RESEARCH ARTICLE

Physics-Coupled Spatio-Temporal Active Learning for Dynamical Systems

YU HUANG[1], (Student Member, IEEE), **YUFEI TANG**[1], (Senior Member, IEEE),
XINGQUAN ZHU[1], (Senior Member, IEEE), **HANQI ZHUANG**[1], (Senior Member, IEEE),
AND LAURENT CHERUBIN[2]

[1]Department of Electrical Engineering and Computer Science, Florida Atlantic University, Boca Raton, FL 33431, USA
[2]Harbor Branch Oceanographic Institute, Florida Atlantic University, Fort Pierce, FL 34946, USA

Corresponding author: Yufei Tang (tangy@fau.edu)

ABSTRACT Spatio-temporal forecasting is of great importance in a wide range of dynamic systems applications, such as earth science, transport planning, etc. These applications rely on accurate predictions of spatio-temporal structured data reflecting real-world phenomena. A stunning characteristic is that the dynamical system is not only driven by some physics laws but also impacted by the localized factor in spatial and temporal regions. One of the major challenges is to infer the underlying causes, which generate the perceived data stream and propagate the involved causal dynamics through the distributed observing units. Another challenge is that the success of machine learning-based predictive models requires massive annotated data for model training. However, the acquisition of high-quality annotated data is objectively manual and tedious as it needs a considerable amount of human intervention, making it infeasible in fields that require high levels of expertise. To tackle these challenges, we advocate a spatio-temporal physics-coupled neural networks (ST-PCNN) model to learn the underlying physics of the dynamical system and further couple the learned physics to assist the learning of the recurring dynamics. To deal with data-acquisition constraints, an active learning mechanism with Kriging for actively acquiring the most informative data is proposed for ST-PCNN training in a partially observable environment. Our experiments on both synthetic and real-world datasets exhibit that the proposed ST-PCNN with active learning converges to near-optimal accuracy with substantially fewer instances.

INDEX TERMS Spatio-temporal modeling, physics-informed neural networks, active learning, Gaussian process model, dynamical systems.

I. INTRODUCTION

Spatio-temporal modeling is essential in many scientific fields ranging from studies in biology [1], information flow in social networks [2], traffic predictions [3], and earth science [4], to recent COVID-19 spread modeling [5]. These applications rely on accurate predictions of spatio-temporal structured data reflecting real-world phenomena. With an unprecedented increase in data accessibility and computational capability, machine learning (ML) has been rapidly explored in spatio-temporal modeling [6], [7], [8], [9], [10].

The associate editor coordinating the review of this manuscript and approving it for publication was Jiajia Jiang.

One of the major challenges of spatio-temporal ML is the high cost of data acquisition. Effective training of advanced ML models requires large amounts of labeled data. However, the acquisition of a large number of high-quality annotated data consumes a lot of human labor, making it unfeasible in fields that require high levels of expertise [11]. This problem is outstanding, especially in atmospheric and oceanographic science, which involves the accurate prediction of some variables of interest [12], [13]. For example, collecting ocean data usually requires trained scientists to travel to sampling locations and deploy in-situ instruments within oceanic structures. The cost further limits the total number of sensors placed/installed to query the data; while the incurring

personnel and equipment costs for data does not necessarily mean improved model predictions.

When and where to query the data within constraints to build efficient and accurate predictive models is an elementary question for spatio-temporal forecasting. To answer this question, active learning (AL) as a special form of weakly supervised learning that attempts to maximize model performance while annotating the fewest samples possible, seems to be a natural alternative in the spatio-temporal domain [11]. AL approaches, from application scenarios [14], can be categorized into membership query synthesis [15], stream-based selective sampling [16], and pool-based AL [17]. Among these approaches, stream-based selective sampling makes an independent evaluation of each sample in the data stream and is more suitable for spatio-temporal scenarios since the whole dataset is not always available and timeliness is required.

Spatio-temporal learning for dynamical systems faces another major challenge to inferring the underlying causes or prime drivers, which generate the perceived data stream and propagate the dynamics that can be observed through the distributed sensors. Stunning characteristics of such scenarios are that the widely distributed sensors share *homogeneity* and *heterogeneity*, where the former is driven by the physical laws governing the systems and the latter is impacted by the spatial and temporal factors in localized regions. One of the critical properties that most of the spatio-temporal processes have in common is that some general underlying principles will apply irrespective of time or location when observing natural processes. For example, many climate models based on mathematical equations that describe the physical processes have been built to predict climate variables while the prediction abilities may still apply to other regions and times with dramatic variations [4]. Physics-informed ML has emerged to build hybrid models for robust predictions recently [18], [19], [20], [21], [22], however, these methods seldom consider the homogeneity and heterogeneity simultaneously when observation data is scarce.

In this paper, we introduce a spatio-temporal physics-coupled neural networks (ST-PCNN) model to capture the spatio-temporal correlations, heterogeneity, and homogeneity in a spatially distributed manner. To bring this technology closer to reality, we further empower ST-PCNN with active learning to address data limitations in practice. The main contributions of this paper are the following:

- We propose ST-PCNN, a novel framework for spatio-temporal modeling of dynamical systems.

  - ST-PCNN learns a forecasting neural network executed at different locations, integrating lateral information interacting with the neighborhoods. ST-PCNN allows efficient parallel computation and captures heterogeneity from all spatial locations.
  - ST-PCNN is embedded with a physics learning module to learn the homogeneity in the form of partial differential equations (PDE). The physics module first fits a Gaussian process to the observations

and then obtains the PDE coefficients by calculating the Gaussian process-derived partial derivatives of the state variable, without the need for initial and boundary conditions.

- We further introduce a stream-based active learning mechanism into the proposed ST-PCNN framework. We design a Kriging sampling method to obtain the critical training data at locations where underlying predictions are in most need of improvement.

The rest of the paper is organized as follows. Section II discusses related work. Section III presents the problem definition followed by the proposed method in Section IV. Section V conducts extensive experiments and comparative studies on both synthetic and real-world datasets, where the former dataset manifests a clear spatio-temporal correlation and the latter presents strong sparsity. Section VI draws conclusions and discusses future works.

II. RELATED WORK
A. ACTIVE LEARNING KRIGING MODEL

When and where to query the data to build an accurate predictive model while staying within a maximum budget for data collection is a rising question for spatio-temporal prediction [23]. Statistical predictive approaches that commonly used in geographic information systems are not applicable, since they require robust data to model first- and second-order spatial effects [24]. In the case of vast amounts of unlabeled data being available and labeled data being scarce since the labeling process is time-consuming and costly, active learning (AL) could be an alternative [17], [23], [25], [26], [27]. The first category of AL methods are based on informativeness that exploits the uncertainty of the classifier predictions to select informative samples [28], and the uncertainty are commonly estimated by least confidence first [29], [30], margin sampling [31], [32], [33], or entropy [34], [35]. The second category AL methods are based on density that aims to enhance the representativeness of the selected samples [26], [27], [36], [37], [38], [39]. Recently, the proposition of the active learning Kriging model is a significant step forward [40], [41]. The general way is to add observations at the locations corresponding to local maxima [42], and a new prediction is then made for the whole space followed by a new local maximum computed to select the next location for additional observations. The local maxima selection criterion is usually based on statistical interpretation. However, due to the constraint set on location number when querying data, in this paper, we improve the active learning Kriging method to utilize only the optimal candidates instead of expanding the dataset as others do.

B. ACTIVE LEARNING WITH LEARNING PHYSICS

Notably, AL is also explored to solve PDEs, e.g., an AL algorithm is proposed for training a neural network to predict PDE solutions over the entire parameter space using training data from a minimal region [43]. AL is also used to quantify
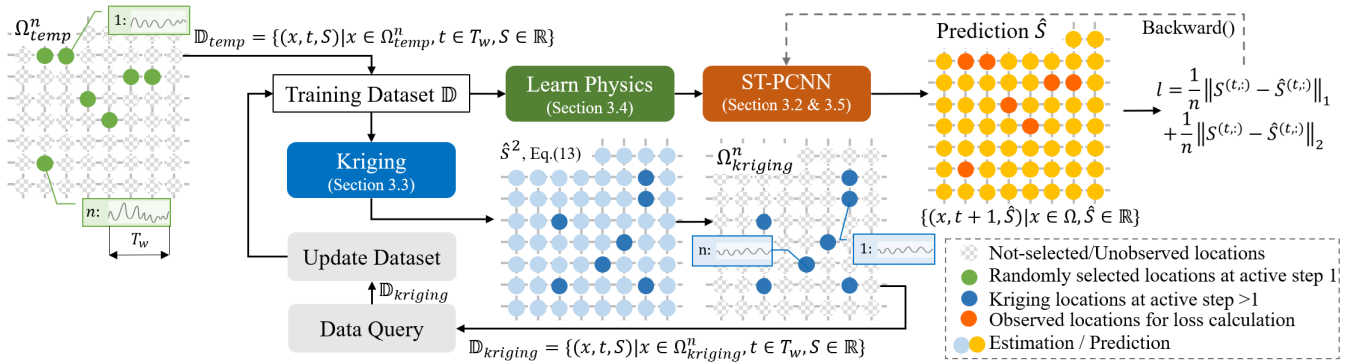
FIGURE 1. Overall framework of the proposed ST-PCNN model with active learning. A system consists of N observable locations, but can only collect values from n locations at any particular time. In the beginning, a set of n random locations are selected (the green-colored dots on the top-left panel) to collect the training dataset $\mathbb{D}$. The lower panel loop denotes Kriging-based active learning, which actively queries locations at the next step, and uses queried samples to update the training set $\mathbb{D}$. Using the updated training dataset, the "Learn Physics" module learns the underlying physics, and combines it with a forecasting network to predict future values of all N observable locations (the orange-colored dots on the top right). Because the red-colored dots contain queried values, they are used for loss calculation.

uncertainty in resulting predictive posterior distributions and naturally leads to adaptive solution refinement, where the observables are scarce and noisy multi-fidelity data [44]. Further, dropout is demonstrated that it can quantify the uncertainty of deep neural networks in solving forward and inverse differential equations and serves as useful guidance for active learning [45]. In this paper, considering the property that the derivative of the Gaussian process (GP) is also a GP, we propose a new method for PDEs learning that uses GP as a basis function. The learned PDE is coupled into our proposed ST-PCNN model to enhance the spatio-temporal modeling.

III. PROBLEM DEFINITION

An observing system (e.g., ocean observation sensor network [46]) consists of N observable locations ($|\Omega| = N$), each produces a set of real valued observations in a temporal order. Due to resource constraints, at any particular data collection period, the system can only collect values from n locations ($0 < n \ll N$), denoted as Ω_{temp}^n. The collected training set at each period is $\mathbb{D}_{temp} = \{(\boldsymbol{x}, t, \mathcal{S}) | \boldsymbol{x} \in \Omega_{temp}^n, t \in T_w, \mathcal{S} \in \mathbb{R}\}$, where $\mathcal{S}$ represents the observation value, $\boldsymbol{x}$ is the location, and T_w is the maximum length of up-to-date data stored in each location due to memory capacity.

Given a dynamical system with N observable locations and their values up to current time t, and a budget n restricting the number of query locations, our research has a **twofold-goal** to (1) find optimal n locations to query their values; and (2) create a model to forecast future values of the whole system, beyond the current time point (w.r.t $t + 1, t + 2, \cdots$), with minimum forecasting errors.

To achieve the goal, our proposed framework has two main components: (1) an active learning Kriging module to query high uncertainty points in the system; and (2) a physics-coupled spatial-temporal learning to combine observed values and learned physics of the system to predict future values. At current time point t, our system has observed $\mathbb{D}_{temp}$ of previous period (i.e., $t - T_w$ to t) from selected sites, the

active learning Kriging module will determines where to query n sites in the next period for best possible predictions. Because such queried values do not guarantee minimum forecasting errors, physics-informed learning leverages observed values to learn physics, and then assist in predicting future values. The query and forecasting are essentially related, so our proposed framework delivers a closed-loop solution to ensure that active learning query and dynamics forecasting are worked synergistically to achieve the designed optimization goal.

IV. PROPOSED METHOD
A. OVERVIEW OF THE FRAMEWORK
To model the dynamical systems considering both where and when to measure and query the data for making the best possible predictions while staying within a maximum budget of sensors and data, a spatio-temporal physics-coupled neural networks (ST-PCNN) model with active learning is proposed.

Active learning is a paradigm in which the network training procedure identifies and requests additional, high-benefit training data from an *oracle*. In our case, the oracle is an observing grid Ω constrained by the maximal data collection cost. The goal is to train the model to accurately predict the system dynamics everywhere within desired area/grid Ω, using training data from a minimal number of locations in Ω_{temp}. Our algorithm falls into the category of stream-based selective sampling active learning, shown in Figure 1 and implemented in Algorithm 1.

B. KRIGING SAMPLING FOR ACTIVE LEARNING
Kriging, also known as Gaussian process regression, is a widely used method in applied mathematics and machine learning for constructing surrogate models, interpolation, supervised learning, and active learning [25]. Kriging constructs a statistical model of a partially observed function of time and/or space, assuming that this function is a realization of a Gaussian process (GP) that is uniquely described by

---

Algorithm 1 Active Learning for ST-PCNN

Input : (1) Ω, a dynamical system as a grid; (2) n, budget restricting maximum # of query locations at each time t;

Output: ST-PCNN, the trained spatio-temporal model

Initialize

- $\Omega_{temp}^n \leftarrow$ Randomly select n locations from Ω to form a temporal observation subset;
- $\mathbb{D} = \mathbb{D}_{temp} \leftarrow$ Collect T_w consecutive observations from each selected locations to form the training data;
- [*Termination Conditions*] $\leftarrow$ Threshold of prediction loss ℓ to terminate active learning;

while *Termination Conditions NOT satisfied* **do**

- **Learn Physics**: $[\boldsymbol{\lambda}] \leftarrow$ learn the physics, *i.e.* coefficients $\boldsymbol{\lambda}$ of PDE, from the existing training data $\mathbb{D}$ by minimizing Eq. (26);
- **ST-PCNN Training**: train the ST-PCNN with the learned physics λ from training data $\mathbb{D}$, see Algorithm 2 with details;
- **Prediction**: $\left[\hat{\boldsymbol{S}}\right] \leftarrow$ make prediction at all locations in the network, $\forall \boldsymbol{x} \in \Omega$;
- **Kriging**: $\Omega_{kriging}^n \leftarrow$ use Kriging to identify n query locations with the largest estimated errors for next active learning step;
- **Data Query**: obtain consecutive observations of window size T_w from the newly selected locations $\mathbb{D}_{kriging} = \{(\boldsymbol{x}, t, \mathcal{S}) | \boldsymbol{x} \in \Omega_{kriging}^n, t \in T_w, \mathcal{S} \in \mathbb{R}\}$;
- **Update Dataset**: add $\mathbb{D}_{kriging}$ to the existing training data $\mathbb{D}$;

end

---

its mean and covariance. The Kriging method used in this paper is the general ordinary Kriging, which is based on the assumption of a stationary Gaussian process. Compared to Universal Kriging, ordinary Kriging assumes stationarity that requires fewer hyperparameters and reduces the model complexity.

Denote the observation locations as $\Omega = \{\boldsymbol{x}^{(i)}\}_{i=1}^N$, where $\boldsymbol{x}^{(i)}$ are d-dimensional vectors in $\mathcal{D} \in \mathbb{R}^d$ and the observed state values at these locations as $\boldsymbol{y} = \{y^{(i)}\}_{i=1}^N$, where $y^{(i)} \in \mathbb{R}$. The Kriging assumes that the observation vector $\boldsymbol{y}$ is a realization of the following N-dimensional random vector that satisfies multivariate Gaussian distribution:

$$\mathcal{Y} = \left(\mathcal{Y}(\boldsymbol{x}^{(1)}), \mathcal{Y}(\boldsymbol{x}^{(2)}), \cdots, \mathcal{Y}(\boldsymbol{x}^{(N)})\right)^T \quad (1)$$

where $\mathcal{Y}(\boldsymbol{x}^{(i)})$ is the concise notation of $\mathcal{Y}(\boldsymbol{x}^{(i)}; \omega)$, which is a random variable defined on a probability space $(\Lambda, \mathcal{E}, \mathcal{P})$ with $\omega \in \Lambda$, where Λ is the sample space, $\mathcal{E}$ is the σ-algebra over Λ, and $\mathcal{P}$ is the probability measure on Λ [40]. Of note, $\boldsymbol{x}(i)$ can be considered as the parameters of the GP $\mathcal{Y} : \mathcal{D} \times \Lambda \to \mathbb{R}$, such that $\mathcal{Y}(\boldsymbol{x}^{(i)}) : \Lambda \to \mathbb{R}$ is a Gaussian random

variable for any $\boldsymbol{x}^{(i)}$ in the set $\mathcal{D}$. Here, $\mathcal{Y}(\boldsymbol{x})$ is denoted as:

$$\mathcal{Y}(\boldsymbol{x}) \sim \mathcal{GP}(\mu(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}')) \quad (2)$$

where $\mu : \mathcal{D} \to \mathbb{R}$ is the mean, $k(\boldsymbol{x}, \boldsymbol{x}') : \mathcal{D} \times \mathcal{D} \to \mathbb{R}$ is the covariance function that provides a measure of closeness between a training point $\boldsymbol{x}$ and a test point $\boldsymbol{x}'$:

$$\mu(\boldsymbol{x}) = \mathbb{E}\{\mathcal{Y}(\boldsymbol{x})\} \quad (3)$$
$$k(\boldsymbol{x}, \boldsymbol{x}') = \mathbb{E}\{(\mathcal{Y}(\boldsymbol{x}) - \mu(\boldsymbol{x}))(\mathcal{Y}(\boldsymbol{x}') - \mu(\boldsymbol{x}'))\} \quad (4)$$

The variance of $\mathcal{Y}(\boldsymbol{x})$ is $k(\boldsymbol{x}, \boldsymbol{x})$ and its standard deviation is $\sigma(\boldsymbol{x}) = \sqrt{k(\boldsymbol{x}, \boldsymbol{x})}$. The covariance matrix $\mathcal{C}$ of a random vector $\mathcal{Y}$ is defined as:

$$\mathcal{C} = \begin{bmatrix} k(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(1)}) & \cdots & k(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(N)}) \\ \vdots & \ddots & \vdots \\ k(\boldsymbol{x}^{(N)}, \boldsymbol{x}^{(1)}) & \cdots & k(\boldsymbol{x}^{(N)}, \boldsymbol{x}^{(N)}) \end{bmatrix} \quad (5)$$

then, the estimation at any new location x^* is given as:

$$\hat{y}(x^*) = \mu(x^*) + \boldsymbol{c}^T \mathcal{C}^{-1}(\boldsymbol{y} - \boldsymbol{\mu}) \quad (6)$$

where $\boldsymbol{\mu} = \left[\mu(\boldsymbol{x}^{(1)}), \cdots, \mu(\boldsymbol{x}^{(N)})\right]^T$ and $\boldsymbol{c}$ is a vector of covariance between the observed data and the predictions:

$$\begin{aligned} \boldsymbol{c} &= c(\boldsymbol{x}^*) \\ &= \left[k(\boldsymbol{x}^{(1)}, \boldsymbol{x}^*), k(\boldsymbol{x}^{(2)}, \boldsymbol{x}^*), \cdots, k(\boldsymbol{x}^{(N)}, \boldsymbol{x}^*)\right]^T \quad (7) \end{aligned}$$

The mean squared error (MSE) of this prediction is defined as $\hat{s}^2(x^*) = \mathbb{E}\{(\hat{y}(x^*) - \mathcal{Y}(x^*))^2\}$ and is calculated as $\hat{s}^2(x^*) = \sigma^2(x^*) - \boldsymbol{c}^T \mathcal{C}^{-1}\boldsymbol{c}$. The prediction and MSE can be derived from the maximum likelihood estimate (MLE) method. The n points with the largest MSE values are selected as the next query points. This policy reassembles to the uncertainty sampling in active learning, so the most uncertain points are selected for future query.

In this context, the Kriging sampling is a process of identifying new locations for critical observations that minimize the prediction error and reduce MSE or uncertainty. Notably, Algorithm 1 is a greedy algorithm to identify observation locations when some observations are affordable. It does not guarantee to identify the actual optimal new observation locations, but the estimated optimal. Our active learning framework, unlike others that keep adding new locations for data collection, is constrained by maximal data collection cost that only a fixed limited number of locations are available at each period. A natural way to improve the predictive performance is to consider leveraging observations at the locations corresponding to local maximum in $\hat{s}^2(x^*)$. Then, we can train our model on those selected locations to make a more accurate prediction and compute a new $\hat{s}^2(x^*)$ to select the locations for the next period's data collection, training, and prediction.

C. PHYSICS LEARNING

Physicists attempt to model natural phenomena in a principled way through analytic descriptions. Conservation laws, physical principles, or phenomenological behaviors are generally formalized using differential equations, which can

best derive the homogeneity of observations. The knowledge accumulated for modeling physical processes in well-developed fields could be a useful guideline for dynamics learning. ST-PCNN includes a physics-aware module (*i.e.*, PN) to learn underlying hidden physics. The objective of the PN is to estimate parameters of a partial differential equation (PDE), given observations of state variable.

The PN requires fitting a GP model to the observations of the state variable. The derivatives of the state variable can be obtained using the property that 'the derivative of a GP is also a GP' [49], and finally adjusting the PDE coefficients so that the GP derived partial derivatives satisfy the PDE [50]. This module does not require initial and boundary conditions.

The Gaussian covariance function, that notationally given by $k(v, v')$, also known as kernel, is:

$$k(v, v') = \sigma_s^2 exp\left(-\frac{1}{2}\sum_{j=1}^{m}\frac{(v_j - v'_j)^2}{l_j^2}\right) \qquad (8)$$

where v is the collection of variables $\{x, t\}$, m is the dimension of the variables space, *i.e.*, the number of independent variables in v and $v_j \in v$, l_j is the length scale for the j-th independent variable v_j, and σ_s is the scale parameter of the covariance function. Together, they are referred as hyperparameters and are represented by $\phi = [l_1, \cdots, l_m, \sigma_s]$. The Gaussian covariance function, also known as squared exponential covariance function, is infinitely differentiable and hence very smooth.

Let the dataset be $\mathbb{D} = \{(v_i, y_i) | i = 1, \cdots, N; v_i \in \mathbb{R}^m; y_i \in \mathbb{R}\}$, where N is the number of observations, v_i is an input vector (independent variable) in m-dimensional space, and y_i is the corresponding observed value of the dependent variable. Also assume that all of the y_i have independent and identically distributed noise, which is assumed to be normally distributed with mean zero and standard deviation σ_y:

$$y_i = \tilde{y}_i + \mathcal{N}(0, \sigma_y^2) \qquad (9)$$

where $\tilde{y}_i$ is the theoretical value of the state variable $v_i = [x_i, t]$ at location x_i and time t. Suppose y follows a Gaussian distribution with mean $\bar{V}$ and covariance matrix $\mathcal{C}$, represented as:

$$y \sim \mathcal{N}(\bar{V}, \mathcal{C}) \qquad (10)$$

where $\bar{V}$ is a $N \times 1$ vector of the priori mean of GP at data points $\mathbb{D}$. In most cases the a priori mean may not be available, but the GP model is powerful enough to capture trends in the data even if the mean of GP is taken as zero or a constant value.

The estimation y^* at new points $v^* \in \mathcal{V}^*$ is given by:

$$y^* \sim \mathcal{N}(\bar{V}^*, \mathcal{C}^{**}) \qquad (11)$$

where $\mathcal{C}^{**} = k(v^*, v^*)$. The joint distribution of y and y^* is given by:

$$\begin{bmatrix} y \\ y^* \end{bmatrix} = \mathcal{N}\left(\begin{bmatrix} \bar{V} \\ \bar{V}^* \end{bmatrix}, \begin{bmatrix} \mathcal{C} + \sigma_y^2 I & \mathcal{C}^* \\ \mathcal{C}^{*T} & \mathcal{C}^{**} \end{bmatrix}\right) \qquad (12)$$

where I is the identity matrix, $\mathcal{C}^* = k(v, v^*)$ and the elements of $\mathcal{C}^*$ are covariance between the ith observed point and the jth new point (i.e., k_{ij}).

The predictive distribution $p(y^*|V^*, \phi, \sigma_y, V, y)$ based on the conditional property of the Gaussian distribution is:

$$p(y^*|V^*, \phi, \sigma_y, V, y) = \mathcal{N}(y^*|\mu, \Sigma) \qquad (13)$$

where

$$\mu = \mathbb{E}[y^*|V^*, \phi, \sigma_y, V, y]$$
$$= \bar{V}^* + \mathcal{C}^{*T}(\mathcal{C} + \sigma_y^2 I)^{-1}(y - \bar{V}) \qquad (14)$$
$$\Sigma = cov[y^*|V^*, \phi, \sigma_y, V, y]$$
$$= \mathcal{C}^{**} - \mathcal{C}^{*T}(\mathcal{C} + \sigma_y^2 I)^{-1}\mathcal{C}^* + \sigma_y^2 I \qquad (15)$$

Estimating the PDE coefficients requires derivatives of the state variable (fitted by a GP model) with respect to independent variables. According to [49], the GP relation Eq. (14) could be straightforwardly differentiated with respect to input variable. The first-derivatives of the covariance function is presented as:

$$\frac{\partial k(v, v*)}{\partial v_j^*} = \frac{\partial}{\partial v_j^*}\left[\sigma_s^2 exp\left(-\frac{1}{2}\sum_{i=1}^{m}\frac{(v_i - v_i^*)^2}{l_i^2}\right)\right]$$
$$= k(v, v*)\frac{(v_j - v_j^*)}{l_j^2} \qquad (16)$$

The second-derivatives of the covariance function is presented as:

$$\frac{\partial^2 k(v, v*)}{\partial v_j^{*2}} = \frac{\partial}{\partial v_j^*}\left[k(v, v*)\frac{(v_j - v_j^*)}{l_j^2}\right]$$
$$= \frac{k(v, v*)}{l_j^2}\left[\frac{(v_j - v_j^*)^2}{l_j^2} - 1\right] \qquad (17)$$
$$\frac{\partial^2 k(v, v*)}{\partial v_j v_j^*} = \frac{\partial}{\partial v_j^*}\left[k(v, v*)\frac{(v_j - v_j^*)}{l_j^2}\right]$$
$$= \frac{k(v, v*)}{l_j^2}\left[1 - \frac{(v_j - v_j^*)^2}{l_j^2}\right] \qquad (18)$$

For a constant mean GP, the first order derivative $\dot{y}_{gp}$ is given by $p(\dot{y}_{gp}|y, \phi) = \mathcal{N}(\dot{\mu}, \dot{\Sigma}_{gp})$, where $\dot{\Sigma}_{gp}$ is the covariance matrix of $\dot{y}_{gp}$. Specifically:

$$\dot{\mu} = \frac{\partial}{\partial v_j^*}\mathbb{E}[y^*|V^*, \phi, \sigma_y, V, y]$$
$$= \dot{\mathcal{C}}^{*T}(\mathcal{C} + \sigma_y^2 I)^{-1}(y - \bar{V}) \qquad (19)$$
$$\dot{\Sigma}_{gp} = \dot{\mathcal{C}}^{**} - \dot{\mathcal{C}}^{*T}(\mathcal{C} + \sigma_y^2 I)^{-1}\dot{\mathcal{C}}^* \qquad (20)$$

where $\dot{\mathcal{C}}^*$ is populated by the mixed covariance function between the state variable and its first order partial derivatives, $cov(y, \frac{\partial}{\partial v_j^*}y^*) = \frac{\partial}{\partial v_j^*}k(v, v^*)$, also known as cross-covariance between the state variable and its derivative. $\dot{\mathcal{C}}^{**}$ is a mixed covariance function between partial derivatives:

$$cov(\frac{\partial}{\partial v_j}y, \frac{\partial}{\partial v_j^*}y^*) = \frac{\partial^2}{\partial v_j \partial v_j^*}k(v, v^*) \qquad (21)$$

Accordingly, the second-derivative of $\mathcal{GP}$ is given by:

$$
\ddot{\bar{y}}_{gp} = \frac{\partial^2}{\partial v_j^{*2}} \mathbb{E}[y^* | V^*, \boldsymbol{\phi}, \sigma_y, V, y]
$$

$$
= \ddot{\mathcal{C}}_*^T (\mathcal{C} + \sigma_y^2 I)^{-1}(y - \bar{\mathcal{V}}) \tag{22}
$$

where $\ddot{\mathcal{C}}_*^T$ is populated by the mixed covariance function between the state variable and its second order partial derivative:

$$
cov(y, \frac{\partial^2}{\partial v_j^{*2}} y^*) = \frac{\partial^2}{\partial v_j^{*2}} k(\boldsymbol{v}, \boldsymbol{v}^*) \tag{23}
$$

Let the hidden PDE be given by an implicit function as:

$$
f(v_1, \cdots, v_m, y, \frac{\partial y}{\partial v_1}, \cdots, \frac{\partial y}{\partial v_m}, \frac{\partial^2 y}{\partial v_1 \partial v_1},
$$

$$
\cdots, \frac{\partial^2 y}{\partial v_1 \partial v_m}, \cdots, \lambda) \tag{24}
$$

where y is the state variable, $v_1, \cdots, v_m$ are independent variables, and λ is a set of PDE coefficients. By given the observations of the state variable and its GP derivatives involved in the PDE, the residual error in the PDE at an observed point is given by:

$$
\epsilon = f\left(v_1, \cdots, v_m, y, \lambda\right.
$$

$$
\left. \mathcal{GP}\{\frac{\partial y}{\partial v_1}, \cdots, \frac{\partial y}{\partial v_m}, \frac{\partial^2 y}{\partial v_1 \partial v_1}, \cdots, \frac{\partial^2 y}{\partial v_1 \partial v_m}, \cdots\}\right) \tag{25}
$$

where the notation $\mathcal{GP}(\cdot)$ incorporates all the partial derivatives of the state variable, means that the arguments are evaluated using the GP relation. The coefficients set λ can be obtained by assuming a distribution for the residual errors and then applying either the method of maximum likelihood or the Bayesian estimation method using Markov chain Monte Carlo sampling. These methods not only provide a point estimate of k but also an estimation interval. However, to keep the parameter estimation simple, we obtain k by minimizing the sum of square of residual error (SSRE) as:

$$
SSRE
$$

$$
= \epsilon^T \epsilon
$$

$$
= \sum_{\forall \boldsymbol{v}, y \in \mathbb{D}} f\left(v_1, \cdots, v_m, y, \lambda, \right.
$$

$$
\left. \mathcal{GP}\{\frac{\partial y}{\partial v_1}, \cdots, \frac{\partial y}{\partial v_m}, \frac{\partial^2 y}{\partial v_1 \partial v_1}, \cdots, \frac{\partial^2 y}{\partial v_1 \partial v_m}, \cdots\}\right)^2 \tag{26}
$$

The minimization is performed using Nelder-Mead algorithm. The GP regression model provides standard error for the estimated state variables and its derivatives. This uncertainty information is propagated to the SSRE by weighting each term, where weights are inversely proportional to the variance of the terms.
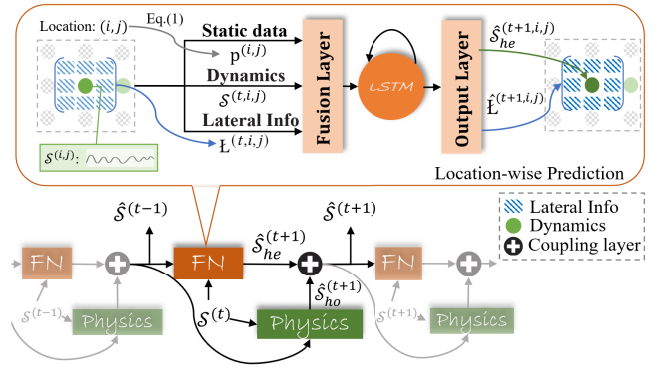

FIGURE 2. Lateral connection schema of forecasting network (FN) and physics network (PN). The upper panel unfolds the structure of the FN network. The green node denotes a center node located at (i, j), the grey (unobserved) and light green (observed) nodes are its neighbors. Three types of information are used to characterize each node: (1) $\mathcal{S}^{(t,i,j)}$: an embedding vector representing dynamics of node at time t, (2) $\vec{\mathbf{p}}^{(i,j)}$: an embedding vector representing node location; and (3) Ł$^{(t,i,j)}$: an embedding vector (dashed dot-square set) capturing interaction (lateral info) between each node and its neighbors.

Algorithm 2 ST-PCNN Training

Input : $\mathbb{D}_{temp} = \{(\boldsymbol{x}, t, \mathcal{S}) | \boldsymbol{x} \in \Omega_{temp}^n, t \in T_w, \mathcal{S} \in \mathbb{R}\}$;
Initialize
    Neural Network parameters: $\theta_{\mathcal{F}}, \theta_{\mathcal{C}}$;
    Static info: $\vec{\mathbf{p}} \leftarrow$ Positional-Encoding(d, Ω);
    Lateral info: Ł $\leftarrow \mathbf{0}$;
for *number of epochs* **do**
    for *t in T_w* **do**
       for *i, j in Ω_{temp}^n* **do**
          $\left[\hat{\mathcal{S}}_{he}^{(t+1,i,j)}, \hat{\text{Ł}}^{(t+1,i,j)}\right] \leftarrow$
          $\mathcal{FN}(\vec{\mathbf{p}}^{(i,j)}, \mathcal{S}^{(t,i,j)}, \text{Ł}^{(t,i,j)}, \theta_{\mathcal{F}})$;
          $\hat{\mathcal{S}}_{ho}^{(t+1,i,j)} \leftarrow PDE(\vec{\mathbf{p}}^{(i,j)}, \hat{\mathcal{S}}^{(t-1,i,j)}, \mathcal{S}^{(t,i,j)}, \lambda)$;
          $\hat{\mathcal{S}}^{(t+1,i,j)} =$
          $Coupling(\hat{\mathcal{S}}_{he}^{(t+1,i,j)}, \hat{\mathcal{S}}_{ho}^{(t+1,i,j)}, \theta_{\mathcal{C}})$;
       end
       Update Ł$^{(t+1,:)} \leftarrow \hat{\text{Ł}}^{(t+1,:)}$;
    end
    loss $\ell \leftarrow \frac{1}{n}\left\|\mathcal{S}^{(t,:)} - \hat{\mathcal{S}}^{(t,:)}\right\|_1 + \frac{1}{n}\left\|\mathcal{S}^{(t,:)} - \hat{\mathcal{S}}^{(t,:)}\right\|_2$;
    Update $\theta_{\mathcal{F}}, \theta_{\mathcal{C}} \leftarrow loss\ \ell.$backward$()$
end

D. FORECASTING NETWORK AND PHYSICS COUPLING

ST-PCNN is a bi-network architecture, shown in Figure 2, including a physics network (PN) and a forecasting network (FN). The FN produces *heterogeneous* prediction leveraging its own specific local attributes only, while the PN generates *homogeneous* solution of the dynamics regularized by the underlain governing physics. The heterogeneous is partially affected by the first- and second-order spatial effects.

Three types of information are fed into the forecasting network, including 1) system dynamic data, that changes over time, 2) static information, that stays constant and

characterizes the location of each FN, and 3) lateral information from neighbors. The output of each FN includes predicted dynamics and additional lateral information denoting interaction between neighbors. Such interaction, which distinguishes our architecture from others, aims to model the location-sensitive transitions between adjacent FNs and thus enable local context-dependent spatial information propagation.

In many natural phenomenons, data are collected distributively and exhibit heterogeneous properties: each of these distributed locations presents a different view of the natural process at the same time, where each view has its own individual representation space and dynamics. Theoretically, each location may contain information that other locations do not have access to. Therefore, all local views must interact in some way in order to describe the global activity comprehensively and accurately.

How to explicitly encode location information into neural networks is critical in our location-wise forecasting. Inspired by the Transformer that encodes word positions in sentences, we extend the absolute positional encoding to represent grid positions. In particular, let i, j be the desired position in a regular grid, $\vec{\mathbf{p}}^{(i,j)} \in \mathbb{R}^d$ be its corresponding encoding, and d be the encoding dimension, then the encoding scheme is defined as:

$$\vec{\mathbf{p}}^{(i,j)} := \begin{cases} \sin(\omega_k, i), \sin(\omega_k, j) & \text{if } i, j = 2k \\ \cos(\omega_k, i), \cos(\omega_k, j) & \text{if } i, j = 2k + 1 \end{cases} \quad (27)$$

where $\omega_k = \frac{1}{10,000^{2k/d}}, k \in \mathbb{N}_{\leq \lceil \frac{d}{2} \rceil}$. The positional embedding as a vector contains pairs of *sines* and *cosines* for each decreasing frequency along the vector dimension.

As illustrated in Figure 2, at each time t, FN is distributively executed at different locations Ω^n_{temp}. FN encodes each view (i.e., static $\vec{\mathbf{p}}$, dynamics $\mathcal{S}$, and Ł of each node) using a fusion layer:

$$f^{(t,i,j)} = [\vec{\mathbf{p}}^{(i,j)}, \mathcal{S}^{(t,i,j)}, Ł^{(t,i,j)}]\mathcal{W}^T_{fusion} + b_{fusion} \quad (28)$$

where $Ł^{(t,i,j)}$ is a vector used to characterize interaction between node at i, j and its neighbors. It is initialized as zeros at the first step and continuously updated by Eq. (30) when $t > 0$.

These features, $f^{(t,i,j)} \in \mathbb{R}^{d_{f_{i,j}}}$, are then fed into an LSTM to model the node-specific interactions over time. The update mechanism of the LSTM cell is defined as:

$$\left[\mathcal{I}^{(t)}; \mathcal{F}^{(t)}; \tilde{\mathcal{C}}^{(t)}; \mathcal{O}^{(t)} \right] = \sigma \left(\mathcal{W}f^{(t,i,j)} + \mathcal{T}h^{(t-1)} \right)$$
$$\mathcal{C}^{(t)} = \tilde{\mathcal{C}}^{(t)} \circ \mathcal{I}^{(t)}$$
$$h^{(t)} = \mathcal{O}^{(t)} \circ \mathcal{C}^{(t)} \quad (29)$$

where $\sigma(\cdot)$ applies sigmoid on the input gate $\mathcal{I}^{(t)}$, forget gate $\mathcal{F}^{(t)}$, and output gate $\mathcal{O}^{(t)}$, and $tanh(\cdot)$ on memory cell $\tilde{\mathcal{C}}^{(t)}$. The parameters are characterized by $\mathcal{W} \in \mathbb{R}^{d_{f_{i,j}} \times d_{h_{i,j}}}$ and $\mathcal{T} \in \mathbb{R}^{d_{h_{i,j}} \times d_{h_{i,j}}}$, where $d_{h_{i,j}}$ is the output dimension. A cell updates

TABLE 1. Statistics of the datasets.

Data Sets	# of Time Points	Grid Size	Sampling Rate
Reflected Wave	8,000	16×16	0.1s
LC of GoM	1,810	29×36	12h

its hidden states $h^{(t)}$ based on the previous step $h^{(t-1)}$ and the current input $f^{(t,i,j)}$.

An output layer is stacked at the end of FN to transform the LSTM output into the expected dynamic prediction and additional lateral information as:

$$\left[\hat{\mathcal{S}}^{(t+1,i,j)}_{he}; \hat{Ł}^{(t+1,i,j)} \right] = Relu(\mathcal{W}_{out} \cdot f^{(t,i,j)} + b_{out}) \quad (30)$$

where $\hat{\mathcal{S}}^{(t+1,i,j)}_{he}$ denotes the *heterogeneous* prediction of the node dynamics at time step $t + 1$. The learnable parameters are characterized by $\mathcal{W}_{out} \in \mathbb{R}^{d_{f_{i,j}} \times d_{y_{i,j}}}$ and $b_{out} \in \mathbb{R}^{d_{y_{i,j}}}$, where $d_{y_{i,j}}$ denotes the total dimensions of the dynamic and the lateral outputs.

Our proposed ST-PCNN models the heterogeneous properties of spatio-temporal data and meanwhile reveals the homogeneous physics from the data. Here, a stacking coupling mechanism is proposed to integrate the obtained physics into the spatio-temporal learning. As shown in Figure 2, at each time step t and location i, j, the FN produces the initial prediction $\hat{\mathcal{S}}^{(t+1,i,j)}_{he}$ based on the current observation $\mathcal{S}^{(t,i,j)}$, the hidden states $h^{(t-1,i,j)}$ from previous-step (within LSTM), and the lateral info from its neighbors. The *heterogeneous* initial prediction leverages its own specific local attributes only. Regarding the integration of PDE function learnt from the PN, the previous-step initial prediction $\hat{\mathcal{S}}^{(t-1,i,j)}$ and the current observations $\mathcal{S}^{(t,i,j)}$ are fed into the learnt PDE to derive the numerical solution $\hat{\mathcal{S}}^{(t+1,i,j)}_{ho}$, which is the *homogeneous* of the dynamics regularized by governing physics. Finally, a coupling layer with parameters $\theta_{\mathcal{C}} = [\mathcal{W}_{\mathcal{C}}, b_{\mathcal{C}}]$ is used to produce the final prediction $\hat{\mathcal{S}}^{(t+1,i,j)}$ by synthesizing $\hat{\mathcal{S}}^{(t+1,i,j)}_{he}$ and $\hat{\mathcal{S}}^{(t+1,i,j)}_{ho}$ as:

$$\hat{\mathcal{S}}^{(t+1,i,j)} = Relu([\hat{\mathcal{S}}^{(t+1,i,j)}_{he}, \hat{\mathcal{S}}^{(t+1,i,j)}_{ho}]\mathcal{W}^T_{\mathcal{C}} + b_{\mathcal{C}}) \quad (31)$$

In this paper, the ST-PCNN training is presented in Algorithm 2 with supervised loss including sum of l_1-norm and l_2-norm loss.

V. EXPERIMENTS

We present two spatial-temporal prediction tasks: (1) reflected wave prediction; and (2) the Gulf of Mexico (GoM) Loop Current (LC) prediction. Since the two tasks have different strengths of spatial-temporal correlation, as well as different scalability and noise levels, they can verify the generalizations of ST-PCNN with active learning and other baselines over different scenarios.

Specifically, in reflected wave prediction, an flow in a position strongly corresponds to the flow in its neighbour position, which indicates a strong correlation. However, the correlation is weaker and more difficult to measure in GoM
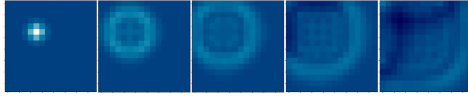
FIGURE 3. Exemplary circular wave with reflecting borders. Plots from left to right denote temporal evolving of the circular wave propagate from the center to boundaries with reflecting effects caused by boundary conditions.

loop current prediction, where the velocity in one position may not indicates changes in other positions.

A. DATASETS

The statistics of the two evaluated datasets are summarized in Table 1 and introduced as below.

Reflected Wave Simulation Data As illustrated in Figure 3, single waves are propagating outwards, where waves are reflected at borders such that wave fronts become interactive. The following 2D wave equation was used for reflected wave data generation:

$$\frac{\partial^2 u}{\partial t^2} = c^2 \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \quad (32)$$

The PDE solutions to generate data were solved numerically using an explicit central difference approach:

$$\frac{\partial^2 u}{\partial b^2} = \frac{u(b+h) - 2u(b) + u(b-h)}{h^2} = u_{bb} \quad (33)$$

where b stands for a variable of function u, and h is the approximation step size. In the case of calculating simulated wave data, we apply Eq. (33) to Eq. (32) to obtain:

$$c^2 (u_{xx} + u_{yy})$$
$$= \frac{u(x, y, t + \Delta t) - 2u(x, y, t) + u(x, y, t - \Delta t)}{\Delta_t^2} \quad (34)$$

which can be solved for $u(x, y, t + \Delta t)$ to obtain an equation for determining state of the field at the next time step $t + \Delta t$ at each point.

Both the boundary conditions (when $x < 0$ or $x > fieldwidth$, analogously for y) and initial condition (in time step 0) are treated as zero. The following variable choices were met: $\Delta_t = 0.1$, $\Delta_x = \Delta_y = 1$ and $c = 3.0$. The field was initialized using a Gaussian distribution:

$$u(x, y, 0) = a exp \left(- \left(\frac{(x - s_x)^2}{2\sigma_x^2} + \frac{(y - s_y)^2}{2\sigma_y^2} \right) \right) \quad (35)$$

with amplitude factor $a = 0.34$, wave width in x and y directions $\sigma_x^2 = \sigma_y^2 = 0.5$, and s_x, s_y being the starting point or center of the circular wave.

Gulf of Mexico Loop Current Data As illustrated in Figure 4, the sensor array are placed in the GoM region, covered from $89°W$ to $85°W$, and $25°N$ to $27°N$ with 30-50 km horizontal resolution, where the LC extended northward and, more importantly, where eddy shedding events occurred most often [46]. This sensor array consisted of
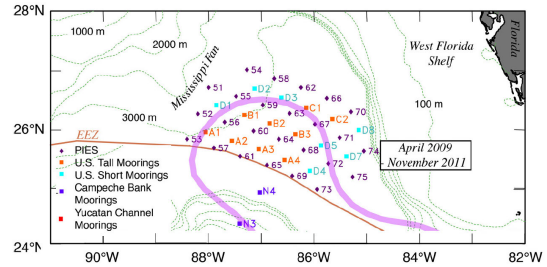

FIGURE 4. Locations of moorings and Pressure-Recording Inverted Echo Sounders (PIES) deployed in the U.S. and Mexican sectors in the eastern GoM.

25 pressure-recording inverted echo sounders (PIES), 9 full-depth tall moorings with temperature, conductivity and velocity measurements, and 7 near bottom current meter moorings deployed under the LC region [46]. The data set contains velocity data gathered from June 2009 to June 2011. Since the sampling frequency from multiple sensors varied from minutes to hours, the dataset was initially processed with a fourth order Butterworth filter and sub-sampled at 12-hours intervals, leading to a total of 1,810 records (905 days). In this experiment, we further subsample the data at 7-days interval.

B. METRICS & BENCHMARK MODELS

We evaluate the models based on the Mean Square Error (MSE). After each active training step, the performance is evaluated by single step prediction on unseen data. Our proposed model is compared by ablation study with the following methodological categories:

1) **Random sampling**: At each data collection step, the positions were randomly selected from Ω at the rate of 10%, 20% and 40%. This is also called passive learning, representing the widely used method in which candidate data points for training a prediction model are chosen at random.

2) **Without physics learning**: The physics network (PN) is disabled in ST-PCNN, denoted as ST-PCNN (remove PN and coupling layer). The prediction is simply produced by the forecasting network (FN).

3) **All data available**: Suppose an optimal case that data from all positions Ω are available, i.e., 100%.

C. IMPLEMENTATION DETAILS

We conduct experiments on a 64-bit Ubuntu 18.04 computer with Intel 3.70GHz and 64GB memory, 2 NVIDIA Quadro RTX 5000 GPUs (16GB DDR6). The ST-PCNN is built on Pytorch. The hyper-parameter setting of ST-PCNN is as follow: The dimension of embedded static vector $\vec{p}$ is set to 4. The FN consists of a fully-connected layer, followed by an LSTM layer with 256 hidden units, and another fully-connected layer. The PN, based on the GP model is developed by GPytorch (a Gaussian process library implemented using PyTorch). In each active step, the training epoch is set to 10, the max iteration steps in GP and in physics learning (mini-
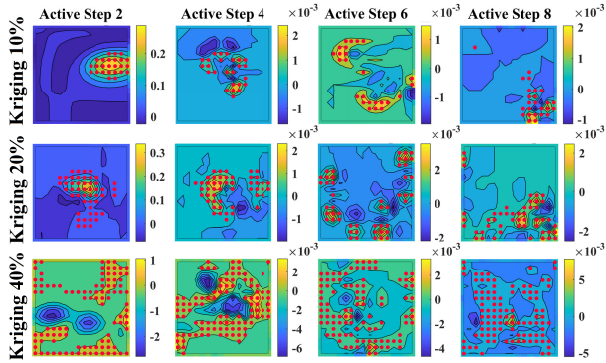
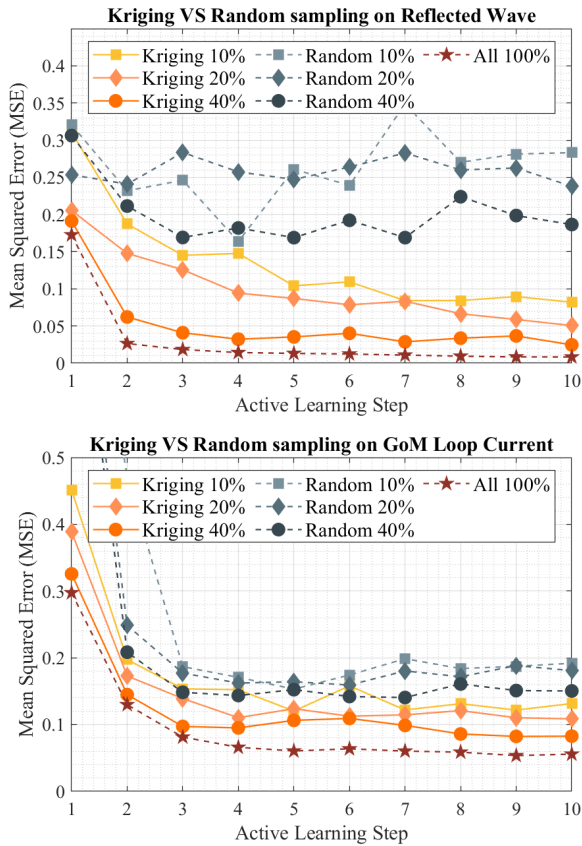**FIGURE 5.** Example of Kriging by $\hat{s}^2$ in active learning on reflected wave.

FIGURE 6. Explore Kriging and Random sampling effect on the performance of ST-PCNN on unseen data.

FIGURE 7. Explore Kriging and Random sampling effect on performance of ST-PCNN on unseen data (the effect on Gulf of Mexico Loop Current refers to Figures 6 and 8).

mizing SSRE) are both set to 200. All the data are normalized within (0, 1). After each active training step for the GoM LC data (a single multivariate time series), the performance is evaluated on the remaining data. For the simulated reflected waves, the model is trained on a single temporal-evolving circular wave propagated from the center while evaluated on 16 circular waves propagated from randomly selected positions.

D. RESULTS AND DISCUSSIONS
1) KRIGING SAMPLING FOR ACTIVE LEARNING RESULTS
In this section, we compare the effect of Kriging sampling-based active learning with random sampling. Our goal is to
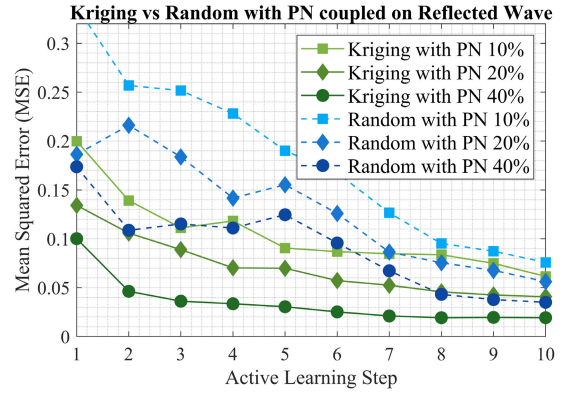
demonstrate whether the proposed active learning approach can effectively identify informative nodes/points and query their values to improve prediction accuracy.

At each data collection period (active learning step), we use Algorithm 1 to perform Kriging by using new observations at Ω_{temp}^n where $\hat{s}^2$ has maximum, which is estimated by the previous step. Figure 5 presents exemplary selected locations (Ω_{temp}^n) by Kriging during active learning. As we can see, the selected locations are expended as the wave flows towards the boundary. The Kriging helps to identify those representative locations to forecast the dynamics as a whole.

In order to understand whether physics learning can indeed help improve the prediction (in combination with the active learning), we carry out an ablation study to understand the interplay between physics network (PN) and active learning, by disabling and enabling PN respectively, and observe the algorithm performance.

In the first experiment, we disable the PN (i.e., ST-PCNN) to observe how sampling affects FN performance. As shown in Figure 6, Kriging outperforms the random sampling in terms of both convergence and MSE. In reflected wave, ST-PCNN with only 10% and 20% sampling does not guarantee convergence in performance, while it converges when the sampling rate rises to 40%. Noticeably, ST-PCNN with Kriging presents a faster convergence and better prediction performance on unseen data with lower MSE. It is worth mentioning that when with a sampling rate of 40%, the performance is close to the optimal case where all data are observable.

In the second experiment, we enable the PN to observe how Kriging gives credits to ST-PCNN. As shown in Figure 7, with the benefits of PN, both Kriging and random sampling present good convergence. The benefit of Kriging is significant in pre-mid active learning. However, the performance of random sampling getting closer to Kriging as active learning goes on, which mainly due to the fact that PN gradually uncovers the underlying hidden physics from observation and thus, is able to assist the spatio-temporal networks to capture the dynamics of reflected waves.
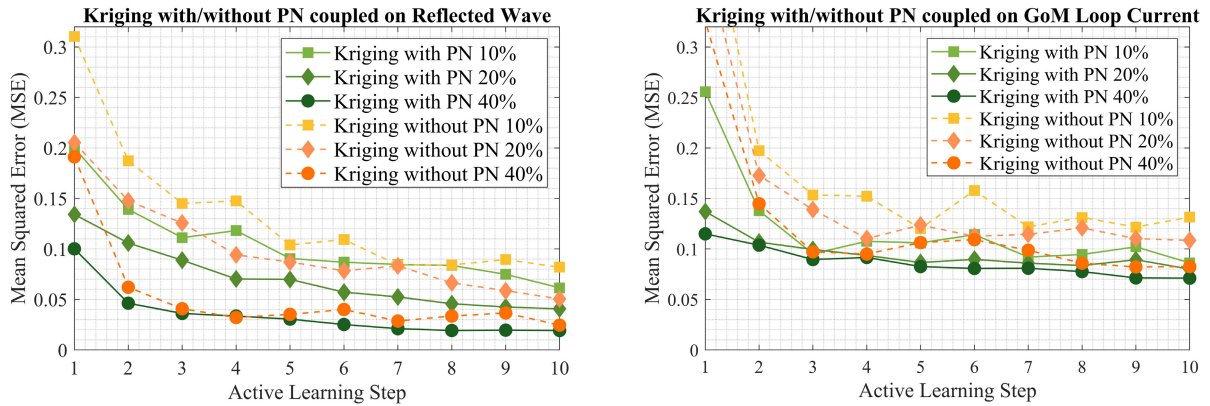
FIGURE 8. Explore physics effect on performance of ST-PCNN (with Kriging) on unseen data.

When comparing Figure 6 (top panel) and Figure 7, we can easily conclude the benefit of using PN for prediction. For example, for the same sampling rate (such as 10%), the MSE after one active learning step is 0.3 in Figure 6 (without PN), where in Figure 7, the MSE is about 0.2. This means that PN helps the model gain more accurate predictions. After 10 active learning steps, the MSE without PN is about 0.09, whereas the one using PN is about 0.06.

2) PHYSICS LEARNING RESULTS

In this section, we further explore the benefits of PN in active learning. To begin with, we consider the optimal case where all data are observed. In this case, the Kriging for active learning is no longer needed (because all nodes are observed). For fair comparisons, we keep the same active learning step and denote by sampling & learning step in Table 2. Because all points are observed in each step, the contribution of PN can be directly observed by including *vs.* excluding PN in the learning.

In Table 2, we report the results from the above experiments. ST-PCNN▲ with wave equation (Eq. (32)) informed is regarded as a reference here, proving that if the governing physics is known, our model could perfectly capture the spatio-temporal dynamics. The MSE score of ST-PCNN is better than that of ST-PCNN at any period, indicating that ST-PCNN (with physics learning) can capture both *homogeneity* (underlying physics) and *heterogeneity* (localized information) in modeling the evolution and dynamics of the observing system.

Then, we explore how PN gives credits to the ST-PCNN model with partially observed data. In the random sampling, by comparing the grey curves in Figure 6 and blue curves in Figure 7, PN boosts the prediction accuracy with fast convergence by uncovering the hidden physics from randomly collected data. Furthermore, from Figure 7, we found that random sampling approximates to Kriging as time goes on. The reason lies in that, during the active learning, as new data being appended to the existing training set at each step, the training set is growing in the time dimension (since location

TABLE 2. Evaluation of physics effect on performance.

Data	Model	Physics	Sampling & Learning Step				
			2	4	6	8	10
Wave	ST-PCNN▲	Eq. (32)	0.0075	0.0034	0.0019	$8.73e^{-5}$	$5.27e^{-6}$
	ST-PCNN	self-learnt	0.0192	0.0088	0.0083	0.0066	0.0057
	ST-PCNN	-	0.0263	0.0142	0.0122	0.0093	0.0082
GoM	ST-PCNN	self-learnt	0.0802	0.0622	0.0410	0.0434	0.0420
	ST-PCNN	-	0.1298	0.0659	0.0636	0.0586	0.0556

number is fixed). The PDE learned by PN is approaching the real one by leveraging the ever-growing dataset. The advantage of Kriging is reduced as ST-PCNN gradually captures the wave dynamics by leveraging the PDE learning. This can also be found in Figure 8, which directly shows the ST-PCNN performance with *vs.* without PN, by using different sampling ratios. In Figure 8, we do not observe significant differences of performance in the late stage. This observation asserts that Kriging helps ST-PCNN capture the dynamics of the system, which is obvious when data are limited, resulting in lower MSE and faster convergence in the early stage.

VI. CONCLUSION

In this paper, we proposed a physics-coupled active learning framework for the accurate prediction of spatio-temporal dynamical systems. The essential goal is to query only a small subset of the observable locations to learn the physics underlying the system and predict future values of the system with minimum errors. We argued that real-world observing systems are often challenged by *heterogeneity* and *homogeneity*. The proposed framework, ST-PCNN, consists of two neural networks, a forecasting network (FN) and a physics network (PN). The FN is learned spatially across all locations, allowing interaction with neighbors, to learn local *heterogeneity*. The PN, on the other hand, is a Gaussian process-based model to uncover *homogeneity* of the system (i.e., estimating the physics parameters). Active learning with Kriging is employed to identify locations for representative observations that minimize the prediction error and reduce MSE with

scarce data and query constraints. We validated ST-PCNN on both synthetic and real-world datasets. The results show that ST-PCNN can provide accurate spatio-temporal predictions for dynamical systems, by querying only a small portion (40% or less) of observable points.

The proposed method exists some potential drawbacks. For example, when implementing active learning using Kriging, the query of locations to get new observations is mainly decided by the estimated maximum errors. Due to the limited number of locations that can query, it is possible that some important locations are not selected, and the information from those locations will not be learned by the model. Information entropy-based approach [47] and attention-based mechanism [48] are able to address this issue and will be tested in future works.

REFERENCES

[1] Y. Li, W. Zheng, L. Wang, Y. Zong, and Z. Cui, "From regional to global brain: A novel hierarchical spatial–temporal neural network model for EEG emotion recognition," *IEEE Trans. Affect. Comput.*, vol. 13, no. 2, pp. 568–578, Apr. 2019.

[2] R. Li, S. Lv, H. Zhu, and X. Song, "Spatial–temporal topic model for cold-start event recommendation," *IEEE Access*, vol. 8, pp. 214050–214060, 2020.

[3] Q. Liu, X. Zheng, H. E. Stanley, F. Xiao, and W. Liu, "A spatio–temporal co-clustering framework for discovering mobility patterns: A study of Manhattan taxi data," *IEEE Access*, vol. 9, pp. 34338–34351, 2021.

[4] Y. Liu, J. Chen, A. Ganguly, and J. Dy, "Nonparametric mixture of sparse regressions on spatio–temporal data–an application to climate prediction," in *Proc. ACM SIGKDD Conf.*, 2019, pp. 2556–2564.

[5] Y. Karadayi, M. N. Aydin, and A. S. Ogrenci, "Unsupervised anomaly detection in multivariate spatio–temporal data using deep learning: Early detection of COVID-19 outbreak in Italy," *IEEE Access*, vol. 8, pp. 164155–164177, 2020.

[6] D. Wang, M. Jiang, M. Syed, O. Conway, V. Juneja, S. Subramanian, and N. V. Chawla, "Calendar graph neural networks for modeling time structures in spatiotemporal user behaviors," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 2581–2589.

[7] R. Dai, S. Xu, Q. Gu, C. Ji, and K. Liu, "Hybrid spatio-temporal graph convolutional network: Improving traffic prediction with navigation data," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 3074–3082.

[8] T. Li, J. Zhang, K. Bao, Y. Liang, Y. Li, and Y. Zheng, "AutoST: Efficient neural architecture search for spatio–temporal prediction," in *Proc. 26th ACM SIGKDD Conf*, 2020, pp. 794–802.

[9] B. Yu, H. Yin, and Z. Zhu, "Spatio–temporal graph convolutional networks: A deep learning framework for traffic forecasting," 2017, *arXiv:1709.04875*.

[10] S. Seo and Y. Liu, "Differentiable physics-informed graph networks," 2019, *arXiv:1902.02950*.

[11] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, B. B. Gupta, X. Chen, and X. Wang, "A survey of deep active learning," 2020, *arXiv:2009.00236*.

[12] Y. Zhao, L. Ye, P. Pinson, Y. Tang, and P. Lu, "Correlation-constrained and sparsity-controlled vector autoregressive model for spatio–temporal wind power forecasting," *IEEE Trans. Power Syst.*, vol. 33, no. 5, pp. 5029–5040, Sep. 2018.

[13] S. Lee, S. Iyengar, M. Feng, P. Shenoy, and S. Maji, "DeepRoof: A data-driven approach for solar potential estimation using rooftop imagery," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 2105–2113.

[14] B. Settles, "Active learning literature survey," Dept. Comput. Sci., Univ. Wisconsin-Madison, 2009. [Online]. Available: https://minds.wisconsin.edu/handle/1793/60660

[15] R. D. King, K. E. Whelan, F. M. Jones, P. G. Reiser, C. H. Bryant, S. H. Muggleton, D. B. Kell, and S. G. Oliver, "Functional genomic hypothesis generation and experimentation by a robot scientist," *Nature*, vol. 427, no. 6971, pp. 247–252, 2004.

[16] V. Krishnamurthy, "Algorithms for optimal scheduling and management of hidden Markov model sensors," *IEEE Trans. Signal Process.*, vol. 50, no. 6, pp. 1382–1397, Jun. 2002.

[17] M. Tsiakmaki, G. Kostopoulos, S. Kotsiantis, and O. Ragos, "Fuzzy-based active learning for predicting Student academic performance," in *Proc. 6th Int. Conf. Eng. MIS*, Sep. 2020, pp. 1–6.

[18] S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Data-driven discovery of partial differential equations," *Sci. Adv.*, vol. 3, no. 4, Apr. 2017, Art. no. e1602614.

[19] J. Berg and K. Nyström, "Data-driven discovery of PDEs in complex datasets," *J. Comput. Phys.*, vol. 384, pp. 239–252, May 2019.

[20] H. Xu, H. Chang, and D. Zhang, "DL-PDE: Deep-learning based data-driven discovery of partial differential equations from discrete and noisy data," 2019, *arXiv:1908.04463*.

[21] Z. Long, Y. Lu, X. Ma, and B. Dong, "PDE-Net: Learning PDEs from data," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 3208–3216.

[22] P. Hu, W. Yang, Y. Zhu, and L. Hong, "Revealing hidden dynamics from time-series data by ODENet," 2020, *arXiv:2005.04849*.

[23] A. Aryandoust and S. Pfenninger, "Active machine learning for spatio–temporal predictions using feature embedding," 2020, *arXiv:2012.04407*.

[24] J. L. Pearce, M. Rathbun, M. Aguilar-Villalobos, and L. P. Naeher, "Characterizing the spatiotemporal variability of PM2. 5 in Cusco, Peru using Kriging with external drift," *Atmospheric Environ.*, vol. 43, no. 12, pp. 2060–2069, 2009.

[25] R. Faghihpirayesh, T. Imbiriba, M. Yarossi, E. Tunik, D. Brooks, and D. Erdoğmuş, "Motor cortex mapping using active Gaussian processes," in *Proc. 13th ACM Int. Conf. Pervasive Technol. Rel. to Assistive Environments*, Jun. 2020, pp. 1–7.

[26] I. Mondal and D. Ganguly, "ALEX: Active learning based enhancement of a model's explainability," 2020, *arXiv:2009.00859*.

[27] V. V. Meduri, L. Popa, P. Sen, and M. Sarwat, "A comprehensive benchmark framework for active learning methods in entity matching," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, Jun. 2020, pp. 1133–1147.

[28] Y. Fu, X. Zhu, and B. Li, "A survey on instance selection for active learning," *Knowl. Inf. Syst.*, vol. 35, no. 2, pp. 249–283, 2013.

[29] A. Agrawal, S. Tripathi, and M. Vardhan, "Active learning approach using a modified least confidence sampling strategy for named entity recognition," *Prog. Artif. Intell.*, vol. 10, no. 2, pp. 113–128, 2021.

[30] A. Culotta and A. McCallum, "Reducing labeling effort for structured prediction tasks," in *Proc. AAAI*, vol. 5, 2005, pp. 746–751.

[31] D. Pereira-Santos, R. B. C. Prudêncio, and A. C. P. L. F. de Carvalho, "Empirical investigation of active learning strategies," *Neurocomputing*, vols. 326–327, pp. 15–27, Jan. 2019.

[32] T. Scheffer, C. Decomain, and S. Wrobel, "Mining the web with active hidden Markov models," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2001, pp. 645–646.

[33] M.-F. Balcan, A. Broder, and T. Zhang, "Margin based active learning," in *Proc. Int. Conf. Comput. Learn. Theory*. Springer, 2007, pp. 35–50.

[34] Y. Siddiqui, J. Valentin, and M. Nießner, "ViewAL: Active learning with viewpoint entropy for semantic segmentation," in *Proc. IEEE/CVF Conf Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 9433–9443.

[35] Z. Qiu, D. J. Miller, and G. Kesidis, "A maximum entropy framework for semisupervised and active learning with unknown and label-scarce classes," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 4, pp. 917–933, Apr. 2016.

[36] B. Settles and M. Craven, "An analysis of active learning strategies for sequence labeling tasks," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2008, pp. 1070–1079.

[37] O. Sener and S. Savarese, "Active learning for convolutional networks: A core-set approach," 2017, *arXiv:1708.00489*.

[38] X. Li, R. Guo, and J. Cheng, "Incorporating incremental and active learning for scene classification," in *Proc. 11th Int. Conf. Mach. Learn. Appl.*, vol. 1, Dec. 2012, pp. 256–261.

[39] S.-J. Huang, R. Jin, and Z.-H. Zhou, "Active learning by querying informative and representative examples," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 23, 2010, pp. 892–900.

[40] X. Yang, G. Tartakovsky, and A. Tartakovsky, "Physics-information-aided Kriging: Constructing covariance functions using stochastic simulation models," 2018, *arXiv:1809.03461*.

[41] Q. Guo, Y. Liu, B. Chen, and Q. Yao, "A variable and mode sensitivity analysis method for structural system using a novel active learning Kriging model," *Rel. Eng. Syst. Saf.*, vol. 206, Feb. 2021, Art. no. 107285.

[42] A. Sobester, A. Forrester, and A. Keane, *Engineering Design Via Surrogate Modelling: A Practical Guide*. Hoboken, NJ, USA: Wiley, 2008.

[43] C. J. Arthurs and A. P. King, "Active training of physics-informed neural networks to aggregate and interpolate parametric solutions to the navier-stokes equations," 2020, *arXiv:2005.05092*.

[44] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Inferring solutions of differential equations using noisy multi-fidelity data," *J. Comput. Phys.*, vol. 335, pp. 736–746, Apr. 2017.

[45] D. Zhang, L. Lu, L. Guo, and G. E. Karniadakis, "Quantifying total uncertainty in physics-informed neural networks for solving forward and inverse stochastic problems," *J. Comput. Phys.*, vol. 397, Nov. 2019, Art. no. 108850.

[46] P. Hamilton, A. Lugo-Fernández, and J. Sheinbaum, "A loop current experiment: Field and remote measurements," *Dyn. Atmos. Oceans*, vol. 76, pp. 156–173, Dec. 2016.

[47] D. W. Chen and Y. H. Jin, "An active learning algorithm based on Shannon entropy for constraint-based clustering," *IEEE Access*, vol. 8, pp. 171447–171456, 2020.

[48] D. Liu, Y. Zhang, J. Zhang, Q. Li, C. Zhang, and Y. Yin, "Multiple features fusion attention mechanism enhanced deep knowledge tracing for student performance prediction," *IEEE Access*, vol. 8, pp. 194894–194903, 2020.

[49] E. Solak, R. Murray-Smith, W. E. Leithead, D. Leith, and C. Rasmussen, "Derivative observations in Gaussian process models of dynamic systems," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 15, 2002.

[50] P. K. Rai and S. Tripathi, "Gaussian process for estimating parameters of partial differential equations and its application to the Richards equation," *Stoch Environ Res Risk Assess*, vol. 33, no. 8, pp. 1629–1649, 2019.

YU HUANG (Student Member, IEEE) received the B.S. and M.S. degrees in manufacturing engineering of aeronautics and astronautics from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2015 and 2018, respectively. He is currently pursuing the Ph.D. degree in electrical engineering with Florida Atlantic University, Boca Raton, FL, USA.

His research interests include prognostics and health management, artificial intelligence, and applications in complex dynamical systems.

YUFEI TANG (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from the University of Rhode Island, Kingston, RI, USA, in 2016. He is currently an Associate Professor with the Department of Electrical Engineering and Computer Science; and a Faculty Fellow with the Institute for Sensing and Embedded Network Systems Engineering, Florida Atlantic University, Boca Raton, FL, USA. His research interests include machine learning, data mining, dynamic systems, and renewable energy.

He was a recipient of the IEEE International Conference on Communications Best Paper Award, in 2014; the National Academies Gulf Research Program Early-Career Research Fellowship, in 2019; and the National Science Foundation CAREER Award, in 2022.

XINGQUAN ZHU (Senior Member, IEEE) received the Ph.D. degree in computer science from Fudan University, Shanghai, China. He is currently a Full Professor with the Department of Electrical Engineering and Computer Science, Florida Atlantic University, Boca Raton, FL, USA. His research interests include data analytics, machine learning, and bio-informatics. Since 2000, he has been authored or coauthored more than 280 refereed journal and conference papers in these areas. He received the four Best Paper Awards, such as PAKDD-21, IRI-18, and PAKDD-13; the Student Paper Awards, such as ICDM-20, ICKG-20, and ICTAI-05; and the three best and ICPR-12. He was an Associate Editor of the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, from 2008 to 2012 and from 2014 to 2021. Since 2017, he has been an Associate Editor of the *ACM Transactions on Knowledge Discovery*.

HANQI ZHUANG (Senior Member, IEEE) received the B.S. degree in electrical engineering from Shanghai University, Shanghai, China, and the M.S. and Ph.D. degrees in electrical engineering from Florida Atlantic University, Boca Raton, FL, USA.

He is currently the Chair at the Electrical Engineering and Computer Science Department, Florida Atlantic University. He is also working on research projects ranging from marine animal detection and classification from acoustic signals to modeling and prediction of the Loop Current System in the Gulf of Mexico. His research interests include acoustic signal processing, deep neural networks, and their applications.

LAURENT CHERUBIN received the B.S. degree in fluid mechanics from the University of Bordeaux—I, France, and the M.S. and Ph.D. degrees in physical and coastal oceanography from the University of Méditerranée, Marseille, France.

He is currently an Associate Research Professor at the Harbor Branch Oceanographic Institute, Florida Atlantic University. He is a physical oceanographer specialized in the understanding of ocean dynamics, which is the study of why the water moves the way it moves. His research interests include dynamics of motions associated with instabilities in coastal currents and eddies, using both analytical and numerical models in the quasi-geostrophic, shallow-water formalisms, and in realistic models. This research provides a deep understanding of the environmental forces that affect ocean ecosystems at multiples levels of the trophic chain. Both observational analysis and numerical modeling involving hydrodynamic (the Regional Oceanic Modeling System (ROMS)) and biophysical models (Connectivity Modeling System (CMS) and Ichthyop) are used to study how environmental drivers shape the oceanic ecosystems.

• • •