

Commun. Theor. Phys. **74** (2022) 055601 (7pp) https://doi.org/10.1088/1572-9494/ac60f9

Less is more: a new machine-learning methodology for spatiotemporal systems

Sihan Feng[1], Kang Wang[1], Fuming Wang[1], Yong Zhang[1,2] and Hong Zhao[1,2]

[1] Department of Physics, Xiamen University, Xiamen 361005, China
[2] Lanzhou Center for Theoretical Physics, Key Laboratory of Theoretical Physics of Gansu Province, Lanzhou University, Lanzhou 730000, China

E-mail: zhaoh@xmu.edu.cn

Received 7 February 2022, revised 24 March 2022
Accepted for publication 25 March 2022
Published 3 May 2022

Abstract

Machine learning provides a way to use only portions of the variables of a spatiotemporal system to predict its subsequent evolution and consequently avoids the curse of dimensionality. The learning machines employed for this purpose, in essence, are time-delayed recurrent neural networks with multiple input neurons and multiple output neurons. We show in this paper that such kinds of learning machines have a poor generalization ability to variables that have not been trained with. We then present a one-dimensional time-delayed recurrent neural network for the same aim of model-free prediction. It can be trained on different spatial variables in the training stage but initiated by the time series of only one spatial variable, and consequently possess an excellent generalization ability to new variables that have not been trained on. This network presents a new methodology to achieve fine-grained predictions from a learning machine trained on coarse-grained data, and thus provides a new strategy for certain applications such as weather forecasting. Numerical verifications are performed on the Kuramoto coupled oscillators and the Barrio−Varea−Aragon−Maini model.

Keywords: machine learning, spatiotemporal systems, prediction, dynamical systems, time series, time-delayed recurrent neural network

(Some figures may appear in colour only in the online journal)

1. Introduction

Predicting the evolution of dynamic systems is important [1–3]. These systems usually have to be solved numerically using spatial and temporal discretization because of the mathematical intractability of obtaining analytical solutions. In this way, a partial differential equation is approximated by a set of ordinary differential equations. The main obstacle to this approach is that it becomes infeasible in higher dimensions due to the need of fine-grained spatial and temporal mesh points, which is known as the 'curse of dimensionality' [4]. For real applications, one has to seek a balance between accuracy and computing cost by applying a relatively coarse mesh [1, 5].

The machine-learning approach developed in recent years provides possible ways to attack this notoriously difficult problem [6–9]. One way is to solve partial differential equations with variables from randomly sampled grid points when the equation of motion is known. Researchers have developed an effective approach called physical informed neural networks [10] for this purpose. Another way, which is also the focus of this paper, is for application scenarios that have data records that are available for a large number of spatial points, while the equations of motion are unknown. A learning machine trained by the time series of such spatial points can predict their subsequent evolution in a model-free way [11, 12]. Using such a strategy, researchers from Google® developed a deep-learning machine called MetNet and improved local weather forecasts greatly [13].

In these applications, a recurrent neural network with Q input and Q output neurons is employed to be the learning machine, where Q is the number of spatial variables whose time series records are used in the training. When training such a learning machine, the data from the Q variables are the Q inputs, and the outputs represent the next-step evolution of these variables. Feeding the outputs back to the inputs, the learning machine can work as an iterative map. In this paper,

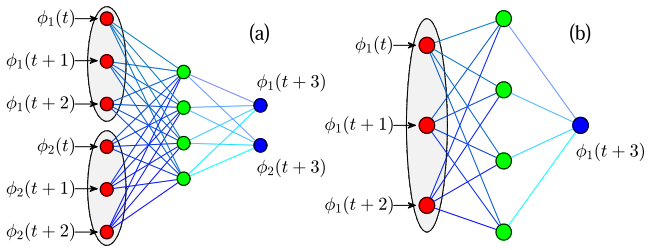

Figure 1. Schematic diagrams of the learning machines. (a) a 2-4-2 QD-TDM, with a delay length of $M = 3$, and (b) a 1-4-1 1D-TDM, with a delay length of $M = 3$. Each gray ellipse represents an input unit for time series from one spatial variable.

we first check the generalization ability of such a learning machine applied to new sets of Q spatial variables that have not been trained with. This ability characterizes the robustness of the machine. For a real system, the system parameters may shift slightly as a function of time. As a result, the learning machine is expected to have a certain degree of robustness to such shifted data sets. Our numerical verification indicates, however, that such kind of learning machines have no such generalization ability.

To solve this problem, we propose, instead, to apply a one-dimensional (1D) time-delayed recurrent neural network as the learning machine to predict the evolution of the underlying system. In the training stage, the learning machine is still trained with time series collected from different spatial variables. In the predicting stage, however, the learning machine can be initiated by the history record measured at any one spatial point. We show that this learning machine can not only predict the evolution of the variables that have been trained with, but also that of variables that have not been trained with before. The advantages of such generalization ability are that the learning machine will be robust against small variations of system parameters, and it is possible to perform fine-grained predictions with a learning machine trained with coarse-grained data sets. This should be useful for applications such as the weather forecast as one can perform the prediction of a local region without the need of learning from the fine-grained mesh of the entire spatiotemporal system. Our numerical studies are performed on models of the Kuramoto coupled oscillators (KCO) [14–16] and the Barrio−Varea−Aragon−Maini equation (BVAM) [17, 18]. The rest of the paper is structured as follows. The machine-learning model and method are introduced in section 2. The main results are reported in section 3, in which the mechanism why the traditional approach has failed in transformational generalization while ours has succeeded is also explained. The summary and discussion are given in section 4.

2. Model and method

The learning machines adopted in this paper are schematically illustrated in figure 1. Figure 1(a) represents an MQ-N-Q neural network (with MQ, N, and Q, respectively, the number of neurons in the input, hidden, and output layer), or equivalently a Q-N-Q time-delayed map with a delay length of M (as illustrated in the gray ellipse, M neurons are utilized as the input units for each of Q input records). Hereafter, it is called the Q dimensional time-delayed machine (QD-TDM), of which the iterated map is given below:

$$\phi_l(n+1) = \sum_{i=1}^{N} \mu_{l,i} f\left(\beta_i \sum_{l=1}^{Q}\left(\sum_{j=0}^{M-1} \nu_{i,j+(l-1)M}\phi_l(n-j)\right) - b_i\right), \qquad (1)$$

where μ, β, ν, and b, are map parameters and f is the neuron transfer function, which is chosen as $f = \exp(-h^2)$ throughout this paper. Equation (1) means that by applying this map the $(n+1)$ th output of a series is determined by its $(n-M+1)$ th to nth historical inputs. This map can also be seen as a three-layer neural network which maps MQ input data points, $\{\phi_l(n-j)\}_{l=1,\cdots,Q;j=0,\cdots,M-1}$ to Q output points $\{\phi_l(n+1)\}_{l=1,\cdots,Q}$. Let $\{x_l(j)\}_{j\ =\ 1,\cdots,P}$ be a segment of a time series of the lth spatial point of a spatiotemporal system, then the $\{x_l(k-i)\}_{l=1,\cdots,Q;i=0,\cdots,M-1}$ is set to be the inputs and $\{x_l(k+1)\}_{l=1,\cdots,Q}$ the expected outputs, we thus get total $P-M$ training samples (each with Q dimensions) numbered by $k = M, \cdots, P-1$.

The cost function is defined as

$$\lambda = \frac{1}{P-M}\sum_{l=1}^{Q}\sum_{k=M}^{P-1}(x_l(k+1) - \phi_l(k+1))^2. \qquad (2)$$

In the training stage, the cost function is minimized for the $P-M$ training samples under a properly chosen set of control parameters: M, N, Q, c_μ, c_ν, c_β, and c_b, and the last four specify the available ranges of the network parameters, i.e., $|\mu_{l,i}| \leqslant c_\mu$, $|\nu_{i,k}| \leqslant c_\nu$, $|\beta_i| \leqslant c_\beta$, $|b_i| \leqslant c_b$. To perform the training, at fixed M, N, and Q, the μ_i, $\nu_{i,k}$, β_i and b_i are randomly initialized in their respective value ranges, then a parameter from μ_i, $\nu_{i,k}$, β_i or b_i is randomly chosen and then randomly mutated in its available range; this mutation is finally accepted if only that it does not increase the cost. Repeat the mutation step over and over again, the cost function will decrease monotonously. Since for each adaptation it needs to renew only a small portion of the network, this algorithm is practical for usual applications, see [19, 20] for more details. Indeed, one can also apply the conventional gradient-based (also known as the back-propagation) algorithm to train our learning machine if removing the restrictions to the value ranges. Researchers in the machine learning community begin to try gradient-free algorithms in recent years, and the training algorithm presented here is one of such an effort. With this gradient-free algorithm one can achieve the goal of training, at least for the usual three-layer neural networks, while gaining the ability to limit the range of the network parameters, and therefore control the structural risk of the network. In addition, the parameters can take assigned discrete values, which may be required for certain practical applications. Under this background, we adopt our gradient-free algorithm instead of the conventional gradient-based algorithms.

In the predicting stage, one just needs to input the last samples of $k = P$ to start equation (1), and feed the outputs back to the inputs correspondingly to make equation (1) a self-evolve iterate map. In the case of figure 1(a), for example, after the first round of iteration one needs to feedback $\phi_1(t+3)$ to the up inputting unit, and $\phi_2(t+3)$ to the bottom inputting one, and turns the inputs to $\phi_1(t+1)$, $\phi_1(t+2)$, $\phi_1(t+3)$, and $\phi_2(t+1)$, $\phi_2(t+2)$, $\phi_2(t+3)$, respectively, to initiate the second round of iteration.

Figure 1(b) represents an M-N-1 neural network, or equivalently a 1-N-1 time-delayed map with a delay length of M (also illustrated in the gray ellipse). We will call it the one-dimensional time-delayed machine (1D-TDM) afterwards, and its evolution dynamics are given by:

$$\phi(n+1) = \sum_{i=1}^{N} \mu_i f(\beta_i(\sum_{j=0}^{M-1} \nu_{i,j}\phi(n-j)) - b_i). \tag{3}$$

This equation appears as a reduced form of equation (1) as the summation of ϕ_l is replaced with a single ϕ. In the training stage, the time series of variables at multiple spatial positions are used sequentially to train the machine so that the dynamic information of those multiple spatial points are all learned by the machine. This is a key difference from previous strategies of training a 1D-TDM-like machine. In more detail, for the lth time series, set the $\{x_l(k-i)\}_{i=0,\cdots,M-1}$ to be the inputs and $\{x_l(k+1)\}$ the expected outputs, we again obtain $P-M$ training samples from one time series l. Apply the above operations to all of the Q time series and we get total $(P-M)$ Q training samples (each in one dimension). The cost function has the same form as equation (2). In the predicting stage, only the last sample from one spatial variable is needed to initiate the learning machine.

We emphasize that those popular learning machines used for model-free prediction of the evolution of the underlying dynamic systems, such as the long short-term memory model [21] and the reservoir computer [22–25] are essentially equivalent to the time-delayed model described by the equation (1). The long short-term memory model manages a set of neurons in the hidden layers to memorize the history (i.e., delayed information) of the inputted time series and thus is obviously equivalent to a time-delayed map. In using the reservoir computer, one needs to iterate the so-called reservoir for a sufficient number of steps before outputting the prediction in the initiation stage. The reservoir network indeed plays the role of storing the delayed information or memory. Thus, it is also a time-delayed map. Therefore, the QD-TDM can generally represent these conventional learning machines.

3. Results

3.1. The KCO model

We first test QD-TDM on the KCO model, which is given by

$$\dot{\theta}_i = \omega_i + \frac{K}{L}\sum_{j=1}^{L}\sin(\theta_j - \theta_i), \tag{4}$$

where ω_i and K, respectively, represent the natural frequency and the coupling coefficient, and L is the total number of oscillators. We see from this equation that the motion of the ith oscillator is determined not only by its natural frequency but also by the coupling from all other oscillators, with the strength being controlled by the parameter K. This model is widely applied to study the collective behavior of complex systems [15, 16]. With strong coupling, the oscillators may show synchronous oscillation and thus are reduced to an effective system with a low dimension. To avoid this case, weak coupling is applied to ensure that all of the oscillators oscillate roughly around their natural frequency ω_i and the system lies in a weak chaotic state. The measurable variable of an oscillator is defined as $x_i(t) = \sin(\theta_i(t)) + \cos(\theta_i(t))$ without loss of generality.

At $L = 20$ and $K = 0.8$, we set ω_i with a random number from the interval (0, 1). It can be checked that the system is chaotic with the largest Lyponov exponent being about 0.005. Evolving the system for a long time (with the time step of integral $h = 0.01$) to collect a sufficiently long time series for each oscillator, then sample these time series with an interval of $\Delta t = 0.1$ to construct the training set. We train the QD-TDM using the first ten oscillators ($i = 1, 2, \ldots, 10$). The width of the learning machine is fixed at $N = 1000$ throughout this paper, which is large enough for achieving the training goal. By searching the control parameter space we find that $M = 2000$, $c_\mu = 0.03$, $c_\nu = 0.45$, $c_\beta = 0.05$, and $c_b = 40$ to be the optimum control parameters for training the learning machine in the case of the KCO data set. After finishing the training, we first apply the learning machine to predict these first ten oscillators, and then apply it to evolve the next ten oscillators that have not been trained with (i.e., inputting the time series from oscillators of $i = 11, 12, \ldots, 20$ respectively to initiate the learning machine).

In figure 2(a), the black lines show segments of time series $x_i(t)$ for $i = 6$ to $i = 15$ as an example. The green and red lines show the predicted evolutions, for oscillators that have been trained with and for those that have not been trained with, respectively. We see that for the oscillators that have been trained with ($i = 6$ to $i = 10$), the predictions are quite good, but it totally loses the ability to predict for the oscillators ($i = 11$ to $i = 15$) that have not been trained with previously.

Then we use the same data of the first ten oscillators to train the 1D-TDM and apply it to predict all the 20 oscillators one by one. We find that not only the oscillators that have been trained with are well predicted, but those that have not been trained with previously are also well predicted to a large extent as can be seen in figure 2(b). Even using fewer oscillators to train this learning machine, the generalization ability is still considerable. Figures 2(c) and (d) show the results of the learning machine trained respectively by the oscillators of $i = 1, 2, 3, 4$ and by only the oscillator of $i = 1$. It can be seen that, although the prediction quality gets slightly worse with the decrease of the number of training oscillators, the difference is not remarkable. Therefore, the 1D-TDM has excellent

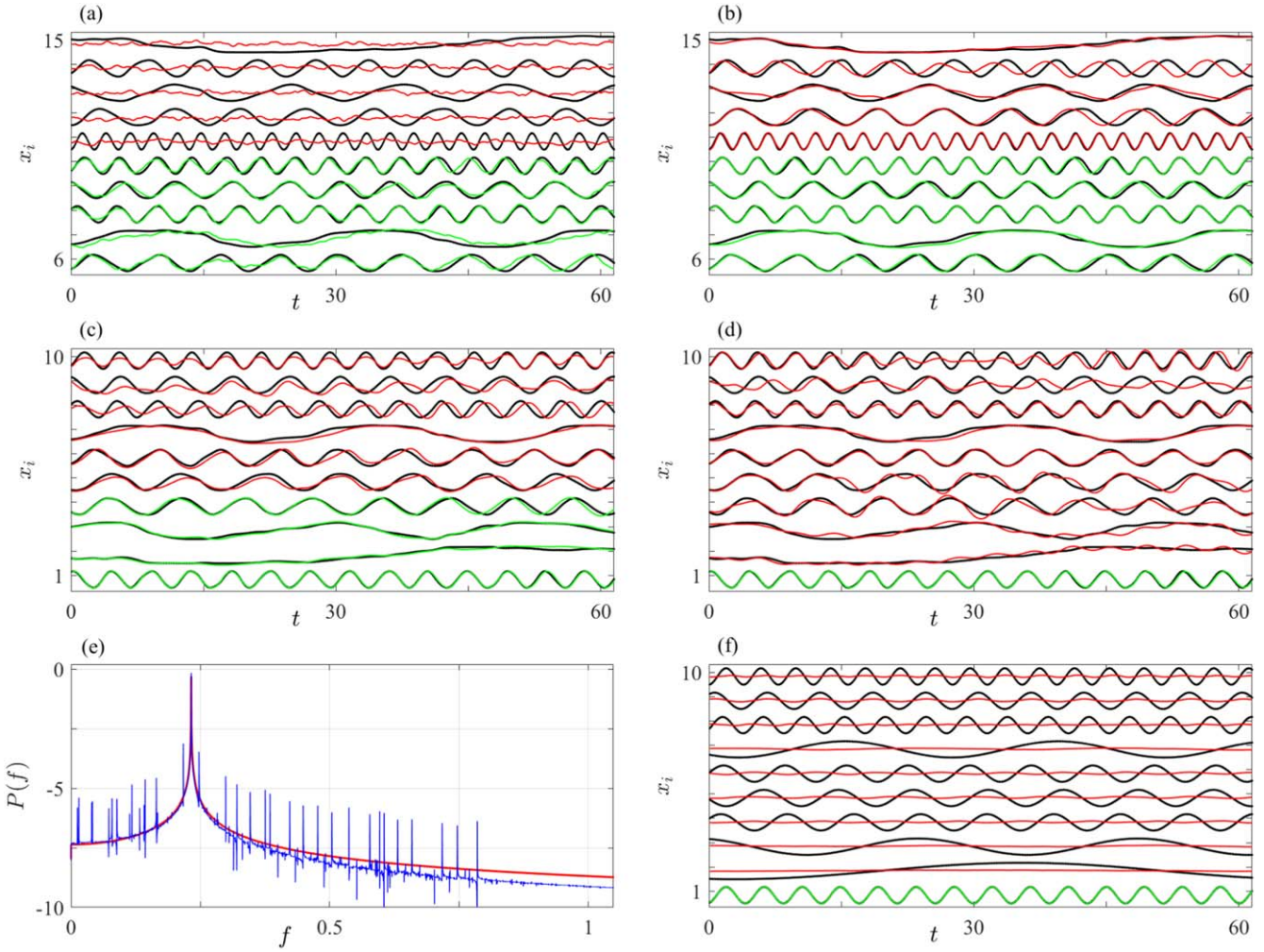

Figure 2. The generalization ability of QD-TDM and 1D-TDM to new variables for the KCO model. (a) Prediction results of QD-TDM trained with oscillators from $i = 1$ to $i = 10$, (b)–(d) prediction results of 1D-TDM trained with oscillators, (b) from $i = 1$ to $i = 10$, (c) from $i = 1$ to $i = 4$, and (d) with only $i = 1$. The black lines represent data x_i of the KCO model, and the green and red lines represent the predictions for oscillators that have been trained with (green) and for those that have not (red). (e) Shows, in semilogarithmic scale, the power spectrum of the first oscillator with control parameter $K = 0$ (red line) and $K = 0.1$ (blue line), and (f) shows the prediction of the 1D-TDM trained with the first oscillator ($i = 1$) with $K = 0$.

generalization ability to the new data set even trained by only few variables.

It is not surprising that a learning machine has the generalization ability to new variables that have not been trained on. According to the Takens theory, for a time series with a sufficient length, one can reconstruct an M-dimensional phase space map, where M is the delay length. If only $M > 2D + 1$, this map should be topologically equivalent to the underlying dynamic system, where D is the fractal dimension of the attractor of the underlying system. This is the well-known phase space reconstruction technique of delay-coordinate embedding [26, 27]. This theory holds because any variable of a dynamic system is considered to be coupled with others, and thus involves the information of other variables.

Figure 2(e) shows the power spectrum of the time series of the first oscillator $x_1(t)$ with $K = 0$ (red line) and $K = 0.1$ (blue line), respectively. It can be seen that without coupling, the power spectrum only the frequency peak of this oscillator (red line). In this case, the learning machine involves no information of the other oscillators. It can be easily checked that the learning machine trained by this time series has no generalization ability to oscillators that have not been trained with previously, see figure 2(f). With a non-vanishing coupling, it is seen that though it is difficult to observe the coupling effect of other oscillators by just looking at the evolution curve (see black lines figure 2(d)), the power spectrum actually includes frequency peaks of the other oscillators (see the blue line in figure 2(e)). These peaks have a very small amplitude (note that the vertical axis is with the logarithmic scale), and differ among different oscillators. The power spectrum confirms that the information of other oscillators are coupled to the first oscillator, which provides the basis that the learning machine trained by data of only one of the oscillators could have the generalization ability to other coupled ones.

The phase space can also be reconstructed using multiple time series with the technique of delay-coordinate embedding [27]. The reason why a 1D-TDM can, while a QD-TDM

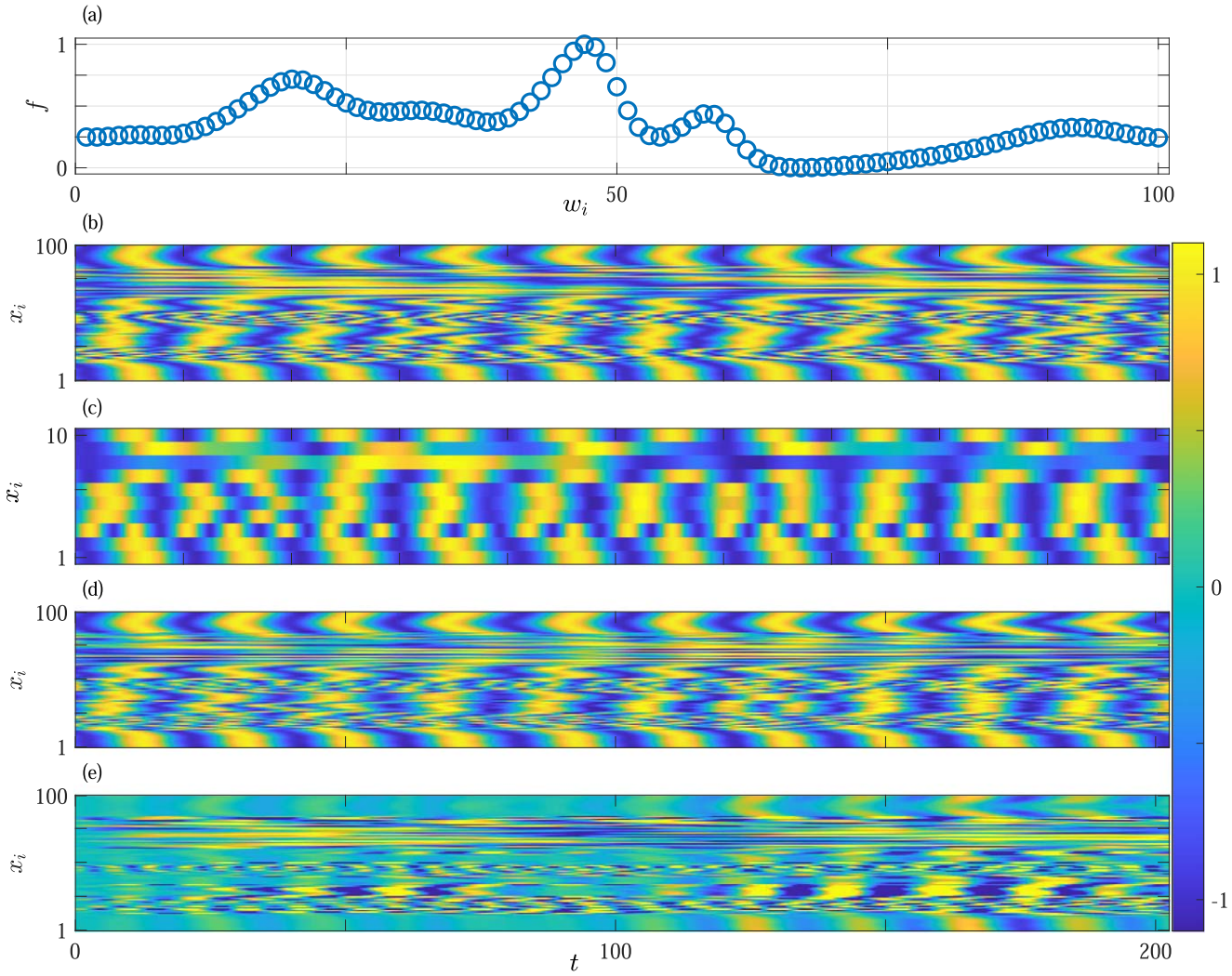

Figure 3. Recovering fine-grained spatiotemporal structures of the underlying system using the 1D-TDM trained with small portions of the variables. (a) ω_i of oscillators, (b) actual fine-grained spatiotemporal pattern of the KCO model, (c) predicted coarse-grained spatiotemporal pattern of the KCO model. (d) Fine-grained spatiotemporal pattern recovered by the 1D-TDM. (e) Prediction errors (panel (b) minus panel (d)).

cannot be applied to spatial points that have not been trained with previously is the following. A remarkable different property of the 1D-TDM from the QD-TDM is that it does not involve the time unit explicitly in its equation of motion; what is inputted to the learning machine is just a discrete time series [see equation (3)]. With this feature, a 1D-TDM can evolve the dynamics of any oscillator once it is initiated by its time series if only the information of this oscillator is correctly coupled into the learning machine. In contrast, the Q time series with different frequencies are needed to be evolved simultaneously in a QD-TDM (see equation (1)), which thus introduces a relative time unit to the machine. If, for example, it is trained by two time series with frequencies ω_1 and ω_2, then a relative time unit is defined by the ratio of ω_2/ω_1. The learning machine could correctly recover the dynamics of two oscillators with frequencies $c\omega_1$ and $c\omega_2$, if these two oscillators belong to the underlying system. In this case, the time series of ω_1 and ω_2 and the time series of $c\omega_1$ and $c\omega_2$ have the same relative time unit. However, if the learning machine is applied to a new set of oscillators with ω_3 and ω_4, the relative time unit changed, and mismatching arises. The mismatching may become more serious for larger Q.

This generalization ability of the 1D-TDM provides a useful strategy for possible applications to perform fine-grained predictions using a learning machine trained on coarse-grained data. To show such a possibility, we construct a KCO model with 100 oscillators. The oscillators take frequencies following the curve shown in figure 3(a). Setting the frequencies in such a way, the KCO model can exhibit fine-grained evolution patterns as shown in figure 3(b) instead of random patterns. Using 10 evenly spaced oscillators to train a 1D-TDM, we find that it can recover the subsequent evolution of these 10 oscillators well, which gives a coarse-grained picture of figure 3(b) in figure 3(c). However, using the trained learning machine to predict every oscillator, we recover figure 3(b) approximately in figure 3(d). Figure 3(e), shows the prediction errors, indicating that the recovery is almost perfect in a long period of an earlier time; the errors increase with the further increase of time, but they are overall

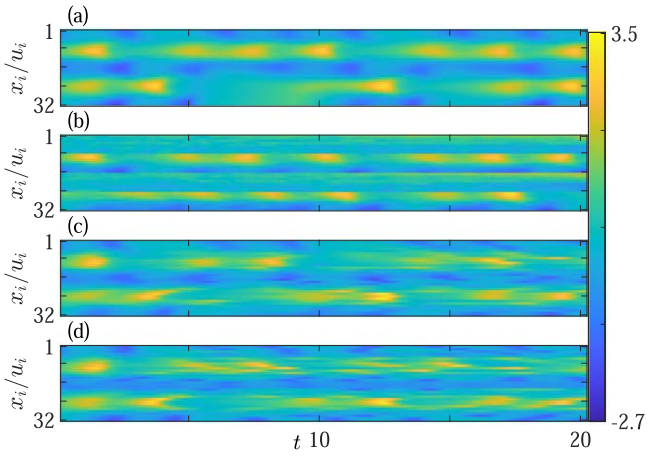

Figure 4. The generalization ability of QD-TDM and 1D-TDM for the BVAM model. (a) Spatiotemporal patterns of the model with the spatial variable u. (b) Predictions of the QD-TDM trained with mesh nodes of $i = 9, 10,\ldots,16$. (c) and (d) are the predictions of the 1D-TDM trained with mesh nodes of $i = 9, 10,\ldots,16$ and $i = 1, 3, 5, 7, 9$, respectively.

relatively small compared to the original variable, demonstrating an excellent prediction ability of the machine.

3.2. The BVAM model

In the KCO model, the amplitudes of oscillators are identical and oscillators are coupled globally. In this case, the time series of any oscillator may fully involve the information of all oscillators. Therefore, the 1D-TDM trained by only the time series of one oscillator may recover the dynamics of all the other oscillators. In other systems, the coupling among different variables may be different and may only couple locally. In this case, however, one has to involve more time series of representative spatial variables to train the learning machine to gain a better generalization ability to other variables. To show this, we study another spatiotemporal system, i.e., the BVAM model, whose equations of motion are given by

$$\begin{aligned}\frac{\partial u}{\partial t} &= D\Delta^2 u + \eta(u + av - Cuv - uv^2),\\ \frac{\partial v}{\partial t} &= \Delta^2 v + \eta(bv + Hu + Cuv + uv^2),\end{aligned} \tag{5}$$

where C, D, H, η, a, and b are system parameters. This model presents the richness of dynamic behavior [18]. Fix the system size at $L_b = 6.4$ with 32 mesh nodes for both u and v, and solve the equation (5) numerically (with the time step of integral $h = 0.01$) in one dimension with zero flux boundary conditions and random initial conditions around the equilibrium point (0, 0), at $D = 0.08$, $H = 3$, $\eta = 1$, $a = -1$, $b = -3/2$, and $C = -1.54$. The evolution of time is shown in figure 4(a), from where typical chaotic spatiotemporal patterns appear, i.e., spatial variables oscillate chaotically with the evolution of time. Typically, it can be seen that the oscillation range is different in different regions, manifested by the bright-yellow and dark-blue colors.

Sample the time series of the mesh node $u_i(t)$ at an interval of $\Delta t = 0.1$ and obtain the measurable time series $x_i(t)$. For this system, the optimum set of control parameters are $M = 300$, $c_\mu = 0.02$, $c_\nu = 1$, $c_\beta = 0.5$, and $c_b = 30$. Using data of $x_i(t)/u_i(t)$ with $i = 9, \cdots, 16$ to train the QD-TDM and the 1D-TDM respectively and then apply them to produce the evolution of the whole system after finishing the training, and the results are shown in figures 4(b) and (c), respectively. It is seen again that the QD-TDM can recover the dynamics of the spatial variables that have been trained with but not to the ones that have not (see figure 4(b)). The 1D-TDM, in contrast, possesses the ability to predict time evolution of variables in the spatial regions that have not been used for training (see figure 4(c)). Since this is a relatively strong chaotic system, the predictable time region is relatively shorter than in the case of the KCO model.

Choosing mesh nodes $x_i(t)/u_i(t)$ with $i = 1, 3, 5, 7, 9$ to be the training time series, the 1D-TDM can still recover the evolution dynamics of the whole system in figure 4(d). Even using only three mesh nodes $x_i(t)$ with $i = 1$, which represents variables in the dark-blue region, $i = 9$, which represents those in the bright-yellow region, and $i = 5$, which represents those in the light-green region, the 1D-TDM still retains some ability to generalize. When using the data from only one spatial variable, the generalization ability remains to some extent for spatial variables that belong to the same class, but not for the other classes.

4. Summary and discussions

In summary, the usual recurrent learning machine, requiring multiple system variables to iterate, almost has no generalization ability to the ones that have not been trained with previously. Applying such a learning machine, the number of sampled variables has to be sufficiently large to cover the major modes of the system, and predictions are only available for those variables that have been trained with. Meanwhile, using the 1D time-delayed map as the learning machine can predict the subsequent evolution of not only variables that have been trained with, but also variables that have not. Specifically, this learning machine requires data from only one variable to iterate, it can be trained with a small set of variables, and it can predict the evolution of the small set of variables it trained with as well as other variables that it did not train with. One of the possible applications of this property is that we may, after being well-trained, apply the learning machine to densely sampled spatial variables in a local region and thus gain the fine-grained evolution patterns of this region. This finding provides a new strategy for special applications such as weather forecasting, where high-resolution prediction may be obtained for a local region using a learning machine trained on low-resolution data from the same region. Presently, this strategy is just illustrated for relatively simple spatiotemporal systems. We hope a more systematic approach can be developed to treat real-life spatiotemporal systems in the future.

Acknowledgments

We acknowledge support from the NSFC (Grants No. 11 975 189, No. 11 975 190).

References

[1] Bauer P, Thorpe A and Brunet G 2015 *Nature* **525** 47
[2] Morim J *et al* 2019 *Nat. Clim. Change* **9** 711
[3] Hochman A, Alpert P, Harpaz T, Saaroni H and Messori G 2019 *Science Advances* **5** eaau0936
[4] Bellman R 1966 *Science* **153** 34
[5] Voosen P 2017 *Science* **356** 128
[6] Ling J, Kurzawski A and Templeton J 2016 *J. Fluid Mech.* **807** 155
[7] Beck C, Weinan E and Jentzen A 2019 *Journal of Nonlinear Science* **29** 1563
[8] Han J, Jentzen A and Weinan E 2018 *Proc. Natl Acad. Sci.* **115** 8505
[9] Sirignano J and Spiliopoulos K 2018 *J. Comput. Phys.* **375** 1339
[10] Raissi M, Perdikaris P and Karniadakis G 2019 *J. Comput. Phys.* **378** 686
[11] Qiu X, Zhang L, Ren Y, Suganthan P N and Amaratunga G 2014 *2014 IEEE symposium on computational intelligence in ensemble learning (CIEL) IEEE* pp 1–6
[12] Grover A, Kapoor A and Horvitz E 2015 *Proceedings of the XXI ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* pp 379–86
[13] Sønderby C K, Espeholt L, Heek J, Dehghani M, Oliver A, Salimans T, Agrawal S, Hickey J and Kalchbrenner N 2020 arXiv:2003.12140
[14] Kuramoto Y 1975 *International Symposium on Mathematical Problems in Theoretical Physics* (Berlin: Springer) pp 420–2
[15] Zhang X, Bi H, Guan S, Liu J and Liu Z 2016 *Phys. Rev.* E **94** 012204
[16] Hu X, Boccaletti S, Huang W, Zhang X, Liu Z, Guan S and Lai C-H 2014 *Sci. Rep.* **4** 7262
[17] Barrio R, Varea C, Aragón J and Maini P 1999 *Bull. Math. Biol.* **61** 483
[18] Aragon J L, Barrio R A, Woolley T E, Baker R E and Maini P K 2012 *Phys. Rev.* E **86** 026201
[19] Zhao H 2017 A general theory for training learning machine arXiv:1704.06885
[20] Zhao H 2021 *Science China Physics, Mechanics & Astronomy* **64** 270511
[21] Hochreiter S and Schmidhuber J 1997 *Neural Comput.* **9** 1735
[22] Pathak J, Hunt B, Girvan M, Lu Z and Ott E 2018 *Phys. Rev. Lett.* **120** 024102
[23] Pathak J, Wikner A, Fussell R, Chandra S, Hunt B R, Girvan M and Ott E 2018 *Chaos* **28** 041101
[24] Zhang H, Fan H, Wang L and Wang X 2021 *Phys. Rev.* E **104** 024205
[25] Fan H, Jiang J, Zhang C, Wang X and Lai Y-C 2020 *Physical Review Research* **2** 012080
[26] Takens F 1981 *Dynamical systems and turbulence, Warwick* 1980 (Berlin: Springer) pp 366–81
[27] Ma H, Leng S, Aihara K, Lin W and Chen L 2018 *Proc. Natl. Acad. Sci.* **115** E9994