

ARTICLE OPEN



Universal scaling between wave speed and size enables nanoscale high-performance reservoir computing based on propagating spin-waves

Satoshi Iihama^{1,2}, Yuya Koike^{2,3,4}, Shigemi Mizukami^{2,5} and Natsuhiko Yoshinaga^{2,4}✉

Physical implementation of neuromorphic computing using spintronics technology has attracted recent attention for the future energy-efficient AI at nanoscales. Reservoir computing (RC) is promising for realizing the neuromorphic computing device. By memorizing past input information and its nonlinear transformation, RC can handle sequential data and perform time-series forecasting and speech recognition. However, the current performance of spintronics RC is poor due to the lack of understanding of its mechanism. Here we demonstrate that nanoscale physical RC using propagating spin waves can achieve high computational power comparable with other state-of-art systems. We develop the theory with response functions to understand the mechanism of high performance. The theory clarifies that wave-based RC generates Volterra series of the input through delayed and nonlinear responses. The delay originates from wave propagation. We find that the scaling of system sizes with the propagation speed of spin waves plays a crucial role in achieving high performance.

npj Spintronics (2024)2:5; <https://doi.org/10.1038/s44306-024-00008-5>

INTRODUCTION

Non-local magnetization dynamics in a nanomagnet, spin-waves, can be used for processing information in an energy-efficient manner since spin-waves carry information in a magnetic material without Ohmic losses¹. The wavelength of the spin-wave can be down to the nanometer scale, and the spin-wave frequency becomes several GHz to THz frequency, which are promising properties for nanoscale and high-speed operation devices. Recently, neuromorphic computing using spintronics technology has attracted great attention for the development of future low-power consumption AI². Spin-waves can be created by various means such as magnetic field, spin-transfer torque, spin-orbit torque, voltage induced change in magnetic anisotropy and can be detected by the magnetoresistance effect³. Therefore, neuromorphic computing using spin waves may have a potential of realizable devices.

Reservoir computing (RC) is a promising neuromorphic computation framework. RC is a variant of recurrent neural networks (RNNs) and has a single layer, referred to as a reservoir, to transform an input signal into an output⁴. In contrast with the conventional RNNs, RC does not update the weights in the reservoir. Therefore, by replacing the reservoir of an artificial neural network with a physical system, for example, magnetization dynamics⁵, we may realize a neural network device to perform various tasks, such as time-series prediction^{4,6}, short-term memory^{7,8}, pattern recognition, and pattern generation. Several physical RC has been proposed: spintronic oscillators^{5,9}, optics¹⁰, photonics^{11,12}, memristor^{13–15}, field-programmable gate array¹⁶, fluids, soft robots, and others (see the following reviews for physical reservoir computing^{17–19}). Among these systems, spintronic RC has the advantage in its potential realization of nanoscale devices at high speed of GHz frequency with low power consumption, which may outperform conventional electric

computers in future. So far, spintronic RC has been considered using spin-torque oscillators^{5,9}, magnetic skyrmion²⁰, and spin waves in garnet thin films^{21–24}.

One of the goals of neuromorphic computing is to realize integrated RC chip devices. To make a step forward in this direction, performance of a single RC unit must be studied and significantly more computational power per unit area might be required. For example, even for the prediction task of one-dimensional spatio-temporal chaos requires 10^4 – 10^5 nodes^{6,10}. Spatio-temporal chaos is an irregular dynamical phenomenon expressed by a deterministic equation. Its time series data is complex but still much simpler than the climate forecast. To increase the number of degrees of freedom N with keeping the total system size, the single RC unit ($N \sim 100$ in this study) should be nanoscale²⁵ ideally ~ 100 nm or less. Also, high-speed operation at GHz frequency compatible with semiconductor devices might be desired.

Spintronics has been a promising technology for the development of nanoscale and high-speed operation nonvolatile memory devices. However, the current performance of spintronic RC demonstrated experimentally still remains poor compared with the Echo State Network (ESN)^{7,8}, idealized RC model implemented in conventional computers with random network and nodes updated by the tanh activation function. The biggest issue is a lack of our understanding of how to achieve high performance in the physical RC systems. It has been well studied that for the ESN, the memory capacity (MC) is bounded by N^7 . MC quantifies how many past steps RC can memorize. For the ESN with linear activation function, MC becomes N as long as each node is independent from other nodes. It has also been shown that the information processing capacity (IPC) of RC is bounded by N^{26} . Here, IPC captures the performance of RC in terms of the memory of past information and nonlinear transformation (see also Methods:

¹Frontier Research Institute for Interdisciplinary Sciences (FRIS), Tohoku University, Sendai 980-8578, Japan. ²WPI Advanced Institute for Materials Research (AIMR), Tohoku University, Katahira 2-1-1, Sendai 980-8577, Japan. ³Department of Applied Physics, Tohoku University, Sendai 980-8579, Japan. ⁴MathAM-OIL, AIST, Sendai 980-8577, Japan. ⁵Center for Science and Innovation in Spintronics (CSIS), Tohoku University, Sendai 980-8577, Japan. ✉email: yoshinaga@tohoku.ac.jp

Learning tasks). Besides MC and IPC, the computational capabilities of RC have been measured by the task of approximating the input-output relation of dynamical system models, such as the nonlinear autoregressive moving average (NARMA) model²⁷. The NARMA model describes noisy but correlated output time series from uniformly random input. Since the performance of MC and IPC is correlated with that of the NARMA task²⁸ (see also Methods: Relation of MC and IPC with learning performance and Supplementary Information Section I and II), it is natural to expect that the higher performance of RC is achieved under larger N .

To make N larger, the reservoir has to have a larger degree of freedom, either by connecting discrete independent units or by using continuum media. Using many units may be impractical because we need to make those units and connect them. The latter approach using continuum media is promising because spatial inhomogeneity may increase the number of degrees of freedom of the physical system. In this respect, wave-based computation in continuum media has attracting features. The dynamics in the continuum media have large, possibly infinite, degrees of freedom. In fact, several wave-based computations have been proposed^{29,30}.

The question is how to use the advantages of both wave-based computation and RC to achieve high-performance computing of time-series data. Even though the continuum media may have a large degree of freedom, in practice, we cannot measure all the information. Therefore, the actual N determining the performance of RC is set by the number of measurements of the physical system. There are two ways to increase N ; by increasing the number of physical nodes, N_p , to measure the system (or the number of discrete units)³¹, and by increasing the number of virtual nodes, N_v ^{32,33}. Increasing N_p is a natural direction to obtain more information from the continuum physical system. For spin wave-based RC, so far, higher performance has been achieved by using a large number of input and/or output nodes^{23,24,34}. However, in practice, it is difficult to increase the number of physical nodes, N_p , because it requires more wiring of multiple nodes. To make computational devices, fewer nodes are preferable in terms of cost and effort to make the devices. With these motivations in our mind, in this study, we propose spin-wave RC, in which we extract the information from the continuum media using a small number of physical nodes.

Along this direction, using N_v virtual nodes for the dynamics with delay was proposed to increase N in³². The idea of the virtual nodes is to use the time-multiplexing approach. By repeating the input N_v times at a certain time, we may treat reservoir states during the N_v steps as states in different *virtual* nodes. This idea was applied in optical fibers with a long delay line¹¹ and a network of oscillators with delay³³. Ideally, the virtual nodes enhance the number of degrees by $N = N_p N_v$. However, the increase of $N = N_p N_v$ with N_v does not necessarily improve performance. In fact, RC based on STO (spin torque oscillator) struggle with insufficient performance both in experiments⁹ and simulations³⁵. On the other hand, for the photonic RC, the virtual nodes work effectively; even though it typically uses only one physical node ($N_p = 1$), the performance is high, for example, $MC \approx N_p N_v$. Still, the photonic RC requires a large size of devices due to the long delay line^{11,12}. To use the virtual nodes as effectively as the physical nodes, we need to clarify the mechanism of high performance using the virtual nodes. Nevertheless, the mechanism of high performance remains elusive, and no unified understanding has been made.

In this work, we show nanoscale and high-speed RC based on spin wave propagation with a small number of inputs can achieve performance comparable with the ESN and other state-of-art RC systems. More importantly, by using a theoretical model, we clarify the mechanism of the high performance of spin wave RC. We show the scaling between wave speed and system size to make virtual nodes effective.

RESULTS

Reservoir computing using wave propagation

The basic task of RC is to transform an input signal U_n to an output Y_n for the discrete step $n = 1, 2, \dots, T$ at the time t_n . For example, for speech recognition, the input is an acoustic wave, and the output is a word corresponding to the sound. Each word is determined not only by the instantaneous input but also by the past history. Therefore, the output is, in general, a function of all the past input, $Y_n = g(\{U_m\}_{m=1}^n)$ as in Fig. 1a. The RC can also be used for time-series prediction by setting the output as $Y_n = U_{n+1}$. In this case, the state at the next time step is predicted from all the past data; namely, the effect of delay is included. The performance of the input-output transformation g can be characterized by how much past information does g have, and how much nonlinear transformation does g perform. We will discuss that the former is expressed by MC³⁶, whereas the latter is measured by IPC²⁶ (see also Methods: Relation of MC and IPC with learning performance).

We propose physical computing based on a propagating wave (see Fig. 1b, c). Time series of an input signal U_n can be transformed into an output signal Y_n (Fig. 1a). As we will discuss in the next paragraph, this transformation requires large linear and nonlinear memories; for example, to predict Y_n , we need to memorize the information of U_{n-2} and U_{n-1} (see also Methods: Relation of MC and IPC with learning performance and Supplementary Information Section I). The input signal is injected in the first input node and propagates in the device to the output node spending a time τ_1 as in Fig. 1b. Then, the output may have past information at $t_n - \tau_1$ corresponding to the step $n - m_1$. The output may receive the information from another input at different time $t_n - \tau_2$. The sum of the two pieces of information is mixed and transformed as $U_{n-m_1} U_{n-m_2}$ either by nonlinear readout or by nonlinear dynamics of the reservoir (see next section: Learning with reservoir computing and Methods). We will demonstrate the wave propagation can indeed enhance MC and learning performance of the input-output relationship.

Before explaining our learning strategy described in next section Spin wave reservoir computing, we discuss how to achieve accurate learning of the input-output relationship $Y_n = g(\{U_m\}_{m=1}^n)$ from the data. Here, the output may be dependent on a whole sequence of the input $\{U_m\}_{m=1}^n = (U_1, \dots, U_n)$. Even when both U_n and Y_n are one-variable time-series data, the input-output relationship $g(\cdot)$ may be T -variable polynomials, where T is the length of the time series. Formally, $g(\cdot)$ can be expanded in a polynomial series (Volterra series) such that $g(\{U_m\}_{m=1}^n) = \sum_{k_1, k_2, \dots, k_n} \beta_{k_1, k_2, \dots, k_n} U_1^{k_1} U_2^{k_2} \dots U_n^{k_n}$ with the coefficients $\beta_{k_1, k_2, \dots, k_n}$ ²⁷. Therefore, even for the linear input-output relationship, we need T coefficients in $g(\cdot)$, and as the degree of powers in the polynomials increases, the number of the coefficients increases exponentially. This observation implies that a large number of data is required to estimate the input-output relationship. Nevertheless, we may expect a dimensional reduction of $g(\cdot)$ due to its possible dependence on the time close to the current step, t_n , and on the lower powers. Still, our physical computers should have degrees of freedom $N \gg 1$, if not exponentially large.

The reservoir computing framework is used to handle time-series data of the input \mathbf{U} and the output \mathbf{Y} . In this framework, the input-output relationship is learned through the reservoir dynamics $X(t)$, which in our case, is magnetization at the detectors. The reservoir state at a time t_n is driven by the input at the n th step corresponding to t_n as

$$X(t_{n+1}) = f(X(t_n), U_n) \quad (1)$$

with nonlinear (or possibly linear) function $f(\cdot)$. The output is

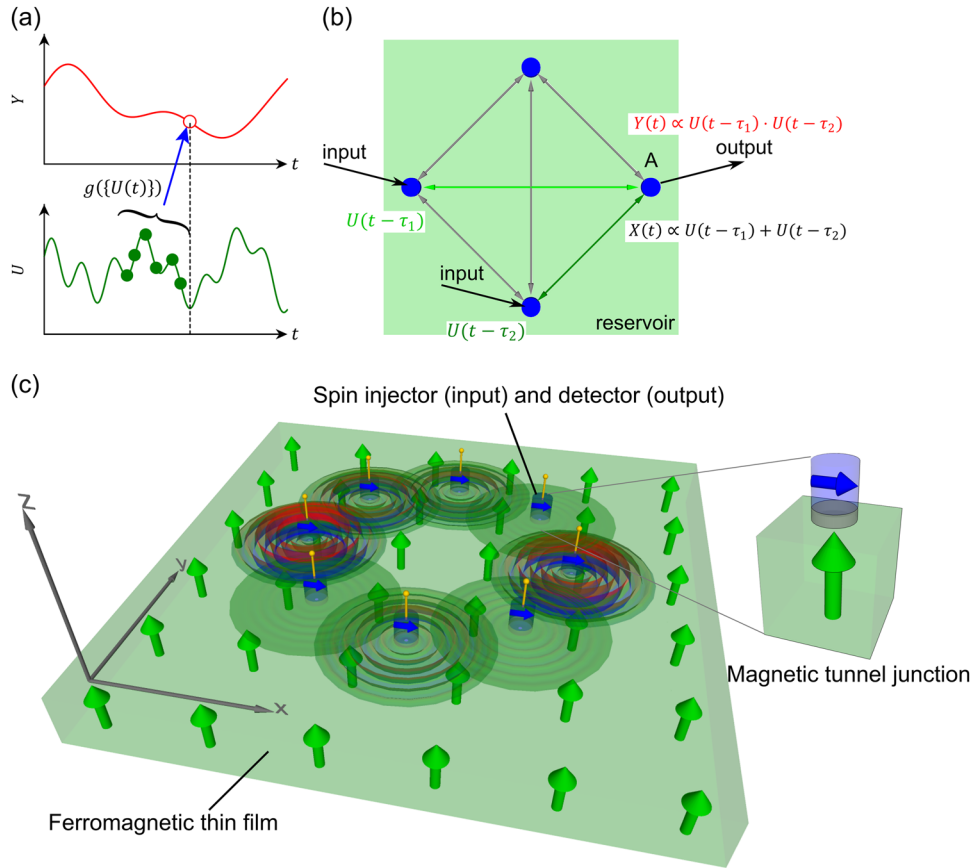


Fig. 1 Illustration of physical reservoir computing and reservoir based on propagating spin-wave network. **a** Schematic illustration of output function prediction by using time-series data. Output signal Y is transformed by past information of input signal U . **b** Schematic illustration of reservoir computing with multiple physical nodes. The output signal at physical node A contains past input signals in other physical nodes, which are memorized by the reservoir. **c** Schematic illustration of reservoir computing based on propagating spin-wave. Propagating spin-wave in ferromagnetic thin film ($\mathbf{m} \parallel \mathbf{e}_z$) is excited by spin injector (input) through spin-transfer torque at multiple physical nodes with reference magnetic layer ($\mathbf{m} \parallel \mathbf{e}_z$) of magnetic tunnel junction. x -component of magnetization is detected by spin detector (output) through magnetoresistance effect using magnetic tunnel junction shown by a cylinder above the thin film at each physical node.

approximated by the readout operator $\psi(\cdot)$ as

$$\hat{Y}_n = \psi(X(t_n)). \quad (2)$$

Our study uses the nonlinear readout $\psi(X(t)) = W_1 X(t) + W_2 X^2(t)$ ^{6,37}. The weight matrices W_1 and W_2 are estimated from the data of the reservoir dynamics $X(t)$ and the true output Y_n , where $X(t)$ is obtained by Eq. (1). With the nonlinear readout, the RC with linear dynamics can achieve nonlinear transformation, as Fig. 1b. We stress that the system also works with linear readout when the RC has nonlinear dynamics³⁷. We discuss this case in Supplementary Information Section V A. In this study, we focus on the second-order polynomial for the readout function. This is because we focus on the second-order IPC to quantify the performance of nonlinear transformation with delay (see Methods: Learning tasks: Information Processing Capacity (IPC)). In Supplementary Information Section I, we clarify that second-order nonlinearity is necessary to achieve high performance for the NARMA10 task. The effect of the order of the polynomials in the readout function is discussed in Supplementary Information Section V B.

Spin wave reservoir computing

The actual demonstration of the spin-wave reservoir computing is shown in Fig. 2 and each step is described in *Learning with reservoir computing* below. First, we demonstrate the spin-wave RC using the micromagnetic simulations (see Methods: Micromagnetic

simulations). Next, to get an insight into the mechanism of learning, we discuss the theoretical model using a response function (see Methods: Theoretical analysis using response function).

Learning with reservoir computing. In this section, we describe each step of our spin-wave RC corresponding to each sub-figure in Fig. 2. Performance of spin-wave reservoir computing is evaluated by NARMA10 task (Fig. 2 (1,5,6)), MC and IPC (Fig. S2 in Supplementary Information Section III), and prediction of chaotic time-series data using Lorenz model (Fig. 5). Details of learning tasks are described in Methods: Learning tasks.

- step 1: preparing input and output time-series data

Our goal using RC is to learn the functional $g(\cdot)$ in $Y_n = g(\{U_m\}_{m=1}^n)$ for the input time series U_n and the output time series Y_n (see Fig. 1a). At the first step, we prepare the ground-truth time series data for the input U_n and output Y_n for the time step $n = 1, \dots, T$. We assume both input and output are discrete time steps in time. In this study, we focus on the continuous value of the input and output, and therefore, $U_n, Y_n \in \mathbb{R}$. To show the performance of our method does not rely on the continuous input, we also demonstrate the performance for the binary and senary inputs in Supplementary Information Section IV. The concrete form of the preparing data will be specified in each task discussed in Methods: Learning tasks.

- step 2: preprocessing the input and translating it into a physical input

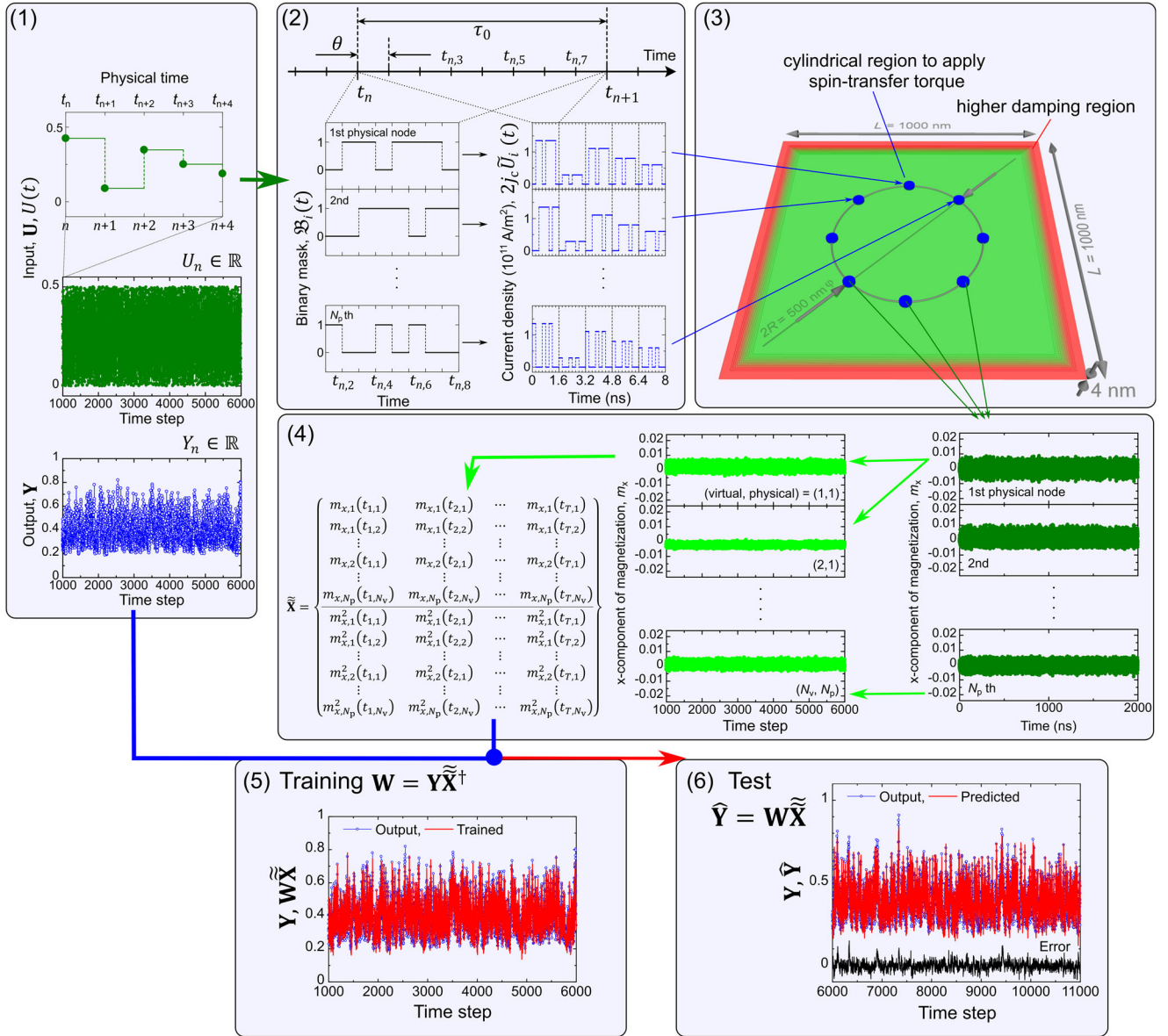


Fig. 2 Schematic illustration of spin-wave reservoir computing using micromagnetic simulation and prediction of NARMA10 task. Input signals \mathbf{U} are transformed into the piece-wise constant input $U(t)$, multiplied by binary mask $B_i(t)$, and transformed into injected current $j(t) = 2j_c \tilde{U}_i(t)$ with $\tilde{U}_i = B_i(t)U(t)$ for the i th physical node. Current is injected into each physical node with the cylindrical region to apply spin-transfer torque and to excite spin-wave. Higher damping regions in the edges of the rectangle are set to avoid reflection of spin-waves. The x -component of magnetization m_x at each physical and virtual node are collected and the extended state $\tilde{\mathbf{X}}$ is constructed from m_x and m_x^2 . Output weights are trained by linear regression. The labels (1) to (6) correspond to step 1 to step 6 in Section Learning with reservoir computing, respectively.

Next, we preprocess the input time series data so that we may drive the physical magnetic system and may perform reservoir computing. The preprocess has two parts: input weights and translation into physical input. First, the input time series U_n is multiplied by the input weight for each physical node. This is necessary because each physical node receives different parts of the input information. In the ESN, the input weights are typically taken from uniform or binary random distribution³⁸. In this study, we use the method of time multiplexing with N_p virtual nodes³². In this case, the input weights are more involved. Details of preprocessing inputs are described in Methods: Details of preprocessing input data

- step 3: driving the physical system

Our RC architecture consists of the update of reservoir state variables $X(t)$ as

$$X(t + \Delta t) = f(X(t), U(t)) \quad (3)$$

and the readout

$$\hat{Y}_n = \mathbf{W} \cdot \tilde{\mathbf{X}}(t_n). \quad (4)$$

We approximate the functional $g(\cdot)$ by the weight \mathbf{W} and the (extended) reservoir states $\tilde{\mathbf{X}}$.

In our system, the reservoir state is magnetization at i th nanocontact (physical node) and at time $t_{n,k}$ (virtual node). The magnetic system is driven by the injected DC current proportional to the input time series \mathbf{U} , described in Methods:

Micromagnetic simulations, with a pre-processing filter. From the resulting spatially inhomogeneous magnetization $\mathbf{m}(\mathbf{x}, t)$, we measure the averaged magnetization at i th nanocontact $\mathbf{m}_i(t)$.

- step 4: measurement, readout, and post-processing

We choose the x -component of magnetization $m_{x,i}$ as a reservoir state, namely, $X_n = \{m_{x,i}(t_{n,k})\}_{i \in [1, N_p], k \in [1, N_v]}$. For the output transformation, we use $\psi(m_{i,x}) = W_{1,i}m_{i,x} + W_{2,i}m_{i,x}^2$. Therefore, the dimension of our reservoir is $2N_pN_v$. The nonlinear output transformation can enhance the nonlinear transformation in reservoir⁶, and it was shown that even under the linear reservoir dynamics, RC can learn any nonlinearity^{37,39}. In Supplementary Information Section V A, we also discuss the linear readout, but including the z -component of magnetization $\mathbf{X} = (m_x, m_z)$. In this case, m_z plays a similar role to m_x^2 .

In our spin wave RC, the reservoir state is chosen as x -component of the magnetization

$$\mathbf{X} = (m_{x,1}(t_n), \dots, m_{x,i}(t_n), \dots, m_{x,N_p}(t_n))^T, \quad (5)$$

for the indices for the physical nodes $i = 1, 2, \dots, N_p$. Here, N_p is the number of physical nodes, and each $m_{x,i}(t_n)$ is a T -dimensional row vector with $n = 1, 2, \dots, T$. We use a time-multiplex network of virtual nodes in RC³², and use N_v virtual nodes with time interval θ . The expanded reservoir state is expressed by $N_pN_v \times T$ matrix $\tilde{\mathbf{X}}$ as (see Fig. 2–(4))

$$\tilde{\mathbf{X}} = \begin{pmatrix} m_{x,1}(t_{n,1}), m_{x,1}(t_{n,2}), \dots, m_{x,1}(t_{n,k}), \dots, m_{x,1}(t_{n,N_v}), \\ \dots, m_{x,i}(t_{n,1}), m_{x,i}(t_{n,2}), \dots, m_{x,i}(t_{n,k}), \dots, m_{x,i}(t_{n,N_v}), \dots, \\ m_{x,N_p}(t_{n,1}), m_{x,N_p}(t_{n,2}), \dots, m_{x,N_p}(t_{n,k}), \dots, m_{x,N_p}(t_{n,N_v}) \end{pmatrix}^T, \quad (6)$$

where $t_{n,k} = ((n-1)N_v + (k-1)\theta)$ for the indices of the virtual nodes $k = 1, 2, \dots, N_v$. The total number of rows is $N = N_pN_v$. We use the nonlinear readout by augmenting the reservoir state as

$$\tilde{\tilde{\mathbf{X}}} = \begin{pmatrix} \tilde{\mathbf{X}} \\ \tilde{\mathbf{X}} \circ \tilde{\mathbf{X}} \end{pmatrix}, \quad (7)$$

where $\tilde{\mathbf{X}}(t) \circ \tilde{\mathbf{X}}(t)$ is the Hadamard product of $\tilde{\mathbf{X}}(t)$, that is, component-wise product. Therefore, the dimension of the readout weights is $N = 2N_pN_v$. As discussed in Eq. (2), we focus on the readout function of the form

$\psi(\mathbf{X}(t)) = W_1\mathbf{X}(t) + W_2\mathbf{X}^2(t)$. When additional nonlinear terms are included in the readout, we may add the rows of $\tilde{\mathbf{X}} \circ \tilde{\mathbf{X}} \cdots \circ \tilde{\mathbf{X}}$ in Eq. (7).

- step 5: training and optimization of the readout weight

The weights in the readout are trained by reservoir variable \mathbf{X} and the output \mathbf{Y} to approximate the ground-truth output \mathbf{Y} by the estimated one $\hat{\mathbf{Y}}$ as Eq. (4). The readout weight \mathbf{W} is trained by the data of the output $\mathbf{Y}(t)$

$$\mathbf{W} = \mathbf{Y} \cdot \tilde{\tilde{\mathbf{X}}}^\dagger \quad (8)$$

where $\tilde{\tilde{\mathbf{X}}}^\dagger$ is pseudo-inverse of $\tilde{\tilde{\mathbf{X}}}$.

Unless otherwise stated, We use 1000 steps of the input time-series as burn-in. After these steps, we use 5000 steps for training and 5000 steps for test for the MC, IPC, and NARMA10 tasks.

- step 6: testing the trained RC

Once we train the output weight \mathbf{W} , we perform a test to estimate the approximated output $\hat{\mathbf{Y}}$ as

$$\hat{\mathbf{Y}} = \mathbf{W} \cdot \tilde{\tilde{\mathbf{X}}}. \quad (9)$$

By comparing the estimated output $\hat{\mathbf{Y}}$ and the ground-truth output \mathbf{Y} , we may quantify the performance of the RC.

Performance of the tasks for MC, IPC, NARMA10, and prediction of chaotic time-series

Figure 3 shows the results of the three tasks. When the time scale of the virtual node θ is small and the damping α is small, the performance of spin wave RC is high. As Fig. 3a shows, we achieve MC ≈ 60 and IPC ≈ 60 . Accordingly, we achieve a small error in the NARMA10 task, NRMSE ≈ 0.2 (Fig. 3c). These performances are comparable with state-of-the-art ESN with the number of nodes ~ 100 . When the damping is stronger, both MC and IPC become smaller. Because the NARMA10 task requires the memory with the delay steps ≈ 10 and the second order nonlinearity with the delay steps ≈ 10 (see Supplementary Information Section I), the NRMSE becomes larger when MC $\lesssim 10$ and IPC $\lesssim 10^2/2$.

In fact, when θ and α are small, MC $\approx N_pN_v$ and IPC $\approx N_pN_v$. The result suggests that our spin-wave RC can use half of the degrees of freedom, respectively, for linear and nonlinear memories. Note that our system has $N = 2N_pN_v$. Around $\theta = 0.1$ ns and $\alpha = 5.0 \times 10^{-4}$, IPC exceeds $N_pN_v = 64$ whereas MC becomes

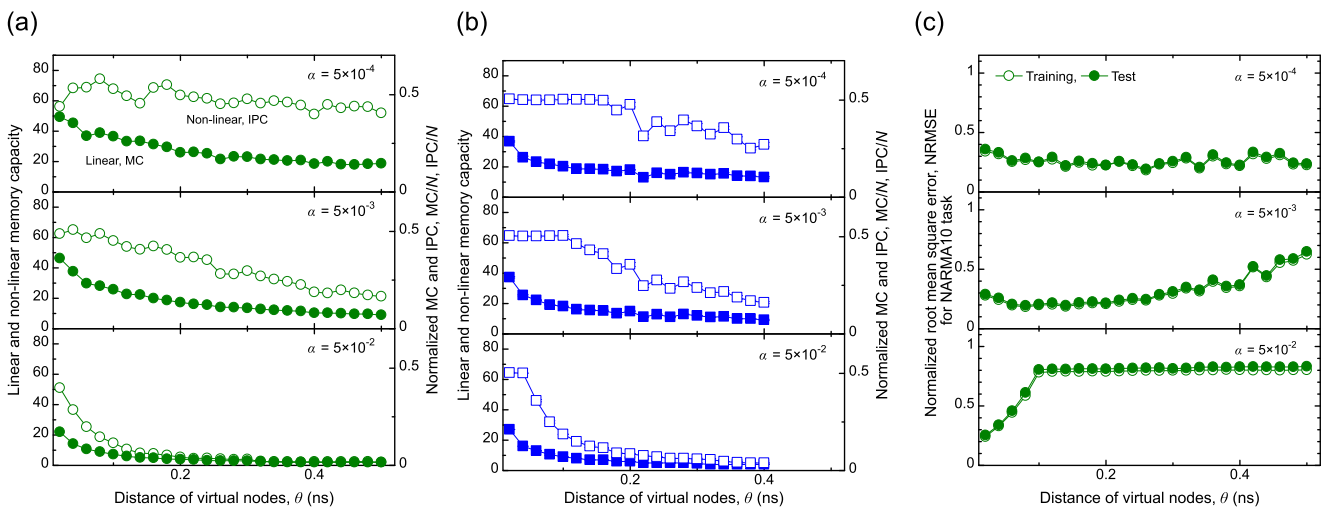


Fig. 3 Virtual node distance θ dependence of MC and IPC for **a** micromagnetics simulation and **b** response function method with different damping parameters α . MC and IPC normalized by the total number of dimensions $N = 2N_pN_v$ are shown in the right axis. **c** Normalized root mean square error, NRMSE for NARMA10 task is plotted as a function of θ with different α .

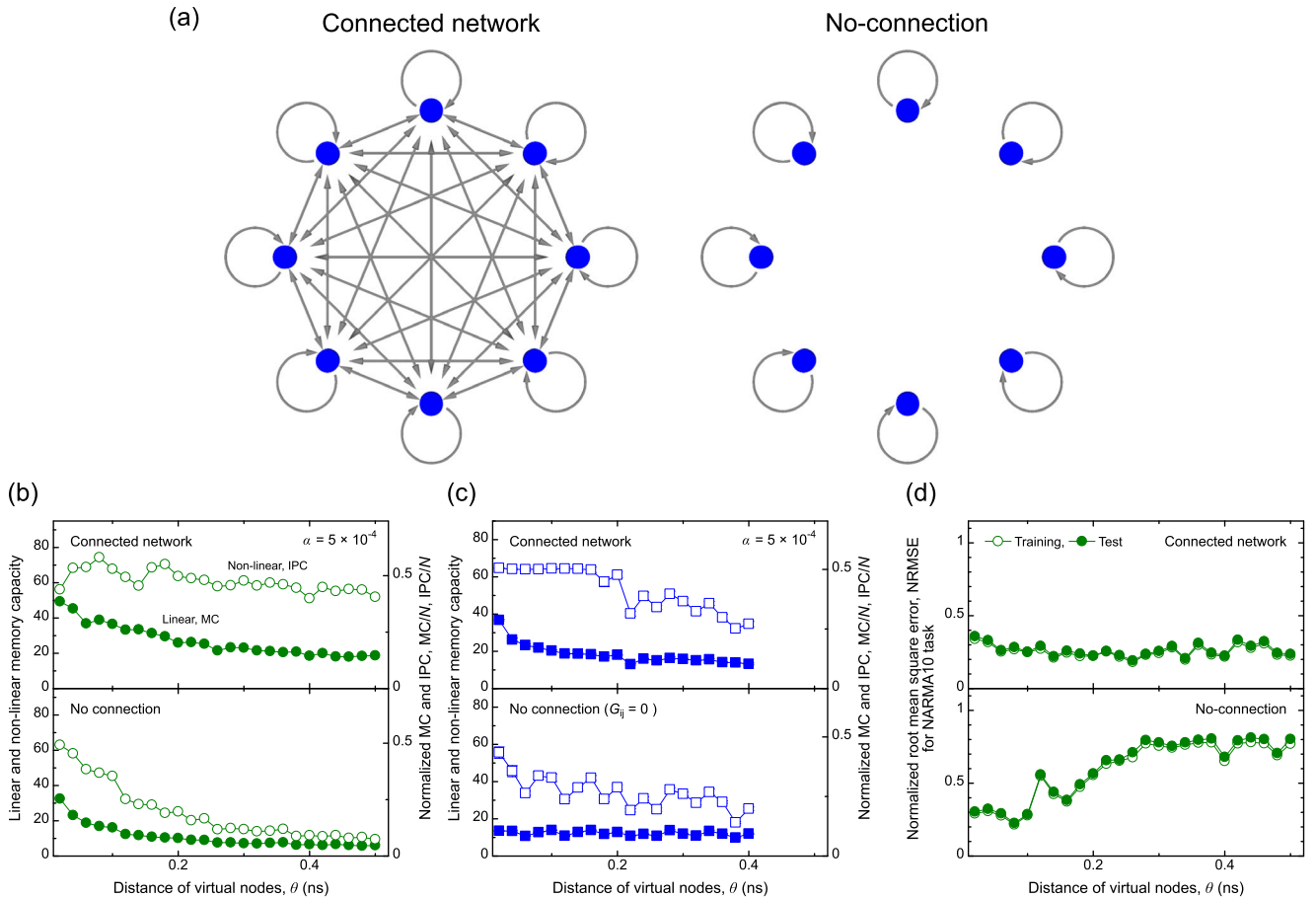


Fig. 4 Effect of the network connection on the performance of reservoir computing. **a** Schematic illustration of the network of physical nodes connected through propagating spin-wave [left] and physical nodes with no connection [right]. **b**, **c** MC and IPC obtained using a connected network with 8 physical nodes [top] and physical nodes with no connection [bottom] calculated by **b** micromagnetics simulation and **c** response function method plotted as a function of virtual node distance θ . $N_v = 8$ virtual nodes are used. **d** Normalized root mean square error, NRMSE for NARMA10 task obtained by micromagnetics simulation is plotted as a function of θ with a connected network [top] and physical nodes with no connection [bottom]. MC and IPC normalized by the total number of dimensions $N = 2N_p N_v$ are shown in the right axis.

smaller than $N_p N_v$. This is consistent with the trade-off between MC and IPC²⁶. The reason why IPC exceeds $N_p N_v$ is as follows; even when we use the linear readout using only m_x , IPC becomes small but nonzero as shown in Supplementary Information Section V A. Therefore, m_x can perform nonlinear transformation because m_x exhibits oscillation with a finite amplitude. However, the nonlinear effect is too weak to have $IPC \approx N_p N_v$, and m_x mainly performs linear memory.

The results of the micromagnetic simulations are semi-quantitatively reproduced by the theoretical model using the response function, as shown in Fig. 3b. This result suggests that the linear response function $G(t, t')$ captures the essential feature of delay $t - t'$ due to wave propagation. Similar to the results of micromagnetic simulations, our spin-wave RC shows $IPC \approx N_p N_v$ when θ and α are small. MC is slightly underestimated but still shows $MC/N > 0.25$, i.e. $MC/(N_p N_v) > 0.5$. When MC and IPC are computed using the response function, each of them is strictly bounded by $N_p N_v$ because the linear part m_x can learn the linear memory (MC) whereas the nonlinear part m_x^2 can learn the nonlinear memory (IPC). In fact, IPC in Fig. 3b is bounded by $N_p N_v = 64$ in contrast with RC using micromagnetic simulations.

Propagating spin waves play a crucial role in high performance. We perform micromagnetic simulations with damping layers between nodes (Fig. 4a). The damping layers inhibit spin

wave propagation. The result of Fig. 4b shows that the MC is substantially lower than that without damping, particularly when θ is small. The NARMA10 task shows a larger error (Fig. 4d). When θ is small, the suppression is less effective. This may be due to incomplete suppression of wave propagation. We also analyze the theoretical model with the response function by neglecting the interaction between two physical nodes, namely, $G_{ij} = 0$ for $i \neq j$. In this case, information transmission between two physical nodes is not allowed. We obtain smaller MC and IPC than the system with wave propagation, supporting our claim (see Fig. 4c).

Our spin wave RC also works for the prediction of time-series data. In the study of ref.⁶, the functional relationship between the state at $t + \Delta t$ and the states before t is learned by the ESN. The schematics of this task are shown in Fig. 5c. The trained ESN can estimate the state at $t + \Delta t$ from the past states from the estimated time series, and therefore, during the prediction step, it can predict the dynamics without the ground-truth time series data (Fig. 5c). In ref.⁶, the prediction for the chaotic time-series data was demonstrated. Figure 5 shows the prediction using our spin wave RC for the Lorenz model. We can demonstrate that the RC shows short-time prediction (Fig. 5a) and, more importantly, reconstruct the chaotic attractor (Fig. 5b). Because the initial deviation exponentially grows in a chaotic system, we cannot reproduce the same time series in the prediction step.

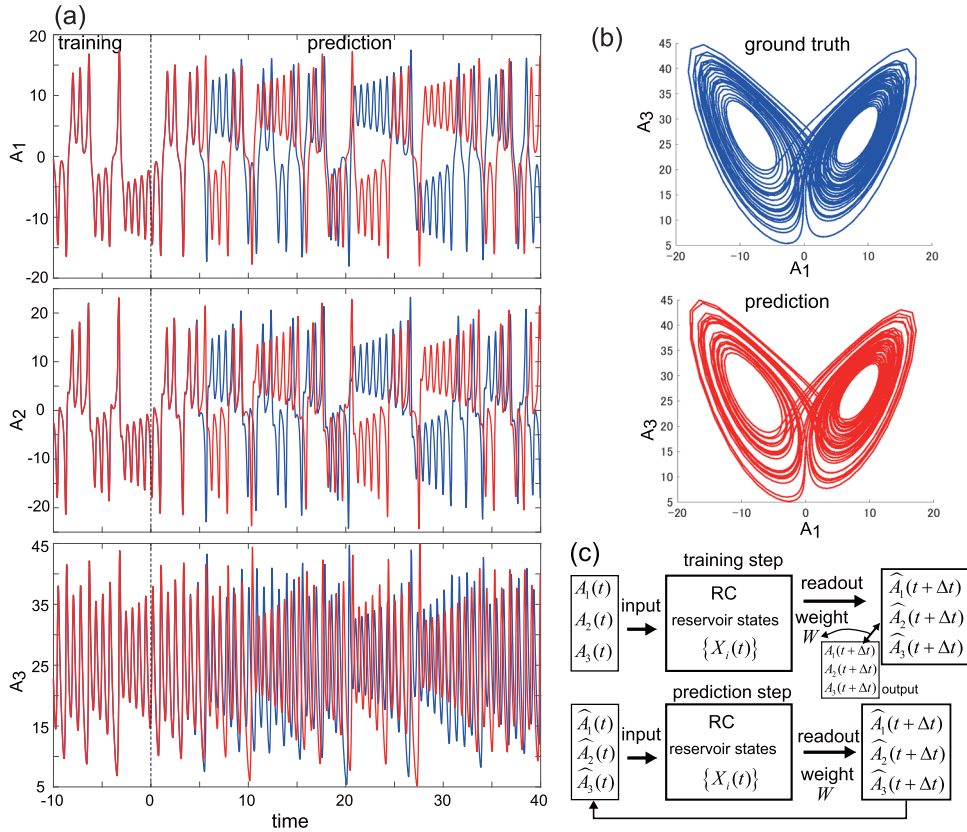


Fig. 5 Prediction of time-series data for the Lorenz system using reservoir computing with micromagnetic simulations. The parameters are $\theta = 0.4$ ns and $\alpha = 5.0 \times 10^{-4}$. **a** The ground truth ($A_1(t)$, $A_2(t)$, $A_3(t)$) and the estimated time series ($\hat{A}_1(t)$, $\hat{A}_2(t)$, $\hat{A}_3(t)$) are shown in blue and red, respectively. The training steps are during $t < 0$, whereas the prediction steps are during $t > 0$. **b** The attractor in the A_1A_3 plane for the ground truth and during the prediction steps for this task. During the training step, RC estimates the time series of the next time step $\hat{A}_i(t + \Delta t)$ ($i = 1, 2, 3$) from the input time series $A_i(t)$ ($i = 1, 2, 3$) by optimizing the readout weights so that the estimated $\hat{A}_i(t + \Delta t)$ approximates the ground truth $A_i(t + \Delta t)$. During the prediction step after the training step, using the optimized readout weights, the estimated $\hat{A}_i(t)$ is transformed into the time series at the next step $\hat{A}_i(t + \Delta t)$, which is further used for the estimation at future time steps $t + 2\Delta t, t + 3\Delta t, \dots$. The ground truth time series is no longer used during the prediction step except at the initial time step.

Nevertheless, we may reproduce the shape of the trajectory in the phase space.

Scaling of system size and wave speed

To clarify the mechanism of the high performance of our spin wave RC, we investigate MC and IPC of the system with different characteristic length scales/and different wave propagating speed v . We choose the characteristic length scale of our spin-wave RC system as the diameter $2R$ of the circle on which inputs are located (see Fig. 2-(3)), namely, $l = 2R$. Hereafter, we denote the characteristic length by l for general systems but interchangeably by l and R for our system. We use our theoretical model with the response function discussed in Methods: Theoretical analysis using response function to compute MC and IPC in the parameter space (v, R). This calculation can be done because the computational cost of our model using response function is much cheaper than numerical micromagnetic simulations. In this analysis, we assume that spin waves are dominated by the dipole interaction. This implies that the speed of the spin wave is constant. The power spectral density shown in Supplementary Information Section VIII A indicates that the amplitude of the spin waves is dominated by small wavenumbers $k < 1/l_0$. Nevertheless, the exchange interaction comes into play at $0 \ll k < 1/l_0$. The effect of the exchange interaction is that the speed of wave propagation exhibits distribution due to the wavenumber-dependent waves,

which are faster than the waves of the dipole-only system. Accordingly, the average speed becomes faster. Although this effect modifies the evaluation of the spin-wave propagation speed, it does not change the scaling discussed in this section. We discuss the effect in Supplementary Information Section VII A.

Figure 6a, b shows that both MC and IPC have maximum when $R \propto v$. The response function for the magnetization dynamics Eqs. (16) and (17) have a peak at some delay time as schematically shown in Fig. 6e (see also Section VII in Supplementary Information). Therefore, the response function plays the role of a filter such that the magnetization of the i th node $m_i(t)$ at time t has the information of the input at time $t - \tau$. The position of the peak is dependent on the distance between two nodes and the wave speed. As a result, $m_i(t)$ may have the information of different delay times (see also Fig. 1c). The maximum delay time can be evaluated from the maximum time that spin wave propagation spends. This is nothing but the time for wave propagation between the furthest physical nodes. RC cannot memorize the information beyond this time scale of the delay. Together with the nonlinear readout, the spin-wave RC can memorize the past information and its nonlinear transformation.

To obtain a deeper understanding of the result, we perform the same analyses for the further simplified model, in which the response function is replaced by the Gaussian function Eq. (19) (see Methods). This functional form has the peak at $t = R_i/v$ and

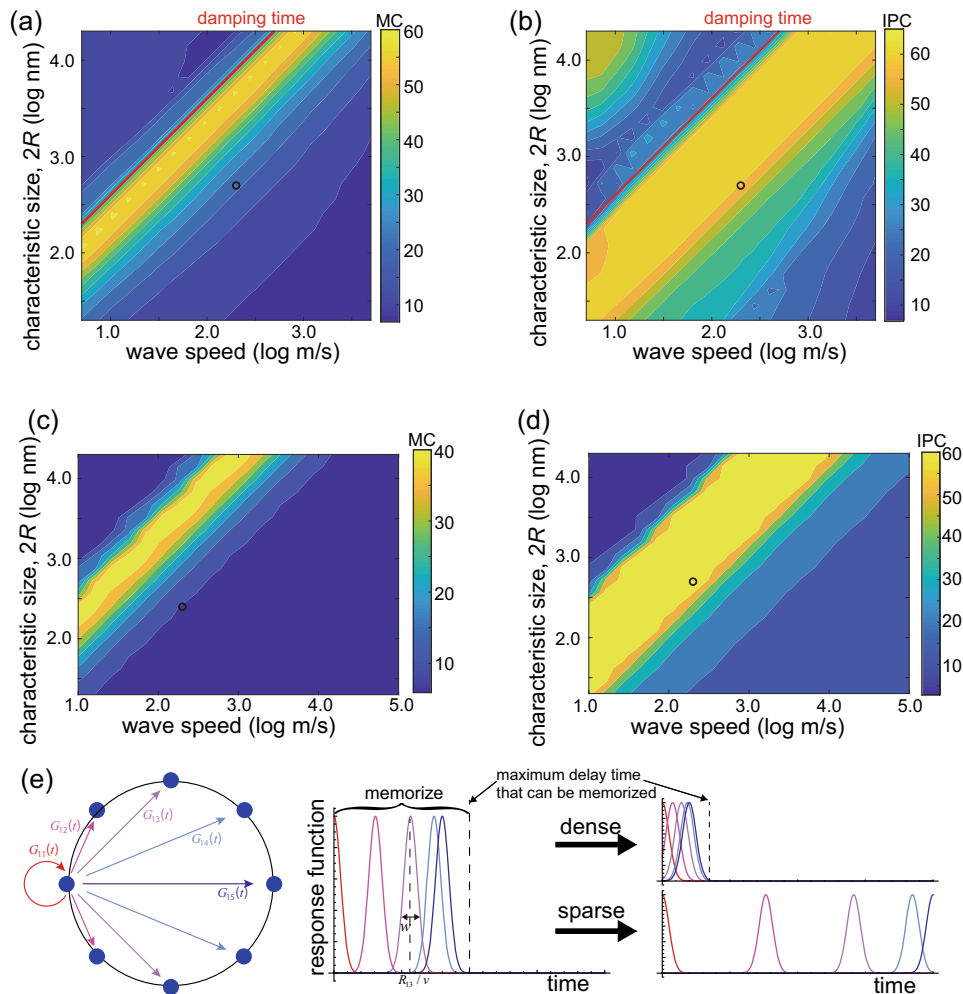


Fig. 6 Scaling between characteristic size and propagating wave speed obtained by response function method. The characteristic size $l = 2R$ is quantified by the radius R of the circle on which inputs are located (see Fig. 2-(3)). **a, c** MC and **b, d** IPC as a function of the characteristic length scale between physical nodes R and the speed of wave propagation v with $\theta = 0.04$ ns and $\alpha = 5.0 \times 10^{-4}$. The results with the response function for the dipole interactions Eqs. (16) and (17) **a, b** and for the Gaussian function Eq. (19) **c, d** are shown. Open circle symbols in **a–d** corresponds to wave speed ($v = 200$ m·s $^{-1}$) and length ($l = 2R = 500$ nm) used in micromagnetic simulation. The damping time shown in **a, b** expresses the length scale obtained from the wave speed multiplied by the time scale of damping associated with α . **e** Schematic illustration of the response function and its relation to wave propagation between physical nodes. When the speed of the wave is too fast, all the response functions are overlapped (dense regime), while the response functions cannot cover the time windows when the speed of the wave is too slow (sparse regime).

the width of the response function is w , where R_{ij} is the distance between i th and j th physical nodes (see Fig. 6e). Even in this simplified model, we obtain $MC \approx 40$ and $IPC \approx 60$, and also the maximum when $l = 2R \propto v$ (Fig. 6c, d). From this result, the origin of the optimal ratio between the length and speed becomes clearer; when $l \ll v$, the response functions under different R_{ij} overlap so that different physical nodes cannot carry the information of different delay times. On the other hand, when $l \gg v$, the characteristic delay time l/v exceeds the maximum delay time to compute MC and IPC, or exceeds the total length of the time series. Note that we set the maximum delay time as 100, which is much longer than the value necessary for the NARMA10 task.

We also analyse the results of micromagnetic simulations along the line with the scaling argument using the response function. As shown in insets of Fig. 7, MC becomes smaller when $v\tau_0$ is larger under the fixed l , and when l is smaller l with fixed $v\tau_0$. These trends of scaling obtained by micromagnetic simulations are comparable with the Gaussian model. However, the position of

the peak deviates from the theoretical value. We speculate this is due to the finite system size, whose effect is not included in the theoretical model.

Reservoir computing scaling and performance in comparison with the literature

Our results suggest the universal scaling between the size of the system and the speed of the RC based on wave propagation. Figure 7 shows reports of reservoir computing in literature with multiple nodes plotted as a function of the length of nodes l and products of wave speed and delay time $v\tau_0$ for both photonic and spintronic RC. MC and NRMSE for NARMA10 tasks using photonic and spintronic RC are reported in refs. ^{12,40–45} for photonic RC and refs. ^{9,23,34,35,46–49} for spintronic RC. Table 1 and 2 summarize the reports of MC for photonic and spintronic RC with different length scales, which are plotted in Fig. 7. We evaluate the characteristic length l in the reported systems by the longest distance between physical nodes for spintronic RC and the size of the feedback loop unit for photonic RC. Our system of the spin wave has a

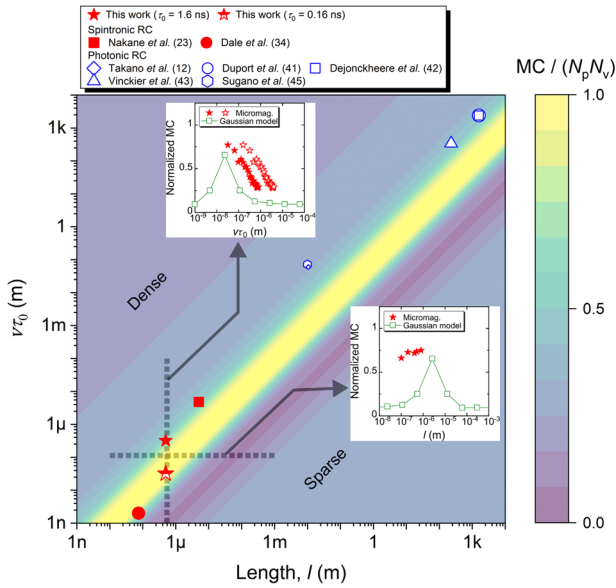


Fig. 7 Reports of reservoir computing using multiple nodes are plotted as a function of the length between nodes (l) and characteristic wave speed (v) times delay time (τ_0) for photonics systems (open symbols) and spintronics systems (solid symbols). The size of symbols corresponds to MC, which is taken from literature^{12,23,34,41–43,45} and this work. The color scale represents MC evaluated by using the response function method. Inset figures show comparison between micromagnetic simulation and response function simulation using Gaussian model Eq. (19). Solid and open star symbols in the insets are $v\tau_0$ evaluated by considering dipole interaction only ($v = 200$ m/s) and with modification of exchange interaction ($v = 1000$ m/s).

Table 1. Report of photonic RC with different length scales used in Fig. 7

Reports	Length, l	Time interval, τ_0	$v\tau_0$	N	MC
Duport et al. ⁴¹	1.6 km	8 μ s	2.4 km	50	21
Dejonckheere et al. ⁴²	1.6 km	8 μ s	2.4 km	50	37
Vincker et al. ⁴³	230 m	1.1 μ s	340 m	50	21
Takano et al. ¹²	11 mm	200 ps	60 mm	31	1.5
Sugano et al. ⁴⁵	10 mm	240 ps	72 mm	240	10

speed of light, $v = 3 \times 10^8$ m/s is used.

Table 2. Report of spintronic RC with different length scales used in Fig. 7

Reports	l	τ_0	v	$v\tau_0$	N	MC
Nakane et al. ²³	5 μ m	2 ns	2.4 km/s	4.8 μ m	72	21
Dale et al. ³⁴	50 nm	10 ps	200 m/s	2 nm	100	35
This work	500 nm	1.6 ns	200 m/s	320 nm	64	26

v is calculated based on magneto-static spin wave described in Supplementary Information Section VIII A.

characteristic length $l \sim 500$ nm and a speed of $v \sim 200$ m s^{-1} . For the spintronic RC, the dipole interaction is considered for wave propagation in which speed is proportional to both saturation magnetization and thickness of the film⁵⁰ (see Supplementary Information Section VIII A). For the photonic RC, the characteristic

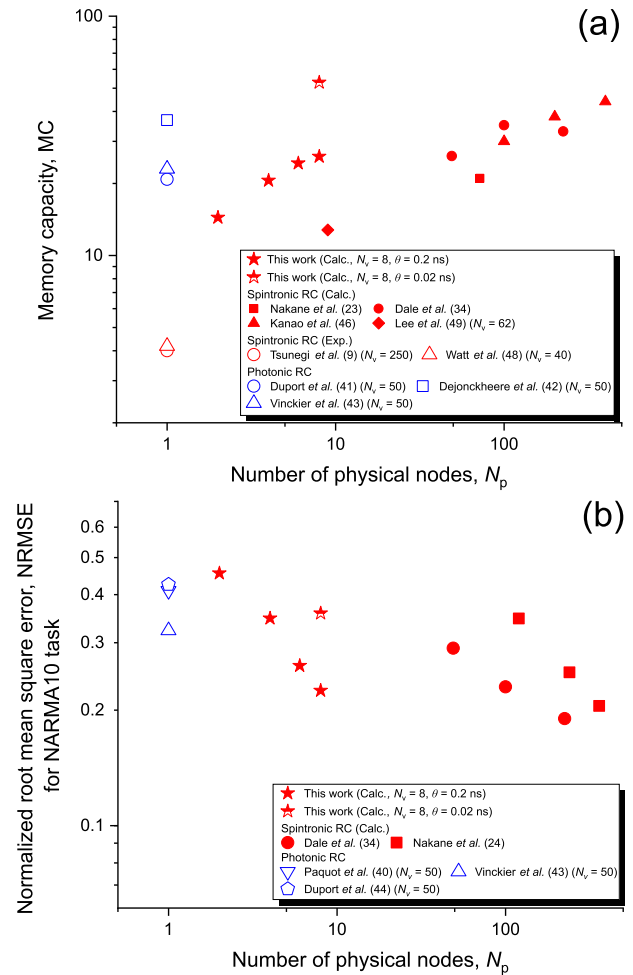


Fig. 8 Reservoir computing performance compared with different systems. **a** MC reported plotted as a function of physical nodes N_p . **b** Normalized root mean square error, NRMSE for NARMA10 task is plotted as a function of N_p . Open blue symbols are values reported using photonic RC while solid red symbols are values reported using spintronic RC. MC and NRMSE for NARMA10 task are taken from refs. ^{9,23,24,34,46,48,49} for spintronic RC and refs. ^{40–44} for photonic RC.

speed is the speed of light, $v \sim 10^8$ m s^{-1} . Symbol size corresponds to MC taken from the literature (see Tables 1 and 2). Plots are roughly on a broad oblique line with a ratio $l/(v\tau_0) \sim 1$. Therefore, the photonic RC requires a larger system size, as long as the delay time of the input $\tau_0 = N_p\theta$ is the same order ($\tau_0 = 0.3–3$ ns in our spin wave RC). As can be seen in Fig. 6, if one wants to reduce the length of physical nodes, one must reduce wave speed or delay time; otherwise, the information is dense, and the reservoir cannot memorize many degrees of freedom (See Fig. 6e). Reducing delay time is challenging since the experimental demonstration of the photonic reservoirs has already used a short delay close to the instrumental limit. Also, reducing wave speed in photonics systems is challenging. On the other hand, the wave speed of propagating spin-wave is much lower than the speed of light and can be tuned by configuration, thickness and material parameters. If one reduces wave speed or delay time over the broad line in Fig. 7, information becomes sparse and cannot be used efficiently (See Fig. 6e). Therefore, there is an optimal condition for high-performance RC.

The performance is comparable with other state-of-the-art techniques, which are summarized in Fig. 8. For example, for the spintronic RC, $MC \approx 30$ ²³ and $NRMSE \approx 0.2$ ³⁴ in the NARMA10 task

are obtained using $N_p \approx 100$ physical nodes. However, the spintronic RC with one physical node and $10^1 - 10^2$ virtual nodes does not show high performance; MC is less than 10 (the bottom left points in Fig. 8). This fact suggests that the spintronic RC so far cannot use virtual nodes effectively, namely, $MC \ll N = N_p N_v$. On the other hand, for the photonic RC, comparable performances are achieved using $N_v \approx 50$ virtual nodes, but only one physical node. As we discussed, however, the photonic RC requires mm system sizes. Our system achieves comparable performances using $\lesssim 10$ physical nodes, and the size is down to nanoscales keeping the 2–50 GHz computational speed. We also demonstrate that the spin wave RC can perform time-series prediction and reconstruction of an attractor for the chaotic data. This has not been done in spintronic RC.

DISCUSSION

Our results of micromagnetic simulations suggest that our system has a potential of physical implementation because all the parameters in this study are feasible using realistic materials. We found that RC performance such as MC can be increased by use of materials with low damping parameter. Thin films of Heusler ferromagnet with lower damping has been reported^{51–53}, considered as material system for spin-wave RC in this study. Nanoscale propagating spin waves in a ferromagnetic thin film excited by spin-transfer torque using nanometer electrical contacts have been observed^{54–56}. Patterning of multiple electrical nanocontacts into magnetic thin films was demonstrated in mutually synchronized spin-torque oscillators⁵⁶. In addition to the excitation of propagating spin-wave in a magnetic thin film, its non-local magnetization dynamics can be detected by tunnel magnetoresistance effect at each electrical contact⁵⁷, as schematically shown in Fig. 1c, which are widely used for the development of spintronics memory and spin-torque oscillators. The size of a magnetic tunnel junction can be 40nm in diameter, and therefore, it is compatible with our proposed system. In addition, virtual nodes are effectively used in our system by considering the speed of propagating spin-wave and distance of physical nodes; thus, high-performance reservoir computing can be achieved with the small number of physical nodes, contrary to many physical nodes used in previous reports. Spin wave can be excited by current pulse with typical time scale of GHz frequency, for example by Heusler ferromagnet alloys discussed above. This leads to the realization of fast and energy-efficient spin-wave RC. Energy consumption needed to process one time step calculated is an order of magnitude smaller than other physical RC such as memristor-based RC, which is mainly due to the fact that shorter input pulse can be used in spin-wave RC (see Supplementary Information Section IX). Therefore, spin-wave RC have potential to be used for future energy-efficient physical RC with spintronics read out network such as spin crossbar arrays^{58,59}.

There is an interesting connection between our study and the recently proposed next-generation RC^{37,60}, in which the linear ESN is identified with the NVAR (nonlinear vectorial autoregression) method to estimate a dynamical equation from data. Our theoretical analyses, shown in Eq. (15), clarify that the magnetization dynamics at physical and virtual nodes can be expressed by the Volterra series of the input, namely, a time delay of U_k with its polynomials (see Methods: Relation of MC and IPC with learning performance). On the other hand, the linear input-output relationship can be expressed with a delay as $Y_{n+1} = a_n U_n + a_{n-1} U_{n-1} + \dots$, or more generally, with the non-linear readout or with higher-order response functions, as $Y_{n+1} = a_n U_n + a_{n-1} U_{n-1} + \dots + a_{n,n} U_n U_n + a_{n,n-1} U_n U_{n-1} + \dots$. These input-output relations are nothing but Volterra series of the output as a function of the input with delay and nonlinearity²⁷. The coefficients of the expansion are associated with the response function. Therefore, the performance of RC

falls into the independent components of the matrix of the response function, which can be evaluated by how much delay the response functions between two nodes cover without overlap. We should stress that the form of expansion is imposed neither in the response function nor in micromagnetic simulations. The structure is obtained from the analyses of the LLG equation. The results would be helpful to a potential design of the network of the physical nodes.

We should note that the polynomial basis of the input-output relation in this study originates from spin wave excitation around the stationary state $m_z \approx 1$. When the input data has a hierarchical structure, another basis may be more efficient than the polynomial expansion. Another setup of magnetic systems may lead to a different basis. We believe that our study shows simple but clear intuition of the mechanism of high-performance RC, that can lead to the exploration of another setup for more practical application of the physical RC.

METHODS

Physical system of a magnetic device

We consider a magnetic device of a thin rectangular system with cylindrical injectors (see Fig. 1c). The size of the device is $L \times L \times D$. Under the uniform external magnetic field, the magnetization is along the z direction. Electric current translated from time series data is injected at the N_p injectors with the radius a and the same height D as the device. The spin-torque by the current drives magnetization $\mathbf{m}(\mathbf{x}, t)$ and propagating spin-waves as schematically shown in Fig. 1c. The magnetization is detected at the same cylindrical nanocontacts as the injectors. The nanocontacts of the inputs and outputs are located on the circle with its radius R (see Fig. 2-(3)). The size of our system is $L = 1000$ nm and $D = 4$ nm. We set $a = 20$ nm unless otherwise stated. We set $R = 250$ nm in Performance of tasks for MC, IPC, NARMA10, and prediction of chaotic time-series section but R is varied in Scaling of system size and wave speed section as a characteristic length scale.

Details of preprocessing input data

In the time-multiplexing approach, the input time-series $\mathbf{U} = (U_1, U_2, \dots, U_T) \in \mathbb{R}^T$ is translated into piece-wise constant time-series $\hat{U}(t) = U_n$ with $t = (n-1)N_v\theta + s$ under $k = 1, \dots, T$ and $s = [0, N_v\theta)$ (see Fig. 2-(1)). This means that the same input remains during the time period $\tau_0 = N_v\theta$. To use the advantage of physical and virtual nodes, the actual input $j_i(t)$ at the i th physical node is $U(t)$ multiplied by τ_0 -periodic random binary filter $\mathcal{B}_i(t)$. Here, $\mathcal{B}_i(t) \in \{0, 1\}$ is piece-wise constant during the time θ . At each physical node, we use different realizations of the binary filter as in Fig. 2-(2). The input masks play a role as the input weights for the ESN without virtual nodes. Because the different masks are used for the different physical nodes, each physical and virtual node may have different information about the input time series.

The masked input is further translated into the input for the physical system. Although numerical micromagnetic simulations are performed in discrete time steps, the physical system has continuous-time t . The injected current density $j_i(t)$ at time t for the i th physical node is set as $j_i(t) = 2j_c \hat{U}_i(t) = 2j_c \mathcal{B}_i(t) U(t)$ with $j_c = 2 \times 10^{-4} / (\pi a^2) A/m^2$. Under a given input time series of the length T , we apply the constant current during the time θ , and then update the current at the next step. The same input current with different masks is injected for different virtual nodes. The total simulation time is, therefore, $T\theta N_v$.

Micromagnetic simulations

We analyze the magnetization dynamics of the Landau-Lifshitz-Gilbert (LLG) equation using the micromagnetic simulator mumax³⁶¹.

The LLG equation for the magnetization $\mathbf{M}(\mathbf{x}, t)$ yields

$$\begin{aligned} \partial_t \mathbf{M}(\mathbf{x}, t) = & -\frac{\gamma \mu_0}{1+\alpha^2} \mathbf{M} \times \mathbf{H}_{\text{eff}} \\ & -\frac{\alpha \gamma \mu_0}{M_s(1+\alpha^2)} \mathbf{M} \times (\mathbf{M} \times \mathbf{H}_{\text{eff}}) \\ & +\frac{\hbar P \gamma}{4M_s^2 e D} J(\mathbf{x}, t) \mathbf{M} \times (\mathbf{M} \times \mathbf{m}_f). \end{aligned} \quad (10)$$

In the micromagnetic simulations, we analyze Eq. (10) with the effective magnetic field $\mathbf{H}_{\text{eff}} = \mathbf{H}_{\text{ext}} + \mathbf{H}_{\text{demag}} + \mathbf{H}_{\text{exch}}$ consists of the external field, demagnetization, and the exchange interaction. The concrete form of each term is

$$\mathbf{H}_{\text{eff}} = \mathbf{H}_{\text{ext}} + \mathbf{H}_{\text{demag}} + \mathbf{H}_{\text{exch}}, \quad (11)$$

$$\mathbf{H}_{\text{ext}} = H_0 \mathbf{e}_z \quad (12)$$

$$\mathbf{H}_{\text{ms}} = -\frac{1}{4\pi} \int \nabla \nabla \frac{\mathbf{M}(\mathbf{x}')}{|\mathbf{x} - \mathbf{x}'|} d\mathbf{x}' \quad (13)$$

$$\mathbf{H}_{\text{exch}} = \frac{2A_{\text{ex}}}{\mu_0 M_s} \Delta \mathbf{M}, \quad (14)$$

where \mathbf{H}_{ext} is the external magnetic field, \mathbf{H}_{ms} is the magnetostatic interaction, and \mathbf{H}_{exch} is the exchange interaction with the exchange parameter A_{ex} . The spin waves are driven by Slonczewski spin-transfer torque⁶², which is described by the last term of Eq. (10). The driving term is proportional to the DC current, namely, $J(\mathbf{x}, t) = j_i(t)$ at the i th nanocontact and $J(\mathbf{x}, t) = 0$ other regions.

Parameters in the micromagnetics simulations are chosen as follows: The number of mesh points is 200 in the x and y directions, and 1 in the z direction. We consider Co_2MnSi Heusler alloy ferromagnet, which has a low Gilbert damping and high spin polarization with the parameter $A_{\text{ex}} = 23.5$ pJ/m, $M_s = 1000$ kA/m, and $\alpha = 5 \times 10^{-4}$ ^{51–53,63,64}. Out-of-plane magnetic field $\mu_0 H_0 = 1.5$ T is applied so that magnetization is pointing out-of-plane. The spin-polarized current field is included by the Slonczewski model⁶² with polarization parameter $P=1$ and spin torque asymmetry parameter $\lambda=1$ with the reduced Planck constant \hbar and the charge of an electron e . The uniform fixed layer magnetization is $\mathbf{m}_f = \mathbf{e}_x$. We use absorbing boundary layers for spin waves to ensure the magnetization vanishes at the boundary of the system⁶⁵. We set the initial magnetization as $\mathbf{m} = \mathbf{e}_z$. The magnetization dynamics is updated by the fourth-order Runge-Kutta method (RK45) under adaptive time-stepping with the maximum error set to be 10^{-7} .

The reference time scale in this system is $t_0 = 1/\gamma\mu_0 M_s \approx 5$ ps, where γ is the gyromagnetic ratio, μ_0 is permeability, and M_s is saturation magnetization. The reference length scale is the exchange length $l_0 \approx 5$ nm. The relevant parameters are Gilbert damping α , the time scale of the input time series θ (see Section Learning with reservoir computing), and the characteristic length between the input nodes R .

Theoretical analysis using response function

To understand the mechanism of high performance of learning by spin wave propagation, we also consider a model using the response function of the spin wave dynamics. By linearizing the magnetization around $\mathbf{m} = (0, 0, 1)$ without inputs, we may express the linear response of the magnetization at the i th readout $m_i = m_{x,i} + im_{y,i}$ to the input as

$$m_i(t) = \sum_{j=1}^{N_p} \int dt' G_{ij}(t, t') U^{(j)}(t'), \quad (15)$$

where the response function G_{ij} for the same node is

$$G_{ij}(t - \tau) = \frac{1}{2\pi} e^{-\hbar(a+i)(t-\tau)} \frac{-1 + \sqrt{1 + \frac{a^2}{(d/4)^2(a+i)^2(t-\tau)^2}}}{\sqrt{1 + \frac{a^2}{(d/4)^2(a+i)^2(t-\tau)^2}}} \quad (16)$$

and for different nodes,

$$\begin{aligned} G_{ij}(t - \tau) = & \frac{a^2}{2\pi} e^{-\hbar(a+i)(t-\tau)} \\ & \times \frac{1}{(d/4)^2(a+i)^2(t-\tau)^2 \left(1 + \frac{|\mathbf{R}_i - \mathbf{R}_j|^2}{(d/4)^2(a+i)^2(t-\tau)^2}\right)^{3/2}}. \end{aligned} \quad (17)$$

The detailed derivation of the response function is shown in Supplementary Information Section VII. Here, $U^{(j)}(t)$ is the input time series at j th nanocontact. The response function has a self part G_{ii} that is, input and readout nanocontacts are the same, and the propagation part G_{ij} , where the distance between the input and readout nanocontacts is $|\mathbf{R}_i - \mathbf{R}_j|$. In Eqs. (16) and (17), we assume that the spin waves propagate only with the dipole interaction. We discuss the effect of the exchange interaction in Supplementary Information Section VII A. We should note that the analytic form of Eq. (17) is obtained under certain approximations. We discuss its validity in Supplementary Information Section VII A.

We use the quadratic nonlinear readout, which has a structure

$$\begin{aligned} m_i^2(t) = & \sum_{j_1=1}^{N_p} \sum_{j_2=1}^{N_p} \int dt_1 \int dt_2 \\ & G_{ij_1 j_2}^{(2)}(t, t_1, t_2) U^{(j_1)}(t_1) U^{(j_2)}(t_2). \end{aligned} \quad (18)$$

The response function of the nonlinear readout is $G_{ij_1 j_2}^{(2)}(t, t_1, t_2) \propto G_{ij_1}(t, t_1) G_{ij_2}(t, t_2)$. The same structure as Eq. (18) appears when we use a second-order perturbation for the input (see Supplementary Information Section V A). In general, we may include the cubic and higher-order terms of the input. This expansion leads to the Volterra series of the output in terms of the input time series, and suggests how the spin wave RC works as shown in Methods: Relation of MC and IPC with learning performance (see also Supplementary Information Section II for more details). Once the magnetization at each nanocontact is computed, we may estimate MC and IPC.

We also use the following Gaussian function as a response function,

$$G_{ij}(t) = \exp\left(-\frac{1}{2w^2} \left(t - \frac{R_{ij}}{v}\right)^2\right) \quad (19)$$

where R_{ij} is the distance between i th and j th physical nodes. The Gaussian function Eq. (19) has its mean $t = R_{ij}/v$ and variance w^2 . Therefore, w is the width of the Gaussian function (see Fig. 6e). We set the width as $w = 50$ in the normalized time unit t_0 . In Supplementary Information Section VII A, we introduce the skew normal distribution, which generalize Eq. (19) to include the effect of exchange interaction.

Learning tasks

NARMA task. The NARMA10 task is based on the discrete differential equation,

$$Y_{n+1} = \alpha Y_n + \beta Y_n \sum_{p=0}^9 Y_{n-p} + \gamma U_n U_{n-9} + \delta. \quad (20)$$

Here, U_n is an input taken from the uniform random distribution $\mathcal{U}(0, 0.5)$, and Y_n is an output. We choose the parameter as $\alpha = 0.3, \beta = 0.05, \gamma = 1.5$, and $\delta = 0.1$. In RC, the input is $\mathbf{U} = (U_1, U_2, \dots, U_T)$ and the output $\mathbf{Y} = (Y_1, Y_2, \dots, Y_T)$. The goal of the NARMA10 task is to estimate the output time-series \mathbf{Y} from the given input \mathbf{U} . The training of RC is done by tuning the weights \mathbf{W} so that the estimated output $\hat{Y}(t_n)$ is close to the true output Y_n in terms of squared norm $|Y(t_n) - Y_n|^2$.

The performance of the NARMA10 task is measured by the deviation of the estimated time series $\mathbf{Y} = \mathbf{W} \cdot \mathbf{X}$ from the true

output \mathbf{Y} . The normalized root-mean-square error (NRMSE) is

$$\text{NRMSE} \equiv \sqrt{\frac{\sum_n (\hat{Y}(t_n) - Y_n)^2}{\sum_n Y_n^2}}. \quad (21)$$

Performance of the task is high when $\text{NRMSE} \approx 0$. In the ESN, it was reported that $\text{NRMSE} \approx 0.4$ for $N = 50$ and $\text{NRMSE} \approx 0.2$ for $N = 200$ ³¹. The number of node $N = 200$ was used for the speech recognition with ≈ 0.02 word error rate³¹, and time-series prediction of spatio-temporal chaos⁶. Therefore, $\text{NRMSE} \approx 0.2$ is considered as reasonably high performance in practical application. We also stress that we use the same order of nodes (virtual and physical nodes) $N = 128$ to achieve $\text{NRMSE} \approx 0.2$.

Memory capacity (MC). MC is a measure of the short-term memory of RC. This was introduced in ref. ⁷. For the input U_n of random time series taken from the uniform distribution, the network is trained for the output $Y_n = U_{n-k}$. Here, U_{n-k} is normalized so that Y_n is in the range $[-1, 1]$. The MC is computed from

$$\text{MC}_k = \frac{\langle U_{n-k}, \mathbf{W} \cdot \mathbf{X}(t_n) \rangle^2}{\langle U_n^2 \rangle \langle (\mathbf{W} \cdot \mathbf{X}(t_n))^2 \rangle}. \quad (22)$$

This quantity is decaying as the delay k increases, and MC is defined as

$$\text{MC} = \sum_{k=1}^{k_{\max}} \text{MC}_k. \quad (23)$$

Here, k_{\max} is a maximum delay, and in this study we set it as $k_{\max} = 100$. The advantage of MC is that when the input is independent and identically distributed (i.i.d.), and the output function is linear, then MC is bounded by N , the number of internal nodes⁷.

We show the example of MC_k as a function of k and IPC_j as a function of k_1 and k_2 in Fig. S2a and b. When the delay k is short, $\text{MC}_k \approx 1$ suggesting that the delayed input can be reconstructed by the input without the delay. When delay becomes longer reservoir does not have capacity to reconstruct delayed input, however, MC_k and IPC_j defined by Eqs. (22) and (25) have long tail small positive values, making overestimation of MC and IPC. To avoid the overestimation, we use the threshold value $\varepsilon = 0.03727$ below which we set $\text{MC}_k = 0$. The value of ε is set following the argument of ref. ²⁶. When the total number of readouts is $N (= 128)$ and the number of time steps to evaluate the capacity is $T (= 5000)$, the mean error of the capacity is estimated as N/T . We assume that the distribution of the error yields χ^2 distribution as $(1/T)\chi^2(N)$. Then, we may set ε such that $\varepsilon = N/T$ where \tilde{N} is taken from the χ^2 distribution and the probability of its realization is 10^{-4} , namely, $P(\tilde{N} \sim \chi^2(N)) = 10^{-4}$. In fact, Fig. S2a demonstrates that when the delay is longer, the capacity converges to N/T and fluctuates around the value. In this regime, we set $\text{MC}_k = 0$ and do not count for the calculation of MC. The same argument applies to the calculation of IPC.

Information processing capacity (IPC). IPC is a nonlinear version of MC²⁶. In this task, the output is set as

$$Y_n = \prod_k \mathcal{P}_{d_k}(U_{n-k}) \quad (24)$$

where d_k is non-negative integer, and $\mathcal{P}_{d_k}(x)$ is the Legendre polynomials of x order d_k . As MC, the input time series is uniform random noise $U_n \in \mathcal{U}(0, 0.5)$ and then normalized so that Y_n is in the range $[-1, 1]$. We may define

$$\text{IPC}_{d_0, d_1, \dots, d_{r-1}} = \frac{\langle Y_n, \mathbf{W} \cdot \mathbf{X}(t_n) \rangle^2}{\langle Y_n^2 \rangle \langle (\mathbf{W} \cdot \mathbf{X}(t_n))^2 \rangle}. \quad (25)$$

and then compute j th order IPC as We may define

$$\text{IPC}_j = \sum_{d_k s, t, j = \sum_k d_k} \text{IPC}_{d_1, d_2, \dots, d_r}. \quad (26)$$

When $j = 1$, the IPC is, in fact, equivalent to MC, because $\mathcal{P}_0(x) = 1$ and $\mathcal{P}_1(x) = x$. In this case, $Y_n = U_{n-k}$ for $d_i = 1$ when $i = k$ and $d_i = 0$ otherwise. Equation (26) takes the sum over all possible delay k , which is nothing but MC. When $j > 1$, IPC_j captures the nonlinear transformation and delays up of the j th polynomial order. For example, when $j = 2$, the output can be $Y_n = U_{n-k_1} U_{n-k_2}$ or $Y_n = U_{n-k}^2 + \text{const}$. In this study, we focus on $j = 2$ because the second-order nonlinearity is essential for the NARMA10 task (see Supplementary Information Section I). In ref. ²⁶, IPC is defined by sum of IPC_j over j . In this study, we focus only on IPC_2 , which plays a relevant role for the NARMA10 task (see Supplementary Information Section I). We denote IPC and IPC_2 interchangeably.

The example of IPC_2 as a function of the two delay times k_1 and k_2 is shown in Fig. S2b. When both k_1 and k_2 are small, the nonlinear transformation of the delayed input $Y_n = U_{n-k_1} U_{n-k_2}$ can be reconstructed from U_n . As the two delays are longer, IPC_2 approaches the value N/T of systematic error. As we performed for the MC task, we define the same threshold ε , and do not count the capacity below the threshold in IPC_2 .

Prediction of chaotic time-series data. Following ref. ⁶, we perform the prediction of time-series data from the Lorenz model. The model is a three-variable system of $(A_1(t), A_2(t), A_3(t))$ yielding the following equation

$$\frac{dA_1}{dt} = 10(A_2 - A_1) \quad (27)$$

$$\frac{dA_2}{dt} = A_1(28 - A_3) - A_2 \quad (28)$$

$$\frac{dA_3}{dt} = A_1 A_2 - \frac{8}{3} A_3. \quad (29)$$

The parameters are chosen such that the model exhibits chaotic dynamics. Similar to the other tasks, we apply the different masks of binary noise for different physical nodes, $\mathcal{B}_i^{(l)}(t) \in \{-1, 1\}$. Because the input time series is three-dimensional, we use three independent masks for A_1, A_2 , and A_3 , therefore, $l \in \{1, 2, 3\}$. The input for the i th physical node after the mask is given as $\mathcal{B}_i(t) \tilde{U}_i(t) = \mathcal{B}_i^{(1)}(t) A_1(t) + \mathcal{B}_i^{(2)}(t) A_2(t) + \mathcal{B}_i^{(3)}(t) A_3(t)$. Then, the input is normalized so that its range becomes $[0, 0.5]$, and applied as an input current. Once the input is prepared, we may compute magnetization dynamics for each physical and virtual node, as in the case of the NARMA10 task. We note that here we use the binary mask of $\{-1, 1\}$ instead of $\{0, 1\}$ used for other tasks. We found that the $\{0, 1\}$ does not work for the prediction of the Lorenz model, possibly because of the symmetry of the model.

The ground-truth data of the Lorenz time-series is prepared using the Runge-Kutta method with the time step $\Delta t = 0.025$. The time series is $t \in [-60, 75]$, and $t \in [-60, -50]$ is used for relaxation, $t \in (-50, 0]$ for training, and $t \in (0, 75]$ for prediction. During the training steps, we compute the output weight by taking the output as $\mathbf{Y} = (A_1(t + \Delta t), A_2(t + \Delta t), A_3(t + \Delta t))$. After training, the RC learns the mapping $(A_1(t), A_2(t), A_3(t)) \rightarrow (A_1(t + \Delta t), A_2(t + \Delta t), A_3(t + \Delta t))$. For the prediction steps, we no longer use the ground-truth input but the estimated data $(\hat{A}_1(t), \hat{A}_2(t), \hat{A}_3(t))$ as schematically shown in Fig. 5c. Using the fixed output weights computed in the training steps, the time evolution of the estimated time-series $(\hat{A}_1(t), \hat{A}_2(t), \hat{A}_3(t))$ is computed by the RC.

Relation of MC and IPC with learning performance

The relevance of MC and IPC is clear by considering the Volterra series of the input-output relation. The input-output relationship is

generically expressed by the polynomial series expansion as

$$Y_n = \sum_{k_1, k_2, \dots, k_n} \beta_{k_1, k_2, \dots, k_n} U_1^{k_1} U_2^{k_2} \dots U_n^{k_n}. \quad (30)$$

Each term in the sum has k_n power of U_n . Instead of polynomial basis, we may use orthonormal basis such as the Legendre polynomials

$$Y_n = \sum_{k_1, k_2, \dots, k_n} \beta_{k_1, k_2, \dots, k_n} \mathcal{P}_{k_1}(U_1) \mathcal{P}_{k_2}(U_2) \dots \mathcal{P}_{k_n}(U_n). \quad (31)$$

Each term in Eqs. (30) and (31) is characterized by the non-negative indices (k_1, k_2, \dots, k_n) . Therefore, the terms corresponding to $j = \sum k_i = 1$ in Y_n have information on linear terms with time delay. Similarly, the terms corresponding to $j = \sum k_i = 2$ have information of second-order nonlinearity with time delay. In this view, the estimation of the output $Y(t)$ is nothing but the estimation of the coefficients $\beta_{k_1, k_2, \dots, k_n}$.

Here, we show how the coefficients can be computed from the magnetization dynamics. In RC, the readout of the reservoir state at i th node (either physical or virtual node) can also be expanded as the Volterra series

$$\tilde{X}^{(i)}(t_n) = \sum_{k_1, k_2, \dots, k_n} \tilde{\beta}_{k_1, k_2, \dots, k_n}^{(i)} U_1^{k_1} U_2^{k_2} \dots U_n^{k_n}, \quad (32)$$

where $\tilde{X}^{(i)}(t_n)$ is measured magnetization dynamics and U_n is the input that drives spin waves. At the linear and second-order in U_k , Eq. (32) is discrete expression of Eqs. (15) and (18). By comparing Eqs. (30) and (32), we may see that MC and IPC are essentially a

reconstruction of $\beta_{k_1, k_2, \dots, k_n}$ from $\tilde{\beta}_{k_1, k_2, \dots, k_n}^{(i)}$ with $i \in [1, M]$. This can be done by regarding $\beta_{k_1, k_2, \dots, k_n}$ as a $T + T(T-1)/2 + \dots$ -dimensional vector, and using the matrix M associated with the readout weights as

$$\beta_{k_1, k_2, \dots, k_n} = M \cdot \begin{pmatrix} \tilde{\beta}_{k_1, k_2, \dots, k_n}^{(1)} \\ \tilde{\beta}_{k_1, k_2, \dots, k_n}^{(2)} \\ \vdots \\ \tilde{\beta}_{k_1, k_2, \dots, k_n}^{(N)} \end{pmatrix}. \quad (33)$$

MC corresponds to the reconstruction of $\beta_{k_1, k_2, \dots, k_n}$ for $\sum_i k_i = 1$, whereas the second-order IPC is the reconstruction of $\beta_{k_1, k_2, \dots, k_n}$ for $\sum_i k_i = 2$. If all of the reservoir states are independent, we may reconstruct N components in $\beta_{k_1, k_2, \dots, k_n}$. In realistic cases, the reservoir states are not independent, and therefore, we can estimate only $<N$ components in $\beta_{k_1, k_2, \dots, k_n}$. For more details, readers may consult Supplementary Information Sections I and II.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request. All codes used in this work are written in MuMax, MATLAB, Fortran, and Python. They are freely available from the corresponding author upon request.

Received: 1 September 2023; Accepted: 14 January 2024;

Published online: 01 March 2024

REFERENCES

- Chumak, A. V., Vasyuchka, V. I., Serga, A. A. & Hillebrands, B. Magnon spintronics. *Nat. Phys.* **11**, 453 (2015).
- Grollier, J. et al. Neuromorphic spintronics. *Nat. Electron.* **3**, 360 (2020).
- Barman, A. et al. The 2021 magnonics roadmap. *J. Phys. Condens. Matter* **33**, 413001 (2021).

- Jaeger, H. & Haas, H. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science* **304**, 78 (2004).
- Torrejon, J. et al. Neuromorphic computing with nanoscale spintronic oscillators. *Nature* **547**, 428 (2017).
- Pathak, J., Hunt, B., Girvan, M., Lu, Z. & Ott, E. Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach. *Phys. Rev. Lett.* **120**, 024102 (2018).
- Jaeger, H. *Short term memory in echo state networks*, Tech. Rep. Technical Report GMD Report 152 (German National Research Center for Information Technology, 2002).
- Maass, W., Natschläger, T. & Markram, H. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Comput.* **14**, 2531 (2002).
- Tsunegi, S. et al. Physical reservoir computing based on spin torque oscillator with forced synchronization. *Appl. Phys. Lett.* **114**, 164101 (2019).
- Rafayelyan, M., Dong, J., Tan, Y., Krzakala, F. & Gigan, S. Large-scale optical reservoir computing for spatiotemporal chaotic systems prediction. *Phys. Rev. X* **10**, 041037 (2020).
- Larger, L. et al. Photonic information processing beyond turing: an optoelectronic implementation of reservoir computing. *Opt. Express* **20**, 3241 (2012).
- Takano, K. et al. Compact reservoir computing with a photonic integrated circuit. *Opt. Express* **26**, 29424 (2018).
- Moon, J. et al. Temporal data classification and forecasting using a memristor-based reservoir computing system. *Nature Electron.* **2**, 480 (2019).
- Milano, G. et al. In materia reservoir computing with a fully memristive architecture based on self-organizing nanowire networks. *Nat. Mater.* **21**, 195 (2022).
- Chen, Z. et al. All-ferroelectric implementation of reservoir computing. *Nat. Commun.* **14**, 3585 (2023).
- Alomar, M. L. et al. Digital implementation of a single dynamical node reservoir computer. *IEEE Trans. Circuits Syst. II: Express Briefs* **62**, 977 (2015).
- Lukoševičius, M. & Jaeger, H. Reservoir computing approaches to recurrent: neural network training. *Comput. Sci. Rev.* **3**, 127 (2009).
- Van der Sande, G., Brunner, D. & Soriano, M. C. Advances in photonic reservoir computing. *Nanophotonics* **6**, 561 (2017).
- Tanaka, G. et al. Recent advances in physical reservoir computing: A review. *Neural Netw.* **115**, 100 (2019).
- Prychynenko, D. et al. Magnetic skyrmion as a nonlinear resistive element: A potential building block for reservoir computing. *Phys. Rev. Appl.* **9**, 014034 (2018).
- Nakane, R., Tanaka, G. & Hirose, A. Reservoir computing with spin waves excited in a garnet film. *IEEE Access* **6**, 4462 (2018).
- Ichimura, T., Nakane, R., Tanaka, G. & Hirose, A. A numerical exploration of signal detector arrangement in a spin-wave reservoir computing device. *IEEE Access* **9**, 72637 (2021).
- Nakane, R., Hirose, A. & Tanaka, G. Spin waves propagating through a stripe magnetic domain structure and their applications to reservoir computing. *Phys. Rev. Res.* **3**, 033243 (2021).
- Nakane, R., Hirose, A. & Tanaka, G. Performance enhancement of a spin-wave-based reservoir computing system utilizing different physical conditions. *Phys. Rev. Appl.* **19**, 034047 (2023).
- Marković, D., Mizrahi, A., Querlioz, D. & Grollier, J. Physics for neuromorphic computing. *Nat. Rev. Phys.* **2**, 499 (2020).
- Dambre, J., Verstraeten, D., Schrauwen, B. & Massar, S. Information processing capacity of dynamical systems. *Sci. Rep.* **2**, 514 (2012).
- S. A., Billings *Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains* (Wiley, 2013).
- Kubota, T., Takahashi, H. & Nakajima, K. Unifying framework for information processing in stochastically driven dynamical systems. *Phys. Rev. Res.* **3**, 043135 (2021).
- Hughes, T. W., Williamson, I. A. D., Minkov, M. & Fan, S. Wave physics as an analog recurrent neural network. *Sci. Adv.* **5**, eaay6946 (2019).
- Marcucci, G., Pierangeli, D. & Conti, C. Theory of neuromorphic computing by waves: Machine learning by rogue waves, dispersive shocks, and solitons. *Phys. Rev. Lett.* **125**, 093901 (2020).
- Rodan, A. & Tino, P. Minimum complexity echo state network. *IEEE Trans. Neural Netw.* **22**, 131 (2011).
- Appeltant, L. et al. Information processing using a single dynamical node as complex system. *Nat. Commun.* **2**, 468 (2011).
- Röhm, A. & Lüdge, K. Multiplexed networks: reservoir computing with virtual and real nodes. *J. Phys. Commun.* **2**, 085007 (2018).
- Dale, M. et al. Reservoir computing with thin-film ferromagnetic devices. arXiv:2101.12700 (2021).
- Furuta, T. et al. Macromagnetic simulation for reservoir computing utilizing spin dynamics in magnetic tunnel junctions. *Phys. Rev. Appl.* **10**, 034063 (2018).

36. Stelzer, F., Röhm, A., Lüdge, K. & Yanchuk, S. Performance boost of time-delay reservoir computing by non-resonant clock cycle. *Neural Netw.* **124**, 158 (2020).
37. Bolt, E. On explaining the surprising success of reservoir computing forecaster of chaos? the universal machine learning dynamical system with contrast to VAR and DMD. *Chaos* **31**, 013108 (2021).
38. Jaeger, H. *Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the "echo state network" approach*, Tech. Rep. Technical Report GMD Report 159 (German National Research Center for Information Technology, 2002).
39. Gonon, L. & Ortega, J.-P. Reservoir computing universality with stochastic inputs. *IEEE Trans. Neural Netw. Learn. Syst.* **31**, 100 (2019).
40. Paquot, Y. et al. Optoelectronic reservoir computing. *Sci. Rep.* **2**, 287 (2012).
41. Duport, F., Schneider, B., Smerieri, A., Haelterman, M. & Massar, S. All-optical reservoir computing. *Opt. Express* **20**, 1958 (2012).
42. Dejonckheere, A. et al. All-optical reservoir computer based on saturation of absorption. *Opt. Express* **22**, 10868 (2014).
43. Vinckier, Q. et al. High-performance photonic reservoir computer based on a coherently driven passive cavity. *Optica* **2**, 438 (2015).
44. Duport, F., Smerieri, A., Akrouf, A., Haelterman, M. & Massar, S. Fully analogue photonic reservoir computer. *Sci. Rep.* **6**, 22381 (2016).
45. Sugano, C., Kanno, K. & Uchida, A. Reservoir Computing Using Multiple Lasers with Feedback on a Photonic Integrated Circuit. *IEEE J. Sel. Top. Quantum Electron.* **26**, 1500409 (2020).
46. Kanao, T. et al. Reservoir computing on spin-torque oscillator array. *Phys. Rev. Appl.* **12**, 024052 (2019).
47. Akashi, N. et al. Input-driven bifurcations and information processing capacity in spintronics reservoirs. *Phys. Rev. Res.* **2**, 043303 (2020).
48. Watt, S., Kostylev, M., Ustinov, A. B. & Kalinikos, B. A. Implementing a magnonic reservoir computer model based on time-delay multiplexing. *Phys. Rev. Appl.* **15**, 064060 (2021).
49. Lee, M. K. & Mochizuki, M. Reservoir computing with spin waves in a Skyrmion crystal. *Phys. Rev. Appl.* **18**, 014074 (2022).
50. B., Hillebrands and J., Hamrle Investigation of spin waves and spin dynamics by optical techniques, in *Handbook of Magnetism and Advanced Magnetic Materials* (John Wiley & Sons, Ltd, 2007).
51. Kubota, T. et al. Half-metallicity and Gilbert damping constant in $\text{Co}_2\text{Fe}_x\text{Mn}_{1-x}\text{Si}$ Heusler alloys depending on the film composition. *Appl. Phys. Lett.* **94**, 122504 (2009).
52. Guillemard, C. et al. Ultralow magnetic damping in Co_2Mn -based Heusler compounds: promising materials for spintronics. *Phys. Rev. Appl.* **11**, 064009 (2019).
53. Guillemard, C. et al. Engineering $\text{Co}_2\text{MnAl}_x\text{Si}_{1-x}$ Heusler compounds as a model system to correlate spin polarization, intrinsic gilbert damping, and ultrafast demagnetization. *Adv. Mater.* **32**, 1908357 (2020).
54. Demidov, V. E., Urazhdin, S. & Demokritov, S. O. Direct observation and mapping of spin waves emitted by spin-torque nano-oscillators. *Nat. Mater.* **9**, 984 (2010).
55. Madami, M. et al. Direct observation of a propagating spin wave induced by spin-transfer torque. *Nat. Nanotechnol.* **6**, 635 (2011).
56. Sani, S. et al. Mutually synchronized bottom-up multi-nanocontact spin-torque oscillators. *Nat. Commun.* **4**, 2731 (2013).
57. Ikeda, S. et al. A perpendicular-anisotropy CoFeB-MgO magnetic tunnel junction. *Nature Materials* **9**, 721 (2010).
58. Jung, S. et al. A crossbar array of magnetoresistive memory devices for in-memory computing. *Nature* **601**, 211 (2022).
59. Deng, Y. et al. Field-free switching of spin crossbar arrays by asymmetric spin current gradient. *Adv. Funct. Mater.* **34**, 2307612 (2024).
60. Gauthier, D. J., Bolt, E., Griffith, A. & Barbosa, W. A. Next generation reservoir computing. *Nat. Commun.* **12**, 1 (2021).
61. Vansteenkiste, A. et al. The design and verification of mumax3. *AIP Adv.* **4**, 107133 (2014).
62. Slonczewski, J. Excitation of spin waves by an electric current. *J. Magn. Magn. Mater.* **195**, L261 (1999).
63. Hamrle, J. et al. Determination of exchange constants of Heusler compounds by Brillouin light scattering spectroscopy: Application to Co_2MnSi . *J. Phys. D: Appl. Phys.* **42**, 084005 (2009).
64. Kubota, T. et al. Structure, exchange stiffness, and magnetic anisotropy of $\text{Co}_2\text{MnAl}_x\text{Si}_{1-x}$ Heusler compounds. *J. Appl. Phys.* **106**, 113907 (2009).
65. Venkat, G., Fangohr, H. & Prabhakar, A. Absorbing boundary layers for spin wave micromagnetics. *J. Magn. Magn. Mater.* **450**, 34 (2018).

ACKNOWLEDGEMENTS

S.M. thanks to CSRN at Tohoku University. Numerical simulations in this work were carried out in part by AI Bridging Cloud Infrastructure (ABCI) at the National Institute of Advanced Industrial Science and Technology (AIST), and by the supercomputer system at the information initiative center, Hokkaido University, Sapporo, Japan. This work is supported by JSPS KAKENHI grant numbers 21H04648, and 21H05000 to S.M., by JST, PRESTO Grant Number JPMJPR22B2 to S.I., X-NICS, MEXT Grant Number JPJ011438 to S.M., and by JST FOREST Program Grant Number JPMJFR2140 to N.Y.

AUTHOR CONTRIBUTIONS

S.M., N.Y. and S.I. conceived the research. S.I., Y.K. and N.Y. carried out simulations. N.Y., S.I. analyzed the results. N.Y., S.I. and S.M. wrote the manuscript. All the authors discussed the results and analysis.

COMPETING INTERESTS

The authors declare no competing interests.

ADDITIONAL INFORMATION

Supplementary information The online version contains supplementary material available at <https://doi.org/10.1038/s44306-024-00008-5>.

Correspondence and requests for materials should be addressed to Natsuhiko Yoshinaga.

Reprints and permission information is available at <http://www.nature.com/reprints>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2024