens. Environ.* 135 (2013) 52–63.
- [47] R.W. Reynolds, A real-time global sea surface temperature analysis, *J. Clim.* 1 (1) (1988) 75–87.
- [48] R.W. Reynolds, T.M. Smith, Improved global sea surface temperature analyses using optimum interpolation, *J. Clim.* 7 (6) (1994) 929–948.
- [49] 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 Trans. Geosci. Remote Sens.* 53 (9) (2015) 5021–5035.
- [50] M. Raissi, H. Babae, G.E. Karniadakis, Parametric Gaussian process regression for big data, *Comput. Mech.* 64 (2019).
- [51] C.E. Rasmussen, C.K.I. Williams, *Gaussian Processes for Machine Learning*, in: Adaptive Computation and Machine Learning, MIT Press, Cambridge, MA, USA, 2006.
- [52] P. Perdikaris, D. Venturi, J.O. Royset, G.E. Karniadakis, Multi-fidelity modelling via recursive co-kriging and Gaussian-Markov random fields, *Proc. R. Soc. A* 471 (2179) (2015) 20150018.
- [53] 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.
- [54] GPy, GPy: A Gaussian process framework in python, 2012, <http://github.com/SheffieldML/GPy>.