Experimental demonstration of adaptive model selection based on reinforcement learning in photonic reservoir computing

Ryohei Mito¹, Kazutaka Kanno¹a), Makoto Naruse², and Atsushi Uchida¹

¹Department of Information and Computer Sciences, Saitama University 255 Shimo-Okubo, Sakura-ku, Saitama City, Saitama, 338-8570, Japan
²Department of Information Physics and Computing, Graduate School of Information Science and Technology, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8656, Japan

¹a) kkanno@mail.saitama-u.ac.jp

Received July 10, 2021; Revised September 1, 2021; Published January 1, 2022

Abstract: Reservoir computing provides superior information processing ability for a time series prediction based on appropriate learning prior to task execution. The performance of reservoir computing, however, may degrade if the characteristics of the input signal drastically change over time because the internal model of reservoir computing deviates from the subjected input signal trains. We propose a method for adaptive model selection using reinforcement learning in electro-optic delay-based reservoir computing. We experimentally show that an adaptive model selection is effective when different dynamical models for the input signals change dynamically over time.

Key Words: reservoir computing, reinforcement learning, semiconductor laser, delayed dynamical system, time series prediction

1. Introduction
In recent years, machine learning has attracted significant attention. Deep learning has been intensively studied and exploited in various fields [1, 2]. In addition to the significant and remarkable development of deep learning, its high learning and energy cost has been seriously debated [3]. Therefore, a variety of research is being explored, focusing on new hardware for deep learning to solve such issues [4]. Furthermore, fundamentally novel architectures have been recently examined to overcome these issues in conventional machine learning.

Hardware implementation for machine learning has been intensively studied in the field of photonics [5]. The advantages of a photonic implementation for machine learning are high speed and high energy efficiency [6]. In recent years, numerous studies on the photonic implementation of machine
learning have been conducted, in association with the concept of optical accelerators [7]. Photonic reservoir computing is one of these hardware implementations for machine learning [8–10]. In this scheme, a network is virtually implemented using an optical system where information processing is conducted based on the system response to a signal injection. As a feature of reservoir computing, coupling weights for input and network do not need to be trained, and only the output weights are trained. Therefore, the learning cost of reservoir computing is much lower than that of conventional deep learning. Reservoir computing has recently been proposed in a variety of physical platforms, where reservoir networks are implemented in nonlinear hardware systems [11–13]. Optical reservoir computing is one such implementation. Fast information processing at processing rates of beyond 1 GB/s has been accomplished using photonic reservoir computing in benchmark tasks such as chaotic time series prediction and speech recognition [8].

Most supervised learning schemes, including reservoir computing, face difficulties when dealing with a variety of input signal models that change over time. For example, consider a situation in which the source model that produces input signals subjected to reservoir computing is dynamically switched over time. In other words, environmental changes cause such variations in the input information, which may deviate significantly from the properties of the training datasets. In this case, reservoir computing cannot process information correctly because the characteristics of the input signals do not meet the property of the data used during the training phase. In addition, it may be difficult to train a single reservoir weight to achieve correct information processing for all different models or arbitrary environmental conditions.

To overcome such an issue, we consider the preparation of multiple reservoir weights, each of which is trained for different input signal models. It is therefore necessary to select a proper reservoir weight when environmental conditions or input models are temporally switched. We recently have proposed an adaptive model selection scheme for selecting a proper reservoir weight [14]. Therein, one of the reservoir weights is selected based on the photonic decision-making method, or photonic reinforcement learning [15]. The scheme of adaptive model selection has been numerically demonstrated in a chaotic time series prediction task, where the input model is switched in time. The scheme successfully demonstrated adaptive model selections for a switching time series. However, the study was limited to two input models, i.e., the case of three or more models has yet to be examined. In addition, the scheme has not been experimentally investigated based on photonic reservoir computing.

In this study, we experimentally demonstrate model selection using reinforcement learning in photonic reservoir computing. Photonic reservoir computing is realized using electro-optic delay systems in our experiment, which has been studied for the implementation of reservoir computing [16]. During the experiment, a temporally switching time series is generated from three or four models, i.e., the Rössler model, Lorenz model, Mackey-Glass model, and electro-optic delay system. A time series prediction for the switching time series is executed using an electric-optic delay system. In our model selection scheme, there are different sets of reservoir weights for each input model, while one of the sets of the reservoir weights is selected for reservoir computing. The number of sets corresponds to the number of input models. The number of weights corresponds to the number of node states. Each set is trained for different input models. To select one of the sets, we used a model selection scheme based on reinforcement learning. We also investigate our model selection scheme for the case of a large number of models through a numerical simulation, where the performance of the scheme depending on the number of input models is examined.

2. Methods
We first describe our scheme for model selection based on reinforcement learning in photonic reservoir computing. Figure 1 shows a schematic diagram of model selection [14]. The scheme is composed of two parts, i.e., photonic reservoir computing and reinforcement learning. We describe both parts in the following subsections.

2.1 Photonic reservoir computing
In this scheme, a reservoir network is replaced with a nonlinear element in photonic reservoir com-
puting with delayed feedback with delay time $\tau$. This concept is called delay-based reservoir computing [17]. Network nodes are virtually implemented in a temporal direction of a nonlinear element, which is called time-multiplexing [17]. In time-multiplexing, we consider a node interval $\theta$. Virtual nodes are defined by temporally dividing the delay time $\tau$ by the node interval $\theta$. The number of nodes $N$ is given by $N = \tau/\theta$. As the advantage of this method, a large number of nodes is unnecessary for implementing a reservoir network.

We consider time-discrete input data $s_n$ ($n = 1, 2, \cdots$ as the discrete time) for reservoir computing. Before the input data are injected, a mask signal $m(t)$ with a period of $T_m$ is multiplied with the input data $s_n$. The mask acts as input weights to the virtual nodes and keeps the dynamics of the reservoir transient states. The mask used in this study is a piecewise step function with node interval $\theta$. The value of the mask is randomly chosen from the set $\{-1, -0.3, 0.3, 1\}$. The input signal multiplied with the mask is given by the following equation:

$$s(t) = \gamma m(t) s_n \quad ((n - 1)T_m \leq t < nT_m),$$

where $\gamma$ is the scaling coefficient. The mask interval $T_m$ is identical to the feedback delay time $\tau$ [17]. However, it is known that a mask interval slightly different from the delay time can improve the performance of reservoir computing [16], where the node interval $\theta$ is introduced for the difference between the such interval and the delay time. We also use the same condition, that is, the mask interval is set to $T_m = \tau + \theta$. Under this condition, the number of nodes is given by $N = (\tau + \theta)/\theta = T_m/\theta$.

When the input signal is injected into the reservoir, a temporal output from the reservoir is produced. Virtual node states $v_j$ ($j = 1, 2, \cdots, N$) are extracted from the temporal output by separating using node interval $\theta$. The reservoir computing output $p(n)$ is given by calculating a weighted linear combination of virtual node states.

$$p(n) = \sum_{j=1}^{N} w_j v_j(n),$$

where $w_j$ is the weight for the $j$-th node state. Here, we consider a vector of which the elements are weights $w_j$ as a set. The weight set is trained by minimizing the mean-square error between the target $\hat{p}(n)$ and the reservoir computing output $p(n)$.

$$\frac{1}{N_{tr}} \sum_{n=1}^{N_{tr}} (p(n) - \hat{p}(n))^2 \to \min,$$

where $N_{tr}$ is the number of input data for training.

### 2.2 Model selection scheme based on reinforcement learning

We describe our model selection scheme based on reinforcement learning. We consider that the task for photonic reservoir computing is a chaotic time series prediction; that is, predicting the signal level at a single point ahead in time. Photonic reservoir computing predicts $u(n+1)$ when $u(n)$ is input into the reservoir. We consider the situation in which the source of the input signal changes over time,
mimicking environmental changes, as shown in Fig. 1. The number of sets of the reservoir weights $w_m$ ($m = 1, 2, \cdots, N_m$, $N_m$ is the number of input models) same as input models is prepared for changes in the input model. Each of these weight sets is trained for different input models. One of the weight sets is selected using adaptive model selection based on reinforcement learning. In adaptive model selection, prediction errors in reservoir computing are utilized to determine which model should be used for time series prediction.

In a previous study [14], the method of decision making based on chaotic laser time series was utilized for two input models, that is, the selection of one of two output weights for reservoir computing was conducted. One of the principles for an adaptive model selection applicable to more than three models is the decision-making method using time-multiplexing of chaotic time series proposed in [18]. We proposed another method for scalable decision making [19], which is used in the present study.

For the method proposed in this study, we prepare the number of different chaotic time series same as the number of input models. The chaotic time series were experimentally generated from a semiconductor laser with optical feedback. The amplitude of the time series is normalized within the range of $0$ to $1$ (see the Appendix A for further details). In the model selection scheme, each time series is assigned to one input model. We denote the signal level of a chaotic time series at $t$ by $c_m(t)$ for the $m$-th input model. Moreover, we introduce a bias $b_m(t)$ for each input model. The initial value of the bias is set to zero, and the sum of the chaotic time series and bias is $f_m(t) = c_m(t) + b_m(t)$. All the time series with bias $f_m(t)$ are compared, and the model with the maximum $f_m(t)$ for $m$ is selected. The selection probabilities of input models are changed by adjusting the bias coefficient $b_m(t)$. For example, when the bias is increased, it is easier to select the model corresponding to the increased bias.

The output of the reservoir computing $p(n)$ is calculated based on the weight corresponding to the selected model. In the previous study, the prediction results with two reservoir weights were continuously calculated [14]. Although it makes the model selection easy, the amount of processing cost increases as the number of models increases, meaning that the scalability is not assured. It should be emphasized that, in the present study, the prediction result $p(n)$ is calculated using only the selected model. The next time step ($n \rightarrow n + 1$) is considered. From the new input data $u(n+1)$ and the predicted output $p(n)$, we obtain the prediction error at time $n$ given by $e(n) = |u(n+1) - p(n)|$. Here, it is necessary to determine whether the model selected in the time step $n$ is correct. A small prediction error indicates that the input signal model and the selected model are matched, whereas a large prediction error implies that a mismatch occurs between them. In the previous study, the correctness of the selected model was determined based on the magnitude of two prediction errors because the prediction results with two reservoir weights were calculated [14]. Conversely, in this study, the prediction is computed using only a single, selected model. To consider the magnitude of the prediction error, we compare it with the average of the past prediction error $e_{ave}(n)$.

$$e_{ave}(n) = \frac{1}{N_{ave}} \sum_{m=1}^{N_{ave}} e(n - m)$$  \hspace{1cm} (4)

where $N_{ave}$ is the number of averaging points. In this study, we fixed the number of averaging points at $N_{ave} = 2$. In the first several steps, the average error $e_{ave}$ cannot be calculated because there are no past prediction results. In this case, we determined that the selected model is inadequate to exploring to explore as many models as possible.

When the input model and the selected model match, it is necessary to proceed with training and achieve an easy selection. With our method, the model selection probability is controlled through the bias $b_m(t)$. In the chaotic time series with bias $f_m(t)$, the time series of laser chaos $c_m(t)$ is the waveform obtained experimentally under the same conditions. Thus, the probability distribution of the amplitudes of the time series is almost identical for each model. If the bias term $b_m(t)$ is zero, all models are selected with identical probability. With our method, the bias is changed based on the prediction error. The model selection probability can be controlled by increasing or decreasing the bias.
We now describe the method for controlling the bias in detail. The bias is updated using the Q-values as follows

$$a_m(t) = Q_m(t) - \frac{1}{n-1} \sum_{n \neq m}^N Q_n(t), \quad (5)$$

$$b_m(t) = \begin{cases} 1 & (ka_m(t) > 1) \\ kb_m(t) & (-1 < ka_m(t) < 1) \\ -1 & (ka_m(t) < -1) \end{cases}, \quad (6)$$

where $Q_m(t)$ is the Q-value of the tug-of-war theory [18, 20]. Here, $k$ is a hyper parameter that adjusts the amount of change in bias and is fixed at $k = 0.6$ in this study. The bias is limited to a value of $-1$ to $1$. The range corresponds to one time series of chaotic waveform $c_m(t)$. Therefore, if the bias becomes 1 for one model and -1 for another, only the model with $b_m(t) = 1$ is selected. The Q-values are updated for the selected and non-selected models as follows:

$$Q_{m'}(t) = \begin{cases} \alpha Q_{m'} + \Delta & (e(t) \leq e_{ave}(t)) \\ \alpha Q_{m'} - \Omega & (e(t) \geq e_{ave}(t)) \end{cases}, \quad (7)$$

$$Q_m(t) = \alpha Q_m(t), \quad (8)$$

where the subscripts $m'$ and $m$ indicate the selected and non-selected models, respectively. Parameter $\alpha$ is referred to as the memory parameter [21, 22] and is set to a value taken from $[0, 1]$. Here, $\alpha = 1$ indicates that the past information of the Q-value does not decay. The value of $\alpha$ is normally set to approximately 1 and in this study was fixed at $\alpha = 0.995$. $\Delta$ and $\Omega$ are the inclement parameters for the Q-values. In the tug-of-war theory, these increment parameters are determined based on the history of probabilities of a correct model selection. In our study, the correctness of the model selection is determined based on the magnitude of the relationship between the past average of the prediction error $e_{ave}$ and the current prediction error $e(t)$. When we describe the estimated probabilities as $P_m$, the increment parameters $\Delta$ and $\Omega$ are given through the following equations:

$$\Omega = P_{top1} + P_{top2},$$

$$\Delta = 2 - (P_{top1} + P_{top2}). \quad (9), (10)$$

The subscripts top1 and top2 represent the highest and second-highest values in the estimated probabilities, respectively. The estimated probability $P_m$ is calculated from $C_m$ and $W_m$. $C_m$ is the number of selections of the input model $m$. $W_m$ is the number of times at which the prediction error obtained from the weight of the selected model $m$ is larger than the past average error given by Eq. (4). The estimated probabilities are given by $P_m = W_m/C_m$. These probabilities refer to the rates at which the current error is larger than the past average error.

The model selection scheme presented here has some important differences from the scheme used in the previous study [14]. The first is the introduction of the bias of chaotic temporal waveforms for reinforcement learning. In the previous study [14], we used a threshold and a laser chaos time series. In that scheme, we selected one of the two models based on the relationship between the threshold and chaotic time series. The scheme can be applied to model selection with two models only. The scheme presented in this study can be applied to three or more models. However, the number of chaotic time series should correspond to the number of input models. The second difference is the calculation of the prediction error corresponding to the selected model and using the average prediction error to determine the correctness of a selected model. In the previous study, two prediction errors corresponding to two models were calculated at every time step, and we determined the correction of the selected model based on the relationship between the prediction errors. The scheme presented in this study needs to calculate only one prediction error, which is effective when the number of input models is large.
3. Experiment on model selection

3.1 Experimental setup for photonic reservoir computing

We show our experimental results on the model selection scheme. In this subsection, we describe our experimental setup for photonic reservoir computing. In the next subsection, experimental results on the model selection scheme in the case of three and four input models are provided.

An electro-optic delay system is one of the implementations of photonic reservoir computing [16]. Electro-optic delay systems have been studied for the investigation of complex dynamics such as dynamical bifurcation and chaos [23]. The experimental setup of the system is shown in Fig. 2. The system is mainly composed of a laser diode (LD), a Mach-Zehnder modulator (MZM), and an optical fiber for the delay. The LD (NTT Electronics, KELD1C5GAAA) is used for an optical source. Two consecutive MZMs (iXBlue, MXAN-LN-10, half-wave-voltage $V_\pi$ of 7 V) are inserted after the LD, where the first one is used for a signal injection, and the second is used for delayed feedback. The MZM provides a nonlinear transfer function $\cos^2(\cdot)$ in the optical response to the input voltage. The optical output from the MZM is divided into two passes by an optical fiber coupler (FC). One is used for signal detection, and the other is used for delayed feedback. In the feedback pass, a fiber with a length of 2 km is inserted for the delay. The total delay time of the system corresponds to 9.8 $\mu$s. The strength of the delayed feedback can be adjusted using an optical attenuator (ATT). The feedback power is detected at a power meter after the fiber coupler (FC). The optical feedback signal is transformed to an electric signal by a photodetector (PD, Newfocus, 1611). The electric signal is fed back into the MZM after the signal is amplified by an electric amplifier (AMP, iXBlue, DR-AN-10-HO).

The feedback strength is important for the performance of reservoir computing [24]. When the feedback strength is increased, the electro-optic system shows a bifurcation in the temporal dynamics, such as periodic and chaos waveforms [25]. In general, system parameters in the reservoir are adjusted to avoid a dynamical bifurcation. Suppose that the reservoir produces periodic and chaotic oscillations. In that case, the reservoir does not have consistency, which is the reproducibility in the temporal response of dynamical systems to the same repeated input signal [26]. Therefore, in this study, the feedback strength is adjusted to the condition just before a dynamical bifurcation occurs.

An input signal for reservoir computing is injected through the first MZM. The signal is generated from an arbitrary waveform generator (AWG, Keysight, 33612A). The amplitude of the signal generated from the AWG is adjusted to the same magnitude as the half-wave voltage of MZM (7 V peak-to-peak). The modulation amplitude corresponds to the range $-\pi/2$ to $\pi/2$ in the response function $\cos^2(\cdot)$ of the MZM when the MZM bias is fixed at $-\pi/4$.

Fig. 2. Experimental setup for photonic reservoir computing. Our reservoir is realized using an electro-optic delay system. In the figure, LD is the laser diode, ISO is the fiber isolator, MZM is the Mach-Zehnder modulator, FC is the fiber coupler, ATT is the fiber attenuator, PM is the optical power meter, PD is the photodetector, AMP is the electric amplifier, and OSC is the digital oscilloscope. The optical fiber with a length of 2 km is introduced for the delay.
Fig. 3. (a) Prediction target waveform generated from three chaotic models. The target waveform is temporally switched every 1000 points. The order of the input models is the Mackey-Glass (MG), Lorenz (L), and Rössler (R) models. (b), (c), and (d) Predicted time series obtained from the weight corresponding to the Rössler, Lorenz, and Mackey-Glass models, respectively. (e), (f), and (g) Prediction errors corresponding to the predicted time series placed above.

A response signal is measured using a digital oscilloscope (OSC, Tektronics, TDS7404B), where the signal is sampled at 100 Megasample/s. The node interval $\theta$ and mask interval $T_m$ for reservoir computing are fixed at 0.2 and 10.0 $\mu$s, respectively. The mask interval is selected as $T_m = \tau + \theta$.

In our experiment, the time series of each model is injected, and the response of the reservoir to the input signal is acquired for training the reservoir weights. In addition, the switching time series for the prediction test is injected into the reservoir, and the response waveform is acquired. For simplicity, in the present study, post-processing (a weighted linear sum of the node states), training for the reservoir weights, and reinforcement learning were conducted offline on a personal computer (PC). Meanwhile, it is possible to conduct the procedure online by controlling the oscilloscope and the arbitrary waveform generator through a PC. A field programmable gate array (FPGA) is another approach for on-line implementation, as reported in [9].

3.2 Experimental results

In the demonstration of our scheme, we use three chaotic models; the Rössler [27], Lorenz [28], and Mackey-Glass models [29]. Three reservoir weights were prepared and is trained for each model. The details of these models are described in the Appendix B. A total 5000 points are used for training. Figure 3(a) shows the temporal switching waveform for prediction test. The waveform is switched at every 1000 points. The prediction test is repeated 100 times, and the order of the models in the switching waveform is changed for each test. The waveform amplitude of the three models was normalized to the range of $-1$ to $1$. The sampling interval for each waveform was determined to match the numbers of points constituting one cycle at the maximum amplitude. The number of points in one average period in the time series is approximately 10 points.

We show the prediction results obtained using the three weights for the switching waveform. Figures 3(b), 3(c), and 3(d) show the predicted waveforms, where the reservoir weight for the Rössler, Lorenz, and Mackey-Glass models is used, respectively. The prediction error corresponding to the predicted waveform is shown in the bottom figures, where the error is calculated from the difference between the target and predicted waveforms $|\hat{p}(n) - p(n)|$. In the bottom figures, it can be seen that the error decreases when the input model corresponding to the reservoir weight is injected. In Fig. 3(e), for example, the prediction error becomes relatively small within $2000 < n \leq 3000$, where
Fig. 4. Example of model selection results. (a) Temporal evolution of the Q-values. (b) Time series of laser chaos with bias. The black, red, and blue curves represent the Rössler, Lorenz, and Mackey-Glass models, respectively. (c) Selected model using our scheme, where the label “R” is the Rössler model, “L” is the Lorenz model, and “M” is the Mackey-Glass model.

We show an example of model selection in Fig. 4. Figure 4(a) shows the temporal evolution of the Q-values. The Q-values evolve based on the prediction errors and through Eq. (7), which means that the Q-value for the selected model increases (decreases) if the prediction error is smaller (larger) than the error averaged for the past. It can be seen that the Q-value corresponding to the input model is the largest. After the Q-value is updated for each time step, the bias $b_m(t)$ varies based on Eqs. (5) and (6), which indicates that the bias increases (decreases) when the corresponding Q-value increases (decreases). Based on the variation of the bias, the chaotic waveform with a bias $f_m(t) = c_m(t) + b_m(t)$ also changes. Figure 4(b) shows the chaotic waveforms with the bias $f(t) = c_m(t) + b_m(t)$. For the first 1000 points, the blue curve, to which the Mackey-Glass model is assigned, is larger than that of the other waveforms. The magnitude relationship between the chaotic waveforms is then switched every 1000 points, where the input model is changed. At each time step, one of the models is selected based on the magnitude relationship between the chaotic waveforms. The selected model at each time step is shown in Fig. 4(c). We observe that the model corresponding to the input model is correctly selected except for several time steps after the input model is switched. All models are selected in a few steps after the input model is switched. This corresponds to the exploration in reinforcement learning. After such an exploration, the model that matches the input model is selected correctly.

The above trials are repeated 100 times to verify different patterns in a chaotic time series. In each trial, different time series are generated using different initial conditions. In the switching time series for testing, the order of the models is randomly changed for each trial. To quantitatively evaluate the correctness of the model selection, we calculate the correct model selection rate, denoted by CMSR$(n)$, which is defined as the ratio of the number of selections of the predicted output corresponding to the target model at time $n$ among 100 trials. If CMSR$(n) = 1$, then the model used for the time series prediction at time $n$ perfectly agrees with the original input model.

Figure 5(a) shows the temporal evolution of CMSR$(n)$ in the case of three input models. The CMSR$(n)$ increases quickly to unity after the prediction begins. When the target model is switched at $n = 1000$ and 2000, CMSR$(n)$ decreases to zero. After the switching, CMSR$(n)$ increases to unity again. Therefore, the correct model is selected adaptively under model switching (i.e., environmental changes). As shown in the figure, the number of time steps required for the CMSR to reach 0.95 for the first time is 55. The average period of a temporal signal obtained from each model is approximately 10 points. Therefore, the CMSR reaches 0.95 for approximately five periods. After the input model is switched at 1000 and 2000 steps, the CMSR reaches at 0.95 again at 1295 and 2292 steps, which corresponds to approximately 300 steps after switching the input model. The number of time steps corresponds to approximately 30 periods in these input models.

It should be noted that the speed of increase in CMSR$(n)$ after the model changes at $n = 1000$ and 2000 decreases in comparison with the increase in CMSR$(n)$ immediately after $n$ is zero. Such behavior is explained as follows. The initial bias value $b_m(t)$ is set to zero. As the learning progresses,
Fig. 5. Temporal evolution of the CMSR in the case of (a) three and (b) four models. The model is switched every 1,000 points.

the bias $b_m(t)$ approaches $+1$ if the model $m$ is the true source model, whereas $b_m(t)$ approaches $-1$ if the model $m$ does not meet the true source model. Therefore, when the source model is switched, certain $b_m(t)$ turns to increase from approximately $-1$ toward $+1$. Hence, it takes more selections until $b_m(t)$ reaches $+1$ after the model switching is induced at $n = 1000$ and 2000, which slows the increase of CMSR($n$) in comparison to the initial startup. We consider that the speed of the adaptation can be accelerated even in the latter switching through further engineering, for example by limiting the biases to a smaller value.

The transition period of CMSR to unity can be accelerated by adjusting the hyper parameters for reinforcement learning. For example, the amount of changes in the biases $k$ can be changed for accelerating the increase of CMSR, where $k$ is given in Eq. (6). If the bias coefficient $k$ is larger, the number of time steps at which the bias saturates to 1 or -1 becomes smaller, which can increase the transition speed of CMSR to unity. However, if the value of $k$ is extremely large, the frequency of exploring the correct model can decrease, which is called the exploration-exploitation dilemma. The coefficient $k$ must be increased under the condition of selecting the correct model. From the viewpoint of reservoir computing, each set of reservoir weights must be customized to the corresponding model to shorten the transition period. In other words, the reservoir parameters are adjusted to reduce the prediction error produced using the weight set corresponding to the input model. In our model selection scheme, reinforcement learning is performed based on the difference in prediction errors. If the difference in prediction errors produced by different weights is large, reinforcement learning becomes easier, and the number of explorations can be reduced. This situation can be achieved by increasing the number of training data points and the number of reservoir nodes.

We also demonstrated the case of four models. The additional model is an optoelectronic delay system [23], which corresponds to the system used in this study except for no external input. The time series used for the prediction test is generated from a numerical model, which is described in the Appendix B. Figure 5(b) shows the temporal evolution of CMSR($n$) in the case of four models. The input model is switched every 1000 steps. The CMSR($n$) increases after the prediction test starts, and the input model is switched. However, the CMSR($n$) does not reach 1 and fluctuates at approximately 0.8. This indicates that the correct model is not always selected. The prediction error of each model can be considered a factor that reduces the CMSR($n$). If the prediction error of each model is insufficiently small, the difference between the prediction errors generated from the reservoir weight corresponding to the correct and different models becomes small. With this method, the correctness of the selected model is decided based on the magnitude relationship between the prediction error of the current model and the averaged error for the recent past. Therefore, if the difference between their prediction errors is small, the model corresponding to the input model is not judged to be correct. This issue is numerically investigated in the next section.

In this study, we apply the model selection scheme to one-step ahead prediction. It is considered that
the scheme can be applied to prediction of several steps if prediction target is within the predictable time, which is defined as the inverse of the maximum Lyapunov exponent. However, the model selection will not work properly if the reservoir computing cannot perform the correct prediction. The model selection quality is related to the prediction error. If the reservoir weights for different models provide almost identical prediction errors, the model selection quality will degrade.

4. Numerical simulation on model selection

4.1 Numerical model

We also numerically demonstrate the model selection scheme to investigate the case of a large number of models and clarify the reason why the CMSR does not reach 1 in the case of four models in the experiment (Fig. 5(b)). The reservoir used in our study is the electro-optic delay system, as shown in Fig. 2. The numerical model corresponding to the experimental setup is shown in Fig. 6. It has been known that the temporal dynamics of the system can be described well by the following delay differential equations [25]:

$$
\tau_L \frac{dx(t)}{dt} = -\left(1 + \frac{\tau_L}{\tau_H}\right)x(t) - y(t) + \beta u(t) \cos^2 \left[\kappa x(t - \tau) + \phi_0\right] + \xi(t),
$$

$$
\tau_H \frac{dy(t)}{dt} = x(t),
$$

where $x$ is the normalized output of the MZM, and $y$ is the integral of $x$. In addition, $\tau_L$ and $\tau_H$ are the time constants describing the low and high-pass filters, which are related to the frequency bandwidth of the components, such as the electric amplifier, MZM, and PD. In addition, $\beta$ is the dimensionless constant that describes the injection strength to the MZM, $\phi$ represent the offset phase of the MZM, $u(t)$ represents signal modulation by the input $s(t)$ through the MZM and is given by $u(t) = \cos^2(\pi/4(s(t) - 1))$. In addition, $\xi(t)$ is white Gaussian noise with the properties $\langle \xi(t) \xi(t_0) \rangle = \delta(t - t_0)$, where $\langle \cdot \rangle$ denotes the ensemble average and $\delta(t)$ is the Dirac’s delta function.

In our numerical simulation, the number of virtual nodes $N = 50$, and the node interval $\theta = 0.20 \mu s$. Then, the delay time is given by $\tau = (N - 1)\theta = 9.80 \mu s$. The offset phase is $\phi = -\pi/4$. We fix the following parameters as $1/(2\pi\tau_L) = 100$ MHz, $1/(2\pi\tau_H) = 5$ kHz, $\beta = 1$, and $\kappa = 0.9$.

4.2 Numerical results

Figure 7(a) shows the CMSR($n$) in the case of four models. The input models are the same as in Fig. 5(b). The input model switches every 1000 points. It can be seen that the CMSR($n$) reaches 1 immediately after the model is switched. During the experiment, the CMSR($n$) does not reach unity in the case of four of the models; as shown in Fig. 5(b), however, the CMSR($n$) reaches unity in the numerical simulation. We also show the case of six models in Fig. 7(b). The additional two models are the Lorenz models with different parameters. The parameter changed in the Lorenz model is represented as $\mu_L$ (see the Appendix for details) and is varied at $\mu_2 = 50$ and $\mu_3 = 140$, where the Lorenz model produces chaotic dynamics in both cases. The parameter value of the Lorenz model used in the first model is represented as $\mu_1 = 28$. In Fig. 7(b), it can be seen that the CMSR($n$) does not reach unity. This indicates that the performance of the model selection deteriorates when the number of input models increases. The dependence of the performance agrees with our experimental result. The reason why the CMSR becomes worse is the fact that the added models are the Lorenz

![Fig. 6. Schematic model of the electro-optic delay system used in our numerical simulation.](image-url)
model with different parameters. In our previous study, the model selection was investigated using the Rössler model with two different parameters, and it was confirmed that model selection is more difficult as the two parameters become close to each other [14]. The temporal dynamics produced from the same model are relatively similar, even if the parameters are different.

Some prediction errors generated using the reservoir weights are shown in Fig. 8(a). The number of input models is six, and the input models used for the calculation are the same as Fig. 7(b). The errors are averaged over the past 20 points. In the figure, black, red, blue, light green, light blue, and orange represent the Rössler, Lorenz 1 (\(\mu_1\)), Mackey-Glass, optoelectronic, Lorenz 2 (\(\mu_2\)), and Lorenz 3 (\(\mu_3\)) models, respectively. The input model is indicated in the top of the figure. The error produced with the weight set corresponding to the input model is the smallest. Figure 8(b) shows the selected model corresponding to Fig. 8(a). Only the model corresponding to the input is selected when the Rössler, Lorenz 1, Mackey-Glass, and optoelectronic models are used as inputs. When Lorenz 2 and 3 are used as inputs, all models are selected, which indicates that the correct model is not selected. When the Lorenz 2 is used as the input model (\(3000 < n \leq 4000\)), the error represented by the light blue curve is close to that of the red curve. It is assumed that these errors are caused by the similar temporal dynamics of Lorenz 1 and 2, which leads to the selection
Table I. The prediction error (NMSE) for each model in the case of $N = 50$ and 200.

| $N$ | Rössler | Lorenz ($\mu_1$) | Mackey-Glass | Electro-optic delay | Lorenz ($\mu_2$) | Lorenz ($\mu_3$) |
|-----|---------|------------------|--------------|---------------------|------------------|------------------|
| 50  | 0.0058  | 0.0083           | 0.0073       | 0.0120              | 0.038            | 0.22             |
| 200 | 0.0020  | 0.0058           | 0.0017       | 0.0081              | 0.026            | 0.12             |

of non-correct model. When the Lorenz 3 model is used as input ($1000 < n \leq 2000$), the error represented by the orange curve is relatively large ($\sim 0.15$). Because of this large error, the differences from other errors become small, which leads to the selection of non-correct model.

From Fig. 7, it was also confirmed in the numerical simulation that the model selection deteriorates. That is, the CMSR does not reach unity when the number of models increases. One of the reasons is the lack of the prediction performance of reservoir computing. Our model selection scheme utilizes the knowledge of the difference between errors in the waveforms predicted using different weights. The correctness of the model selection is decided based on the magnitude relationship between prediction errors. When the difference between the prediction errors is small, the selected model can be judged as a wrong model even if the selected model is correct. Therefore, we investigate the dependence of the performance of the model selection on the prediction error of the reservoir computing. The prediction error can be varied by changing the number of nodes in reservoir computing. The number of nodes is directly related to the performance of reservoir computing, and a large number of nodes can improve such performance [30]. The number of nodes $N$ is increased from 50 to 200. To increase the number of nodes to 200, the feedback delay time of the reservoir and the node interval are set to $\tau = 9.95 \mu s$ and $\theta = 0.05 \mu s$, respectively.

The prediction error, i.e., normalized mean squared error (NMSE) is calculated from the following equation:

$$\text{NMSE} = \frac{1}{N_{\text{test}} \sigma^2} \sum_{n=1}^{N_{\text{test}}} (\hat{p}(n) - p(n))^2,$$

where $N_{\text{test}}$ is the number of points in a predicted time series, and $\sigma$ is the standard deviation of the target waveform $\hat{p}(n)$. Table I shows the prediction error of each model when the number of nodes $N$ is 50 and 200. It can be seen that the prediction error (NMSE) is reduced by increasing the number of nodes to 200 in all models.

The results of the model selection (CMSR) for different numbers of nodes are shown in Fig. 9(a). The black and red curves show the cases of $N = 50$ and $N = 200$, respectively. It can be seen that the CMSR for the red curve ($N = 200$) reaches close to 1, compared with the black curve ($N = 50$). Therefore, the increase in the number of nodes improves the performance of the model selection. Although having a small perturbation, the CMSR can converge to 1 if the number of nodes further increases.

We quantitatively evaluate the dependence of the CMSR on the number of reservoir nodes of $N = 50$ and 200. Figure 9(b) shows the CMSR for different numbers of models, where the CMSR is averaged for 100 time steps just before the input model is switched. The black curve with the circles and the red curve with the diamonds represent numbers of nodes $N = 50$ and $N = 200$, respectively. We also show the result of $N = 500$ as the blue curve with the triangles, where we use a feedback delay time of $\tau = 9.98 \mu s$ and a node interval of $\theta = 0.02 \mu s$ to increase the number of nodes. In addition, an additional model is introduced to investigate the case of the seven models. The additional model is the Rössler model with a different parameter value, where $\sigma_2 = 0.6$ is used for parameter $\sigma$ (see the Appendix for detail). The original parameter value is represented as $\sigma_1 = 0.2$.

We can observe that the CMSR decreases as the number of models increases. In the case of $N = 50$, the CMSR decreases from unity when the number of models is more than four. With $N = 200$ and 500, the CMSR can reach unity for five and six models, respectively. A slight improvement can be observed when the number of models is six for $N = 500$, as compared with $N = 200$. That is, the increase in the number of nodes in the reservoir improves the performance of the model selection. However, the CMSR does not perfectly reach unity when the number of models is seven, even when the number of nodes is $N = 500$. In this case, incorrect selections are frequently observed when the
Fig. 9. (a) CMSR in the case of six input models. The black and red curves represent \( N = 50 \) and 200, respectively. (b) Dependence of the CMSR on the number of input models. The CMSR is averaged for 100-time steps just before the input model is switched. The black solid curve with circles, the red dashed curve with diamonds, and the blue dotted curve with triangles represent reservoir node numbers of \( N = 50, 200, \) and 500, respectively.

input model is the Rössler model with a parameter of \( \sigma_1 = 0.2 \). Incorrect selections are observed under the condition in which the Rössler model with a parameter of \( \sigma_2 = 0.6 \) is introduced. Thus, the model selection is considered difficult because the same model is added. In particular, the Rössler model with the two different parameter values produces similar temporal dynamics. The prediction error for the model with a different parameter value is relatively small. Therefore, the correct model cannot be selected using the Rössler model with two different parameter values. This may be able to be solved using additional information, such as another dynamical variable. However, based on a numerical simulation, we show that our model selection scheme can be successful when applying up to six models.

5. Conclusions

We experimentally and numerically demonstrated a dynamic model selection using reinforcement learning in photonic reservoir computing. Chaotic time series prediction was demonstrated using photonic reservoir computing, where chaotic models used for generating the time series subjected to the reservoir were switched over time. Appropriate pre-trained models were required to minimize the prediction errors. Three and four chaotic models for the prediction task were used during the experiment. It was shown that the correct model was selected using our model selection scheme even if the input models were temporally switched. In the numerical study, we accomplished correct model selections for up to six models. In the case of seven models, our scheme achieved a high correct model selection rate, although the rate did not reach unity; namely, some imperfections remains. We analyzed the underlying reason why the rate did not reach unity and concluded that it was due to the similarity of the input models used for generating the switching time series when applying the Rössler model with different parameter values. This issue can be improved using additional input information (for example, other dynamical variables) for reservoir computing to identify the temporal dynamics of the same model with different parameter values. Our proposed scheme can be applied to applications such as load forecasting, multi-objective control, and signal recovery in communications when environmental changes or diverse types of input signals are expected.

Acknowledgments

This work was supported in part by Grants-in-Aid for Scientific Research (JSPS KAKENHI, Grant Nos. JP19H00868, JP20K15185, and JP20H00233) from the Japan Society for the Promotion of Science, (JST CREST, Grant No. JPMJCR17N2), and the Telecommunications Advancement Foundation.
Appendix

A. Generation of laser chaotic outputs for decision making

The chaotic time series of laser outputs used for decision making were experimentally obtained from a semiconductor laser with optical feedback. Semiconductor lasers can produce chaotic intensity fluctuations by delayed optical feedback. Optical feedback is realized using an external fiber reflector [31]. The chaotic output was detected using a photodetector and sampled by a high-speed digital oscilloscope. The sampling interval of the digital oscilloscope was 10 ps. In this study, decision making is conducted at a sampling interval of 50 ps.

B. Chaotic models for prediction targets

The Rössler, Lorenz, Mackey-Glass, and optoelectronic delay systems were used as input models for the prediction targets. The Rössler and Lorenz models can produce chaotic dynamics. The Mackey-Glass model is a delayed dynamical system and has been used for prediction tests in reservoir computing. The optoelectronic delay system is also a delayed dynamical system and is used as the reservoir in this study.

The temporal dynamics of the Rössler model is represented in the following equations [27]:

\[
\frac{dx_R}{dt} = -y_R - z_R, \tag{B-1}
\]
\[
\frac{dy_R}{dt} = x_R + 0.2y_R, \tag{B-2}
\]
\[
\frac{dz_R}{dt} = \sigma + x_R z_R - 5.7z_R, \tag{B-3}
\]

where \(\sigma\) is the parameter and is set to \(\sigma_1 = 0.2\) or \(\sigma_2 = 0.6\) for two different input models. The time series of \(x_R\) is used for the prediction test in reservoir computing.

The temporal dynamics of the Lorenz model is represented in the following equations [28]:

\[
\frac{dx_L}{dt} = 10(y_L - x_L), \tag{B-4}
\]
\[
\frac{dy_L}{dt} = -x_L z_L + \mu x_L - y_L, \tag{B-5}
\]
\[
\frac{dz_L}{dt} = x_L y_L - \frac{8}{3}z_L, \tag{B-6}
\]

where \(\mu\) is the parameter and is set to \(\mu_1 = 28, \mu_2 = 50,\) or \(\mu_3 = 140\) for three different input models. The time series of \(x_L\) is used for the prediction test in reservoir computing.

The temporal dynamics of the Mackey-Glass is represented in the following equation [29]:

\[
\frac{dx_M}{dt} = \frac{0.2x_M(t - \tau_M) - 0.1x_M(t)}{1 + x_M^10(t - \tau_M)} - 0.1x_M(t), \tag{B-7}
\]

where \(\tau_M\) is the feedback delay time and fixed at \(\tau_M = 17.0\).

The temporal dynamics of the electro-optic delay system is represented in the following equations [32]:

\[
\tau_L \frac{dx_o}{dt} = \left(1 + \frac{\tau_H}{\tau_L}\right) x_o(t) - y_o(t) + \beta \cos^2 \left[x_o(t - \tau_o) - \frac{\pi}{4}\right], \tag{B-8}
\]
\[
\tau_H \frac{dy_o}{dt} = x_o(t) \tag{B-9}
\]

where the parameter values are fixed at \(\tau_o = 9.0\) ns, \(1/(2\pi\tau_L) = 100\) MHz, \(1/(2\pi\tau_H) = 100\) kHz, and \(\beta = 3.4\). The time series of \(x_o\) is used for the prediction test in reservoir computing.

The time series generated from these models are normalized to the range \([-1, 1]\) using the maximum and minimum time series. In each model, the sampling interval used for generating the time series is adjusted such that the number of points in one average period of the time series is approximately 10 points.
References

[1] A. Shrestha and A. Mahmood, “Review of deep learning algorithms and architectures,” IEEE Access, vol. 7, pp. 53040–53065, April 2019.
[2] F. Emmert-Streib, Z. Yang, H. Feng, S. Tripathi, and M. Dehmer, “An introductory review of deep learning for prediction models with big data,” Frontiers in Artificial Intelligence, vol. 3, p. 4, February 2020.
[3] N.C. Thompson, K. Greenewald, K. Lee, and G.F. Manso, “The computational limits of deep learning,” Preprint at https://arxiv.org/abs/2007.05558, July 2020.
[4] S.K. Esser, P.A. Merolla, J.V. Arthur, A.S. Cassidy, R. Appuswamy, A. Andreopoulos, D.J. Berg, J.L. McKinstry, T. Melano, D.R. Barch, C. diNolfo, P. Datta, A. Amir, B. Tabata, M.D. Flickr, and D.S. Modha, “Convolutional networks for fast, energy-efficient neuromorphic computing,” Proceedings of the National Academy of Sciences, vol. 113, no. 41, pp. 11441–11446, October 2016.
[5] Y. Shen, N.C. Harris, S. Skirlo, M. Prabhu, T. Baehr-Jones, M. Hochberg, X. Sun, S. Zhao, H. Larochelle, D. Englund, and M. Sojačić, “Deep learning with coherent nanophotonic circuits,” Nature Photonics, vol. 11, no. 7, pp. 441–446, June 2017.
[6] B.J. Shastry, A.N. Tait, T. Ferreira deLima, W.H.P. Pernice, H. Bhaskaran, C.D. Wright, and P.R. Prucnal, “Photonics for artificial intelligence and neuromorphic computing,” Nature Photonics, vol. 15, no. 2, pp. 102–114, January 2021.
[7] K. Kitayama, M. Notomi, M. Naruse, K. Inoue, S. Kawakami, and A. Uchida, “Novel frontier of photonics for data processing–photonic accelerator,” APL Photonics, vol. 4, no. 9, p. 090901, September 2019.
[8] D. Brunner, M.C. Soriano, C.R. Mirasso, and I. Fischer, “Parallel photonic information processing at gigabyte per second data rates using transient states,” Nature Communications, vol. 4, no. 1, p. 1364, June 2013.
[9] P. Antonik, F. Duport, M. Hermans, A. Smerieri, M. Haelterman, and S. Massar, “Online training of an opto-electronic reservoir computer applied to real-time channel equalization,” IEEE Transactions on Neural Networks and Learning Systems, vol. 28, no. 11, pp. 2686–2698, November 2017.
[10] C.R.S. Williams, T.E. Murphy, R. Roy, F. Sorrentino, T. Dahms, and E. Schöll, “Experimental observations of group synchrony in a system of chaotic optoelectronic oscillators,” Phys. Rev. Lett., vol. 110, no. 6, p. 064104, February 2013.
[11] G. Tanaka, T. Yamane, J.B. Héroux, R. Nakane, N. Kanazawa, S. Takeda, H. Numata, D. Nakano, and A. Hirose, “Recent advances in physical reservoir computing: A review,” Neural Networks, vol. 115, pp. 100–123, March 2019.
[12] K. Nakajima, H. Hauser, T. Li, and R. Pfeifer, “Information processing via physical soft body,” Scientific Reports, vol. 5, no. 1, p. 10487, May 2015.
[13] K. Nakajima, “Physical reservoir computing—an introductory perspective,” Japanese Journal of Applied Physics, vol. 59, no. 6, p. 060501, May 2020.
[14] K. Kamio, M. Naruse, and A. Uchida, “Adaptive model selection in photonic reservoir computing by reinforcement learning,” Scientific Reports, vol. 10, no. 1, p. 10062, June 2020.
[15] M. Naruse, Y. Terashima, A. Uchida, and S.J. Kim, “Ultrafast photonic reinforcement learning based on laser chaos,” Scientific Reports, vol. 7, no. 1, p. 8772, April 2017.
[16] Y. Paquot, F. Duport, A. Smerieri, J. Dambre, B. Schrauwen, M. Haelterman, and S. Massar, “Optoelectronic reservoir computing,” Scientific Reports, vol. 2, no. 1, p. 287, February 2012.
[17] L. Appeltant, M.C. Soriano, G.V. derSande, J. Danckaert, S. Massar, J. Dambre, B. Schrauwen, C.R. Mirasso, and I. Fischer, “Information processing using a single dynamical node as a complex system,” Nature Communications, vol. 2, no. 1, p. 468, September 2011.
[18] M. Naruse, T. Mihana, H. Hori, H. Saigo, K. Okamura, M. Hasegawa, and A. Uchida, “Scalable photonic reinforcement learning by time-division multiplexing of laser chaos,” Scientific Reports, vol. 8, no. 1, p. 10890, July 2018.
[19] K. Morijiri, T. Mihana, A. Oda, R. Iwami, K. Kanno, M. Naruse, and A. Uchida, *Proc. NOLTA ’20*, vol. 1, pp. 290–293, November 2020.

[20] S.J. Kim, M. Aono, and E. Nameda, “Efficient decision-making by volume-conserving physical object,” *New Journal of Physics*, vol. 17, no. 8, p. 083023, August 2015.

[21] S.J. Kim, M. Naruse, M. Aono, M. Ohtsu, and M. Hara, “Decision maker based on nanoscale photo-excitation transfer,” *Scientific Reports*, vol. 3, no. 1, p. 2370, December 2013.

[22] T. Mihana, Y. Terashima, M. Naruse, S.J. Kim, and A. Uchida, “Memory effect on adaptive decision making with a chaotic semiconductor laser,” *Complexity*, vol. 2018, p. 4318127, April 2018.

[23] L. Larger and J.M. Dudley, “Nonlinear dynamics: Optoelectronic chaos,” *Nature*, vol. 465, no. 7294, pp. 41–42, May 2010.

[24] Y.K. Chembo, D. Brunner, M. Jacquot, and L. Larger, “Optoelectronic oscillators with time-delayed feedback,” *Rev. Mod. Phys.*, vol. 91, no. 3, p. 035006, September 2019.

[25] T.E. Murphy, A.B. Cohen, B. Ravoori, K.R.B. Schmitt, A.V. Setty, F. Sorrentino, C.R.S. Williams, E. Ott, and R. Roy, “Complex dynamics and synchronization of delayed-feedback nonlinear oscillators,” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 368, no. 1911, pp. 343–366, January 2010.

[26] A. Uchida, R. McAllister, and R. Roy, “Consistency of nonlinear system response to complex drive signals,” *Phys. Rev. Lett.*, vol. 93, no. 24, p. 244102, December 2004.

[27] O.E. Rössler, “An equation for continuous chaos,” *Physics Letters A*, vol. 57, no. 5, pp. 397–398, July 1976.

[28] E.N. Lorenz, “Deterministic nonperiodic flow,” *Journal of the Atmospheric Sciences*, vol. 20, pp. 130–141, August 1963.

[29] M.C. Mackey and L. Glass, “Oscillation and chaos in physiological control systems,” *Science*, vol. 197, no. 4300, pp. 287–289, July 1977.

[30] M. Dale, S. O’Keefe, A. Sebald, S. Stepney, and M.A. Trefzer, “Reservoir computing quality: connectivity and topology,” *Natural Computing*, vol. 20, no. 2, pp. 205–216, June 2021.

[31] A. Uchida, *Synchronization of chaos in lasers -Applications of Nonlinear Dynamics and Synchronization-*, Wiley-VCH, 2012.

[32] J. Bueno, D. Brunner, M.C. Soriano, and I. Fischer, “Conditions for reservoir computing performance using semiconductor lasers with delayed optical feedback,” *Optics Express*, vol. 25, no. 3, pp. 2401–2412, February 2017.