

Device Variability and Uncertainty Propagation in Memristive Reservoir Computing

Original

Device Variability and Uncertainty Propagation in Memristive Reservoir Computing / Rossetti, D., Corinto, F.. - (2026).
(2026 15th International Conference on Modern Circuits and Systems Technologies (MOCASST) Palma (Spa) 10-12 June 2026) [10.1109/MOCASST70204.2026.11626415].

Availability:

This version is available at: 11583/3014856 since: 2026-09-26T18:25:08Z

Publisher:

IEEE

Published

DOI:10.1109/MOCASST70204.2026.11626415

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

IEEE postprint/Author's Accepted Manuscript

©2026 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

Device Variability and Uncertainty Propagation in Memristive Reservoir Computing

Davide Rossetti

*Dept. Electronic and Telecommunications
Politecnico di Torino
Turin, Italy
davide.rossetti@polito.it*

Fernando Corinto

*Dept. Electronic and Telecommunications
Politecnico di Torino
Turin, Italy
fernando.corinto@polito.it*

Abstract—Memristive crossbar arrays provide an efficient hardware substrate for in-memory and neuromorphic computing architectures. Reservoir Computing (RC) is particularly well suited for such systems because only the output layer requires training, making it attractive for hardware implementations. However, memristive devices exhibit intrinsic variability due to fabrication mismatch, cycle-to-cycle fluctuations, and conductance drift, which introduce stochasticity in the reservoir dynamics. In this work we develop a probabilistic formulation of memristive reservoir computing that explicitly models conductance variability in the crossbar array. Analytical expressions are derived to describe how device uncertainty propagates through the reservoir states and determines the predictive distribution at the readout layer. This framework enables the analysis of dynamical regimes in which hardware variability either degrades performance or can instead enhance the diversity of the reservoir dynamics. The theoretical results are validated through numerical experiments on chaotic time-series prediction tasks, including the Mackey–Glass benchmark, showing how device variability affects both prediction accuracy and predictive uncertainty. These results provide theoretical insight into uncertainty propagation in memristive reservoirs and establish connections between variability-aware reservoir computing and stochastic random feature methods.

Index Terms—Reservoir Computing, Memristors, In-Memory Computing, Time-Discrete Dynamical Systems, Uncertainty Propagation, Device Non-idealities

I. INTRODUCTION

Reservoir Computing (RC) has emerged as an efficient framework for processing temporal and sequential data while avoiding the expensive training procedures required by conventional recurrent neural networks [1, 2]. In RC architectures, a fixed nonlinear dynamical system projects the input signal into a high-dimensional state space, while only the output layer is trained, significantly reducing training complexity and making RC particularly suitable for hardware implementations [3]. In recent years, the concept of *physical* or *next-generation reservoirs* has attracted increasing attention, where the reservoir is implemented directly using physical dynamical systems rather than simulated neural networks [4]. Various physical implementations have been explored, including photonic and optical systems [5, 6], which exploit the intrinsic dynamics of physical devices to perform temporal information processing. Among the different hardware candidates, memristive crossbar arrays have emerged as a promising platform for

neuromorphic and in-memory computing [7, 8]. These systems enable massively parallel matrix–vector multiplications directly in the analog domain and offer high computational density and low power consumption, making them attractive for edge AI and neuromorphic applications [9, 10]. However, memristive devices exhibit intrinsic physical variability arising from fabrication mismatch, stochastic switching behavior and conductance drift [11]. While these effects are typically treated as non-idealities, randomness and variability can play a constructive role in dynamical computing systems by enriching the diversity of internal representations [3]. Recent works have demonstrated memristor-based reservoir computing systems capable of performing temporal processing tasks [12]. Nevertheless, the role of device variability in shaping the stochastic dynamics of memristive reservoirs remains insufficiently understood. Motivated by this observation, this work develops a probabilistic formulation of memristive reservoir computing that explicitly incorporates device variability into the dynamical model. The conductance matrices of the memristive crossbar are modeled as random variables, allowing us to analyze how hardware variability propagates through the reservoir states and affects the predictive distribution at the readout layer. The main contributions of this work can be summarized as follows:

- We introduce a time-discrete formulation of memristive reservoir computing that explicitly models device variability in memristive arrays.
- We derive analytical expressions describing how conductance uncertainty propagates through the reservoir states and affects the predictive distribution at the readout layer.
- Through numerical experiments on chaotic time-series prediction tasks, we investigate how device variability influences prediction accuracy and uncertainty, and how multiple stochastic reservoir realizations can be combined through ensemble averaging.

II. TIME-DISCRETE MEMRISTIVE RESERVOIR MODEL

Memristive crossbar reservoirs exhibit intrinsic device variability. To capture these effects, the reservoir is modeled as a time-discrete dynamical system with stochastical variation where both the reservoir state and the memristive conductances evolve in time. The coupled dynamics can be written as

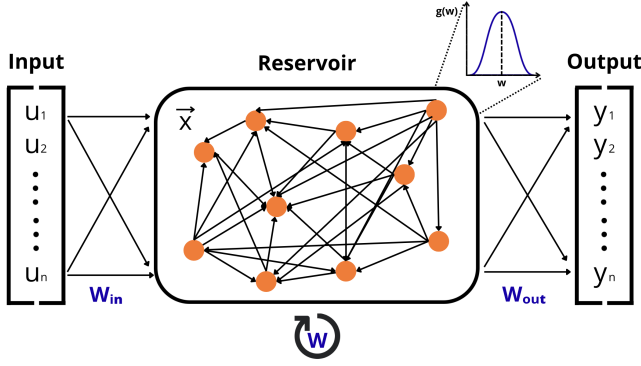


Fig. 1. The crossbar conductance matrix represents the adjacency matrix of the reservoir graph. Each memristive conductance is modeled as a Gaussian random variable in order to account for intrinsic device variability. The readout layer performs a linear projection of the reservoir states.

$$W_{t+1} = W_t + F(W_t, x_t) + \varepsilon_0 \quad (1)$$

$$W_{t+1}^{in} = W_t^{in} + F_1(W_t^{in}, u_t) + \varepsilon_0^{in} \quad (2)$$

$$x_{t+1} = f(W_t x_t + W_t^{in} u_t) \quad (3)$$

where $x_t \in \mathbb{R}^{N_r}$ denotes the reservoir state and u_t the input signal. The matrices W_t and W_t^{in} correspond to the conductance matrices implemented in a memristive crossbar array which directly encodes the adjacency matrix of the reservoir graph, where each memristive device represents a weighted connection between two reservoir nodes. Fully connected reservoirs can be obtained by programming all conductances, while sparse topologies can be realized by setting selected conductances to zero. The stochastic terms ε_0 and ε_0^{in} model initial intrinsic hardware variability and are assumed to follow Gaussian distributions.

$$\varepsilon_0 \sim \mathcal{N}(0, \Sigma_W), \quad \varepsilon_0^{in} \sim \mathcal{N}(0, \Sigma_{W^{in}}).$$

This formulation captures two distinct timescales typical of memristive systems: a fast reservoir state evolution governed by the nonlinear activation function $f(\cdot)$, and a slower conductance evolution described by the functions $F(\cdot)$ and $F_1(\cdot)$.

A. Slow Reservoir Dynamics

For the analysis presented in this work, we consider a quasi-static variability regime in which memristor conductances evolve on a much slower timescale than the reservoir state dynamics. Under this assumption, the weight matrices can be approximated as random but quasi-static variables

$$W = \bar{W} + \Delta W \quad W^{in} = \bar{W}^{in} + \Delta W^{in} \quad (4)$$

where $\Delta W \sim \mathcal{N}(0, \Sigma_W)$ and $\Delta W^{in} \sim \mathcal{N}(0, \Sigma_{W^{in}})$. This approximation captures stationary device variability while preserving the fast reservoir dynamics.

B. Linearized State Dynamics

The reservoir update is given by

$$x_{t+1} = f(Wx_t + W^{in}u_t). \quad (5)$$

Define the pre-activation variable

$$z_t = Wx_t + W^{in}u_t. \quad (6)$$

Let $\mu_x = \mathbb{E}[x_t]$ denote the mean reservoir state and $\mu_z = \bar{W}\mu_x + \bar{W}^{in}u_t$ the corresponding mean pre-activation variable. To study the propagation of device variability through the reservoir, the nonlinear activation function is linearized around the mean operating point:

$$f(z_t) \approx f(\mu_z) + J_f(z_t - \mu_z) \quad (7)$$

where

$$J_f = \left. \frac{\partial f}{\partial z} \right|_{\mu_z}. \quad (8)$$

denotes the Jacobian of the activation function evaluated at the mean operating point. Let $\delta x_t = x_t - \mu_x$ denote the deviation of the reservoir state from its mean value. Substituting the decomposition of the weight matrices $W = \bar{W} + \Delta W$ and $W^{in} = \bar{W}^{in} + \Delta W^{in}$ into the reservoir update equation and retaining only first-order terms yields the approximate fluctuation dynamics

$$\delta x_{t+1} = J_f (\bar{W}\delta x_t + \Delta W\mu_x + \Delta W^{in}u_t). \quad (9)$$

C. Uncertainty Propagation and Stability

Let

$$\Sigma_x(t) = \mathbb{E}[\delta x_t \delta x_t^T] \quad (10)$$

denote the reservoir state covariance. Using the linearized fluctuation dynamics and assuming that the conductance perturbations have zero mean and are independent of the state fluctuations, the covariance evolution can be obtained by taking the expectation of the outer product $\delta x_{t+1} \delta x_{t+1}^T$. This yields

$$\Sigma_x(t+1) = A_x \Sigma_x(t) A_x^T + \Gamma \quad (11)$$

where $A_x = J_f \bar{W}$ and $\Gamma = J_f (\Sigma_W \mu_x \mu_x^T + \Sigma_{W^{in}} u_t u_t^T) J_f^T$. The matrix Γ captures the contribution of intrinsic device variability to the reservoir state covariance, where Σ_W and $\Sigma_{W^{in}}$ represent the statistical fluctuations of memristive conductances arising from device-level non-idealities (e.g., mismatch, stochastic switching, and drift), while J_f determines how such perturbations propagate through the nonlinear activation. The stability of the linearized stochastic reservoir is determined by the spectral radius of the effective state matrix A_x . The dynamics remains stable when

$$\rho(A_x) < 1. \quad (12)$$

Under this condition, fluctuations induced by device variability remain bounded and the covariance of the reservoir state converges to a stationary solution. The reservoir output is computed through a linear readout

$$y_t = W^{out} x_t. \quad (13)$$

Since the reservoir state is approximated as a Gaussian random variable under the linearized stochastic dynamics $x_t \sim \mathcal{N}(\mu_x, \Sigma_x)$, the output distribution is also Gaussian $y_t \sim \mathcal{N}(\mu_y, \Sigma_y)$ with

$$\mu_y = W^{out} \mu_x \quad (14)$$

$$\Sigma_y = W^{out} \Sigma_x W^{out T}. \quad (15)$$

Therefore, the predictive uncertainty at the output layer is directly determined by the covariance of the reservoir state induced by intrinsic device variability. In classical reservoir computing, the readout matrix W^{out} is typically trained using ridge regression on the reservoir states. The stochastic formulation introduced in this work also enables alternative uncertainty-aware training strategies. In particular, three approaches can be considered:

- **Mean-based training:** the readout is trained using only the mean reservoir state μ_x , corresponding to standard deterministic reservoir computing.
- **Weighted least squares:** prediction errors are weighted by the inverse predictive variance, giving lower importance to uncertain reservoir states.
- **Probabilistic training:** the readout parameters are optimized by maximizing the Gaussian likelihood of the predicted output distribution.

In the experiments presented in this paper, the readout is trained using the standard ridge regression formulation for computational efficiency, while the predictive covariance Σ_y is evaluated analytically using the proposed uncertainty propagation framework in order to quantify the prediction uncertainty.

D. Relation to Random Feature Kernels

Reservoir computing can be interpreted as a nonlinear random feature mapping $\phi(u_t) = x_t$, followed by a linear readout layer. The resulting kernel between two inputs u_i and u_j is therefore

$$k(u_i, u_j) = \phi(u_i)^T \phi(u_j). \quad (16)$$

When the reservoir dynamics is stochastic, the reservoir state becomes a random variable $x_t \sim \mathcal{N}(\mu_x, \Sigma_x)$. In this case the expected kernel can be written as

$$k(u_i, u_j) = \mathbb{E}[x_i^T x_j] = \mu_x(u_i)^T \mu_x(u_j) + \text{Tr}(\Sigma_{x,ij}), \quad (17)$$

showing that device variability directly contributes to the effective kernel induced by the reservoir. This interpretation links stochastic reservoir dynamics to random feature kernel

methods, where device variability enriches the effective kernel representation. From this perspective, different stochastic realizations of the reservoir can be interpreted as different random feature mappings of the input signal.

III. NUMERICAL EXPERIMENTS

To evaluate the proposed stochastic formulation of memristive reservoir computing, we perform numerical experiments on the prediction of chaotic time series. The experiments are conducted on the Mackey–Glass time series, which is a widely used benchmark in reservoir computing and nonlinear time-series prediction [13, 14]. The system is governed by the delay differential equation

$$\frac{dx(t)}{dt} = \beta \frac{x(t-\tau)}{1 + x(t-\tau)^n} - \gamma x(t), \quad (18)$$

where $\beta = 0.2$, $\gamma = 0.1$, $n = 10$, and $\tau = 17$ are the system parameters chosen following the standard configuration adopted in the reservoir computing literature which results in chaotic dynamics. The task consists of predicting the 20th future value of the signal from its past observations. In the experiments, the reservoir consists of $N_r = 64$ internal nodes with a spectral radius $\rho(W) = 0.9$ smaller than one in order to ensure the echo-state property. The readout layer is trained using ridge regression with regularization parameter $\lambda = 5 \cdot 10^{-7}$. The input to the reservoir is constructed from a delay embedding of the time series, and the readout layer is trained to predict a future sample of the sequence.

A. Exploiting Device Variability through Reservoir Ensembles

Each stochastic (or hardware) realization of the reservoir produces a different state trajectory of the input signal into the reservoir state space. Formally, the i -th realization generates a feature map $\phi^{(i)}(u_t) = x_t^{(i)}$, where $x_t^{(i)}$ denotes the state produced by the i -th stochastic reservoir realization. From the perspective of random feature methods, each reservoir therefore provides a different projection of the input signal into a high-dimensional nonlinear representation space. To exploit this diversity, we combine the predictions of multiple stochastic reservoirs through ensemble averaging. Given M independent reservoirs, the ensemble prediction is defined as

$$\hat{y}_t = \frac{1}{M} \sum_{i=1}^M y_t^{(i)}. \quad (19)$$

This approach aggregates multiple random feature mappings generated by different dynamical realizations of the reservoir [15]. In addition to enlarging the representation space, ensemble averaging reduces the variance of the estimator, leading to progressively more stable predictions. Under the assumption of independent realizations, the variance of the ensemble prediction scales approximately as

$$\text{Var}(\hat{y}_t) \approx \frac{1}{M} \text{Var}(y_t). \quad (20)$$

Fig. 2 illustrates the prediction of the Mackey–Glass time series obtained with the reservoir model. The top panel shows

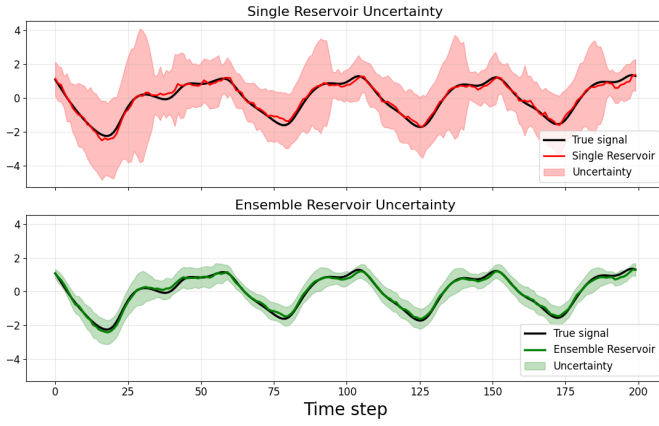


Fig. 2. Prediction of the Mackey–Glass time series using a stochastic reservoir. Solid curves denote the mean predicted trajectory, while shaded regions indicate uncertainty bands due to device variability. In hardware, the actual trajectory corresponds to a single stochastic realization within these bands. Top: single reservoir. Bottom: ensemble prediction obtained by averaging multiple reservoirs, resulting in reduced variance and narrower uncertainty bands.

the prediction produced by a single reservoir together with the corresponding uncertainty band. In the simulations, memristive device variability is modeled as Gaussian perturbations in the reservoir weights with a standard deviation equal to 10% of the nominal conductance values, consistent with experimentally reported variability levels in memristive devices, which can range from a few percent to several tens of percent depending on technology and operating conditions. This value represents a moderate variability regime, and is used to highlight the impact of stochasticity on the reservoir dynamics, while the proposed framework remains applicable to arbitrary variability levels. The bottom panel shows the prediction obtained by averaging multiple reservoir realizations. This can be interpreted as combining the outputs of several hardware reservoirs with slightly different device configurations. Ensemble averaging reduces both the prediction error and the predictive uncertainty. In our simulations, using an ensemble of ten reservoirs reduces the mean squared error by approximately 30% compared to a single reservoir. Under the assumption of independent realizations, the variance of the ensemble prediction decreases approximately as $1/M$, implying that the width of the uncertainty band scales as $1/\sqrt{M}$. From a hardware perspective, each stochastic reservoir corresponds to a different physical realization of the memristive crossbar arising from intrinsic device variability. Combining multiple realizations therefore provides a richer set of nonlinear features while simultaneously reducing prediction variance. These results suggest that device variability, often considered a limitation in memristive hardware, can instead be exploited as a computational resource.

IV. CONCLUSION

In this work we presented a dynamical framework to analyze the impact of device variability in memristive reservoir computing systems. By modeling the conductance matrices

of the reservoir as random variables, we derived analytical expressions describing how hardware uncertainty propagates through the reservoir dynamics and determines the predictive distribution at the readout layer. This formulation represents the reservoir state through its mean trajectory and covariance, enabling an explicit quantification of prediction uncertainty. Numerical experiments on the Mackey–Glass benchmark show that stochastic reservoirs naturally generate uncertainty bands reflecting the intrinsic variability of memristive devices. While individual realizations may produce fluctuating predictions, ensemble averaging significantly improves robustness and reduces prediction variance. These results indicate that device variability, often considered a limitation of emerging hardware technologies, can instead be exploited as a computational resource that enriches reservoir dynamics and expands the effective representation space. The proposed framework also establishes a connection between variability-aware reservoir computing and stochastic random feature methods. Future work will investigate experimental implementations of memristive reservoirs and strategies to exploit device variability in large-scale neuromorphic hardware.

REFERENCES

- [1] W. Maass, T. Natschlager, and H. Markram, “Real-time computing without stable states: a new framework for neural computation based on perturbations,” *Neural Computation*, vol. 14, no. 11, pp. 2531–2560, 2002.
- [2] H. Jaeger and H. Haas, “Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication,” *Science*, vol. 304, no. 5667, pp. 78–80, 2004.
- [3] M. Lukosevicius and H. Jaeger, “Reservoir computing approaches to recurrent neural network training,” *Computer Science Review*, vol. 3, no. 3, pp. 127–149, 2009.
- [4] G. Tanaka et al., “Recent advances in physical reservoir computing: A review,” *Neural Networks*, vol. 115, pp. 100–123, 2019.
- [5] L. Appeltant et al., “Information processing using a single dynamical node as complex system,” *Nature Communications*, vol. 2, p. 468, 2011.
- [6] K. Vandoorne et al., “Experimental demonstration of reservoir computing on a silicon photonics chip,” *Nature Communications*, vol. 5, p. 3541, 2014.
- [7] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, “The missing memristor found,” *Nature*, vol. 453, no. 7191, pp. 80–83, 2008.
- [8] M. Prezioso et al., “Training and operation of an integrated neuromorphic network based on metal-oxide memristors,” *Nature*, vol. 521, no. 7550, pp. 61–64, 2015.
- [9] C. Li et al., “Analogue signal and image processing with large memristor crossbars,” *Nature Electronics*, vol. 1, pp. 52–59, 2018.
- [10] S. Ambrogio et al., “Equivalent-accuracy accelerated neural-network training using analogue memory,” *Nature*, vol. 558, no. 7708, pp. 60–67, 2018.
- [11] D. Ielmini and H.-S. P. Wong, “In-memory computing with resistive switching devices,” *Nature Electronics*, vol. 1, pp. 333–343, 2018.
- [12] Y. Zhong et al., “A memristor-based analogue reservoir computing system for real-time and power-efficient signal processing,” *Nature Electronics*, vol. 5, no. 10, pp. 672–681, 2022.
- [13] D. Badalotti, C. Baldassi, M. Mézard, M. Scardecchia, and R. Zecchina, “Dynamical learning in deep asymmetric recurrent neural networks,” *arXiv preprint arXiv:2509.05041*, 2025.
- [14] M. C. Mackey and L. Glass, “Oscillation and chaos in physiological control systems,” *Science*, vol. 197, no. 4300, pp. 287–289, 1977.
- [15] H. Jaeger, “A tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the echo state network approach,” GMD-Forschungszentrum Informationstechnik, 2002.