2024 IEEE 63rd Conference on Decision and Control (CDC)
December 16-19, 2024. MiCo, Milan, Italy

A Surrogate Prediction Model for Systems Governed by Partial Differential Equations*

Wei Kang[1] Liang Xu[2] and Hong Zhou[3]

Abstract— **This work is dedicated to the real-time prediction of dynamical system trajectories using sensor data. Our approach introduces a learning-based surrogate prediction model tailored for forecasting the state of partial differential equations (PDEs) within a limited area. Different from existing learning based methods of solving PDEs, the prediction is made based on observation data, without the necessity of knowing the initial condition and precise lateral boundary conditions for the limited-area model, both in online and offline computations. The design of our surrogate prediction model hinges on two pivotal concepts: predictability and effective region. Predictability enables us to quantitatively assess whether the observation data is sufficient for accurate prediction. Concurrently, the effective region concept decreases the computational burden associated with determining predictability and generating training data. Compared to the conventional two-stage approach—first employing data assimilation followed by prediction through differential equation integration—commonly utilized in control systems and numerical weather prediction, our surrogate prediction model offers real-time forecasting in a single step, namely, evaluating a neural network.**

I. Introduction

Real-time prediction of dynamical system trajectories based on sensor data stands as a crucial element in a diverse array of scientific and engineering applications. These applications span domains such as model predictive control [1], real-time or faster-than-real-time simulation for large-scale power systems to predict dynamics under disturbances [2], and numerical weather prediction [3]–[5]. Over the years, a conventional approach is to combine a nonlinear filter, such as a Kalman filter or a Luenberger observer, with a numerical model of the dynamical system. The nonlinear filter furnishes a state estimate grounded in the system's output, notably sensor data. Subsequently, this estimated state serves as the initial condition within the numerical model, facilitating the prediction of system trajectories within a specified time window. While these methods have found widespread and successful applications in various engineering and scientific domains, they have drawbacks. The challenges have grown significantly, especially in the face of the continuously escalating scale and complexity of systems under study, including but not limited to networked systems and distributed parameter systems. In estimating a system's states, existing algorithms for nonlinear filters require running the numerical model of the dynamical system multiple times in each estimation cycle. Additionally, the prediction is computed by integrating the model over the forward time window. For high-dimensional systems, these computational tasks become bottlenecks that can render conventional approaches infeasible.

In this paper, we introduce a learning-based surrogate prediction model for state estimation in a limited-area in space for dynamical systems governed by PDEs. The surrogate model employs deep learning techniques, where a neural network is trained using sensor data as inputs and facilitating the prediction of the system's states as outputs. Compared to the conventional two-stage approach—first employing data assimilation followed by prediction through differential equation integration—commonly utilized in control systems and numerical weather prediction, our surrogate prediction model offers real-time forecasting in a single step, namely, evaluating a neural network. This approach significantly reduces online computational burden as evaluating the surrogate prediction model does not require propagating the dynamic model in real-time. Unlike numerical PDE algorithms that require boundary conditions, evaluating neural networks does not have this limit. This property makes the learning-based method attractive for applications in which prediction is focused on variables in a limited area of interest around which the lateral boundary condition [6] is unknown. Different from existing learning based methods of solving PDEs [7], the surrogate prediction model is based on observation data, without the necessity of knowing the initial condition.

The design of the surrogate prediction model is built upon a robust theoretical framework that leverages two fundamental concepts: predictability and effective region. Predictability analysis enables us to quantitatively determine the amount of observation data necessary for accurate prediction. The effective region substantially reduces the computational burden associated with computing predictability and generating data for neural network training and validation.

II. Predictability

The concept of predictability is defined for discrete-time dynamical systems. It will be applied to discretized PDEs in the sections that follow. Consider a discrete-time dynamical

*This work was supported in part by U.S. Naval Research Laboratory - Monterey, CA, Air Force Office of Scientific Research # F4FGA03243G001 and National Science Foundation # 2202668.

[1,3]Wei Kang and Hong Zhou are with Faculty of Applied Mathematics, Naval Postgraduate School, Monterey, California, USA wkang@nps.edu, hzhou@nps.edu

[2]Liang Xu is with the Marine Meteorology Division, Naval Research Laboratory, Monterey, California, USA liang.xu.civ@us.navy.mil

system

$$\begin{aligned}
\mathbf{u}(k+1) &= \mathbf{f}(\mathbf{u}(k)), && \mathbf{u} \in \mathbb{R}^n, \quad k \in \mathbb{N}_0, && \text{(1a)}\\
\mathbf{y}(k) &= \mathbf{h}(\mathbf{u}(k)) + \boldsymbol{\nu}(k), && \mathbf{y} \in \mathbb{R}^m, && \text{(1b)}\\
z(k) &= w(\mathbf{u}(k)), && z \in \mathbb{R}, && \text{(1c)}
\end{aligned}$$

where $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$, $\mathbf{h} : \mathbb{R}^n \to \mathbb{R}^m$ and $w : \mathbb{R}^n \to \mathbb{R}$ are Lipschitz continuous functions, $\mathbf{u}$ is the state variable of the system, $\mathbf{y}$ is the output, or observation, whose value is known or can be measured, $\boldsymbol{\nu}$ is the observation uncertainty, z is the variable to be estimated. In this section, we develop a theoretical framework to answer the following question: given an interval $[0, K]$, the system's output $\{\mathbf{y}(k)\}_{k=0}^K$ and an integer $\Delta K > 0$, is it possible to accurately approximate the value of $z(K + \Delta K)$?

Inspired by the definition of observability in [8], [9], we first consider the following linear system. Nonlinear systems are addressed in III.

$$\begin{aligned}
\mathbf{u}(k+1) &= A\mathbf{u}(k), && \mathbf{u} \in \mathbb{R}^n, \quad k \in \mathbb{N}_0, && \text{(2a)}\\
\mathbf{y}(k) &= H\mathbf{u}(k) + \boldsymbol{\nu}(k), && \mathbf{y} \in \mathbb{R}^m, && \text{(2b)}\\
z(k) &= W\mathbf{u}(k), && z \in \mathbb{R}, && \text{(2c)}
\end{aligned}$$

where A, H, and W are matrices and vectors of appropriate dimensions. The predictability of $z(K + \Delta K)$ is measured by the value of ρ defined as follows. Consider a nominal trajectory, $\mathbf{u}(0), \cdots, \mathbf{u}(K + \Delta K)$, around which we study the predictability. Define

$$\rho^2 = \max_{\hat{\mathbf{u}}(0),\cdots,\hat{\mathbf{u}}(K+\Delta K)} \{(W\hat{\mathbf{u}}(K + \Delta K) - z(K + \Delta K))^2\}, \tag{3a}$$

subject to

$$\hat{\mathbf{u}}(k+1) = A\hat{\mathbf{u}}(k), \qquad k = 0, 1, \cdots, K + \Delta K - 1, \tag{3b}$$

$$\sum_{k=0}^{K} \|H\hat{\mathbf{u}}(k) - H\mathbf{u}(k)\|_2^2 \le \epsilon^2. \tag{3c}$$

In this definition, $\epsilon > 0$ is a constant that represents an upper bound of observation uncertainty. A trajectory, $\hat{\mathbf{u}}(0), \cdots, \hat{\mathbf{u}}(K + \Delta K)$, satisfying (3c) is said to be *indistinguishable* because its output $\hat{\mathbf{y}}$ is too close to that of the nominal trajectory. Its prediction, $W\hat{\mathbf{u}}(K + \Delta K)$, is called an *indistinguishable prediction*. In this definition, the value of ρ represents the maximum potential error in predicting $z(K + \Delta K)$ among all indistinguishable predictions.

A trajectory of the dynamical system can be uniquely determined by its initial state $\hat{\mathbf{u}}_0$,

$$\hat{\mathbf{u}}(k) = A^k \hat{\mathbf{u}}_0,\ H\hat{\mathbf{u}}(k) = HA^k\hat{\mathbf{u}}_0. \tag{4}$$

Substituting (4) into (3) yields a quadratically constrained quadratic programming (QCQP),

$$\rho^2 = \max_{\Delta\mathbf{u}_0} \Delta\mathbf{u}_0^\intercal (W(K + \Delta K))^\intercal W(K + \Delta K)\Delta\mathbf{u}_0, \tag{5a}$$

subject to

$$\Delta\mathbf{u}_0^\intercal G \Delta\mathbf{u}_0 = \epsilon^2, \tag{5b}$$

where $W(k) = WA^k$ and G is the observability Gramian,

$$G = \sum_{k=0}^{K} (A^\intercal)^k H^\intercal H A^k. \tag{6}$$

The computation of predictability boils down to solving the QCQP (5). If the observability Gramian has full rank, the solution of (5) has an explicit expression.

Theorem 1: Let $\sigma_1, \sigma_2, \cdots, \sigma_n$ be the eigenvalues of G, the observability Gramian defined in (6). Let $T \in \mathbb{R}^{n\times n}$ be a matrix in which the ith column is a unit eigenvector associated with σ_i, $1 \le i \le n$. Suppose that G has full rank. Then

$$\rho^2 = \epsilon^2 \sum_{i=1}^{n} \frac{\bar{w}_i^2}{\sigma_i}, \tag{7}$$

where $\bar{w}_i$ is the ith component of the vector

$$\bar{W} = W(K + \Delta K)T. \tag{8}$$

Proof. This result generalizes a theorem in [9] in which $\Delta K = 0$. To solve (5), define the Lagrangian

$$\begin{aligned}
L(\Delta\mathbf{u}_0, \lambda) = \Delta\mathbf{u}_0^\intercal (W(K + \Delta K))^\intercal W(K + \Delta K)\Delta\mathbf{u}_0\\
-\lambda(\Delta\mathbf{u}_0^\intercal G\Delta\mathbf{u}_0 - \epsilon^2),
\end{aligned}$$

where $\lambda \in \mathbb{R}$ is the Lagrange multiplier. At the solution of (5), denoted by $\Delta\mathbf{u}_0^*$, there exists λ^* such that

$$\frac{\partial L}{\partial \Delta\mathbf{u}_0}(\Delta\mathbf{u}_0^*, \lambda^*) = 0.$$

Therefore, we have

$$(W(K + \Delta K))^\intercal W(K + \Delta K)\Delta\mathbf{u}_0^* = \lambda^* G\Delta\mathbf{u}_0^*, \tag{9a}$$

$$(\Delta\mathbf{u}_0^*)^\intercal G\Delta\mathbf{u}_0^* = \epsilon^2. \tag{9b}$$

Multiplying $(\Delta\mathbf{u}_0^*)^\intercal$ to (9a), we have $\rho^2 = \lambda^*(\Delta\mathbf{u}_0^*)^\intercal G\Delta\mathbf{u}_0^*= \lambda^*\epsilon^2$. Therefore,

$$\lambda^* = \rho^2/\epsilon^2. \tag{10}$$

Because T is the matrix of unit eigenvectors and G is symmetric, we have

$$T^\intercal G T = diag\left(\begin{bmatrix} \sigma_1 & \sigma_2 & \cdots & \sigma_n \end{bmatrix}\right).$$

Multiplying $T^\intercal$ to (9a), applying $\bar{W}$ in (8) and the fact that $T^\intercal$ is the inverse matrix of T, yield

$$\bar{W}^\intercal \bar{W} T^\intercal \Delta\mathbf{u}_0^* = \lambda^* diag\left(\begin{bmatrix} \sigma_1 & \sigma_2 & \cdots & \sigma_n \end{bmatrix}\right) T^\intercal \Delta\mathbf{u}_0^*.$$

Because $\bar{W}T^\intercal\Delta\mathbf{u}_0^*$ is a scalar, we have

$$\lambda^* T^\intercal \Delta\mathbf{u}_0^* = (\bar{W}T^\intercal\Delta\mathbf{u}_0^*)\begin{bmatrix} \dfrac{\bar{w}_1}{\sigma_1} & \cdots & \dfrac{\bar{w}_n}{\sigma_n} \end{bmatrix}^\intercal.$$

Multiplying this equation by $\bar{W}$, we have

$$\lambda^* \bar{W} T^\intercal \Delta\mathbf{u}_0^* = (\bar{W}T^\intercal\Delta\mathbf{u}_0^*)\bar{W}\begin{bmatrix} \dfrac{\bar{w}_1}{\sigma_1} & \cdots & \dfrac{\bar{w}_n}{\sigma_n} \end{bmatrix}^\intercal. \tag{11}$$

If $\bar{W}T^\intercal\Delta\mathbf{u}_0^* \neq 0$, then (7) is proved by cancelling the nonzero scalar $\bar{W}T^\intercal\Delta\mathbf{u}_0^*$ in (11) and applying (10). If $\bar{W}T^\intercal\Delta\mathbf{u}_0^* = 0$, then

$$0 = (\bar{W}T^\intercal\Delta\mathbf{u}_0^*)^2 = (W(K + \Delta K)\Delta\mathbf{u}_0^*)^2 = \rho^2. \tag{12}$$

Because $\rho = 0$, (5) implies $W(K + \Delta K) = 0$ and $\bar{W} = 0$. Therefore, (7) holds true.

Remark: For limited-area prediction to be addressed in the next section, the sensors are located around the region of interest. As a result, the collected data may not necessarily contain adequate information to estimate the full dynamical system. In this case, G either does not have full rank or it has a high condition number. To overcome the problem caused by the singularity of G, we have to take additional information into consideration. Following the same idea presented in [9], let $\delta > 0$ be a known error upper bound of the initial state in the estimation, i.e. $||\Delta \mathbf{u}_0||_2 < \delta$. It is explained in [9] that the observability Gramian can be modified, denoted by G_δ, which has full rank and

$$\rho^2 = \sum_{i=1}^{n} \bar{w}_i^2 \min\{\frac{\epsilon^2}{\sigma_i}, \delta^2\}. \tag{13}$$

III. Nonlinear systems and effective region

The surrogate prediction model to be introduced in the next section is a neural network trained using simulation data. Before we can generate data for training and validation, we must overcome two obstacles. Firstly, the dynamical systems that we are interested in are nonlinear. However, the results in Section II are all based on linear systems. Secondly, generating data using the simulation of PDEs can be computationally formidable because the discretized system has a high dimension. In the following, we introduce an empirical observability Gramian that is applicable to nonlinear systems and the concept of effective region that leads to a reduced order model for the purpose of generating data.

For nonlinear systems, we still use (13) to compute ρ as a measure of observability. We assume that the observability of the linearized system along the nominal trajectory approximates the observability of the nonlinear system. Therefore, the computed ρ represents only the local observability. To approximate G for the nonlinear system, we use the empirical observability Gramian, which has been studied in a variety of applications [9]–[11]. Essentially, the empirical observability Gramian is the observability Gramian of the linearized system along the nominal trajectory. Consider a nonlinear system (1). Adopting the formulation in [9], let $\{\mathbf{u}(k)\}_{k=0}^{K}$ be a nominal solution around which we analyze its observability. Let $h > 0$ be a small number, let $\{\hat{\mathbf{u}}_i(k)\}_{k=0}^{K}$ be the solution of (1) with initial state $\hat{\mathbf{u}}_i(0) = \mathbf{u}(0) + h\mathbf{e}_i$, where $\mathbf{e}_i$ is the ith unit vector in $\mathbb{R}^n$. We define the empirical observability Gramian, G, as follows

$$\begin{aligned}
&D_i\mathbf{y}(k) = (\mathbf{h}(\hat{\mathbf{u}}_i(k)) - \mathbf{h}(\mathbf{u}(k)))/h, \;\; i \in \mathcal{I},\\
&D\mathbf{y}(k) = \left[\begin{array}{cccc} D_{i_1}\mathbf{y}(k) & D_{i_2}\mathbf{y}(k) & \cdots & D_{i_{n_\mathcal{I}}}\mathbf{y}(k) \end{array}\right],\\
&G = \sum_{k=0}^{K} (D\mathbf{y}(k))^\intercal D\mathbf{y}(k),
\end{aligned} \tag{14}$$

where $\mathcal{I}$ is a subset of $\{1, 2, \cdots, n\}$ and $n_\mathcal{I}$ is the number of elements in $\mathcal{I}$. The empirical observability Gramian is a $n_\mathcal{I} \times n_\mathcal{I}$ matrix. If $n_\mathcal{I} = n$, then G has full size, $n \times n$. The matrix $W(K + \Delta K)$ in (8) is computed using finite difference,

$$\begin{aligned}
&D_i w = (w(\mathbf{u}_i(K + \Delta K)) - w(\mathbf{u}(K + \Delta K)))/h,\\
&W(K + \Delta K) = \left[\begin{array}{ccc} D_{i_1} w & \cdots & D_{i_{n_\mathcal{I}}} w \end{array}\right], i_j \in \mathcal{I}.
\end{aligned} \tag{15}$$

Based on the empirical observability Gramian defined in (14), we evaluate the predictability of z by computing ρ using (13).

For high dimensional problems, we choose an index subset $\mathcal{I}$ so that $n_\mathcal{I} < n$ to reduce the computational load. This is based on the concept of the effective region, which is illustrated using the following example. Consider the Burgers equation in $\mathbb{R}^2$,

$$\begin{aligned}
&\frac{\partial U}{\partial t} + U\frac{\partial U}{\partial x_1} + V\frac{\partial U}{\partial x_2} = \kappa\left(\frac{\partial^2 U}{\partial x_1^2} + \frac{\partial^2 U}{\partial x_2^2}\right),\\
&\frac{\partial V}{\partial t} + U\frac{\partial V}{\partial x_1} + V\frac{\partial V}{\partial x_2} = \kappa\left(\frac{\partial^2 V}{\partial x_1^2} + \frac{\partial^2 V}{\partial x_2^2}\right),
\end{aligned} \tag{16}$$

where $(x_1, x_2) \in [0, 2\pi] \times [0, 2\pi]$, $t \in [0, T]$. To numerically solve the PDE, we discretize the equation over a grid. Notation-wise, the grid point in space with coordinates $(i\Delta x_1, j\Delta x_2)$ is denoted by (i, j). The value of $U(x_1, x_2, t)$ at a grid point (i, j) at time $t = k\Delta t$ is denoted by $u_{i,j}(k)$. Similarly, we define $v_{i,j}(k)$. The discretized approximation of (16) is a system that can be represented in the form of (1). The state variable is a vector in $\mathbb{R}^n$, where $n = 2(n_{x_1} \times n_{x_2})$, and n_{x_1} and n_{x_2} are the numbers of grid points in the x_1 and x_2 directions, respectively. We set $\kappa = 0.14$ and $T = 5$. The initial condition is

$$\begin{aligned}
&U(x_1, x_2, 0) = V(x_1, x_2, 0) = g(x_1, x_2),\\
&g(x_1, x_2) = \begin{cases} \alpha(x_1^3(4 - x_1)^3 x_2^3(4 - x_2)^3), & 0 \leq x_1, x_2 \leq 4\\ 0 & \text{otherwise} \end{cases}
\end{aligned} \tag{17}$$

where α rescales the function so that the maximum value of g is 1. The equation is discretized on a uniform grid that has 50 points in both x_1 and x_2 directions. Assume that the values of U and V are measured at 12 locations. The grid and sensor locations are shown in Figure 1. As an example, we compute the predictability of U at the grid point $(25, 25)$, which is the red point in Figure 1. To compute the predictability, let $\mathcal{I}$ in (14) consist of all integer pairs, (i, j), that represent the grid

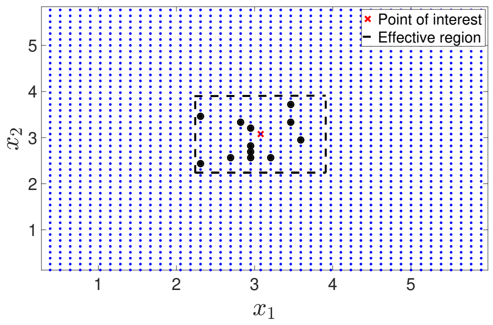


Fig. 1: The sensor locations (large dots) and the point of interest (red point) at which the state is predicted.

points located inside a square region around the center point $(25, 25)$. The radius of the region is defined as the number of grid points from $(25, 25)$ to the boundary, including the points on the boundary. For example, the radius of the region bounded by the dotted line in Figure 1 is R = 6. The value of ρ is computed based on (13)-(15) in regions with radius $1 \leq R \leq 24$ (Figure 2). Numerical scheme exploited is a first order upwind in time and space for the first order partial derivatives on the left-hand side of the equation, with an implicit central difference for the second order spatial derivatives on the right-hand side [12]. The time step size is $\Delta t = 0.05$. The sensor error bound is set to be $\epsilon = 0.065$. The upper bound of $\Delta \mathbf{u}_0$ is $\delta = 1.0$. The time window is $K = 15$.

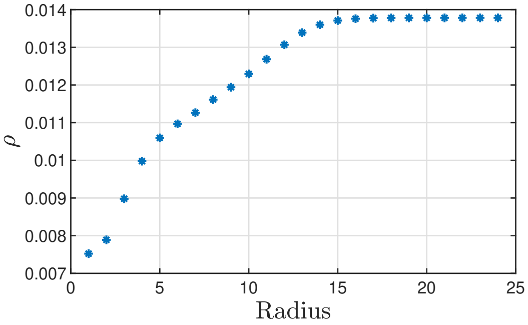


Fig. 2: Predictability: the value of ρ in regions with $R = 1, \cdots, 24$.

The value of ρ is stabilized after $R \geq 15$. We define an effective region as follows: *If the value of ρ computed in a region is stabilized, i.e., the value of ρ computed in any larger region has negligible change, then this region is called an effective region.* The determination of an effective region depends on the threshold set for the variation of ρ. In this example, any region with $R \geq 15$ is an effective region because the value of ρ does not increase when R is larger than 15. One may relax the threshold. For instance, set $R = 10$ as an effective region if the value of ρ is considered stable with a variation less than 1.5×10^{-3}.

If the effective region is much smaller than the domain of the PDE, the dimension of the empirical observability Gramian is significantly reduced. This is critical for high dimensional problems, such as the atmospheric models in numerical weather prediction, for which the eigenvalues of G in its full dimension are impossible to compute. Consequently, the computation of predictability can be confined to an effective region rather than the entire domain of the PDE. Additionally, by utilizing an effective region, data generation for training a surrogate prediction model can be confined to a smaller area, thus eliminating the need to integrate the PDE across its entire domain. This will be exemplified in the next section.

IV. The surrogate prediction model of dynamical systems

A surrogate prediction model proposed in this paper is a neural network whose input is the sensor information in a given time interval. The output of the neural network is the value of a variable to be predicted. It has a similar structure to the deep filters introduced in [9], [13], [14]. For example, the 12 sensors shown in Figure 1 for system (16) measure $u_{i,j}(k)$ and $v_{i,j}(k)$, where (i, j) represents the location of sensors, $u_{i,j}$ and $v_{i,j}$ represent the value of $U(x, y, t)$ and $V(x, y, t)$ at the grid point (i, j), and $0 \leq k \leq K$. For simplicity of discussion, we assume that sensors are located at grid points. The sensor information in each data point is a vector of dimension $24(K + 1)$. In this example, we predict the value of $u_{25,25}(K + \Delta K)$. It is straightforward to generalize the idea to predicting multiple variables at a set of grid points.

A. Training and validating the surrogate prediction model

We generate data through simulations for the training of the surrogate prediction model. Rather than solving the PDE in the entire domain, simulations are carried out within an effective region. Consequently, the computational requirements can be significantly reduced by eliminating the need to solve the PDE in a substantially larger spatial domain.

Another advantage of concentrating on an effective region, which will be addressed in the rest of this paper, is the PDE's insensitivity to boundary condition around an effective region. This property enables us to generate data by utilizing a *fabricated boundary condition* around the effective region, i.e., the boundary condition is pre-set, but its value may not always reflect the truth. It eliminates the requirement of providing precise lateral boundary conditions, as commonly needed in limited-area models utilized in applications such as numerical weather prediction. For instance, in the following examples, we set the boundary condition around the effective region invariant with time, i.e.,

$$\begin{aligned} U(x_1, x_2, t) &= U(x_1, x_2, 0), \\ V(x_1, x_2, t) &= V(x_1, x_2, 0), \end{aligned} \tag{18}$$

for (x_1, x_2) on the boundary. We will demonstrate that the surrogate prediction model, trained using data with fabricated boundary conditions, can accurately predict the state variable, even when the validating trajectory does not adhere to the fabricated boundary condition. This is possible due to an intriguing characteristic of the effective region: the trajectory of the state variable at the center of the region is insensitive to boundary conditions. An explanation is that, by definition, ρ can be determined by computing the empirical observability Gramian solely within the effective region, without the necessity of exploring state variations across the entire spacial domain. In other words, alterations in the boundary conditions around the effective region do not significantly impact the worst indistinguishable prediction; the effect of boundary conditions has limited impact on prediction accuracy.

We first computed $10,000$ solutions of the discretized Burgers equation in $t \in [0, T]$, $T = 5$, with the following random initial values around the nominal solution (17),

$$U(x_1, x_2, 0) = g(x_1, x_2) + \sum_{l,s=1}^{N_F} a_{ls}^U \sin(\frac{lx_1}{2}) \sin(\frac{sx_2}{2}),$$
$$V(x_1, x_2, 0) = g(x_1, x_2) + \sum_{l,s=1}^{N_F} a_{ls}^V \sin(\frac{lx_1}{2}) \sin(\frac{sx_2}{2}), \tag{19}$$

where $g(x_1, x_2)$ is defined in (17), a_{ls}^U and a_{ls}^V have a normal distribution with a mean of zero and standard deviation of 0.1, and we set $N_F = 3$. The data set for training is generated from these solutions at uniformly distributed random time $t_0 = k_0 \Delta t$, $\Delta t = 0.05$, and k_0 varies randomly for each data point. The time window length of sensor data is $K = 15$, and the prediction horizon is $\Delta K = 6$. The total number of data points is about $30,000$. When generating the training data, we solved the PDE in the effective region with $R = 15$. Additionally, for comparison purposes, we generated data by solving the PDE in regions with $R = 6$ and $R = 10$. For validation, the dataset was generated using initial conditions defined over the original spatial domain of the PDE with a 50×50 grid. The goal is to assess the performance of the surrogate prediction model on trajectories representing the overall dynamical system across the entire spatial domain.

The surrogate prediction model is a neural network that has 8 hidden layers with a width of 16 neurons. Each neuron is a hyperbolic tangent. In the supervised learning, the neural network is trained using the standard mean square error as the loss function. The neural network is trained in TensorFlow utilizing L-BFGS optimization. Although the training data is generated from small regions, we validate the surrogate prediction model using data computed across the entire domain to ensure the accuracy of predictions for the original problem defined within a larger domain. The prediction errors of the three surrogate prediction models trained using data from regions with $R = 6, 10, 15$, represented by SPM6, SPM10 and SPM15, are summarized in the following table.

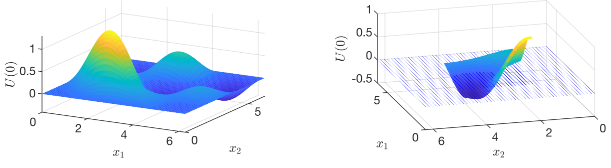
SPM	SPM6	SPM10	SPM15
RMSE	0.0252	0.0158	0.0091
Maximum error	0.1276	0.0575	0.0504

TABLE I: Prediction errors.

The errors are measured by root-mean-square error (RMSE) and the maximum absolute error over validation data. According to Figure 2, ρ is stabilized when $R = 15$. We expect that data generated from this region results in the most accurate surrogate prediction model. When $R = 6$, ρ is far from stabilized. The data generated in such a small region does not represent the family of all solutions of the PDE. At $R = 10$, while ρ is not finally stabilized, it is close. The difference between the value of ρ at $R = 10$ and $R = 15$ is less than 1.5×10^{-3}, which is considered small relative to the sensor noise standard deviation ($\sigma = 0.065$). The results shown in Table I reflect the above analysis. The accuracy of the surrogate prediction model trained in the region $R = 6$ exhibits the worst performance, with the maximum error more than double the error trained in the regions with $R = 10$ and $R = 15$. The RMSE for $R = 6$ is about 60% higher than that of $R = 10$ and 170% higher than that of $R = 15$.

B. Insensitivity to the boundary condition around an effective region

In IV-A, the validation data set is generated using initial conditions defined over the original spatial domain of the PDE with 50×50 grid points. A zero boundary condition is applied. This is significantly different from the training data that is generated in regions $R = 6, 10, 15$ with the fabricated boundary condition (18). Two examples of typical initial conditions are shown in Figure 3. The differences are obvious. The results in Table I demonstrate that the surrogate prediction model trained in an effective region ($R = 10, 15$) is insensitive to the inaccuracy of the boundary condition. The predictions are relatively accurate without a known boundary condition, as commonly needed in conventional numerical models.



(a) (b)

Fig. 3: An example of initial condition, validation data in (a) and training data in (b).

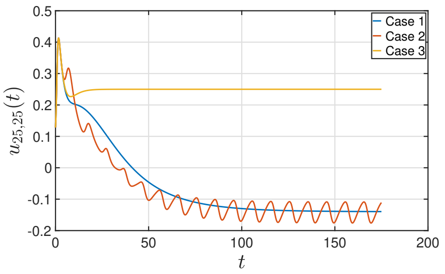


Fig. 4: Three trajectories of $u_{25,25}$ with different nonzero boundary conditions. On the boundaries, U and V equal: $0.4 \tanh(t) \sin(z)$ (Case 1), $0.8 \sin(t\pi/5) \sin(z)$ (Case 2), and $0.25 \tanh(t)$ (Case 3), where $z = x$ or y along boundaries parallel to x- or y- axes, respectively

For further verification of the insensitivity to boundary conditions around effective regions, we generated more validating data based on nonzero boundary conditions in the domain of the PDE. Three trajectories of $u_{25,25}$ are shown in Figure 4. They are dramatically different from the training data used in IV-A. The accuracy of the surrogate prediction models in IV-A is summarized in Table II. Relative to the sensor noise standard deviation, $\sigma = 0.065$, the surrogate prediction models trained in the effective regions with $R = 10, 15$ are acceptable or good. Case 2 is most challenging due to the oscillating behavior. While the RMSEs are relatively small, the maximum errors of both SPM10 and SPM15 show that the predictions contain outliers as large as the standard deviation of the sensor noise.

Error	Case 1	Case 2	Case 3
RMSE (SPM10)	0.0189	0.0245	0.0093
Maximum error (SPM10)	0.1005	0.0805	0.0385
RMSE (SPM15)	0.0110	0.0125	0.0061
Maximum error (SPM15)	0.0476	0.0610	0.0219

TABLE II: Prediction errors.

C. Shifting the point of interest in space

For PDEs in which coefficients are independent of the spatial variables, such as the Burgers equation (16), a surrogate prediction model trained in one effective region can be applicable to another area as long as the relative positions of the point of interest and the sensor locations remain unchanged. For example, let us consider the surrogate prediction models demonstrated in IV-A, which are trained in effective regions centered around the grid point $(25, 25)$. Now, we employ the same surrogate prediction model to estimate the state at a different location, $(20, 25)$, assuming that the relative locations of sensors are unchanged, i.e. the sensors are also shifted to the left by 5 grid points along the x-axis. The validation data is generated in the same way as in IV-A except that the target function to be predicted is $u_{20,25}$. The results are summarized in Table III.

SPM	SPM10	SPM15
RMSE	0.0238	0.0125
Maximum error	0.1000	0.0570

TABLE III: Prediction errors

The performance of the surrogate prediction model trained in the region with $R = 15$ is comparable to that presented in Table I. Shifting the point of interest in prediction does not significantly affect prediction accuracy. However, shifting the point of interest appears to cause a notable increase in error for the surrogate prediction model trained in the region with $R = 10$, where the value of ρ is close to the point of stability but has not quite stabilized yet.

V. Conclusions

The proposed surrogate prediction model for dynamical systems does not require multiple integrations of a numerical model during online computations. Instead, the online computation merely involves evaluating a feedforward neural network, which is significantly less computationally expensive than integrating high-dimensional numerical models. The offline computation involves neural network training and data generation. Due to the property of insensitivity to boundary conditions, the data is generated using a fabricated boundary condition around an effective region. It does not require an accurate lateral boundary condition, as conventional algorithms commonly need to solve PDEs. Furthermore, a reduced-order model can be employed to generate the dataset, focusing only on the effective region and avoiding to integrate the PDE across the entire domain.

Prediction of dynamical systems based on integrating a differential equation requires an initial condition. If the initial condition is not directly available, a data assimilation algorithm, such as the Luenberger observer, Kalman filter, or 4D-Var, has to be applied to estimate the initial condition using sensor information. This two-stage approach, i.e., data assimilation first, then prediction by integrating the differential equation is commonly used in control systems and numerical weather prediction. For high-dimensional problems, this approach is complicated and computationally demanding. The surrogate prediction model proposed in this paper is fundamentally different in the sense that the prediction is carried out in one single step, evaluating a neural network. As a result, updating predictions based on incoming sensor data is computationally less expensive than using algorithms that require integrating the system model.

While the examples of the Burgers equation demonstrate encouraging outcomes, for future research, additional examples and applications as well as statistical data analysis are necessary to strengthen the evidence and ensure the practical applicability of the proposed methods.

Acknowledgment: Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the NSF, the AFOSR, Naval Postgraduate School and the NRL.

References

[1] F. Allgöwer and A. Zhang, *Nonlinear Model Predictive Control*. Birkhäuser Basel, 2000.

[2] D. Osipov and K. Sun, "Adaptive nonlinear model reduction for fast power system simulation," *IEEE Trans. Power Syst.*, vol. 33, no. 6, pp. 6746–6754, 2018.

[3] T. T. Warner, *Numerical Weather and Climate Prediction*, illustrated ed. Cambridge University Press, 2011.

[4] N. Gustafsson, X.-Y. Huang, X. Yang, K. Mogensen, M. Lindskog, O. Vignes, T. Wilhelmsson, and S. Thorsteinsson, "Four-dimensional variational data assimilation for a limited area model," *Tellus A*, vol. 64, 2012, 14985.

[5] Z. Meng and F. Zhang, "Limited-area ensemble-based data assimilation," *Monthly Weather Review*, vol. 139, pp. 2025 – 2045, 2011.

[6] N. Gustafsson, E. Källén, and S. Thorsteinsson, "Sensitivity of forecast errors to initial and lateral boundary conditions," *Tellus A*, vol. 50, pp. 167 – 185, 1998.

[7] L. Lu, X. Meng, Z. Mao, and G. E. Karniadakis, "Deepxde: A deep learning library for solving differential equations," *arXiv:1907.04502*, 2019.

[8] W. Kang and L. Xu, "Computational analysis of control systems using dynamic optimization," *arXiv: 0906.0215v2*, 2009.

[9] W. Kang, L. Xu, and H. Zhou, "Observability and effective region of partial differential equations with application to data assimilation," *SIAM Journal on Scientific Computing*, 2024, to appear.

[10] W. Kang and L. Xu, "Observability and optimal sensor placement," *International Journal of Sensors, Wireless Communication and Control*, vol. 1, no. 2, pp. 93 – 101, 2011.

[11] A. J. Krener and K. Ide, "Measures of unobservability," in *48th IEEE Conference on Decision and Control*, 2009, pp. 6401–6406.

[12] R. J. LeVeque, *Numerical Methods for Conservation Laws*. Birkhauser Verlag, 1992.

[13] W. Kang, L. Xu, and H. Zhou, "The observability in unobservable systems," in *IEEE 17th International Conference on Control & Automation*, 2022, pp. 315–319.

[14] H. Qian, G. Yin, and Q. Zhang, "Deep filtering with adaptive learning rates," *IEEE Trans. Automat. Control*, vol. 68, no. 6, 2023.