Forecasting Buoy Observations Using Physics-Informed Neural Networks

Austin B. Schmidt[1], Pujan Pokhrel[1], Mahdi Abdelguerfi[1], Elias Ioup[1], and David Dobson[1]

[1]Affiliation not available

April 18, 2024

Abstract

Methodologies inspired by physics-informed neural networks (PINNs) were used to forecast observations recorded by stationary ocean buoys. We combined buoy observations with numerical models to train surrogate deep learning networks that performed better than with either data alone. Numerical model outputs were collected from two sources for training and regularization: the hybrid circulation ocean model and the fifth ECMWF reanalysis experiment. A hyperparameter determines the ratio of observational and modeled data to be used in the training procedure, so we conducted a grid search to find the most performant ratio. Overall, the technique improved the general forecast performance compared with nonregularized models. Under specific circumstances, the regularization mechanism enabled the PINN models to be more accurate than the numerical models. This demonstrates the utility of combining various climate models and sensor observations to improve surrogate modeling.

Forecasting Buoy Observations Using Physics-Informed Neural Networks

Austin Schmidt*, Pujan Pokhrel†, Mahdi Abdelguerfi‡,
Elias Ioup §, and David Dobson ¶

Canizaro Livingston Gulf States Center for Environmental Informatics*†‡
University of New Orleans
New Orleans, Louisiana 70148
Email: {*aschmid2, †ppokhre1, ‡mahdi}@uno.edu

Center for Geospatial Sciences Naval Research Laboratory§¶
Stennis Space Center, Mississippi 39529
Email: {§elias.ioup, ¶david.dobson}@nrlssc.navy.mil

Abstract

Methodologies inspired by physics-informed neural networks (PINNs) were used to forecast observations recorded by stationary ocean buoys. We combined buoy observations with numerical models to train surrogate deep learning networks that performed better than with either data alone. Numerical model outputs were collected from two sources for training and regularization: HYCOM and ERA5. A hyperparameter determines the ratio of observational and modeled data to be used in the training procedure, so we conducted a grid search to find the most performant ratio. Overall, the technique improved the general forecast performance compared to non-regularized models. Under specific circumstances, the regularization mechanism enabled the PINN models to be more accurate than the numerical models. This demonstrates the utility of combining various climate models and sensor observations to improve surrogate modeling.

Index Terms

PINN, Deep Learning, HYCOM, ERA5, Recurrent Model, Surrogate Model

I. INTRODUCTION

OCEAN parameter forecasting is studied for various applications, like climate modeling, marine life population surveying, and water quality monitoring. There is a clear need across industries to have fast and far-reaching forecasts. As such, research and improvements in ocean and climate modeling tools have continued to be interesting and necessary in literature. Well-studied numerical solutions for this task include Navier-Stokes and advection-diffusion, which are formulated as sets of partial differential equations (PDEs) for modeling flow systems. Building primitive equations into a more complex model yields global ocean and climate models for accurate, full-coverage simulations [1] [2] [3]. The initial values and boundary conditions of the modeled system are important for accurately modeling physical behaviors in this way [4]. Initial values are recorded as sparse observations across the world's oceans using different methods. These methods include free-floating buoys that record data by following ocean currents, stationary buoys for monitoring fixed locations, and satellites for collecting global imagery [5]. As the viability of the modeled forecasts greatly depends on accurate estimations of the initial values, data assimilative systems have been a point of research, and assimilating observations with numerical models has shown improved results [6]. In the case of the United States Navy, researchers have developed the global coupled atmosphere-ocean-sea ice forecasting system called the Navy Earth System Prediction Capability (Navy-ESPC) where modeled data is assimilated with observations for an improved result [7]. However, observations can be missing such that there is no data availability. In this situation, the data assimilation scheme cannot be taken advantage of. Therefore, there exists some motivation to generate discrete observation forecasts for their integration into an assimilation pipeline. To this end, we investigate a generalized procedure to predict sparse ocean observation values.

Surrogate deep learning models are trained using available historical data to model a system given prior input values. The main benefit of this technique is that forecasts are generated more quickly than when evolving a numerical model. Recurrent network architectures like long short-term memory (LSTM) networks and Transformers are used to propagate information forward when making long-term predictions, making them popular choices for modeling ocean parameters as surrogate models [5]. When surrogate modeling ocean parameters, data is required from recorded observations, numerical model outputs, or both. In this work, we take particular interest in two data assimilated numerical models which provide training and regularization data. The

Hybrid Circulation Ocean Model (HYCOM) is a hybrid isopycnic model which sees improvement over its predecessor in shallow water and unstratified ocean regions. [1]. ERA5 is the fifth reanalysis experiment of the European Centre for Medium-Range Weather Forecasts (ECMWF) model for global climate and weather features. [2].

By combining the numerical models with buoy-collected observation data, we show how a physics-regularized approach can be used to improve observation forecasting. Thus, we consider physics-informed neural networks (PINNs) for approximating numerical models to accurately forecast a single discrete point (i.e., an observation). A PINN is a neural network which is regularized at training time by applying penalties in the loss function. The penalties are scored by comparing adherence to a PDE-based numerical model [4]. We investigate if the forecasting result of real-world sensor data collected by stationary ocean buoys can be more accurately forecasted when regularized by the prior mentioned numerical models. Since reanalysis data exists for many ocean and climate features, we use the high-quality numerical model outputs to regularize our PINN model.

As far as we know, we are the first to integrate HYCOM and ERA5 data as a regularizing source in a PINN-inspired network. We show that the physical models may be used with recorded buoy data to provide more stable long-term predictions due to the regularization support. Our methodology differs from other PINN research by modeling only observations and, more importantly, by the way in which we implement the loss function. These differences will be discussed further in the upcoming Related Works section. To assess our models, sea surface temperature (SST), gust strength, and air pressure are sparsely forecasted using our technique. The main contributions of this paper are as follows. We train deep learning models to recursively forecast physical parameters as recorded by free-floating ocean buoys. We define a custom loss function to use numerically modeled data and observation data as sources for training physics regularized models. The methodology is capable of handling situations where a physical parameter is available from both sources or a single source. When both sources of data are available for a feature, we show how the surrogate may be trained using a ratio of the training errors from each source. The most performant surrogate for the test data is found through a grid search of the static regularization term, λ, which controls the ratio of errors. We demonstrate the flexibility of PINNs to combine different numerical models using a surrogate deep learning model, which outperforms the non-regularized deep learning models. We discuss the numerical models and their effect on the rolling forecast ability of our surrogate model for up to 24 hours.

The rest of this paper is formatted as follows: II. Related Works; III. Methods; IV. Results; and V. Conclusions.

II. Related Works

Ocean surrogate models have been advancing with the advent of deep learning, and more refined machine learning approaches [5][8]. Research into deep learning surrogate modeling of SST shows promising results as SST can be forecasted as discrete points [9], as a field [10], or as a super-resolution field [11]. Instead of directly solving intractable formulations like Naiver-Stokes or other prognostic equations for ocean modeling, a data-driven surrogate model is trained using the substantial amounts of historical training data available via numerical models or raw observations [12]. The use of observation assimilated models to train deep learning surrogates has been seen multiple times using both HYCOM [13][14] and ERA5 [15][16][17] models. Through back propagation a deep learning model learns a parameterized representation of the underlying physical phenomenon which are otherwise modeled numerically. Surrogate models may be preferred over traditional models due to faster outputs once the model has been trained [8]. For example, in [18], approaching hurricane parameters are forecasted in seconds. Machine learning surrogate models will generally have more numerical instability when compared to numerical models in forecasting experiments. This speed and accuracy trade-off is seen in the conclusions of surrogate modelling studies for data assimilation in dynamic subsurface flow [12] and regional wind/wave forecasting [19]. In both papers, the forecast accuracy was similar or lower than numerical models, but the computational speed was greatly improved. One keynote on numerical stability and model accuracy is that the generalization of machine learning surrogate modeling is not assured for all cases. Authors observe the stability difference in operational planning with dynamic constraints where the forecasting stability is very good for some deep learning surrogate models but unstable when using other machine learning techniques [20]. This forecasting stability problem is also considered in [21] where outputs of physics-based numerical models are combined and used as supervised learning training sets to promote more accurate forecasts than when used independently. Furthermore, the surrogate modeling task can be used with data assimilation to correct numerical model error in an online fashion [22]. As such, surrogate models have a place among the more carefully calculated simulation-based numerical models, like HYCOM and ERA5. This is especially true in applications where numerical solutions are too complex or computationally intensive for real time analysis and the acceptable error threshold is high.

Physics-informed neural networks are referred to as such because they leverage physical constraints within the model's loss function during training to enforce convergence to governing physical laws. This type of network was popularized in the deep learning community by Raissi et al. in 2017 and 2019 [23]. The introduction of differential equations that define physical phenomenon to the training procedure is found to improve the model's resilience to noise [24]. PINNs are regularized in training by comparing model performance to the adherence of the introduced PDEs while also fitting data points to unique solutions [25]. The result of these forecasting models is that we can incorporate noisy data into existing algorithms, ignore complex mesh generation, and tackle high-dimensional problems governed by parameterized PDEs. Originally, research has focused on surrogate modeling with PINNs for solving systems governed by the Burgers' and Navier-Stokes equations [26]. PINNs have recently been investigated in industry informatics settings such as modeling flow equations for ocean models [24], modeling crack propagation [27][28], modeling leakage [29], modeling faults [30], and modeling electric loads [31]. Forecasting SST is commonly found as a full-coverage modeling problem combining either generative models [32][33] or convolutional neural networks [34] with various PDEs. Continual discussion on PINNs and the types of equations usually solved can be reviewed in [4] and [35].

We have not seen any other works that use a ratio of numerical model data and observations to train and regularize a deep neural network for surrogate modeling. Our methods share similarities with [21], who utilizes numerical models as training data for surrogate models. However, we employ our PINN-inspired approach to regularize models by combining both observations and numerical outputs. Furthermore, our work differs methodologically from the prior mentioned PINN research in two significant ways. First, there is no differentiation or simulation step to solve selected PDEs within the surrogate training procedure. This is the case because the numerical model pipeline is too computationally intensive for this to be feasible. Instead, the selected climate and oceanography models, HYCOM and ERA5, have already undergone comprehensive modeling and data assimilation processes which provide high quality, historical simulation data. Using the pre-computed data instead of directly solving PDEs means the numerical model can be arbitrarily complex and we do not need to implement the formulation for use in our framework. The second divergence is the role of the hyperparameter λ within the PINN loss function. The traditional PINN training loss function sums the performance of the surrogate model and the divergence when compared to the numerical solution of selected PDEs. In that case, λ is used as the multiplicative weighting term to determine how much of a contribution the divergence from the numerical solution has on the final loss output. Instead, we use λ as a mechanism to control a weighted ratio of observation versus modeled data in training. This ratio of loss from multiple sources improves the training process when numerical data, observational data, or both are noisy. The proposed buoy forecasting task is inspired by [36], but we forecast multiple buoy parameters, test additional numerical models (ERA5 and HYCOM), and apply our physics-regularized training methodology, as main differences. So, we show, in an experimental approach, that we may use complex solutions calculated by numerical climatology and ocean flow models as a means of regularizing surrogate PINN models. We aim to demonstrate that a PINN can internalize the simulated outputs of ocean and climate models to be more capable of forecasting unseen buoy values.

III. Methods

In this section, we discuss the methodologies utilized in investigating our PINN-inspired surrogate models. The models are trained to forecast ocean observations at fixed locations given prior conditions. The numerical models, HYCOM and ERA5, regularize the model at training time and offer additional input features. The section is organized as follows: A. Numerical Models Overview; B. Data and Feature Processing; C. Deep Learning Models; and D. Metrics and Testing Strategy.

A. Numerical Models Overview

The Hybrid Circulation Ocean Model (HYCOM) system is a primitive equation model for general ocean circulation that evolved from the Miami Isopycnic-Coordinate Ocean Model (MICOM) system developed by Rainer Bleck and associates [1] [3]. HYCOM, like MICOM, is a primitive-equation model containing five prognostic equations. Two equations for the horizontal velocity components, a mass continuity or layer thickness tendency equation, and two conservation equations for a pair of thermodynamic variables, such as salt and temperature or salt and density. The authors also define several diagnostic equations to control the spacing and movement of layer interfaces. This includes the hydrostatic equation which links temperature, salinity, and pressure, alongside an equation prescribing the vertical mass flux through a surface. A hybrid grid-generating technique determines whether isopycnal or inflated non-isopycnal layers are specified [1]. Beyond the general governing equations and gridding algorithm, HYCOM has specialized mixing processes, many of which are shared with the MICOM implementation. Temperature and salinity profiles are assimilated into the ocean flow model to improve initial analysis. The specific HYCOM implementation we use for data is the 41-layer HYCOM + NCODA Global $1/12°$ Reanalysis experiment.

ERA5 is the fifth ECMWF reanalysis for global climate and weather features. The atmospheric global reanalysis (HRES) includes the period from January 1950 to the present year. ERA5 reanalysis is produced using the 4D-Var data assimilation technique and model forecasts with 137 hybrid vertical sigma/pressure levels [2]. The data assimilation of ERA5 also contains an ensemble system (EDA) of ten members for providing background error estimates. The model assimilates as many observations as possible in the upper air and near-surface regions. This forecasting system includes over a decade of research and development for all components: atmosphere, land, and ocean waves. The integrated forecast system (IFS) implemented by ECMWF has its equations expertly discussed in the documentation manual [37] and is more generally discussed in [2]. We specifically use the ERA5 hourly data on single levels from 1959 to the present [38], which is a data assimilative reanalysis that uses the 2016 version of the ECMWF numerical weather prediction model and data assimilation system (IFS Cy41r2). The ERA5 implementation is modeled at $1/4°$ latitude/longitude increments. Thus, the resolution of ERA5 is lower than that of HYCOM.

Given these arbitrarily complex numerical models, which are pre-computed, we do not need to implement the PDEs which govern the models directly. Instead, we will use the outputs from both models as training and regularization data within our deep learning models. To yield discrete value forecasting in a generic manner, we only need the values which are geographically closest to the latitude and longitude of the buoy observations. Likewise, we collect the discrete time step temporally closest to the observations we are interested in. Therefore, we consider a generic method for retrieving data from full-coverage numerical models in (1).

$$f_m(t, x, y) = v \tag{1}$$

For a sufficiently complex model f_m, we input the desired period t and the closest possible latitude and longitude, x and y. This yields whichever set of scalar features v are desired from the numerical model. These values can then be used as regularization data, training data, or both for a deep learning PINN model. This formulation is useful in our methodology where we want to train a neural network on the observations themselves while regularizing with numerical model data. This differs to similar PINNs that provide full-coverage modeling of ocean and climate features, where the training data is limited to full-coverage reanalysis and the regularizing PDEs are formulated from simpler equations as seen in [32] [33] [34].

B. Data and Feature Processing

Both buoy observations and numerical model outputs are publicly available and have decades worth of data. In this study, we select dates from January 1st, 2011, to December 31st, 2011. The buoy data, which comprises the observation data for this study, comes from three-meter discus Self-Contained Ocean Observations Payload (SCOOP) sensor package buoys and Waverider buoys. We select 124 candidate buoys from around the United States East and West Coasts, the Caribbean, and the Gulf of Mexico. The buoy data is collected from the National Oceanic and Atmospheric Administration (NOAA) public data center. NOAA arranges individual buoys systematically by assigning each one a distinct ID number. The specific ID corresponding to each buoy selected for analysis is found in the Appendix. Water temperature, air pressure, and gust strength are extracted from the buoy feature set to provide the real-world recorded result. Since HYCOM and ERA5 are both gridded datasets, we select the data points which match the latitude and longitude as closely as possible to each buoy position. HYCOM snapshots are taken every three hours, and most buoys are recorded at the 50th minute of each hour. Therefore, we forecast buoy features in three-hour increments. To facilitate the coupling of the numerical models and buoy data, we select buoy features that have matching modeled numerical features. Out of the eighteen selected features, water temperature, gust strength, and air pressure are shared by the numerical models and the buoys, so they will be coupled in training time, as described by the loss function. We display all features recorded from the buoys and numerical models in Table I along with their original units.

It is possible that data is missing from our data sources in two separate ways. A value may be missing temporally such that no data is recorded at all for a particular time step. This is most common in the NOAA buoy data where, for example, a buoy faces mechanical failure and cannot record observations for days to months at a time. Therefore, our training and testing data is limited by the amount of available buoy-recorded data. The numerical models do not leave a time step without data except in one case, a 24 hour gap found within the HYCOM dataset. Since this represents only eight data points, we cover the temporal gap by replacing the missing time steps with the previous 24 hour period. Otherwise, for a given time step, features may be

TABLE I

DATA FEATURES AND THEIR SOURCES. IN BOLD ARE NUMERICAL MODEL FEATURES TO BE COUPLED AS A REGULARIZATION MECHANISM WHEN FORECASTING BUOY OBSERVATIONS.

Feature Name	Feature Units	Feature Source
Water Temperature	°C	Buoy
Gust Strength	m/s	Buoy
Air Pressure	hPa	Buoy
Water Temperature	°C	**HYCOM**
Salinity	psu	HYCOM
Surf Elevation	m	HYCOM
Water Eastern Flow (U)	m/s	HYCOM
Water Northern Flow (V)	m/s	HYCOM
Wind Eastern Flow (U)	m/s	ERA5
Wind Northern Flow (V)	m/s	ERA5
Evaporation	m of w.e.	ERA5
Gust Strength	m/s	**ERA5**
Mean evaporation Rate	kg/($m^{-2}s^{-1}$)	ERA5
Mean Runoff Rate	kg/($m^{-2}s^{-1}$)	ERA5
Sea-Ice Cover (%)	[0-1]	ERA5
Air Pressure	hPa	**ERA5**
Cloud Cover	[0-1]	ERA5
Precipitation	m	ERA5

missing data and are replaced with fill values of 99, 999, 9999, or -32767, depending on the data source and feature. Each of our sources of data exhibits at least some fill data, depending on the geographical region or time of year. We remove all fill values from the data and, in their place, linearly interpolate the missing values forwards and backwards for that individual buoy or numerical model. If any numerical model data source is composed of more than 20% fill values, we disregard that corresponding buoy from the training and testing pipeline. No buoys are discarded for having too many fill values for the purpose of preserving as much data for training and testing as possible. It is important to note that the retention of buoys with interpolated values can have an impact on model accuracy.

The processed data is split into three datasets for training, validation, and testing. As each buoy is missing various days, we select the train, test, and validation splits by date. Therefore, all members of the training data are chosen from January 1st to September 13th. The validation data is from September 13th to October 20th. The testing data includes the remainder of the year. Since the buoys are missing data at separate times of the year, a buoy may occasionally contribute to one dataset but not another. We specify the buoy selection in Table VI where we display the number of buoys allowed into each dataset. There are 148,365 training instances, 23,118 validation instances, and 48,039 testing instances. Among the original 124 buoys selected for processing, only 86 buoys had training, validation, and testing data available. Each feature is independently normalized between -1 to 1 before training, using the training data minimum and maximum values. This approach is essential in deep learning to prevent data with varying scales from dominating the network's performance. As our network is trained on scaled data, we transform the network's output to its original scale for meaningful result comparison.

To understand the impact of first-order differenced data on our regularizing technique, we studied two separate setups. In the first, we train the models using the original values recorded by the data sources. Subsequently, we take the first-order difference to train the model on the differences between time steps. Training with differenced values to make the data stationary is seen for non-regularized RNNs [39] and physics regularized RNNs [40] when forecasting time series. Stationarity means that a time series has been stabilized such that it has consistent statistical properties, like mean and variance [41]. Non-stationary data contains trends and seasonality that may introduce bias to the surrogate models. Taking the first-order difference of our data removes trends in the training data and makes the analysis problem more forgiving. The result is that modeling using the differenced data will result in higher accuracy and a more stable forecast. The more consistent statistics also imply more accurate scaling when normalizing the test data. Non-stationary data is still useful for models with longer context windows or the addition of features which are embedded in time, so testing both data representations is worthwhile. In our experiments, we will clearly denote the data used when training or evaluating a surrogate model as either original data or differenced data. When comparing models which forecast the differences in data rather than the original data, we need to transform the resulting forecast back to the original scale. This transformation is computed by summing the forecast f_t with the initial conditions x_{t-1}, then that value is summed iteratively with each following difference forecast in the horizon window.

C. Deep Learning Models

A PINN is made up of any general network architecture. Since we are forecasting time series, we experiment on architectures that utilize GRU units, LSTM units, and Transformer units. Layers of these units are accompanied by dense fully connected layers, normalization layers, and training dropout layers. Each layer includes a non-linear activation function except for some dense layers, which are linear in the Transformer architecture. Between the layers, we add dropout layers with 5% dropout rate during training for the Transformer and 10% for the LSTM. Similarly, we apply a normalization layer in between dense and LSTM layers to prevent exploding or vanishing gradients. The Transformer block is made of ten attention heads. The exact summary of the LSTM-based and Transformer-based models can be seen in Tables II and III. The GRU-based model architecture is the same as the LSTM model. The number of trainable parameters is lesser for the GRU compared to the LSTM but is otherwise the same structure. The GRU and LSTM models have much fewer weights than the Transformer based model, which takes longer to train. We include each layer of the model, the number of trainable parameters, and the activation at that layer, if any. The GRU and LSTM models are trained for 100 epochs while the Transformer model is trained for 200 epochs, due to the increased number of trainable weights. A data batch size of 256 was used in all cases. To optimize the value in each epoch of back-propagation, the Adam optimizer is selected for the Transformer model and RMSProp for the LSTM and GRU networks. The models are always trained using the same random seed to ensure experiments are as uniform as possible.

Each model, once initialized, is trained to accept the 18 specified features as input and produce the predicted next step for each feature as output. Since each model is trained to produce the same outputs it requires as inputs, this is considered a rolling forecast model. In this approach, to forecast further into the future, we may use the model's own outputs from time t as inputs for forecasting time $t + 1$. This forecasting technique depends on accurate initial values. Only the first forecast in a period, t_0, is provided with initial conditions, and as time progresses, inherent chaos or model error will compound within forecasts. This method yields models which are not constrained to a single forecast horizon. Instead, the models are more flexible, and can generically forecast any number of desired periods, once provided initial values. Using the numerical model data as inputs to our deep learning models may be considered self-fulfilling because reanalysis data includes high-quality features assimilated with ground truths not yet

TABLE II

LSTM MODEL ARCHITECTURE. THERE ARE 24 TOTAL LAYERS WITH 2,348,546 TRAINABLE PARAMETERS. N REPRESENTS A VARIABLE BATCH SIZE.

Layer Type	Output Shape	Param #	Activation
Input Layer	(N, 18, 1)	0	None
Reshape	(N, 1, 18)	0	None
Dense	(N, 1, 256)	4864	Tanh
Batch Normalization	(N, 1, 256)	1024	None
Dropout	(N, 1, 256)	0	None
LSTM	(N, 1, 256)	525312	Tanh
Dropout	(N, 1, 256)	0	None
LSTM	(N, 1, 256)	525312	Tanh
Dense	(N, 1, 256)	65792	Tanh
Batch Normalization	(N, 1, 256)	1024	None
Dropout	(N, 1, 256)	0	None
LSTM	(N, 1, 256)	525312	Tanh
Dropout	(N, 1, 256)	0	None
LSTM	(N, 256)	525312	Tanh
Dropout	(N, 256)	0	None
Dense	(N, 200)	51400	Tanh
Dropout	(N, 200)	0	None
Dense	(N, 200)	40200	Tanh
Dropout	(N, 200)	0	None
Dense	(N, 200)	40200	Tanh
Dropout	(N, 200)	0	None
Dense	(N, 200)	40200	Tanh
Dropout	(N, 200)	0	None
Dense	(N, 18)	3618	Tanh

observed. We point out that the assimilated data and observations are only used in training time and when seeding initial values into the model. The subsequent predictions use the results from the previous prediction cycle. All else is kept equal among the models, so we may measure the effects of our methodology across multiple experiments.

To train the models, the loss function for our PINN is designed such that the outputs from numerical models are coupled with buoy-extracted real-world values. To do this, a weighted ratio term is used to determine how much of the calculated error comes from the residual of buoy observations versus the residual of the HYCOM and ERA5 modeled features. This combination

TABLE III

TRANSFORMER MODEL ARCHITECTURE. THERE ARE 21 TOTAL LAYERS WITH 13,619,190 TOTAL TRAINABLE PARAMETERS. N REPRESENTS A VARIABLE BATCH SIZE.

Layer Type	Output Shape	Param #	Activation
Input Layer	(N, 18, 1)	0	None
Reshape	(N, 1, 18)	0	None
Dense	(N, 1, 512)	9728	Linear
Batch Normalization	(N, 1, 512)	2048	None
Transformer Block	(N, 1, 512)	11016692	Selu
Dropout	(N, 1, 512)	0	None
LSTM	(N, 1, 512)	2099200	Tanh
Dropout	(N, 1, 512)	0	None
Dense	(N, 1, 512)	262656	Linear
Dropout	(N, 1, 512)	0	None
Batch Normalization	(N, 1, 512)	992	None
Dense	(N, 1, 200)	2048	Linear
Dropout	(N, 1, 200)	0	None
Dense	(N, 1, 200)	102600	Linear
Dropout	(N, 1, 200)	0	None
Dense	(N, 1, 200)	40200	Linear
Dropout	(N, 1, 200)	0	None
Dense	(N, 1, 200)	40200	Linear
Dropout	(N, 1, 200)	0	None
Flatten Layer	(N, 200)	0	None
Dense	(N, 18)	3618	Linear

is completed for all coupled buoy features, i.e., water temperature, gust strength, and surface air pressure. Thus, the piece-wise cost can be calculated as follows in Equations (2)-(7).

$$\Delta_1 = |\hat{y}_{\text{obs}} - y_{\text{obs}}| \tag{2}$$

$$\Delta_2 = |\hat{y}_{\text{obs}} - f_m(t, x, y)| \tag{3}$$

$$\Omega_{\text{coupled feature loss}} = \lambda * \Delta_1 + (1 - \lambda) * \Delta_2 \tag{4}$$

The two Δ terms defined in (2) and (3) represent the absolute error between the predicted observation and the observation ground truth followed by the absolute error of the predicted observation and the numerical model output as defined in (1). The two error terms are weighted by λ, as seen in (4). The selected λ value represents a ratio to determine how much weight is provided to each ground truth. This coupled feature loss is only calculated for those features which have both an observational and modeled collection of data available. Through additional feature collection, the technique can be extended to couple any number of observation features to numeric models.

$$\Omega_{\text{modeled feature loss}} = |\hat{y}_{\text{model}} - f_m(t, x, y)| \tag{5}$$

$$\Omega_{\text{observed feature loss}} = |\hat{y}_{\text{obs}} - y_{\text{obs}}| \tag{6}$$

The remaining uncoupled features, as seen in (5) and (6), are used to collect loss in a more traditional way. Excluding the coupled features from the calculation, numerical feature forecasts are measured against numerical model values only and forecasted observational data are measured against observational ground truth only. We include additional numerical features in our setup, which were identified in Table I. There do not exist any non-coupled observational features, so $\Omega_{\text{observation forecast loss}} = 0$, in this experiment. There is no λ controlling the coupling ratio in the case of (5) and (6). The final loss function which combines the disparate loss calculations can be summarized in (7).

$$\Omega_{\text{total loss}} = \Omega_{\text{coupled forecast loss}} + \Omega_{\text{numeric forecast loss}} + \Omega_{\text{observation forecast loss}} \tag{7}$$

The addition of a coupled loss component is rationalized by considering that as the λ value approach 0.0, we are training our model to behave more like the numerical model, $f_m(t, x, y)$. Conversely, as the λ values approach 1.0, we are promoting forecasts which more closely resemble the observations, y_{obs}. Expanding the example, when $\lambda = 0.5$, the model balances agreement between both sources equally. In our experiments, the ground truth is measured using y_{obs}, so when $\lambda = 1.0$, we are essentially training a model while using no regularization strategy.

D. Metrics and Testing Strategy

For the original data and differenced data setups the SST, gust strength, and air pressure are forecasted over the reserved testing data for final evaluations of each model. Test horizon windows are conducted from one period to eight periods, where an individual period measures data collected every three hours. Therefore, this manifests as a one-step three hour forecast through an eight-step 24 hour forecast since each forecast step is three hours apart. Using the rolling forecast property mentioned, we record the mean absolute error (MAE) and root mean square error (RMSE) for each forecast period. The MAE is calculated as follows for an individual buoy $\frac{1}{N}\sum_{i=1}^{N}(|Y_i^p - Y_i^t|)$, where N is the total number of time steps forecasted, Y^p is the collection of predicted ocean features, and Y^t is the collection of ground truth ocean observations. Similarly, the RMSE is computed as $\sqrt{\frac{1}{N}\sum_{i=1}^{N}((Y_i^p - Y_i^t)^2)}$. In analysis, the total MAE and RMSE from our test results are collected from each buoy and then averaged to find the global mean metrics. The best possible model will provide low value metrics for all forecast periods and features. To verify whether the coupled loss component works as a regularization mechanism, we evaluate for λ values between 0.0 and 1.0 with 0.1 step intervals. Next, we evaluate around the best scoring λ values using 0.02 step intervals. The results gathered in this way may be contrasted with the numerical model outputs from HYCOM and ERA5, which are scored using the same metrics. Using this grid search technique, we are not guaranteed to find the λ value which yields global minimal error, so we aim to highlight two behaviors instead. The first is that there exists a value of λ, where the RMSE, MAE, or both are lesser than $\lambda = 1.0$ (no regularization), for at least one feature per model. The second is that the selection of best λ is influenced by inconsistencies in the observation data, misalignment in the numerical model data, and the PINN architecture.

IV. Results

We consider which experiments yield the lowest error metrics given various PINN model setups, our three physical features of interest, and whether the data has been differenced or not. Beyond providing an accurate forecast, we are primarily interested in the regularization ability of the PINN's specialized loss function. As such, we begin by considering which values of λ yield the lowest error metrics. Then, the general forecasting ability of our highest performing models will be considered for further context. Finally, we will examine the buoy accuracy given its geographical region to consider where our method may struggle to provide high-quality outputs. In the Appendix, we supply Tables VII-XII to display the RMSE results gathered from our PINN models trained on various λ values. In the Tables, each feature from horizons starting with three hours (one period) and up to 24 hours (eight periods) are given to see the evolution of error over time.

A. Selection of Best λ Values

We present the best value for λ given variations in our PINN models and the selected coupled feature. A series of Figures display each λ value and corresponding error metrics per model and feature. We consider the original data best λ results for the GRU model in Figure 1, the LSTM model in Figure 2, and the Transformer model in Figure 3. The λ-based ratio regularization successfully managed to reduce the MAE and RMSE of 24 hour forecasts when compared to $\lambda = 1.0$ (no regularization). For the GRU and LSTM figures, each evaluated feature displays at least one value for λ which yielded more performant metrics. Using the Transformer model, the PINN-style regularization yields explicitly worse forecasts for SST and Gust, but air pressure has a reduced error when $\lambda = 0.9$. In this sense, each model has displayed the property of MAE and RMSE reduction for at least one feature, using the regularization technique. The reason that the Transformer model performs well in the $\lambda = 1.0$ case is because the architecture is sufficiently complex enough to generalize the observations when trained using large amounts of data. However, the results of the air pressure forecasts imply some features benefit from the coupled loss function regardless of model complexity. The LSTM and GRU models are less complex and achieve worse test results overall, so the regularization has a larger effect on error reduction. For this reason, there exists a best performing model when $\lambda < 1.0$ in all features.

We highlight that the best λ values are unique for each experiment. This is true when comparing the separate features in the same model and when comparing the same feature from each model. For example, the best λ values found in the GRU features are 0.9, 0.84, and 0.96, for SST, gust strength, and air pressure, respectively. When comparing by model, the best λ for SST is largely separated at 0.9, 0.68, and 1.0 for GRU, LSTM, and Transformer models, respectively. The uniqueness of each λ selection is problematic in situations where the best λ value significantly differs between features. Each feature is coupled using the same λ value, although an optimal choice for one feature may not be optimal for all features. A multiple λ setup could allow more flexibility towards this problem.

In observing the change between λ values and their error metrics, we see some trends in each feature. The SST feature in GRU and LSTM models is inconsistent with many local minima observed. The gust strength feature displays error that is mostly consistent regardless of the selection of λ. However, there is a noticeable decrease in error as λ approaches the discovered minimal value. The most obvious trend that occurs in all PINN models is the sharp decrease in error of the air pressure feature as λ increases. This is the sole case where a regularized Transformer model outperforms the $\lambda = 1.0$ case. This is likely caused by misalignment in the ERA5 model when compared to the ground truth. Extremely divergent outliers in ERA5 mean that training the surrogate model using numerical model data is a poor choice compared to the observations. So, error decreases when $\lambda > 0.5$ and the PINN produces forecasts more aligned with the observations. Still, the ERA5 data is well-fitted outside of outlier conditions, so $\lambda < 1.0$ promotes a regularizing effect on the model. This is an example of how our methodology can combine multiple data sources to improve results when each has their own biases.

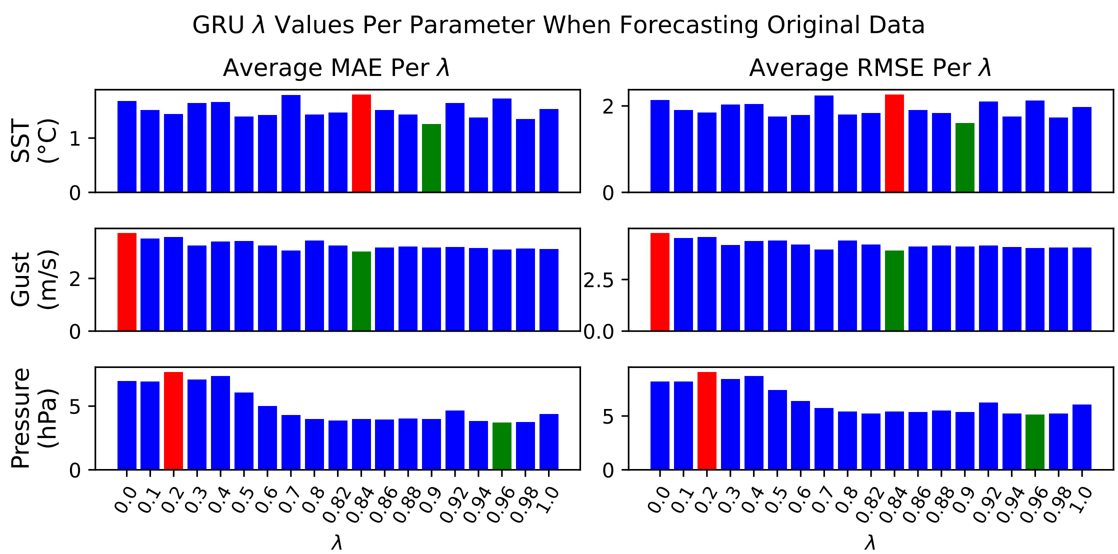


Fig. 1. MAE and RMSE for GRU forecasts from $\lambda = 0.0$ to $\lambda = 1.0$ (no regularization). The lowest scoring λ value is displayed in green while the highest is red. Forecasts are given as the original values.

Comparing the experimental results of the original data scheme to the results of the differenced data scheme shows varying results. We present the differenced data best λ results for the GRU model in Figure 4, the LSTM model in Figure 5, and the Transformer model in Figure 6. The λ-based ratio regularization scheme reduces MAE and RMSE in all but one case. As before, the Transformer yields strictly better results when $\lambda = 1.0$ for SST. However, the ERA5 features show strictly best results when $\lambda = 0.0$, achieving lowest scores when the model is only trained on numerical data. Considering the GRU and LSTM figures, each feature displays a minimizing λ that yields lower error metrics than the $\lambda = 1.0$ case. The best λ values found overall are typically closer to $\lambda = 0.0$. This is the exact opposite behavior when compared to the original results, and the trend is most obvious when considering the air pressure feature. Lower values of λ yield

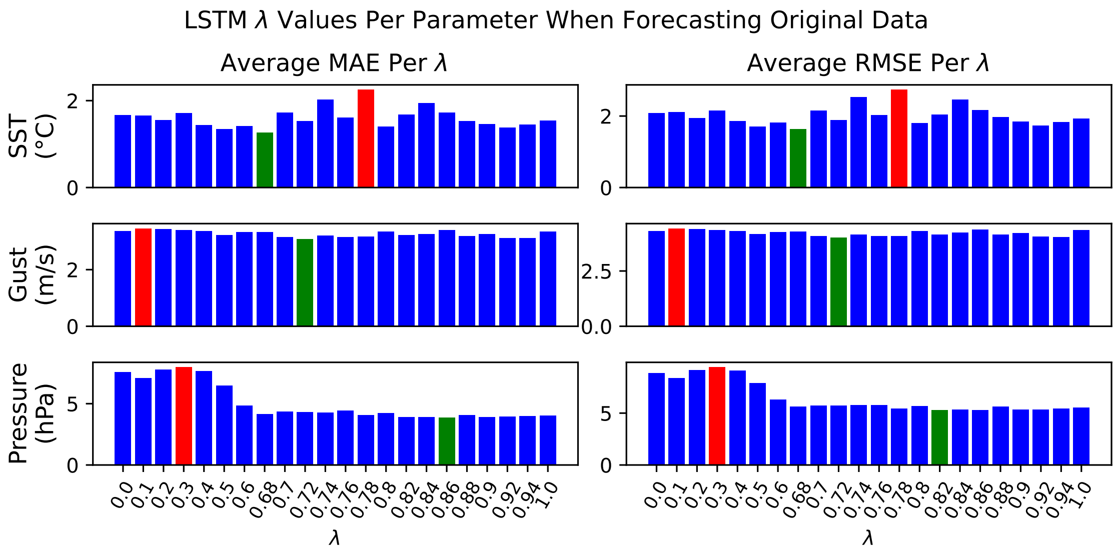


Fig. 2. MAE and RMSE for LSTM forecasts from $\lambda = 0.0$ to $\lambda = 1.0$ (no regularization). The lowest scoring λ value is displayed in green while the highest is red. Forecasts are given as the original values.

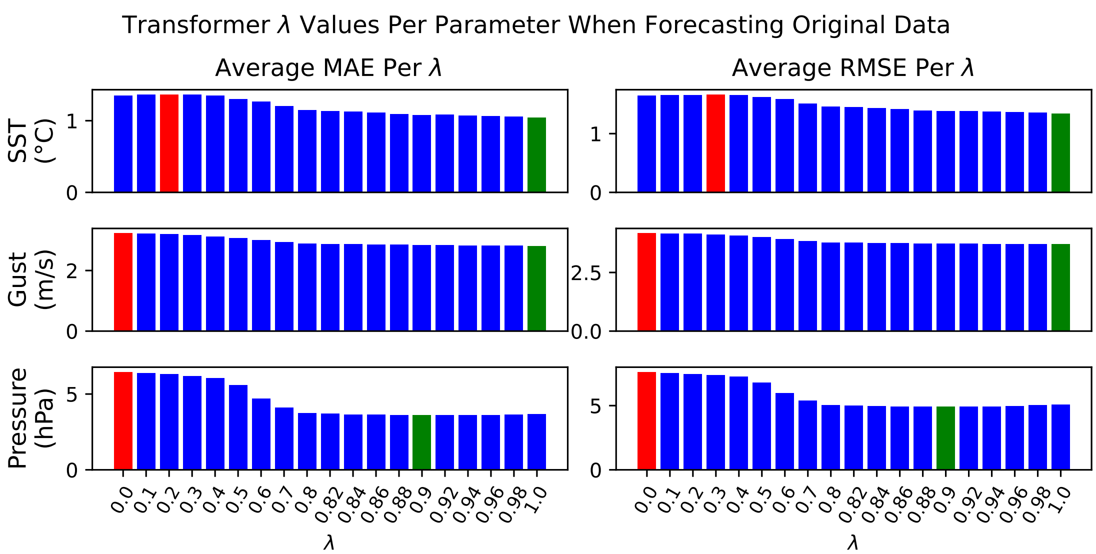


Fig. 3. MAE and RMSE for Transformer forecasts from $\lambda = 0.0$ to $\lambda = 1.0$ (no regularization). The lowest scoring λ value is displayed in green while the highest is red. Forecasts are given as the original values.

more performant results, although the absolute difference in error is small. Most importantly, each model has shown error reduction for at least two features using the regularization technique.

The λ values for SST are chaotic, like before, and the best value varies greatly per model. Conversely, the error metrics are much lower overall due to the differenced data representation. The behavior of λ regarding the gust strength feature is similar to the original data figures for the

GRU and LSTM models. In all, the selection of a wider variety of lower λ values suggests that the rate of change in both datasets are alike. The numerical models also have less interpolated data which promotes more stable training. Once again, we find that most results display best λ values which are different between features and models. The one outlier comes from the Transformer model, where SST maintains a best result at $\lambda = 1.0$. Wind gust strength and air pressure both display similar values of λ between the GRU and LSTM models, but the SST varies drastically between each. This discussion underpins the idea that both the feature, the model, and the data representation influence the selection of best λ.

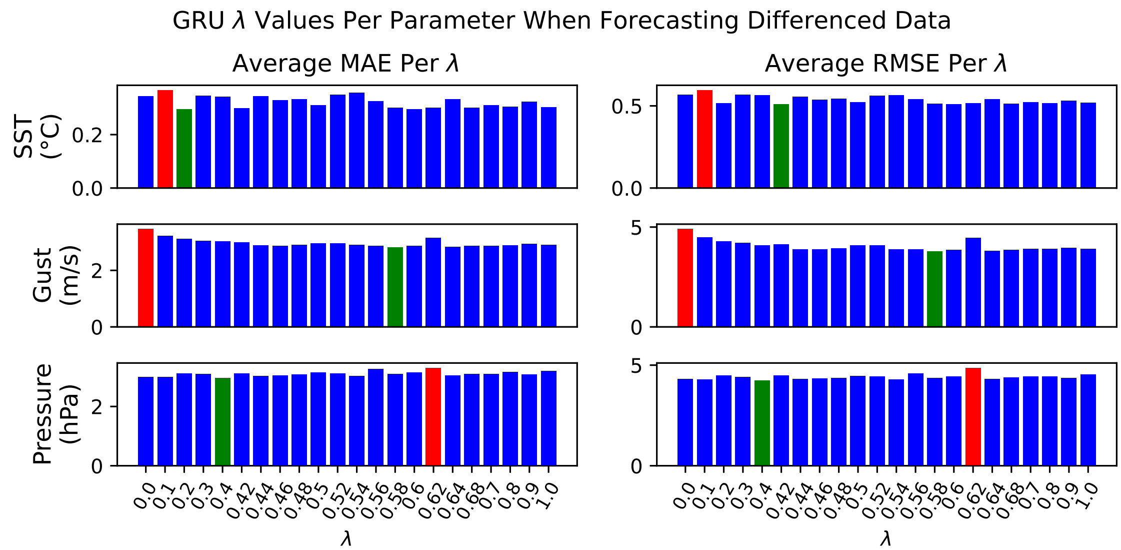


Fig. 4. MAE and RMSE for GRU forecasts from $\lambda = 0.0$ to $\lambda = 1.0$ (no regularization). The lowest scoring λ value is displayed in green while the highest is red. Forecasts are given as first-order differenced values.

In this section we considered how the selection of the best λ differs as the parameters of our experiments change. The Transformer model received the least benefit from $\lambda < 1.0$ overall. For the Transformer, the SST feature never benefits from the coupled loss, air pressure is always improved, and gust speed depends on whether the data is differenced or not. Both other models benefit at least somewhat from the regularization in all cases. We learned the benefit of the regularization and the corresponding selection of best λ are tied to the complexity of the model, where models with fewer weights benefit more when using this methodology. Another observation is that values approaching 0.0 for λ tend to yield worse results unless we are considering the differenced data representation. This is due to the way each model is trained to forecast the change between time steps. When taking a first-order difference of the data, a larger

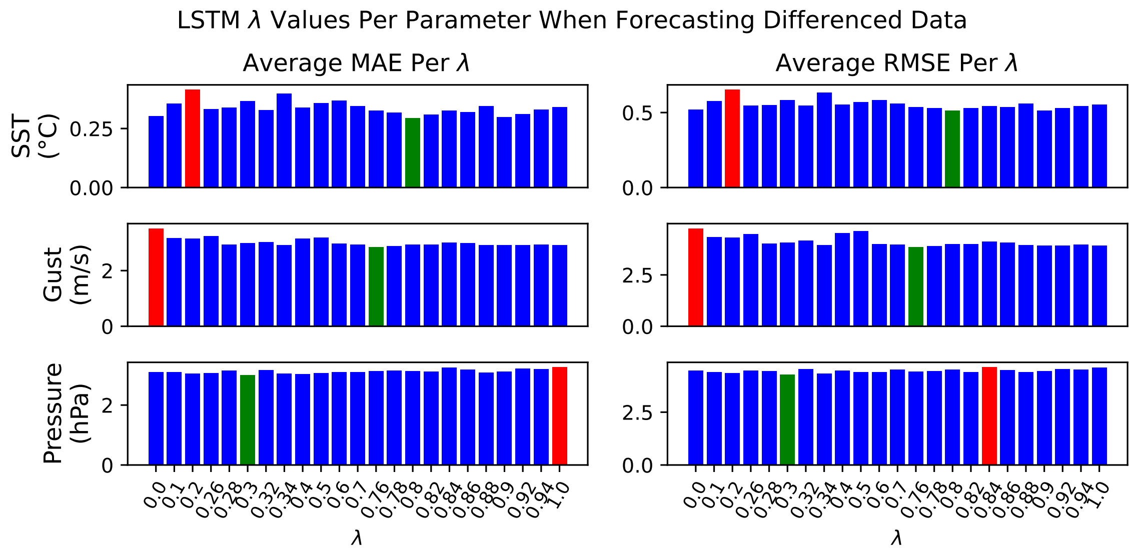


Fig. 5. MAE and RMSE for LSTM forecasts from $\lambda = 0.0$ to $\lambda = 1.0$ (no regularization). The lowest scoring λ value is displayed in green while the highest is red. Forecasts are given as first-order differenced values.

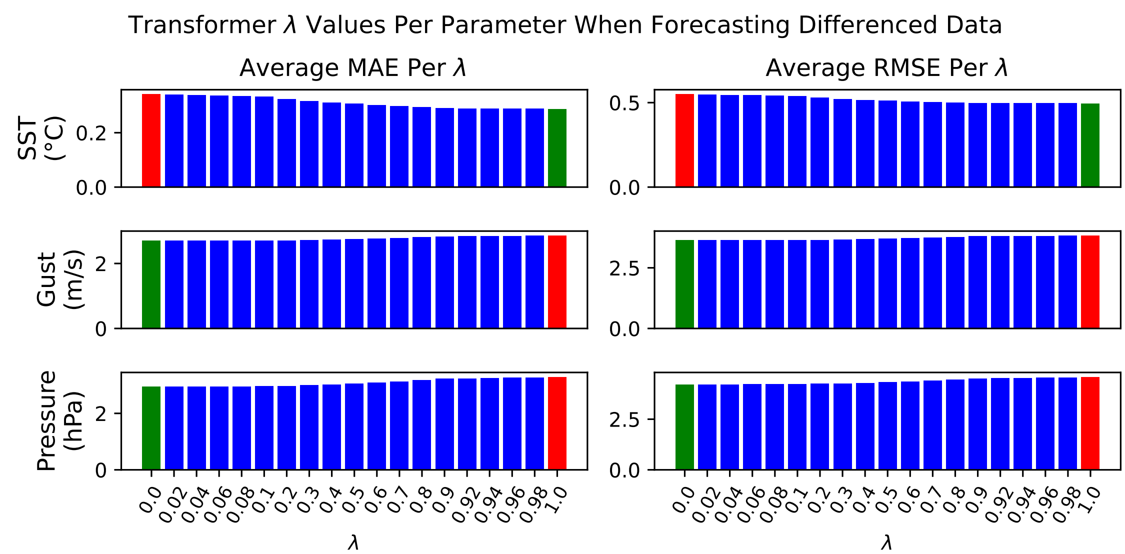


Fig. 6. MAE and RMSE for Transformer forecasts from $\lambda = 0.0$ to $\lambda = 1.0$ (no regularization). The lowest scoring λ value is displayed in green while the highest is red. Forecasts are given as first-order differenced values.

number of interpolated buoy observation values produces an uninformative training environment for differenced data. The numerical models, have fewer interpolated values and more accurately reflect change from one time to another. Therefore, PINNs which act more like the numerical model are more performant in this case. Finally, by examining the way the best λ changes in each experiment, we find that the feature, the model, and the data representation all influence the selection of best λ. Otherwise, the best λ selections would be more homogeneous overall.

B. General Forecast Accuracy

By examining the general forecast accuracy of our models, we gain additional insights into the coupled loss technique used and the stability of our PINN models. To begin, we consider the measured RMSE for the best found λ per feature. We compare this error to those derived from the $\lambda = 1.0$ case and from the numerical models for additional context. To facilitate this comparison, we introduce Tables IV for the original value forecasts and V for the differenced value forecasts. In these tables, we compare the percent change in RMSE between the best λ value and $\lambda = 1.0$ in the fourth column. In the final column, we compare the best $lambda$ value to the numerical models. These values are calculated using the RMSE as found in the eight-step forecast from the Appendix Tables VII-XII. Negative values indicate a reduced error when comparing the best λ value to the $\lambda = 1.0$ case or the numerical models. Positive values show when the best λ results are worse than the compared source of error. When the percentage is zero, the best value of λ for that experiment was $\lambda = 1.0$.

Examining the original value forecast results in Table IV shows that this method is rarely more performant than the numerical models. The feature SST is worse than the numerical model by at least 100%, which implies the HYCOM model is well-calibrated to local conditions. When comparing the lower resolution ERA5 model, air pressure and gust strength are less aligned with the recorded observations. As a result, the feature gust speed is up to 37% less accurate when using the PINN models and results are more accurate using all architectures for air pressure. This is encouraging and suggests that our surrogate modeling technique can produce permissible forecasts depending on the feature. The comparison of the best surrogate model to the non-regularized surrogate when $\lambda = 1.0$ is more favorable. From the Table, we show that there is a percent decrease in error for most cases. The GRU and LSTM models are more accurate when compared to the non-regularized versions. The air pressure results show that the surrogate outperforms the numerical model only after finding the best λ value. That is, we only outperform the numerical model due to the coupled loss function. The Transformer models showed improved forecasts for air pressure alone. This indicates that a large network with many trainable parameters can still benefit from our technique, but the reduction in error will be less, if there is any at all.

Continuing, we consider the percent change in RMSE when experimenting with the differenced data representation in Table V. Overall, when comparing the PINN models to the numeric

TABLE IV
ORIGINAL VALUE FORECAST % CHANGE IN RMSE WHEN COMPARING THE BEST FOUND λ AGAINST $\lambda = 1.0$ (NO REGULARIZATION) AND THE NUMERICAL MODEL (HYCOM/ERA5)

Model	Best λ	Feature	$\lambda = 1.00$	Numerical Model
GRU	0.90	SST (°C)	-18.44%	+141.47%
	0.96	Pressure (hPa)	-14.83%	-3.08%
	0.84	Gust (m/s)	-3.98%	+33.11%
LSTM	0.68	SST (°C)	-15.42%	+145.45%
	0.82	Pressure (hPa)	-4.48%	-0.78%
	0.72	Gust (m/s)	-7.62%	+37.25%
Transformer	1.00	SST (°C)	0.0%	+102.02%
	0.90	Pressure (hPa)	-3.06%	-7.58%
	1.00	Gust (m/s)	0.0%	+26.44%

model, we see improvement when using this data representation. The only comparison which is still worse than the numerical models is when forecasting the gust speed feature, although the percentage of error is slightly decreased. Almost all the features show decrease in error when comparing the best λ to the model trained when $\lambda = 1.0$. The spread of the decrease in error is lesser than when forecasting the original data, with the highest at about 8% and the lowest at 1.6%. There is no situation for this data where the best λ directly causes improvement over the numerical model, but we find an increased performance gap between the deep learning and numerical models in most cases.

We also consider the stability of the forecasts, given a single example buoy. In Figure 7 and Figure 8 we show how the error of our PINNs evolves over the forecast period of 24 hours given chaotic features, model architectures, and data representations. These figures capture a subset of 10 forecast periods, from time step 40 to time step 120, for a single buoy. The ground truth values are reinitialized into the model every eighth time step, hence the ten forecast periods. To select the λ value to represent in the figures, we use the best λ value found for SST. When SST does not have a best $\lambda < 1.0$ then the best value for gust strength or air pressure was chosen. This highlights the limiting factor of our technique in its current form, as it cannot utilize multiple values for λ. Future explorations into this technique might consider a multiple λ setup for more flexibility.

TABLE V

DIFFERENCED VALUE FORECAST % CHANGE IN RMSE WHEN COMPARING THE BEST FOUND λ AGAINST $\lambda = 1.0$ (NO REGULARIZATION) AND THE NUMERICAL MODEL (HYCOM/ERA5)

Model	Best λ	Feature	$\lambda = 1.00$	Numerical Model
GRU	0.42	SST (°C)	-1.64%	-23.43%
	0.40	Pressure (hPa)	-6.30%	-20.10%
	0.58	Gust (m/s)	-2.89%	+29.73%
LSTM	0.80	SST (°C)	-7.45%	-23.19%
	0.30	Pressure (hPa)	-7.18%	-19.66%
	0.76	Gust (m/s)	-2.18%	+31.46%
Transformer	1.00	SST (°C)	0.0%	-25.63%
	0.00	Pressure (hPa)	-7.93%	-20.72%
	0.00	Gust (m/s)	-4.80%	+24.45%

When examining the original data forecast results for buoy 42002 in Figure 7, it is expected for error to increase over the period. Ideally, the error of the best found λ will increase more slowly than when $\lambda = 0.0$ or $\lambda = 1.0$, for each feature. From this figure, we can observe that error increases until the model is realigned with fresh initial values. We see that the forecasts are often worse than the numerical model. They are typically most performant around time steps one or two, when the initial values are still relatively recent. Comparing models and features shows a wide variety of behaviors. The most similar forecasts are found when considering the Transformer, when each of the PINN models performs almost identically. The GRU models tend to disagree the most between each of the specific experiments, which makes sense considering it achieves the highest reduction in forecast error overall. PINNs are traditionally used to reduce numerical instability, and this behavior can be seen when forecasting air pressure using the GRU model. Between time steps 56 and 64, the best-selected λ shows significantly reduced error when comparing to the $\lambda = 1.0$ case. The same temporal region in the Transformer forecast displays the opposite behavior where the non-regularized model performs better than any regularized version. This is due to the complexity of the Transformer-based architecture which causes the model to generalize underlying behaviors more effectively than the GRU or LSTM architectures.

Finally, we compare the differenced value forecast MAE scores for buoy 42002 from the Figure 8. In the case of the Transformer model, we show $\lambda = 0.5$ because each feature's best λ

lies on the extreme end of either $\lambda = 0.0$ or $\lambda = 1.0$. The main benefit of using the differenced data representation is displayed by the reduction in overall error across all models. The Figure demonstrates how the λ forces the PINN to behave more like one data source or the other, evidenced by the fact that the MAE found tends to be bound by the other error sources. Overall, error increases more slowly in regions where the forecasted feature remains highly stable over time. Once again, we see that refreshing the initial values reduces error significantly, which is the expected behavior. The error spread between the PINN is much more similar in this case because the models rely more on autocorrelation between forecast periods. Error reduction is significant enough to suggest the regularized models make more informed forecasts on average. It is significant to note that individual plots of forecasts from the best λ model may be less accurate than other setups in specific instances, but error is reduced overall when considering all buoys.

In this section, we analyzed the forecasting ability of our models by considering percent reduction in errors and the forecast of a single buoy via different experimental permutations. The selection of λ and total amount of error reduction was shown to depend on the model, the features examined, and the data representation used. When compared to models where $\lambda = 1.0$, percentage reductions in error were as low as 1.6% and as high as 18.4%. When using the Transformer model, the feature SST never showed improvement over the $\lambda = 1.0$ case. The surrogate models always outperform the numerical model for the air pressure feature and outperform in SST forecasting depending on the data representation. We never outperform the numerical model when forecasting gust strength. In the case of feature air pressure, the error reduction from selecting λ through a grid search allows the surrogate PINN model to out-perform the numerical model. It is important to restate that the interpolated values in the ground truth provide some bias in the test by penalizing the numerical models when comparing to those interpolated values. In addition, inference based on differenced inputs produces more stable estimates of local conditions, i.e., the observations. Our surrogate models benefit from both points which explains the general improvement when compared to the numerical model. More importantly, selecting the best regularization parameter, λ, yields models that achieve higher accuracy, and this is consistent across both data representations. We showed how the error in forecasts are reduced on average by training the surrogate model using the selected λ value. This revealed the way model selection and data representation affects the numerical stability over the forecast period. The differenced data representation simplifies the problem for the surrogate

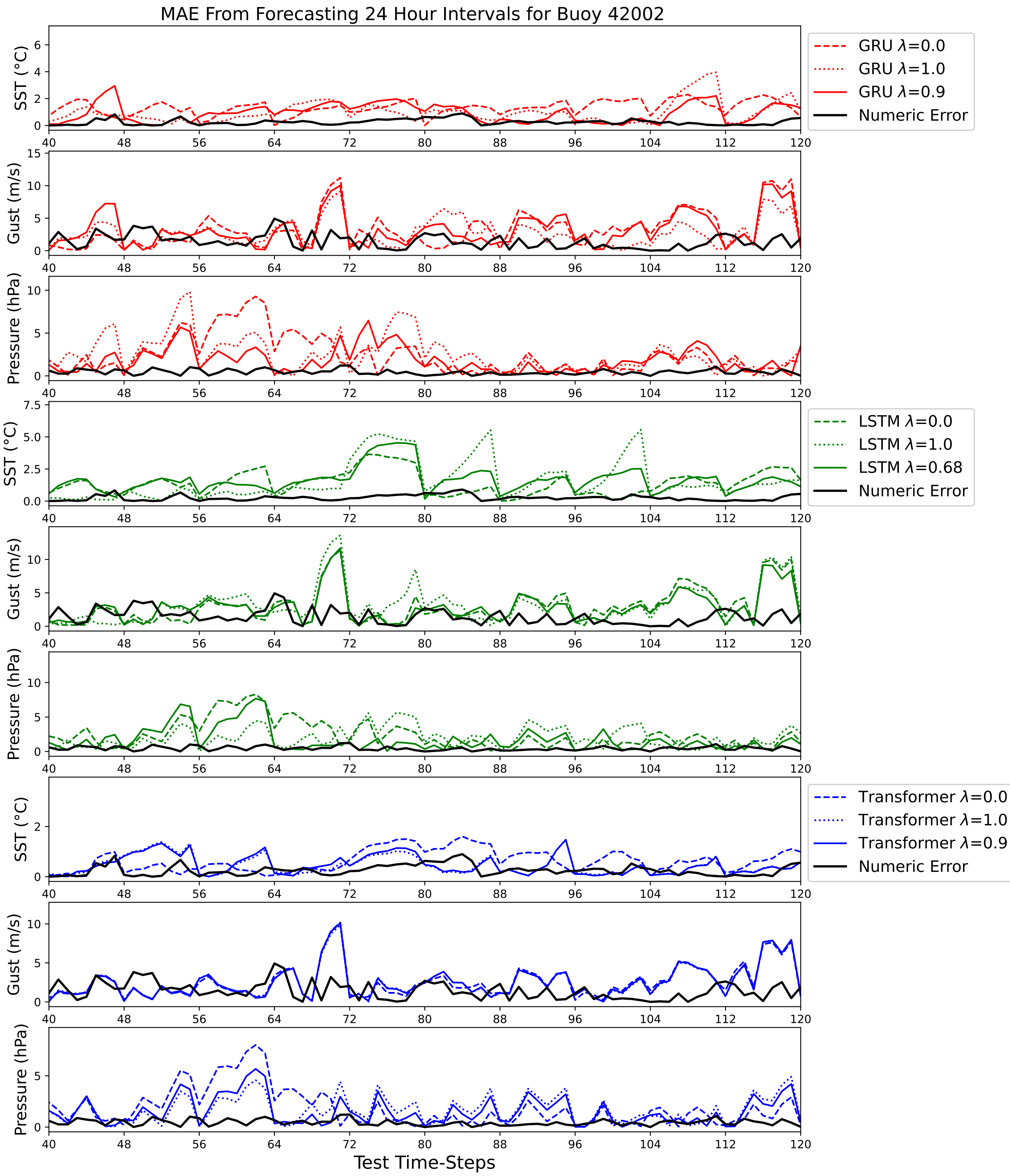


Fig. 7. The numerical and surrogate model MAE for each feature over ten 24-hour forecast periods is displayed. We include each PINN with $\lambda = 0.0$, $\lambda = 1.0$ (no regularization), and the best found λ. The PINNs are reinitialized with new starting values every eighth period.

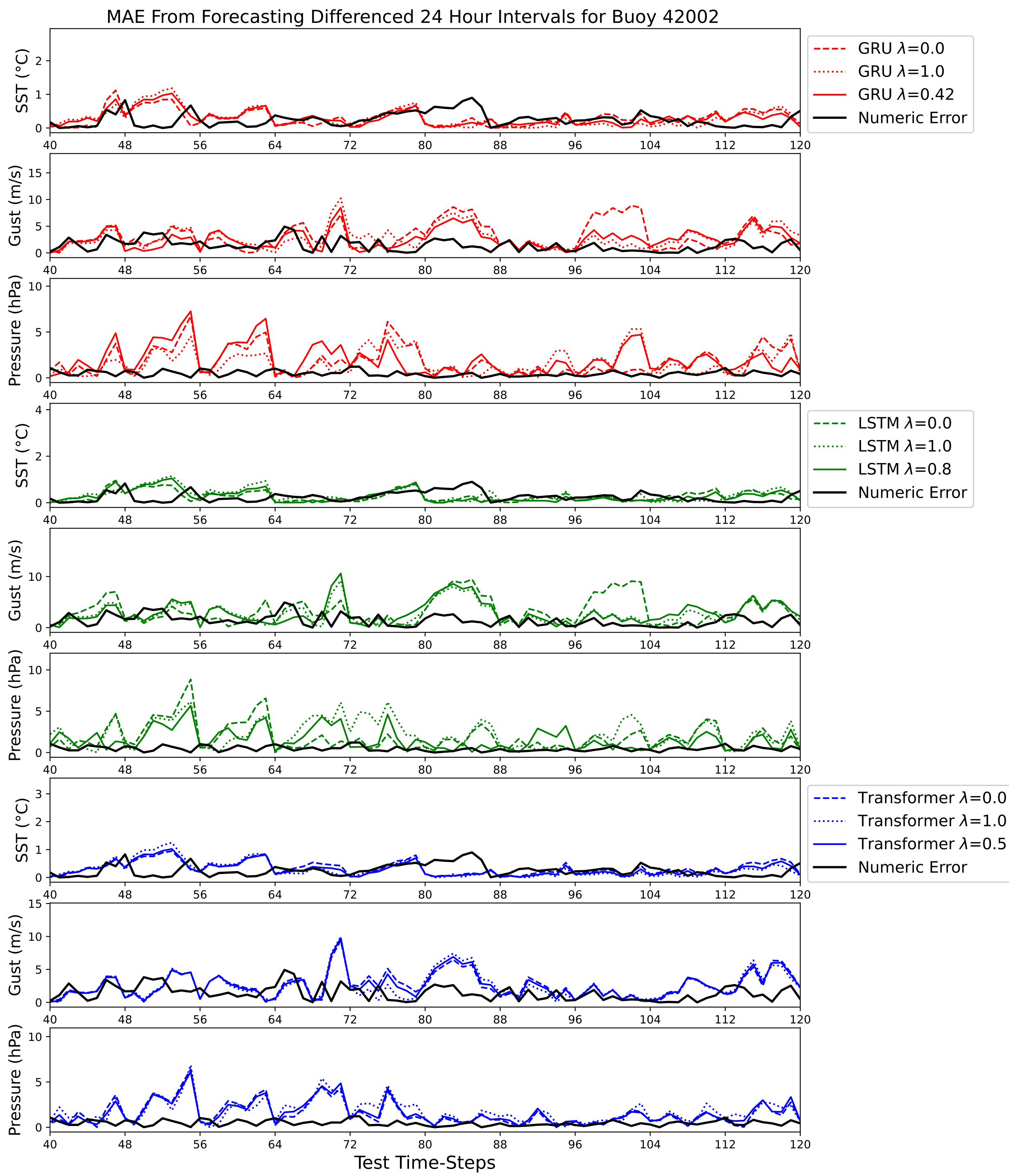


Fig. 8. The numerical and surrogate model MAE for each feature over ten 24-hour forecast periods is displayed. Differenced value forecasts have been transformed back to the original scale before finding the error. We include each PINN with $\lambda = 0.0$, $\lambda = 1.0$ (no regularization), and the best found λ. The PINNs are reinitialized with new starting values every eighth period.

models, so the forecast stability remains similar between models and features. The opposite is true in the original data forecasts, which is more chaotic and showed disagreements. In all, the analysis of these results suggest that our model is relatively stable over 24 hour periods, but error is often worse than the reanalysis models when they are well-fitted to the observation data.

C. Geographical Error Analysis

Our final method for comparing the numerical models with our PINNs involves an analysis of buoy RMSE per their geographical position. To this end, we provide two figures which represent a grid of our models as rows with the forecasted feature as columns. Positional markers reference the latitude and longitude of each buoy, and there is overlap due to the number of buoys. The color bar represents the amount of RMSE calculated for a buoy and is normalized column-wise by the minimum and maximum error generated for the feature by each model. In Figure 9 we show the results from the original data forecast and in Figure 10 we show the results from the differenced dataset. One caveat to these figures is that we cap the error of the air pressure feature in both figures to a max value of 10. This is because the ERA5 has an extreme misalignment in outlier areas, which dominates the color interpolation. We cap the error derived from SST to a max value of one in the differenced Figure 10 for the same reason.

The original values forecast results in Figure 9 show there are some trends among the models. First, the best performing region for all features are the forecasts of buoys clustered around the Caribbean. The Gulf of Mexico region performs similarly but can be slightly less accurate depending on the experiment. The least performant regions tend to be along the North Atlantic East-Coast and various regions around the Pacific West-Coast. The numerical models are, on average, are extremely well fitted to real-world observations. Although, there are cases, possibly due to resolution constraints of grid data, where massive influxes of error are found. This misalignment shows the benefit of local condition forecasting. For example, the numerically modeled outliers for air pressure are along the West-Coast. These same regions perform well using our technique because we model the forecast based on local observed conditions. Geographic regions which are poorly forecasted by a PINN model tend to cluster among similarly performing regions. We do not observe alternating high and low error regions, which would imply random forecasts. Instead, we very consistently see gradients of low to high error regions. This may be explained by considering that some regions may pose a modeling challenge due to geography, river runoff, human operations, lack of data, and so on.

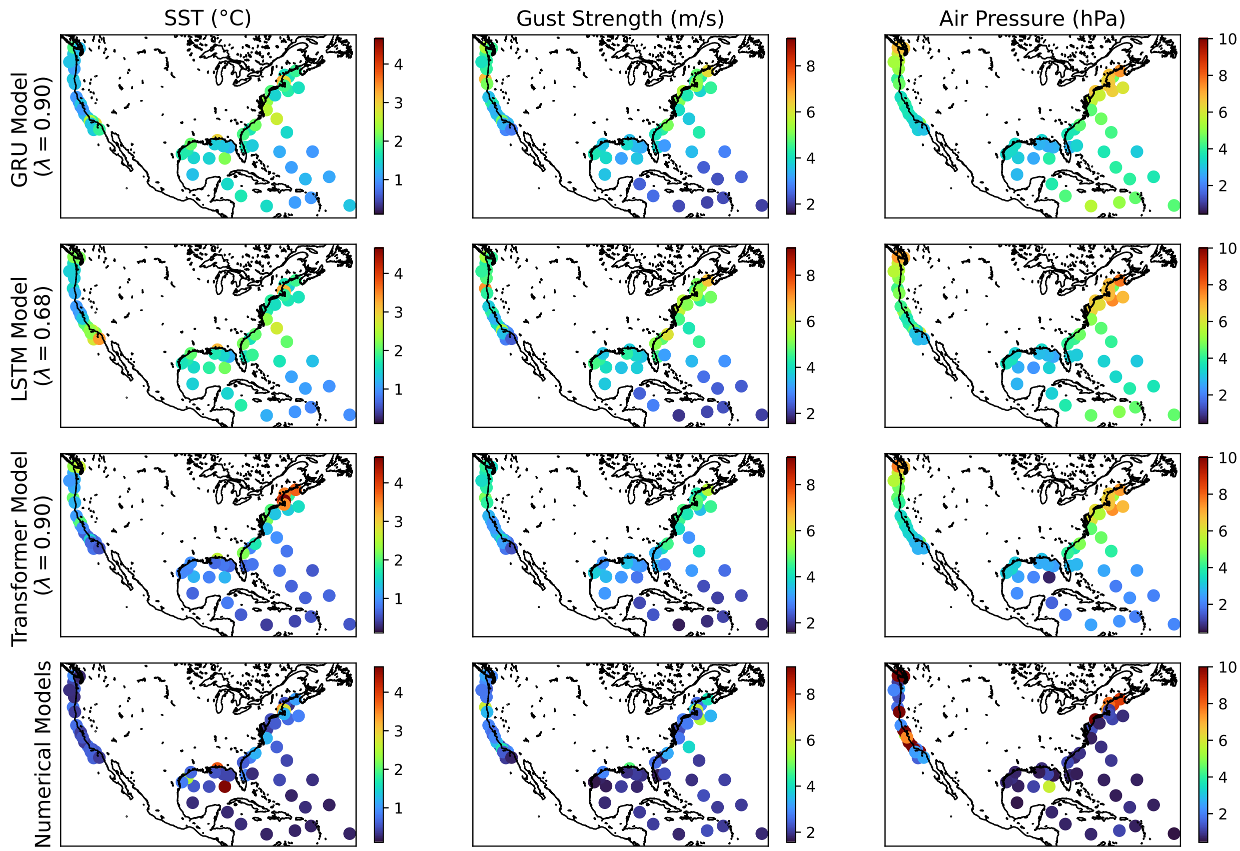


Fig. 9. Analyzed original features (columns) compared to the generating model (rows) by RMSE given at the geographical buoy location. Error is capped for SST and air pressure for visualization purposes. Color maps are normalized by each feature for comparative evaluation.

Next, we analyze the difference valued forecast results in Figure 10. The results are more homogeneous and more accurate across all models and features. Compared to the original forecast, similar geographical zones display relatively high errors, showing these are likely regions of high change. Each of the PINN models yields similar error scores which suggests that they rely on low-change forecasts to accurately describe the true value. Therefore, the models produce more similar results and are more sensitive to chaotic regions. From the Figure, we can pick out an instance of an outlier buoy in the center of the Caribbean region, when forecasting the SST parameter. There, error from HYCOM is high while the error from each PINN model is low. In this case, the numerical model represents real world conditions and error is calculated through interpolated initial values, causing inflated metrics. However, this is not the reason for all outliers. In the case of air pressure, most high-error regions are a case of misalignment in the numerical model.

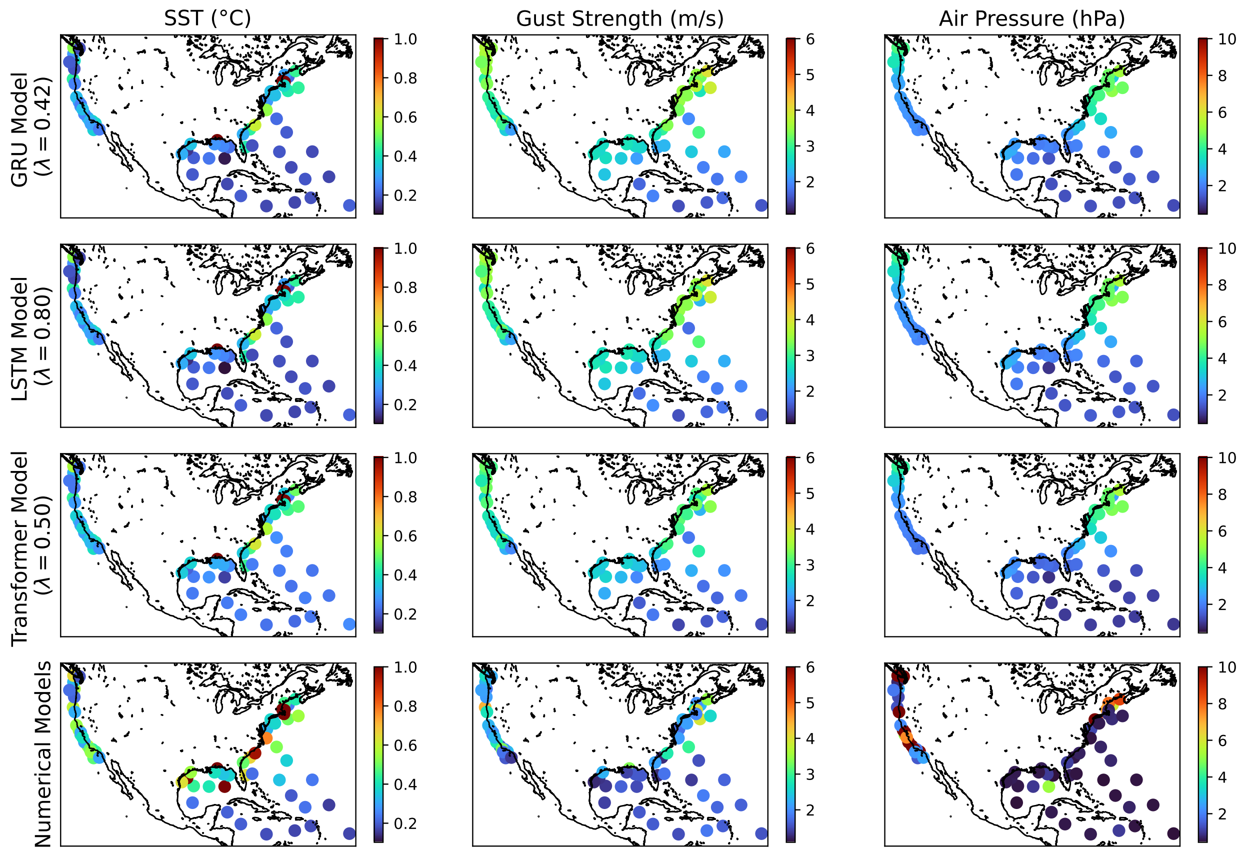


Fig. 10. Analyzed differenced features (columns) compared to the generating model (rows) by RMSE given at the geographical buoy location. Error is capped for SST and air pressure for visualization purposes. Color maps are normalized by each feature for comparative evaluation.

By examining the individual buoy error, we learned which geographic regions are most difficult to model. We also revealed patterns in the similarities between our PINN experiments and the numerical models. The figures revealed that the numerical models have some regions with high error. The error is mainly found when there is misalignment in the numerical models. Some error was introduced through our interpolation scheme, such as the SST outlier in the Gulf of Mexico. Buoys which received low accuracy forecasts tend to be surrounded by buoys with similar metrics, which implies they are within difficult-to-model geographical regions. Although the error for the differenced data representation is lower than when forecasting the original values, the buoys with the highest error come from similar regions. When comparing our sparse forecasting technique to a full-coverage model, our method is not constrained to a grid region, and any arbitrary point may be modeled. Therefore, error may be reduced when forecasting regions between vertices, without relying on interpolation techniques. The drawback of using this sparse forecasting technique is that greater spatial conditions cannot be deciphered by the observations alone. In this way, we trade off providing regional context to the PINN model for increased forecasting flexibility. The PINN architecture bases the forecast off current conditions alone and is independent of the buoy's geography.

V. Conclusions

We investigated the ability of the ocean flow model HYCOM and the climate model ERA5 to be used as regularization data for PINN-inspired deep learning models. A special formulation of the loss function yielded comprehensive models for forecasting any number of physical parameters in a sequence-to-sequence model. The techniques demonstrated how multiple ocean and climate features may be forecasted and combined using deep LSTM, GRU, and Transformer physics-informed networks. Our sparse feature forecasting approach yielded more flexible, generalized models, which are less constrained to predefined regions. In contrast to other PINN models, we train the models using observation data while regularizing with pre-computed numerical models. The significance of this is that we do not need to implement the numerical formulation for use in our framework. In most cases, we improved the surrogate model performance by combining the observation data and numerical models. To assess the models, we set up experimental sparse sequential forecasting procedures for SST, air pressure, and gust strength as observed by free floating buoys. Two separate data representations were investigated which included the original observed/modeled data and first order differenced versions of the data. Over these experiments, the hyperparameter λ was fine-tuned between 0.0 and 1.0 to find the best possible data ratio. We found that models which have a less complex architecture improved the most from the inclusion of the numerical model regularization. This was shown explicitly by comparing the results of the least complex and most complex architectures of the GRU and Transformer models. The GRU and LSTM models showed improvements after tuning for λ in every case while the Transformer models showed improvement for fewer features. Further, the selection of λ significantly altered the behavior of the PINN models. As the λ value approaches 0.0, the trained model produced results more like the numerical models, while the opposite is true when λ approaches 1.0. Depending on the experiment, we saw improvements over the numerical model in forecast error. In favor of our method, the PINN forecasting of air pressure showed improvement over the numerical models when the best selection of λ was chosen. Overall, our method improved the numerical stability of the forecasts on average over the horizon period. In the case of the differenced data representation, we saw the stability of each PINN model was similar. Lower valued λ values were most performant in this case, which suggests the numerical model data was more informative overall. This is likely due to fewer interpolated values from the numerical models when compared to the buoy observations. The differenced data forecasts are the most accurate overall, but the amount of error reduction found when using this data representation was less. Exploring the error geographically showed us that modeling high-change areas of interest is difficult for both the numerical models and our PINNs. This methodology can be used to forecast observations between the vertices of grid-based numerical models. The trade-off of the increased flexibility is the loss of context of spatial conditions beyond the immediate forecast region. Ongoing work on this methodology continues in several ways. Because the selection of λ changes on a feature-by-feature basis, we should investigate an approach to allow an independent selection of λ values on a per-feature case. Using a grid search for selecting the best λ value is currently inefficient. Future improvements to our technique will revolve around fine-tuning the λ selection approach to reduce computational overhead of the models. Moreover, since we formulate new models that combine numerical models with observations, our framework leaves room to explore integration into a data assimilation scheme. The methodology should be expanded to combine multiple numerical models with relevant PDEs to see if similar improvements can be found when forecasting full-coverage models also. Different domain problems and experimental setups will yield further insight into this procedure for combining multiple sources of data when each has inherent limitations.

Acknowledgment

This work was partly supported by the U.S. Department of the Navy, Office of Naval Research (ONR), and Naval Research Laboratory under contracts N00173-20-2-C007 and N00173-20-2-C007, respectively. The work of Austin Schmidt was funded by a SMART (Science, Mathematics and Research for Transformation) Department of Defense (DoD) scholarship for service.

References

[1] R. Bleck, “An oceanic general circulation model framed in hybrid isopycnic-cartesian coordinates,” *Ocean modelling*, vol. 4, no. 1, pp. 55–88, 2002.

[2] H. Hersbach, B. Bell, P. Berrisford, *et al.*, “The era5 global reanalysis,” *Quarterly Journal of the Royal Meteorological Society*, vol. 146, no. 730, pp. 1999–2049, 2020.

[3] G. R. Halliwell, "Evaluation of vertical coordinate and vertical mixing algorithms in the hybrid-coordinate ocean model (hycom)," *Ocean Modelling*, vol. 7, no. 3-4, pp. 285–322, 2004.

[4] S. Cuomo, V. S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli, "Scientific machine learning through physics–informed neural networks: Where we are and what's next," *Journal of Scientific Computing*, vol. 92, no. 3, p. 88, 2022.

[5] M. Haghbin, A. Sharafati, D. Motta, N. Al-Ansari, and M. H. M. Noghani, "Applications of soft computing models for predicting sea surface temperature: A comprehensive review and assessment," *Progress in earth and planetary science*, vol. 8, no. 1, pp. 1–19, 2021.

[6] S. Zhang, Z. Liu, X. Zhang, *et al.*, "Coupled data assimilation and parameter estimation in coupled ocean–atmosphere models: A review," *Climate Dynamics*, vol. 54, no. 11, pp. 5127–5144, 2020.

[7] N. Barton, E. J. Metzger, C. A. Reynolds, *et al.*, "The navy's earth system prediction capability: A new global coupled atmosphere-ocean-sea ice prediction system designed for daily to subseasonal forecasting," *Earth and Space science*, vol. 8, no. 4, e2020EA001199, 2021.

[8] S. Razavi, B. A. Tolson, and D. H. Burn, "Review of surrogate modeling in water resources," *Water Resources Research*, vol. 48, no. 7, 2012.

[9] X. Yu, S. Shi, L. Xu, Y. Liu, Q. Miao, and M. Sun, "A novel method for sea surface temperature prediction based on deep learning," *Mathematical Problems in Engineering*, vol. 2020, 2020.

[10] C. Xiao, N. Chen, C. Hu, *et al.*, "A spatiotemporal deep learning model for sea surface temperature field prediction using time-series satellite data," *Environmental Modelling & Software*, vol. 120, p. 104 502, 2019.

[11] A. Ducournau and R. Fablet, "Deep learning for ocean remote sensing: An application of convolutional neural networks for super-resolution on satellite-derived sst data," in *2016 9th IAPR Workshop on Pattern Recogniton in Remote Sensing (PRRS)*, IEEE, 2016, pp. 1–6.

[12] M. Tang, Y. Liu, and L. J. Durlofsky, "A deep-learning-based surrogate model for data assimilation in dynamic subsurface flow problems," *Journal of Computational Physics*, vol. 413, p. 109 456, 2020.

[13] G.-Q. Jiang, J. Xu, and J. Wei, "A deep learning algorithm of neural network for the parameterization of typhoon-ocean feedback in typhoon forecast models," *Geophysical Research Letters*, vol. 45, no. 8, pp. 3706–3716, 2018.

[14] Y. Zhu, G. Cao, Y. Wang, *et al.*, "Variability of the deep south china sea circulation derived from hycom reanalysis data," *Acta Oceanologica Sinica*, vol. 41, no. 7, pp. 54–64, 2022.

[15] B. Kesavakumar, P. Shanmugam, and R. Venkatesan, "Enhanced sea surface salinity estimates using machine-learning algorithm with smap and high-resolution buoy data," *IEEE Access*, vol. 10, pp. 74 304–74 317, 2022.

[16] D.-H. Kim and H. M. Kim, "Deep learning for downward longwave radiative flux forecasts in the arctic," *Expert Systems with Applications*, vol. 210, p. 118 547, 2022.

[17] R. Zhang, Q. Liu, R. Hang, and G. Liu, "Predicting tropical cyclogenesis using a deep learning method from gridded satellite and era5 reanalysis data in the western north pacific basin," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–10, 2021.

[18] S.-W. Kim, J. A. Melby, N. C. Nadal-Caraballo, and J. Ratcliff, "A time-dependent surrogate model for storm surge prediction based on an artificial neural network using high-fidelity synthetic hurricane modeling," *Natural Hazards*, vol. 76, no. 1, pp. 565–585, 2015.

[19] L. Huang, Y. Jing, H. Chen, L. Zhang, and Y. Liu, "A regional wind wave prediction surrogate model based on cnn deep learning network," *Applied Ocean Research*, vol. 126, p. 103 287, 2022.

[20] G. Qiu, Y. Liu, J. Zhao, *et al.*, "Analytic deep learning-based surrogate model for operational planning with dynamic ttc constraints," *IEEE Transactions on Power Systems*, vol. 36, no. 4, pp. 3507–3519, 2021. DOI: 10.1109/TPWRS.2020.3041866.

[21] S. C. James, Y. Zhang, and F. O'Donncha, "A machine learning framework to forecast wave conditions," *Coastal Engineering*, vol. 137, pp. 1–10, 2018.

[22] P. Pokhrel, M. Abdelguerfi, and E. Ioup, "A machine learning and data assimilation forecasting framework for surface waves," *Quarterly Journal of the Royal Meteorological Society*, 2023.

[23] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational physics*, vol. 378, pp. 686–707, 2019.

[24] T. de Wolff, H. Carrillo, L. Marti, and N. Sanchez-Pi, "Towards optimally weighted physics-informed neural networks in ocean modelling," *arXiv preprint arXiv:2106.08747*, 2021.

[25] C. Dong, G. Xu, G. Han, B. J. Bethel, W. Xie, and S. Zhou, "Recent developments in artificial intelligence in oceanography," *Ocean-Land-Atmosphere Research*, vol. 2022, 2022.

[26] M. A. Nabian and H. Meidani, "Physics-informed regularization of deep neural networks," *arXiv preprint arXiv:1810.05547*, 2018.

[27] J. Tu, C. Liu, and P. Qi, "Physics-informed neural network integrating pointnet-based adaptive refinement for investigating crack propagation in industrial applications," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 2, pp. 2210–2218, 2022.

[28] H. Sun, L. Peng, J. Lin, S. Wang, W. Zhao, and S. Huang, "Microcrack defect quantification using a focusing high-order sh guided wave emat: The physics-informed deep neural network guwnet," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 5, pp. 3235–3247, 2021.

[29] H. Sun, L. Peng, S. Huang, *et al.*, "Development of a physics-informed doubly fed cross-residual deep neural network for high-precision magnetic flux leakage defect size estimation," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 3, pp. 1629–1640, 2021.

[30] J. Zhao, W. Li, X. Yuan, *et al.*, "An end-to-end physics-informed neural network for defect identification and 3-d reconstruction using rotating alternating current field measurement," *IEEE Transactions on Industrial Informatics*, pp. 1–10, 2022. DOI: 10.1109/TII.2022.3217820.

[31] G. Huang, Z. Zhou, F. Wu, and W. Hua, "Physics-informed time-aware neural networks for industrial nonintrusive load monitoring," *IEEE Transactions on Industrial Informatics*, 2022.

[32] J. Rice, W. Xu, and A. August, "Analyzing koopman approaches to physics-informed machine learning for long-term sea-surface temperature forecasting," *arXiv preprint arXiv:2010.00399*, 2020.

[33] S. Yuan, X. Feng, B. Mu, B. Qin, X. Wang, and Y. Chen, "A generative adversarial network–based unified model integrating bias correction and downscaling for global sst," *Atmospheric and Oceanic Science Letters*, p. 100 407, 2023.

[34] T. Yuan, J. Zhu, W. Wang, *et al.*, "A space-time partial differential equation based physics-guided neural network for sea surface temperature prediction," *Remote Sensing*, vol. 15, no. 14, p. 3498, 2023.

[35] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," *Nature Reviews Physics*, vol. 3, no. 6, pp. 422–440, 2021.

[36] P. Pokhrel, E. Ioup, J. Simeonov, M. T. Hoque, and M. Abdelguerfi, "A transformer-based regression scheme for forecasting significant wave heights in oceans," *IEEE Journal of Oceanic Engineering*, vol. 47, no. 4, pp. 1010–1023, 2022.

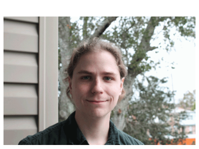
[37] "Ifs documentation cy41r2 - parts 1-5," in *IFS Documentation CY41R2* (IFS Documentation 1), IFS Documentation 1. ECMWF, 2016.

[38] H. H., *Era5 hourly data on single levels from 1959 to present. copernicus climate change service (c3s) climate data store (cds).* (Accessed on 14-APR-2021), 10.24381/cds.adbb2d47, 2018.

[39] E. C. Eze and C. R. Chatwin, "Enhanced recurrent neural network for short-term wind farm power output prediction," *J. Appl. Sci*, vol. 5, no. 2, pp. 28–35, 2019.

[40] Y. Yu, H. Yao, and Y. Liu, "Structural dynamics simulation using a novel physics-guided machine learning method," *Engineering Applications of Artificial Intelligence*, vol. 96, p. 103 947, 2020.

[41] R. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice*, 3rd. Melbourne, Australia: OTexts, 2021, https://otexts.com/fpp3.



Austin B. Schmidt completed his M.S. degree in computer science in 2021 and is a current awardee of the prestigious SMART scholarship, awarded by the DoD. Mr. Schmidt is currently pursuing a Ph.D. in engineering and applied sciences at the University of New Orleans. Alongside working on degree requirements, he conducts machine learning research at the Canizaro Livingston Gulf States Center for Environmental Informatics (GulfSCEI).

Pujan Pokhrel received his Ph.D. from the University of New Orleans in engineering and applied sciences and conducts research with the Canizaro Livingston Center for Gulf Informatics. He is currently a Software Engineer at Amazon Web Services. His research interests lie in machine learning, artificial intelligence, computational fluid dynamics, numerical models, and optimization.

Mahdi Abdelguerfi is currently a professor of computer science at the University of New Orleans (UNO) and the founder and executive director of the interdisciplinary Canizaro-Livingston Gulf States Center for Environmental Informatics (GulfSCEI - pronounced Gulf Sea), whose primary goal is to assist state, federal agencies and non-government organizations solve environmental problems of the coastal margin of the State of Louisiana as well as other Gulf States. Dr. Abdelguerfi is the recipient of the 2022 UNO's Career Research Award. He was awarded the 2003 and 2016 Alan Berman Annual Research Publication Award from the Naval Research Laboratory (NRL) - Department of the Navy for research performed jointly with scientists from NRL. He received the 2016 UNO's Competitive Funding Prize in recognition of outstanding and innovative work in support of UNO's research mission. He was also the recipient of UNO's Early Career Achievement Award for Excellence in Research.

Elias Ioup received a Ph.D. degree in engineering and applied science from The University of New Orleans. He is currently a Computer Scientist and the Head of the Geospatial Computing Section, U.S. Naval Research Laboratory. His research interests include high-performance geospatial data processing, geospatial and environmental web services, and geospatial data visualization.

David Dobson received a Ph.D. degree in computational science from The University of Southern Mississippi. He currently leads a cloud computing and security research team at the U.S. Naval Research Laboratory. His research interests include machine learning on big data and cloud computing security.

APPENDIX

124 selected buoy observations from the NOAA archive for potential inclusion into train, validation, and test datasets. The numbers selected into each set are displayed in VI.

- 51001, 41002, 41004, 41008, 41009, 41010, 41013, 41025, 41040, 41041, 41043, 41044, 41046, 41047, 41048, 41049, 42001, 42002, 42003, 42012, 42019, 42020, 42035, 42036, 42039, 42040, 42055, 42056, 42057, 42058, 42059, 42060, 44005, 44007, 44008, 44009, 44011, 44013, 44014, 44017, 44018, 44020, 44025, 44027, 44065, 44066, 45001, 45002, 45003, 45004, 45005, 55039, 45006, 45007, 45008, 45012, 46001, 46002, 46005, 46006, 46011, 46012, 46013, 46014, 46015, 46022, 46025, 46026, 46027, 46028, 46029, 46035, 46041, 46042, 46047, 46050, 46053, 46054, 46059, 46060, 46061, 46066, 46069, 46070, 46071, 46072, 46073, 46075, 46076, 46077, 46078, 46080, 46081, 46082, 46083, 46084, 46085, 46086, 46087, 46088, 46089, 51000, 51001, 51002, 51003, 51004, 51101, 46221, 46214, 46211, 46224, 46215, 46222, 46213, 46235. 46239, 46240, 46243, 46244, 46232, 44095, 44100, 42099, and 44024.

TABLE VI

NUMBER OF BUOYS DISTRIBUTED INTO EACH DATASET. THERE ARE 127 BUOYS SORTED IN TOTAL.

Subset Contributions by Buoy	Total Number
Total Buoys	124
Train Only	3
Val Only	0
Test Only	1
Train and Test Only	2
Val and Test Only	1
Train/Test/Val Included	86
Not Included At All	31

TABLE VII

GRU ORIGINAL FORECASTS PER $\lambda \in [0, 1]$
RMSE RESULTS OVER 8 FORECAST PERIODS (24 HOURS)

Feature	λ	1.000	2.000	3.000	4.000	5.000	6.000	7.000	8.000
SST (°C)	0.00	1.117	1.327	1.507	1.663	1.800	1.923	2.035	2.138
	0.10	1.029	1.216	1.370	1.503	1.619	1.724	1.818	1.907
	0.20	1.006	1.178	1.324	1.452	1.567	1.670	1.764	1.850
	0.30	0.986	1.195	1.372	1.529	1.670	1.798	1.918	2.029
	0.40	0.978	1.198	1.387	1.552	1.697	1.826	1.941	2.045
	0.50	0.855	1.038	1.194	1.329	1.449	1.558	1.660	1.757
	0.60	0.828	1.030	1.197	1.342	1.471	1.587	1.691	1.785
	0.70	0.882	1.143	1.370	1.574	1.761	1.932	2.091	2.238
	0.80	0.851	1.067	1.239	1.384	1.508	1.618	1.714	1.801
	0.90	0.781	0.977	1.134	1.262	1.369	1.460	1.539	1.607
	1.00	0.887	1.133	1.332	1.497	1.640	1.763	1.872	1.970
Pressure (hPa)	0.00	6.223	6.663	7.011	7.306	7.569	7.805	8.016	8.202
	0.10	6.240	6.702	7.054	7.344	7.593	7.812	8.004	8.175
	0.20	6.393	7.038	7.536	7.947	8.297	8.599	8.858	9.081
	0.30	6.072	6.643	7.077	7.432	7.736	7.999	8.224	8.419
	0.40	5.746	6.424	6.972	7.437	7.837	8.179	8.467	8.713
	0.50	4.446	5.194	5.753	6.202	6.579	6.898	7.169	7.402
	0.60	2.896	3.632	4.252	4.798	5.285	5.711	6.079	6.401
	0.70	2.343	2.968	3.507	4.013	4.508	4.971	5.383	5.754
	0.80	2.302	2.882	3.378	3.831	4.273	4.692	5.073	5.420
	0.96	2.072	2.657	3.148	3.598	4.037	4.447	4.817	5.154
	1.00	2.119	2.832	3.452	4.034	4.600	5.136	5.617	6.051
Gust (m/s)	0.00	3.044	3.399	3.709	3.975	4.205	4.405	4.580	4.738
	0.10	2.917	3.256	3.554	3.811	4.029	4.212	4.366	4.501
	0.20	2.957	3.312	3.616	3.873	4.090	4.271	4.425	4.560
	0.30	2.809	3.124	3.388	3.606	3.787	3.938	4.065	4.176
	0.40	2.789	3.138	3.438	3.691	3.903	4.077	4.223	4.348
	0.50	2.683	3.076	3.404	3.678	3.906	4.094	4.251	4.387
	0.60	2.538	2.963	3.285	3.541	3.747	3.916	4.059	4.182
	0.70	2.412	2.806	3.107	3.347	3.541	3.700	3.833	3.947
	0.84	2.396	2.782	3.077	3.309	3.497	3.650	3.781	3.894
	0.90	2.415	2.841	3.167	3.429	3.640	3.813	3.958	4.081
	1.00	2.378	2.778	3.102	3.368	3.587	3.768	3.923	4.055

TABLE VIII

LSTM ORIGINAL FORECASTS PER $\lambda \in [0, 1]$
RMSE RESULTS OVER 8 FORECAST PERIODS (24 HOURS)

Feature	λ	1.000	2.000	3.000	4.000	5.000	6.000	7.000	8.000
SST (°C)	0.00	1.089	1.282	1.459	1.616	1.754	1.875	1.983	2.080
	0.10	1.031	1.237	1.418	1.583	1.733	1.867	1.989	2.102
	0.20	1.052	1.241	1.408	1.551	1.672	1.776	1.865	1.943
	0.30	1.120	1.344	1.533	1.694	1.833	1.954	2.060	2.153
	0.40	0.900	1.101	1.272	1.418	1.545	1.658	1.760	1.856
	0.50	0.813	1.005	1.165	1.298	1.415	1.519	1.613	1.700
	0.60	0.788	1.015	1.201	1.360	1.497	1.617	1.723	1.817
	0.68	0.756	0.962	1.127	1.263	1.377	1.475	1.560	1.633
	0.80	0.773	1.001	1.186	1.342	1.478	1.597	1.705	1.805
	0.90	0.798	1.033	1.226	1.388	1.527	1.648	1.753	1.848
	1.00	0.850	1.097	1.296	1.462	1.603	1.727	1.835	1.931
Pressure (hPa)	0.00	6.706	7.270	7.691	8.020	8.288	8.510	8.698	8.858
	0.10	6.371	6.854	7.220	7.517	7.770	7.987	8.176	8.343
	0.20	6.493	7.150	7.666	8.079	8.418	8.700	8.938	9.140
	0.30	6.334	7.070	7.646	8.117	8.517	8.862	9.162	9.424
	0.40	5.788	6.556	7.155	7.653	8.083	8.460	8.791	9.084
	0.50	4.557	5.382	6.017	6.532	6.960	7.316	7.613	7.865
	0.60	2.675	3.410	4.037	4.596	5.101	5.546	5.932	6.269
	0.70	2.472	3.119	3.670	4.163	4.617	5.026	5.385	5.703
	0.82	2.241	2.832	3.319	3.762	4.190	4.594	4.954	5.276
	0.90	2.215	2.817	3.315	3.767	4.205	4.616	4.983	5.310
	1.00	2.038	2.656	3.186	3.682	4.183	4.672	5.120	5.524
Gust (m/s)	0.00	2.944	3.240	3.499	3.717	3.904	4.062	4.197	4.315
	0.10	2.991	3.323	3.602	3.831	4.021	4.179	4.310	4.422
	0.20	2.931	3.260	3.536	3.767	3.962	4.128	4.273	4.402
	0.30	2.836	3.169	3.455	3.697	3.902	4.075	4.225	4.355
	0.40	2.768	3.107	3.399	3.647	3.857	4.034	4.185	4.319
	0.50	2.666	3.018	3.314	3.557	3.756	3.919	4.054	4.168
	0.60	2.535	2.976	3.315	3.584	3.805	3.986	4.136	4.264
	0.72	2.440	2.840	3.148	3.397	3.598	3.762	3.898	4.015
	0.80	2.458	2.914	3.270	3.559	3.798	3.998	4.169	4.318
	0.90	2.413	2.842	3.191	3.478	3.714	3.911	4.077	4.217
	1.00	2.386	2.834	3.202	3.514	3.776	3.998	4.186	4.346

TABLE IX

TRANSFORMER ORIGINAL FORECASTS PER $\lambda \in [0, 1]$
RMSE RESULTS OVER 8 FORECAST PERIODS (24 HOURS)

Feature	λ	1.000	2.000	3.000	4.000	5.000	6.000	7.000	8.000
SST (°C)	0.00	0.918	1.027	1.134	1.240	1.345	1.448	1.547	1.644
	0.10	0.893	1.008	1.121	1.233	1.344	1.451	1.556	1.658
	0.20	0.863	0.985	1.102	1.219	1.334	1.446	1.554	1.659
	0.30	0.829	0.957	1.081	1.204	1.324	1.441	1.555	1.664
	0.40	0.784	0.916	1.044	1.171	1.296	1.418	1.537	1.652
	0.50	0.722	0.852	0.982	1.114	1.244	1.372	1.498	1.620
	0.60	0.668	0.808	0.943	1.078	1.211	1.340	1.464	1.583
	0.70	0.615	0.753	0.885	1.017	1.147	1.273	1.395	1.512
	0.80	0.587	0.724	0.853	0.981	1.107	1.230	1.348	1.463
	0.90	0.572	0.699	0.818	0.936	1.052	1.166	1.276	1.383
	1.00	0.568	0.691	0.805	0.918	1.030	1.138	1.243	1.344
Pressure (hPa)	0.00	6.204	6.447	6.663	6.872	7.076	7.269	7.448	7.613
	0.10	6.060	6.334	6.565	6.783	6.995	7.195	7.379	7.548
	0.20	5.878	6.192	6.441	6.671	6.893	7.100	7.290	7.464
	0.30	5.628	6.005	6.284	6.533	6.767	6.984	7.180	7.358
	0.40	5.240	5.735	6.070	6.353	6.611	6.845	7.053	7.242
	0.50	4.050	4.817	5.326	5.718	6.051	6.338	6.587	6.807
	0.60	2.684	3.421	3.995	4.490	4.934	5.325	5.666	5.966
	0.70	2.231	2.877	3.397	3.865	4.306	4.709	5.063	5.379
	0.80	2.003	2.592	3.069	3.512	3.947	4.354	4.716	5.040
	0.90	1.885	2.446	2.904	3.336	3.775	4.196	4.574	4.914
	1.00	1.824	2.391	2.867	3.324	3.799	4.263	4.685	5.070
Gust (m/s)	0.00	2.901	3.145	3.367	3.567	3.745	3.902	4.042	4.169
	0.10	2.844	3.102	3.332	3.537	3.721	3.882	4.027	4.158
	0.20	2.778	3.052	3.292	3.505	3.696	3.864	4.014	4.150
	0.30	2.699	2.992	3.240	3.459	3.653	3.823	3.974	4.111
	0.40	2.595	2.916	3.177	3.403	3.601	3.774	3.927	4.064
	0.50	2.466	2.827	3.107	3.342	3.545	3.720	3.873	4.009
	0.60	2.321	2.707	3.005	3.251	3.459	3.637	3.791	3.927
	0.70	2.229	2.615	2.917	3.165	3.374	3.551	3.704	3.840
	0.80	2.175	2.555	2.855	3.103	3.311	3.488	3.641	3.777
	0.90	2.150	2.522	2.818	3.064	3.270	3.445	3.596	3.730
	1.00	2.138	2.505	2.798	3.042	3.246	3.418	3.567	3.698

TABLE X

GRU DIFFERENCED FORECASTS PER $\lambda \in [0, 1]$

RMSE RESULTS OVER 8 FORECAST PERIODS (24 HOURS)

Feature	λ	1.000	2.000	3.000	4.000	5.000	6.000	7.000	8.000
SST (°C)	0.00	0.220	0.339	0.400	0.451	0.493	0.525	0.549	0.569
	0.10	0.221	0.342	0.410	0.467	0.513	0.549	0.575	0.595
	0.20	0.211	0.330	0.384	0.424	0.455	0.479	0.498	0.515
	0.30	0.225	0.349	0.414	0.463	0.501	0.529	0.551	0.569
	0.40	0.218	0.341	0.402	0.448	0.486	0.517	0.543	0.566
	0.50	0.211	0.330	0.382	0.421	0.453	0.480	0.502	0.523
	0.60	0.219	0.339	0.392	0.430	0.458	0.480	0.496	0.510
	0.70	0.218	0.339	0.393	0.432	0.462	0.486	0.505	0.522
	0.80	0.216	0.336	0.389	0.426	0.455	0.479	0.499	0.517
	0.90	0.218	0.337	0.391	0.430	0.461	0.487	0.510	0.532
	1.00	0.222	0.340	0.391	0.427	0.455	0.478	0.499	0.518
Pressure (hPa)	0.00	1.044	1.498	2.003	2.516	3.028	3.504	3.923	4.304
	0.10	1.043	1.474	1.970	2.490	3.010	3.491	3.911	4.291
	0.20	1.059	1.496	1.994	2.514	3.052	3.571	4.049	4.496
	0.30	1.101	1.582	2.094	2.620	3.153	3.635	4.048	4.422
	0.40	1.085	1.536	2.021	2.517	3.016	3.474	3.878	4.249
	0.50	1.149	1.631	2.132	2.647	3.167	3.646	4.069	4.464
	0.58	1.151	1.697	2.214	2.705	3.183	3.622	4.015	4.377
	0.70	1.183	1.699	2.198	2.678	3.162	3.622	4.037	4.429
	0.80	1.224	1.801	2.316	2.796	3.262	3.695	4.078	4.433
	0.90	1.231	1.797	2.297	2.754	3.195	3.613	3.996	4.360
	1.00	1.276	1.864	2.365	2.808	3.257	3.720	4.151	4.534
Gust (m/s)	0.00	2.089	2.668	3.176	3.623	4.006	4.343	4.637	4.901
	0.10	2.122	2.645	3.095	3.481	3.803	4.073	4.298	4.496
	0.20	2.032	2.489	2.880	3.219	3.517	3.786	4.035	4.273
	0.30	2.052	2.511	2.903	3.244	3.535	3.786	4.005	4.201
	0.40	2.101	2.539	2.914	3.228	3.492	3.718	3.912	4.082
	0.50	2.120	2.532	2.883	3.188	3.454	3.687	3.895	4.081
	0.60	2.154	2.541	2.870	3.143	3.369	3.556	3.716	3.856
	0.70	2.222	2.579	2.891	3.158	3.385	3.578	3.747	3.897
	0.80	2.318	2.650	2.947	3.204	3.422	3.605	3.763	3.900
	0.90	2.434	2.759	3.048	3.297	3.508	3.682	3.832	3.962
	1.00	2.446	2.756	3.032	3.269	3.470	3.639	3.782	3.908

TABLE XI

LSTM DIFFERENCED FORECASTS PER $\lambda \in [0, 1]$
RMSE RESULTS OVER 8 FORECAST PERIODS (24 HOURS)

Feature	λ	1.000	2.000	3.000	4.000	5.000	6.000	7.000	8.000
SST (°C)	0.00	0.214	0.330	0.384	0.424	0.456	0.481	0.501	0.518
	0.10	0.222	0.341	0.405	0.454	0.493	0.525	0.551	0.574
	0.20	0.225	0.356	0.431	0.493	0.546	0.587	0.621	0.652
	0.30	0.220	0.344	0.409	0.458	0.498	0.530	0.557	0.581
	0.40	0.211	0.332	0.389	0.433	0.470	0.501	0.528	0.553
	0.50	0.216	0.336	0.392	0.435	0.474	0.509	0.541	0.570
	0.60	0.222	0.348	0.412	0.459	0.497	0.529	0.557	0.583
	0.70	0.218	0.341	0.397	0.439	0.474	0.505	0.533	0.559
	0.80	0.218	0.338	0.389	0.425	0.452	0.475	0.494	0.511
	0.90	0.215	0.336	0.388	0.424	0.452	0.476	0.495	0.512
	1.00	0.230	0.352	0.408	0.448	0.480	0.507	0.531	0.552
Pressure (hPa)	0.00	1.027	1.463	1.974	2.502	3.035	3.548	4.022	4.461
	0.10	1.065	1.526	2.058	2.612	3.152	3.634	4.038	4.400
	0.20	1.067	1.553	2.068	2.587	3.102	3.567	3.968	4.331
	0.30	1.104	1.579	2.087	2.592	3.085	3.525	3.912	4.272
	0.40	1.097	1.542	2.030	2.542	3.064	3.557	4.015	4.456
	0.50	1.160	1.655	2.147	2.629	3.117	3.576	3.993	4.386
	0.60	1.176	1.710	2.219	2.711	3.195	3.639	4.026	4.387
	0.70	1.180	1.706	2.199	2.666	3.144	3.628	4.088	4.518
	0.76	1.219	1.759	2.262	2.737	3.200	3.635	4.037	4.418
	0.90	1.246	1.795	2.271	2.708	3.159	3.620	4.045	4.434
	1.00	1.320	1.913	2.415	2.865	3.322	3.788	4.215	4.603
Gust (m/s)	0.00	2.059	2.590	3.056	3.469	3.834	4.161	4.463	4.758
	0.10	2.067	2.584	3.023	3.392	3.701	3.957	4.170	4.349
	0.20	2.073	2.561	2.982	3.338	3.633	3.887	4.112	4.313
	0.30	2.077	2.539	2.922	3.240	3.502	3.720	3.906	4.067
	0.40	2.106	2.557	2.968	3.344	3.683	3.993	4.280	4.536
	0.50	2.133	2.584	3.000	3.383	3.729	4.050	4.354	4.638
	0.60	2.197	2.580	2.911	3.198	3.443	3.653	3.840	4.009
	0.70	2.267	2.635	2.955	3.229	3.463	3.660	3.831	3.982
	0.80	2.369	2.703	3.003	3.267	3.492	3.683	3.850	3.998
	0.90	2.393	2.713	3.006	3.260	3.473	3.652	3.806	3.939
	1.00	2.470	2.782	3.061	3.299	3.498	3.664	3.806	3.931

TABLE XII

TRANSFORMER DIFFERENCED FORECASTS PER $\lambda \in [0, 1]$
RMSE RESULTS OVER 8 FORECAST PERIODS (24 HOURS)

Feature	λ	1.000	2.000	3.000	4.000	5.000	6.000	7.000	8.000
SST (°C)	0.00	0.206	0.322	0.380	0.426	0.463	0.494	0.522	0.549
	0.10	0.205	0.321	0.378	0.421	0.456	0.486	0.512	0.539
	0.20	0.204	0.321	0.376	0.417	0.450	0.479	0.504	0.530
	0.30	0.204	0.320	0.374	0.413	0.445	0.472	0.497	0.522
	0.40	0.203	0.320	0.373	0.410	0.441	0.468	0.492	0.516
	0.50	0.202	0.320	0.371	0.408	0.438	0.464	0.488	0.511
	0.60	0.202	0.320	0.370	0.406	0.435	0.461	0.485	0.507
	0.70	0.201	0.319	0.370	0.404	0.433	0.458	0.482	0.504
	0.80	0.201	0.320	0.369	0.404	0.432	0.457	0.479	0.500
	0.90	0.201	0.320	0.369	0.403	0.431	0.456	0.477	0.497
	1.00	0.201	0.320	0.370	0.404	0.431	0.455	0.476	0.495
Pressure (hPa)	0.00	0.933	1.410	1.940	2.470	2.980	3.440	3.847	4.216
	0.10	0.950	1.438	1.966	2.494	3.005	3.463	3.865	4.231
	0.20	0.965	1.465	1.993	2.519	3.029	3.484	3.883	4.246
	0.30	0.984	1.501	2.031	2.553	3.060	3.511	3.904	4.265
	0.40	1.007	1.544	2.079	2.596	3.095	3.541	3.931	4.289
	0.50	1.032	1.594	2.134	2.646	3.136	3.575	3.964	4.320
	0.60	1.059	1.649	2.197	2.700	3.178	3.614	4.003	4.357
	0.70	1.091	1.712	2.269	2.761	3.227	3.662	4.054	4.404
	0.80	1.125	1.783	2.348	2.826	3.281	3.723	4.119	4.461
	0.90	1.160	1.855	2.426	2.889	3.335	3.787	4.184	4.517
	1.00	1.197	1.933	2.511	2.958	3.396	3.859	4.256	4.578
Gust (m/s)	0.00	1.820	2.225	2.573	2.869	3.115	3.319	3.491	3.640
	0.10	1.852	2.251	2.592	2.883	3.125	3.326	3.495	3.642
	0.20	1.885	2.278	2.614	2.901	3.139	3.336	3.503	3.649
	0.30	1.923	2.312	2.643	2.924	3.157	3.351	3.515	3.658
	0.40	1.966	2.351	2.676	2.952	3.181	3.372	3.533	3.673
	0.50	2.012	2.393	2.713	2.984	3.209	3.396	3.554	3.692
	0.60	2.061	2.437	2.751	3.017	3.239	3.422	3.578	3.713
	0.70	2.114	2.484	2.794	3.055	3.272	3.453	3.605	3.738
	0.80	2.171	2.532	2.836	3.093	3.307	3.485	3.635	3.765
	0.90	2.230	2.581	2.880	3.133	3.343	3.518	3.665	3.794
	1.00	2.290	2.630	2.923	3.172	3.379	3.552	3.697	3.824