Chaos
An Interdisciplinary Journal of Nonlinear Science

RESEARCH ARTICLE | JULY 05 2023

Memory-based parameterization with differentiable solver: Application to Lorenz '96 FREE

Special Collection: Theory-informed and Data-driven Approaches to Advance Climate Sciences

Mohamed Aziz Bhouri ; Pierre Gentine

Chaos 33, 073116 (2023)
https://doi.org/10.1063/5.0131929
CHORUS

Memory-based parameterization with differentiable solver: Application to Lorenz '96

Cite as: Chaos 33, 073116 (2023); doi: 10.1063/5.0131929
Submitted: 25 October 2022 · Accepted: 6 June 2023 ·
Published Online: 5 July 2023

Mohamed Aziz Bhouri[a] and Pierre Gentine[b]

AFFILIATIONS

Department of Earth and Environmental Engineering, Columbia University, New York, New York 10027, USA

Note: This article is part of the Focus Issue, Theory-informed and Data-driven Approaches to Advance Climate Sciences.
[a]**Author to whom correspondence should be addressed:** mb4957@columbia.edu
[b]Department of Earth and Environmental Sciences, Columbia University, New York, USA. **Electronic mail:** pg2328@columbia.edu

ABSTRACT

Physical parameterizations (or closures) are used as representations of unresolved subgrid processes within weather and global climate models or coarse-scale turbulent models, whose resolutions are too coarse to resolve small-scale processes. These parameterizations are typically grounded on physically based, yet empirical, representations of the underlying small-scale processes. Machine learning-based parameterizations have recently been proposed as an alternative solution and have shown great promise to reduce uncertainties associated with the parameterization of small-scale processes. Yet, those approaches still show some important mismatches that are often attributed to the stochasticity of the considered process. This stochasticity can be due to coarse temporal resolution, unresolved variables, or simply to the inherent chaotic nature of the process. To address these issues, we propose a new type of parameterization (closure), which is built using memory-based neural networks, to account for the non-instantaneous response of the closure and to enhance its stability and prediction accuracy. We apply the proposed memory-based parameterization, with differentiable solver, to the Lorenz '96 model in the presence of a coarse temporal resolution and show its capacity to predict skillful forecasts over a long time horizon of the resolved variables compared to instantaneous parameterizations. This approach paves the way for the use of memory-based parameterizations for closure problems.

Turbulence, ocean, weather, and climate models involve physical processes across different scales. Given the available computational resources, small-scale processes are typically not resolved in those models but are rather represented by parameterization schemes. Machine learning has been recently used to improve existing parameterization approaches, yet those methods still show some important mismatches that are often attributed to stochasticity in the considered processes. This stochasticity can be due to noisy and sparse data, unresolved physical variables, or simply to the inherent chaotic nature of the process. In this work, we develop a memory-based parameterization scheme that is trained while solving the parameterized dynamical system. The resulting parameterized model is capable of predicting skillful forecasts of the resolved physical variables compared to instantaneous parameterizations. This approach paves the way for the use of memory-based approaches for parameterization problems.

I. INTRODUCTION

Parameterization schemes, or closures, are approximate representations of unresolved subgrid processes in turbulence, ocean, weather, and climate models and are the most dominant source of uncertainty in model predictions. The corresponding errors have been reduced throughout the improvements of existing (physically based) parameterizations and the development of new schemes, yet these errors cannot be completely eliminated because of inherent model structural errors, as they try to approximate complex physical processes. To address these structural errors, recently, several groups have started developing machine learning-based parameterizations, which have been shown to dramatically improve the representation of subgrid physical processes and strongly reduce parameterization structural errors compared to standard parameterizations.[1–5] Another source of uncertainty in models stems from the inherent stochastic nature of many physical processes in nature.[6–9]

Different approaches have been proposed for parameterization schemes. We can distinguish between approaches that are extensively used thanks to their physically based implementation and model prediction improvements[10–13] and statistical schemes, such as for convection and cloud formation.[14–20] Data-driven approaches are another option to build parameterizations based on observations or high-resolution model simulations used as truth, in order to account for the subgrid variability that is unresolved in the lower-resolution models.[21–26] The major limitation of these approaches is that they only have access to coarse-scale variables to represent unresolved small-scale processes.

Machine learning-based methods have been successfully developed in order to parameterize various atmospheric[1–4,27–29] and oceanic processes,[5] including turbulence,[30,31] by inferring instantaneous closure terms that depend mostly on the current model time step. These models might not capture any memory effect, such as in ocean eddies or decaying turbulence. Other approaches relied on model order reduction techniques to parameterize dynamical systems, and among these methods, the memory dependence is understood within the framework of the Mori–Zwanzig formalism.[16,32–36] However, most of these existing techniques require training data for the closure term and do not learn directly from the state variable observations.[32,33] Some of them are also specifically developed for first-order forward time integration with limited stability and accuracy.[33] Others use memory terms that are limited to a single time-lag or are implemented in Fourier space with modes truncation instead of a direct implementation to the time-dependent differential equation.[32] It is worth noting that the NARMAX (nonlinear autoregression moving average with exogenous input)-based parameterization scheme, which accounts for memory, allows more flexibility in terms of the number of time-lags and time integration scheme.[37] However, it is based on a predefined form of the closure term with a limited number of parameters to be tuned and does not take advantage of recent development in machine learning, including deep neural networks and their proven expressiveness. Many physical processes, such as turbulence, clouds, or ocean eddies, may have substantial memory. Excluding memory from the parameterization (closure) can lead to further uncertainties and appear as a source of stochasticity, as the model will generate varying predictions given only the current time state. Such modeling choice might limit considerably the online forecasts of the parameterized model.

To overcome these limitations, we develop a memory-based parameterization that depends not only on the current state of the resolved variables but also on their previous states. We also rely on a differentiable solver of the parameterized model in order to learn the parameterization implicitly by requiring only observations of the resolved variables and not of the closure term. Similarly to previous work on differential programming for dynamical systems identification,[38,39] a differentiable solver is used in order to back-propagate the gradient of the loss function with respect to the tunable parameters within the differential equation solver. The model is further evaluated in an online setting, i.e., when integrating (in time) the parameterized dynamical system based only on the resolved variables. The time-lags defining the memory terms of the parameterization and the time step used to solve the parameterized model are judiciously chosen in order to avoid data interpolation during temporal integration of the differentiable parameterized model. We will show that the parameterization is capable of not only implicitly learning the coupling terms but also of accounting and correcting for the numerical errors introduced by the temporal discretization in online setting. We apply the proposed memory-based stochastic parameterization to the Lorenz '96 model using simulated observational data with a coarse temporal resolution and show its capability of producing accurate temporal forecasts for the resolved variables compared to instantaneous parameterizations, such as the Wilks scheme[40] or an instantaneous feedforward neural network.

The Lorenz '96 model and the proposed memory-based parameterization scheme are detailed in Sec. II. The forecast results and study of numerical error and stability of the proposed parameterization schemes are presented in Sec. III. The proposed memory-based parameterization scheme is evaluated against non-memory-based (instantaneous) parameterization schemes. Finally, in Sec. IV, we summarize the proposed parameterization schemes and their results, discuss shortcomings of the proposed approach, and carve out directions for future investigation.

II. METHODS

A. Lorenz '96 model

The Lorenz '96 model is a two time-scale dynamical system, which mimics the non-linear dynamics of the extratropical atmosphere with a simplified representation of multiscale interactions and nonlinear advection.[6] It consists of a set of equations coupling variables evolving over slow X_k and fast timescales Y_j,

$$\frac{dX_k}{dt} = -X_{k-1}(X_{k-2} - X_{k+1}) - X_k + F - \frac{hc}{b}\sum_{j=J(k-1)+1}^{kJ} Y_j,\ k = 1,\dots,K, \tag{1}$$

$$\frac{dY_j}{dt} = -cbY_{j+1}(Y_{j+2} - Y_{j-1}) - cY_j + \frac{hc}{b}X_{\lfloor (j-1)/J \rfloor + 1},\ j = 1,\dots,JK. \tag{2}$$

The model includes K large-scale, low-frequency (slow-varying) variables X_k, $k = 1,\dots,K$. Each slow-varying variable X_k is coupled to a larger number of small-scale, high-frequency (fast-varying) variables Y_j, $j = J(k-1)+1,\dots,kJ$. The fast time scales impact the slow variables through the coupling term $\sum_{j=J(k-1)+1}^{kJ} Y_j$ and the coupling strength depends on three key parameters: b, c, and h. Parameter b determines the magnitude of the non-linear interactions between the fast variables. Parameter c controls how rapidly the fast-varying variables fluctuate compared to the slow-varying variables. Finally, the parameter h governs the strength of the coupling between the slow- and fast-varying variables.

The chaotic dynamical system Lorenz '96 is a useful ansatz for testing different numerical methods in atmospheric modeling thanks to its transparency, low computational cost, and simplicity compared to full-blown Global Climate Models (GCMs). The interaction between variables of different scales makes the Lorenz '96 model of particular interest when evaluating new parameterization

methodologies. As such, it was used in assessing different techniques that could later be incorporated into more complex models.[41,42]

The Lorenz '96 model has been extensively used and studied as a test bed in various studies, including data assimilation approaches,[43,44] stochastic parameterization schemes,[37,45,46] and machine learning-based parameterizations.[28,29,47,48] The full Lorenz '96 system (1) and (2) including the fast variables is considered as the "true" model and is used to generate the dataset.

B. Standard (instantaneous) parameterizations

The main objective of the parameterizations for the Lorenz '96 model is to replace the coupling term as a function of the resolved slow-varying variables X_k, $k = 1, \ldots, K$. In an instantaneous parameterization, this is done using the resolved variables at the current time step,

$$\frac{dX_k^*}{dt} = -X_{k-1}^*(X_{k-2}^* - X_{k+1}^*) - X_k^* + F + P(X_k^*\ bi\theta),\ k = 1, \ldots, K, \tag{3}$$

where X_k^*, $k = 1, \ldots, K$ is the forecast estimate of X_k based on the parameterized subgrid tendency $P(\cdot;\cdot)$ and $\boldsymbol{\theta}$ is a vector of unknown parameters for the parameterization, which are learned given the available training dataset.

In our study, the goal is to infer a surrogate model for the parameterization (closure) that replaces the coupling term. We also aim to build a parameterization that remains accurate over long time periods when tested online, i.e., when integrating (in time) the parameterized dynamical system based only on the resolved variables. As we will show, this task is typically unfeasible unless we allow the parameterized subgrid tendency term to depend not only on the resolved variables evaluated at the current time but also on the previous time steps. Indeed, relying on an instantaneous parameterization in the form of Eq. (3) will likely not provide an online forecast estimate X_k^*, $k = 1, \ldots, K$, which "matches" the "true" model variables X_k, $k = 1, \ldots, K$, since the state of these variables at a time instance t does not only depend on their state at a previous time instance but also on the state of the fast-varying variables Y_j, $j = 1, \ldots, JK$, at the same previous time instance. In other words, in a forecast setting, the next time step prediction for X_k^*, $k = 1, \ldots, K$, does not only depend on its current time state but also on the current state of the inaccessible small-scale variables Y_j, $j = 1, \ldots JK$. Our working hypothesis is that including previous observations of coarse-scale variables may provide more information about the current state of unobserved short-scale variables. Equivalently, the variance of the coupling term is expected to be lower when being conditioned on a history of observations rather being dependent only on the instantaneous resolved variable at the current time.

C. Memory-based parameterization

To remedy the limitations of having a parameterization that depends only on the current state of the resolved variables, we propose a parameterized subgrid tendency that depends not only on the resolved variables evaluated at the current time steps but also on their states at previous time steps as follows:

$$\frac{dX_k^*}{dt} = -X_{k-1}^*(X_{k-2}^* - X_{k+1}^*) - X_k^* + F + P\big(X_k^*(t), X_k^*(t-\tau_1), \ldots, X_k^*(t-\tau_{n_h}); \boldsymbol{\theta}\big),\ k = 1, \ldots, K, \tag{4}$$

where τ_i, $i = 1, \ldots, n_h$ are the time-lags that define the previous time steps to consider for the parameterization. With such a parameterization, we want to assess whether the effect of the unresolved fast-varying variables on the forecast of the resolved slow-varying variables is (partly) embedded within the historical (previous time steps) evolution of the resolved coarse variables. Examples of such behavior would be convective aggregation,[49–51] (decaying) turbulence, or ocean eddies. Such memory-based inference can be carried out using machine learning techniques based on a temporal recurrence/memory.

One technical challenge that may result from considering a memory-based parameterization is that the resulting dynamical system (4) will consist of a Delay Differential Equation (DDE) instead of an Ordinary Differential Equation (ODE). Using Runge–Kutta schemes for DDEs may result in interpolating the existing data in order to perform time marching,[52,53] which can be an additional significant source of error for the inference task. Fourth-order Runge–Kutta (RK4) schemes are the standard time-stepping methods that are used not only to solve the "true" model but also to solve the parameterized one.[6,29,40,46,54] Explicit RK4 schemes are suitable for such a parameterization problem given their stability, and the fact that using implicit schemes would result in a computational bottleneck when using machine learning surrogate models for the parameterization. In such a case, performing time marching with an implicit time-stepping scheme would require solving nonlinear equations depending on the output of the machine learning surrogate models that are used for the parameterization, which is generally infeasible when using deep neural networks.

In order to explain the choice of the time-lags and time-stepping for the parameterized model that are needed to avoid data interpolation, we re-write the DDE (4) as follows:

$$\frac{d\boldsymbol{X}^*}{dt} = f\big(\boldsymbol{X}^*(t), \boldsymbol{X}^*(t-\tau_1), \ldots, \boldsymbol{X}^*(t-\tau_{n_h}); \boldsymbol{\theta}\big), \tag{5}$$

where $\boldsymbol{X}^*$ refers to the K-dimensional vector concatenating X_k^*, $k = 1, \ldots, K$ and $f(\cdot)$ refers to the whole right-hand-side term of Eq. (4).

If one chooses the time-lags as multiples of the time step Δt for which the data are available: $\tau_i = i\Delta t$, $i = 1, \ldots, n_h$, then applying RK4 with a time step equal to Δt to Eq. (5) gives the following time-stepping:

$$\boldsymbol{X}^*(t+\Delta t; \boldsymbol{\theta}) = \boldsymbol{X}^*(t) + \frac{1}{6}(r_1 + 2\,r_2 + 2\,r_3 + r_4), \tag{6}$$

$$r_1 = \Delta t\, f\big(\boldsymbol{X}^*(t), \boldsymbol{X}^*(t-\Delta t), \ldots, \boldsymbol{X}^*(t-n_h\Delta t); \boldsymbol{\theta}\big), \tag{7}$$

$$r_2 = \Delta t\, f\left(\boldsymbol{X}^*(t) + \frac{r_1}{2}, \boldsymbol{X}^*\left(t - \frac{\Delta t}{2}\right), \boldsymbol{X}^*\left(t - \frac{3\Delta t}{2}\right), \ldots, \boldsymbol{X}^*\left(t - \frac{(2n_h-1)\Delta t}{2}\right); \boldsymbol{\theta}\right), \tag{8}$$

$$r_3 = \Delta t f\Big(\boldsymbol{X}^*(t) + \frac{r_2}{2}, \boldsymbol{X}^*\left(t - \frac{\Delta t}{2}\right), \boldsymbol{X}^*\left(t - \frac{3\Delta t}{2}\right), \ldots, \boldsymbol{X}^*\left(t - \frac{(2n_h - 1)\Delta t}{2}\right); \boldsymbol{\theta}\Big), \tag{9}$$

$$r_4 = \Delta t f\Big(\boldsymbol{X}^*(t) + r_3, \boldsymbol{X}^*(t), \boldsymbol{X}^*(t - \Delta t), \ldots, \boldsymbol{X}^*\big(t - (n_h - 1)\Delta t\big); \boldsymbol{\theta}\Big). \tag{10}$$

This means that if we want to solve the parameterized model in order to fit the predicted next time-step variables, we would need the midpoint value of the available trajectory evaluated at a time grid with a time step equal to Δt. In order to avoid interpolating data, we actually double the time step to integrate the parameterized model and consider the time-lags as multiples of $2\ \Delta t$: $\tau_i = 2i\Delta t$, $i = 1, \ldots, n_h$. Using these settings, the RK4 time-stepping of Eq. (5) becomes

$$\boldsymbol{X}^*(t + 2\ \Delta t; \boldsymbol{\theta}) = \boldsymbol{X}^*(t) + \frac{1}{6}(r_1 + 2\ r_2 + 2\ r_3 + r_4), \tag{11}$$

$$r_1 = 2\ \Delta t f\big(\boldsymbol{X}^*(t), \boldsymbol{X}^*(t - 2\ \Delta t), \ldots, \boldsymbol{X}^*(t - 2n_h \Delta t); \boldsymbol{\theta}\big), \tag{12}$$

$$r_2 = 2\ \Delta t f\Big(\boldsymbol{X}^*(t) + \frac{r_1}{2}, \boldsymbol{X}^*(t - \Delta t), \boldsymbol{X}^*(t - 3\ \Delta t), \ldots, \boldsymbol{X}^*\big(t - (2n_h - 1)\Delta t\big); \boldsymbol{\theta}\Big), \tag{13}$$

$$r_3 = 2\ \Delta t f\Big(\boldsymbol{X}^*(t) + \frac{r_2}{2}, \boldsymbol{X}^*(t - \Delta t), \boldsymbol{X}^*(t - 3\ \Delta t), \ldots, \boldsymbol{X}^*\big(t - (2n_h - 1)\Delta t\big); \boldsymbol{\theta}\Big), \tag{14}$$

$$r_4 = 2\ \Delta t f\Big(\boldsymbol{X}^*(t) + r_3, \boldsymbol{X}^*(t), \boldsymbol{X}^*(t - 2\Delta t), \ldots, \boldsymbol{X}^*\big(t - (2n_h - 2)\Delta t\big); \boldsymbol{\theta}\Big). \tag{15}$$

This means that using the points $\{\boldsymbol{X}(t - 2n_h\Delta t), \boldsymbol{X}(t - (2n_h - 1)\Delta t), \ldots, \boldsymbol{X}(t - \Delta t), \boldsymbol{X}(t)\}$ from the available dataset, we can predict the point $\boldsymbol{X}^*(t + 2\ \Delta t)$ and fit the model by matching it with the data-point $\boldsymbol{X}(t + 2\ \Delta t)$. We illustrate the computational stencils associated with the time-stepping schemes (6)–(10) and (11)–(15) in Figs. 1(a) and 1(b), respectively.

The time-stepping scheme detailed in (11)–(15) provides a differentiable time-solver with respect to the unknown parameters $\boldsymbol{\theta}$. Note that in some works, an explicit second-order Runge–Kutta (RK2) scheme was considered to solve the parameterized model in order to represent the temporal discretization of the equations representing the resolved dynamics in an atmospheric forecasting model.[29] The numerical property detailed above for RK4 that does not require any time interpolation still applies for explicit RK2. We do not detail the corresponding algebra for the sake of clarity and conciseness, as it is similar to the derivation of the RK4 case.

As a summary, fitting discrete observations of the resolved slow-varying variables that are available with a time step Δt by discretizing the DDE (4) with an explicit RK4 can be carried out without interpolating data by choosing the time-lags as multiples of $2\ \Delta t$: $\tau_i = 2i\Delta t$, $i = 1, \ldots, n_h$ and discretizing the DDE (4) with the time step $2\ \Delta t$.

D. Machine learning formulation

A system identification task can be formulated as follows: Given some observations $\mathcal{D} = \{X(t_i), t_i = i\Delta t, i = 0, \ldots, n - 1\}$, we can learn the vector of the model parameters $\boldsymbol{\theta}$ that best parameterizes the underlying dynamics (4). In practice, the observations in $\mathcal{D}$ are different states from a single trajectory that is obtained by integrating the "true" model (1) and (2). In order to learn the model parameters $\boldsymbol{\theta}$, we define a loss function $\mathcal{L}$ that measures the discrepancy between the observed data and the parameterized model predictions: $\mathcal{L}\Big(\boldsymbol{X}(t_{j+n_f+1}), \boldsymbol{X}^*(t_j + (n_f + 1)\Delta t; \boldsymbol{\theta})\Big)$. $\boldsymbol{X}^*(t_j + (n_f + 1)\Delta t; \boldsymbol{\theta})$ is the parameterized prediction obtained for the data-point $\boldsymbol{X}(t_{j+n_f+1})$ by integrating (4) n_f times and starting from the data-point $\boldsymbol{X}(t_j)$, $j = 0, \ldots, n - n_f - 2$. In this work, we consider the L_2 norm of the error in the loss formulation (mean squared error). n_f is a hyperparameter that is tuned such that the model's accuracy is optimized for its integration over different time scales by changing n_f. In practice, the parameters $\boldsymbol{\theta}$ are learned by minimizing the loss function $\mathcal{L}$ evaluated on a batch of data points of size n_b from the training dataset. We note that the loss function can be evaluated for different n_f values simultaneously. In this context, using decaying weights as inference is conducted further in time can help remedy the issue of systematically predicting the mean state for too large n_f.[55]

Numerical integration (e.g., doubling the time step) of the parameterized model may be a source of additional numerical error. However, as we will show in the numerical results, Sec. III, the machine learning surrogate model P used in Eq. (4) is actually not only capable of learning the coupling (closure) term but also of correcting the numerical errors introduced by the discretization of the parameterized model compared to the accuracy of the data on which the model is trained. Indeed, the training data considered are generated by integrating the "true" model (1) and (2) with a fine time step equal to $\Delta t/2$ and then removing half of the points to retain trajectories with a time step equal to Δt. On one hand, we show that the parameterized model (4) discretized with a time step equal to $2\ \Delta t$ and trained with such a dataset can generate forecasts that closely match the "true" model trajectories. On the other hand, using the RK4 to solve the "true" model (1) and (2) with a time step equal to $2\ \Delta t$ is not even computationally stable. Hence, the parameterized model is still numerically stable even when it is discretized with a larger time step (time step equal to $2\ \Delta t$) for training and forecasting. This inference approach can also be applied for standard instantaneous parameterizations described in Sec. II B by discretizing the parameterized dynamical system (3) with a time step Δt.

III. RESULTS

The instantaneous and memory-based parameterizations detailed in Secs. II B and II C, respectively, are applied to the Lorenz '96 model (1) and (2) in the chaotic regime. Fully connected neural networks are used as machine learning surrogate models for the parameterizations. A Wilks parameterization,[40] consisting of a fourth-order polynomial closure and a first-order autoregressive stochastic process, is also considered as a reference for another instantaneous parametric parameterization. Although the proposed memory-based parameterization is only formulated

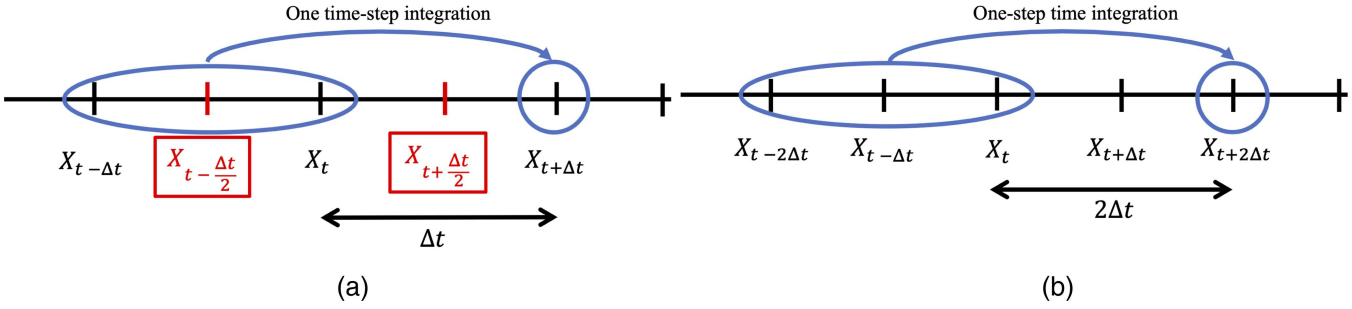

FIG. 1. Computational stencils for explicit RK4 applied to a DDE of the form (5): n_h is taken equal to 1 for clarity. (a) RK4 with a time step Δt and time-lags as multiples of Δt as detailed in Eqs. (6)–(10): red points indicate the needed points for time-stepping but which are not available among the dataset. (b) RK4 with a time step 2 Δt and time-lags as multiples of 2 Δt as detailed in Eqs. (11)–(15).

within a deterministic framework, it is still informative to investigate its performance against this other scheme. All codes and data presented in this section are made publicly available at https://github.com/bhouri0412/Hist_Bayesian_Closure.

A. Problem setup

Similarly to other studies,[29] we choose as the number of slow-varying variables $K = 8$ and as the number of fast-varying variables per low-frequency variable $J = 32$, which gives a Lorenz '96 system of total dimension equal to 264. The coupling constant is set to $h = 1$, the spatial scale ratio to $b = 10$, and the temporal scale ratio to $c = 10$ similarly to previous works on Lorenz '96 model.[6,29,40,46] The forcing term F is taken large enough to ensure chaotic behavior for the resolved variables by taking it equal to the values of 15 and 18. Such a parameter setting results in a model time unit (MTU) that is approximately equivalent to five atmospheric (ATM) days.[6,29,46] The online parameterization schemes, detailed in Secs. II B and II C, are built using only data of the $K = 8$ resolved slow-varying variables for the closure and resulting in a dynamical system that only depends on these resolved variables.

The "true" model (1) and (2) is integrated using the RK4 scheme with a time step equal to $dt = 0.005$ from $t = 0$ to $t = 100$ MTU (500 ATM days). In order to account for potential coarse temporal resolution, for instance, such as when using satellite data, we assume that we only have access to data at each time step equal to $\Delta t = 2dt = 0.01$. This means that the memory-based parameterized model (4) is integrated with a time step equal to $2\Delta t = 0.02$. The resulting training dataset has a total of 9995 points for the memory-based parameterization and 9999 points for the instantaneous parameterizations. We will show that even though the parameterized model's time step is 4 times the time step used to integrate the "true" model, the proposed parameterization is capable of returning time forecasts that are faithful and sufficiently accurate compared to the "true" model's trajectories in online testing.

B. Model initialization and hyperparameter tuning

The minimization of the loss $\mathcal{L}$ is carried out using the Adam optimization.[56] The proposed memory-based parameterization is evaluated against two instantaneous parameterizations, where the coupling term is modeled as a neural network similarly to the memory-based scheme and as a fourth-order polynomial term and a first-order auto-regressive stochastic process as proposed by the Wilks parameterization.[40] The only numerical difference in the problem setting between the instantaneous parameterizations and the memory-based closure consists in using the time step $\Delta t = 0.01$ for the former, and $2\Delta t = 0.02$ for the latter, as justified by the numerical integration scheme (see Sec. II C).

Neural network-based parameterizations are modeled by considering fully connected neural networks. For the examples considered in this study, we did not face an issue of vanishing gradients, which may justify using Residual Neural Networks (RNNs)[57] or Long Short-Term Memory (LSTM) networks,[58] for instance, if encountered in other problems settings. For both parameterizations, the estimate of the model parameter $\boldsymbol{\theta}$ is obtained using 15×10^3 stochastic gradient descent iterations with $n_f = 1$, then starting from the inferred parameter $\boldsymbol{\theta}$, 30×10^3 stochastic gradient descent iterations are conducted with $n_f > 1$ as this allows improving the model accuracy for long-time integration in the online setting.

We note that the loss for the instantaneous parameterization is slightly different than the memory-based one's as the time integration is conducted with a time step of Δt instead of $2\Delta t$. We conducted a (non-exhaustive) hyperparameters' search for the training, and the corresponding results are detailed in Table I.

C. Forecasts results

The accuracy of the parameterized models forecasts will be measured by evaluating the model root mean square errors (RMSE). The lower the RMSE, the more accurate the forecast. The RMSE for a trajectory defined at the time instances $t_i, i = 1, \ldots, N$ is given by

$$\mathrm{RMSE}(t_i) = \sqrt{\sum_{j=1}^{i} ||\boldsymbol{X}(t_j) - \boldsymbol{X}^*(t_j)||_2^2} \Big/ \sqrt{\sum_{j=1}^{N} ||\boldsymbol{X}(j_i)||_2^2}, \quad (16)$$

where $\boldsymbol{X}(t_i), i = 1, \ldots, N$ are the "true" model points and $\boldsymbol{X}^*(t_i)$, $i = 1, \ldots, N$ are the parameterized model points obtained by integrating the parameterized model starting from the initial condition (i.e., online forecasting task). Note that the metrics defined in (16) does account for memory since for a time instance t_i, it does not

TABLE I. Hyperparameters search for deterministic training of memory-based and non-memory-based parameterizations.

Hyperparameter	Range	Best for memory-based parameterization	Best for non-memory-based parameterization
Architecture (number of layers x number of nodes per layer)	$\{2 \times 16, 2 \times 32, 6 \times 64, 6 \times 128, 12 \times 128, 8 \times 256, 6 \times 512\}$	6×64	6×64
Learning rate	$\{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}\}$	10^{-4}	10^{-4}
Batch size	{128, 256, 512, 1024, 2048, 4096}	512	512
n_f	$\{2, 3, \ldots, 9, 10\}$	5	4
n_h	{1, 2, 3, 4, 5, 10}	2	N/A

measure the instantaneous RMSE but rather the accumulation of error between the "true" model points and the predictions throughout the time integration starting from the initial condition up to time t_i. Hence, the metrics considered here (16) is a non-decreasing function of the evaluation time.

Given the chaotic nature of the Lorenz '96 model, the accuracy of the parameterized models forecasts is also measured by evaluating a statistical metric taken as the correlation coefficient between the "true" model points and the parameterized model predictions obtained with online forecasting task. The correlation coefficient is constrained between 0 and 1. The higher the correlation coefficient, the more accurate the forecast. Similarly to the RMSE (16), the correlation coefficient between the "true" model points and the parameterized model predictions at a time instance t_i is computed by considering all points from the initial condition up to the time instance t_i in order to account for the memory effect.

The parameterizations' performance is compared in an online setting (i.e., in which the parameterization is coupled to the resolved variables equations and integrated forward in time) by solving the parameterized differential equations with the inferred closure parameters. The parameterized models are integrated in time until $t = 20$ MTU $= 100$ ATM days starting from the first and last point of the training dataset.

The first two columns of Table II summarize the final correlation coefficients for the resolved variables when integrating dynamical systems starting from the first and last training points for forcing $F = 15$ and 18, respectively. Figures 2(a) and 2(b) show the temporal evolution of the resolved variables' RMSE starting from the first training points for forcing $F = 15$ and 18, respectively. All these results correspond to the errors obtained after temporal integration of the parameterized dynamical systems. When integrating the dynamical systems starting from the first training point, the instantaneous parameterization schemes are capable of being as accurate as the memory-based one. Indeed, for $F = 15$, the instantaneous NN-based parameterization shows similar performance to the memory-based scheme while the Wilks parameterization clearly underperforms the other parameterizations, and for $F = 18$, the Wilks parameterization shows similar performance to the memory-based scheme while the instantaneous NN-based parameterization clearly underperforms. For a forcing $F = 15$, the Wilks parameterization was trained since the corresponding values are not available for such a forcing value.[40] However, for a forcing $F = 18$, we used the Wilks parameterization results detailed in Ref. 40. Note that the Wilks scheme in Ref. 40 was trained on trajectories with a time step equal to $\Delta t_w = 0.005$, while the proposed memory-based parameterization is trained with data with a larger time step equal to $\Delta t = 0.01$, which may explain the competitive results observed for the Wilks scheme in this test.

On the other hand, the last two columns of Table II show the final correlation coefficients for the resolved variables in the temporal extrapolation setting where parameterized models are integrated in time starting from the last training point. Figures 3(a) and 3(b) show the temporal evolution of the resolved variables' RMSE starting from the last training points for forcing $F = 15$ and 18, respectively. For this test, the memory-based parameterization clearly outperforms both instantaneous schemes (the instantaneous NN-based parameterization and Wilks scheme) for both values of forcing considered. Notably, in the temporal extrapolation setting and for $F = 18$, the final correlation coefficient for the memory-based parameterization is twice and

TABLE II. Parameterizations' performance in an online task: final correlation coefficient for the resolved variables $\boldsymbol{X}$ with $F = 15$ and $F = 18$. Best parameterization performances for each task are highlighted with bold for clarity.

Parameterization type	Corr. coeff. start. from first tr. pt. for $F = 15$	Corr. coeff. start. from last tr. pt. for $F = 15$	Corr. coeff. start. from first tr. pt. for $F = 18$	Corr. coeff. start. from last tr. pt. for $F = 18$
Memory-based	**0.883**	**0.825**	**0.488**	**0.675**
Instantaneous NN-based	**0.923**	0.457	0.432	0.316
Wilks	0.373	0.652	**0.487**	0.189

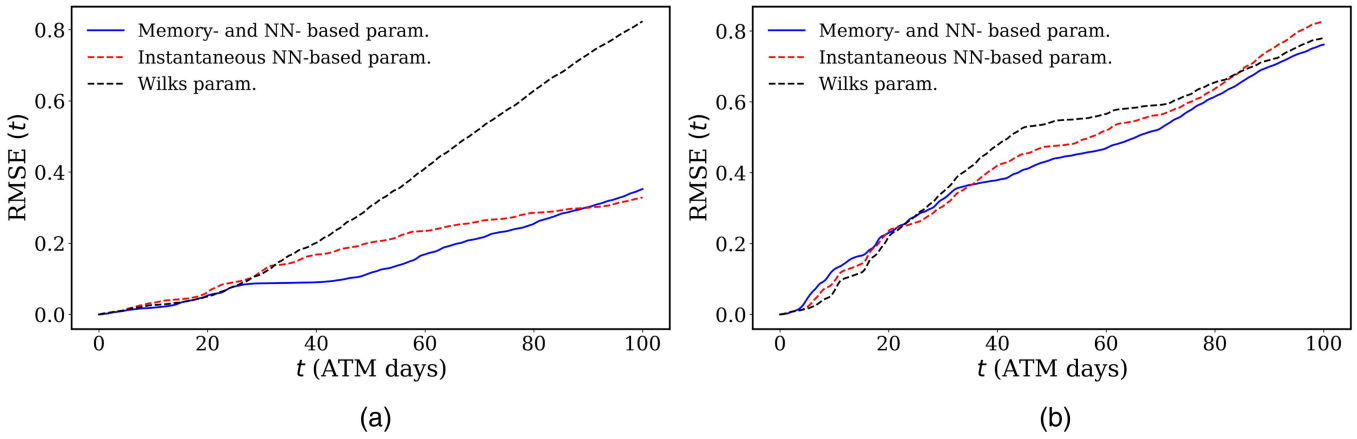

FIG. 2. Temporal evolution of the resolved variables' RMSE in an online task starting from the first training point for different parameterization schemes. (a) $F = 15$ and (b) $F = 18$.

3.6 times the final correlation coefficient obtained for the instantaneous NN and Wilks schemes, respectively. Comparing the results obtained when integrating the parameterized models from the first and last training points forward suggests that the instantaneous parameterization schemes tend to overfit the training data, while the memory-based scheme is able to better distill the temporal dependency of the parameterization term on the previous states of the resolved variables and, hence, returns more accurate predictions for future states. This claim will also be justified in Sec. III E where the parameterized models are evaluated online (i.e., coupled to the coarse-scale dynamics) starting from random and unseen initial conditions.

Figures 4 and 5 show the online predictions for the resolved slow-varying variables and the closure terms, respectively, using the memory-based parameterization and starting from the last training point. The good agreement between the true trajectory and the parameterized model prediction confirms the ability of the proposed framework to learn a parameterization that is stable and also sufficiently accurate when tested online. It also shows that the model is capable of implicitly inferring the closure term correctly, without using any data or information on closure or on the unresolved fast-varying variables, but by only being trained on resolved variables trajectories with coarse temporal resolution.

As expected, the actual closure (based on the fast variables) displays higher frequencies than the inferred trajectories from the parameterization as shown in Fig. 5. This is due to the fact that the actual closure terms have a higher frequency than the resolved slow-varying variables since they depend on the fast-varying

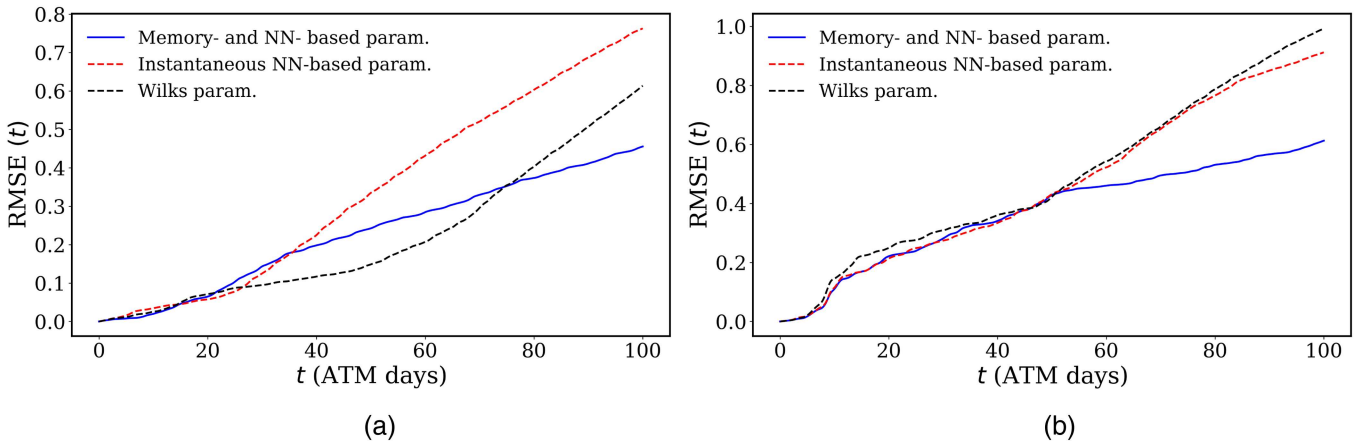

FIG. 3. Temporal evolution of the resolved variables' RMSE in an online task starting from the last training point for different parameterization schemes. (a) $F = 15$ and (b) $F = 18$.

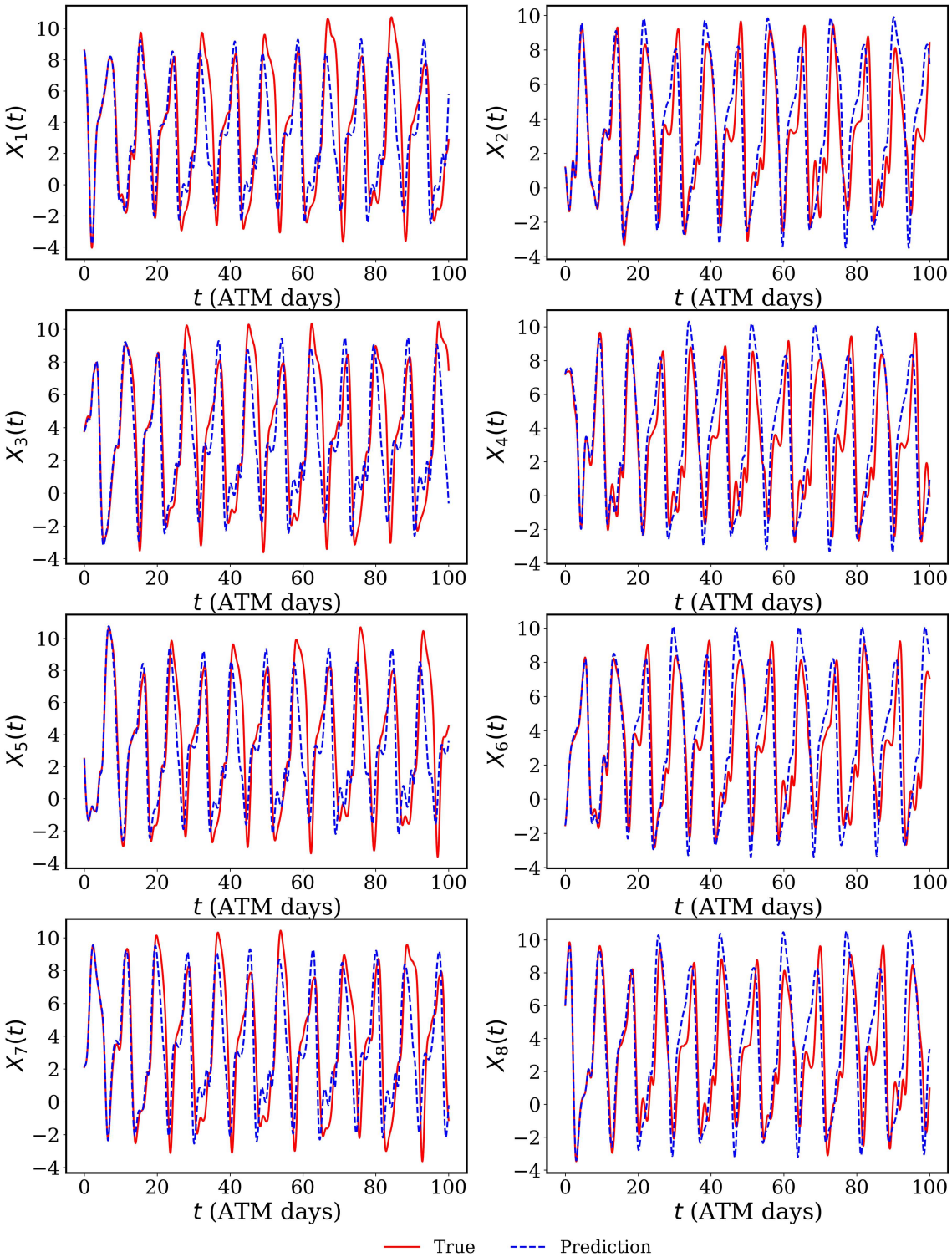

FIG. 4. Online predictions for the resolved variables using the memory-based parameterization starting from the last training point for $F = 15$: Predictions correspond to solutions of the parameterized DDE (4), while true trajectories correspond to solutions of the "true" model (1) and (2).

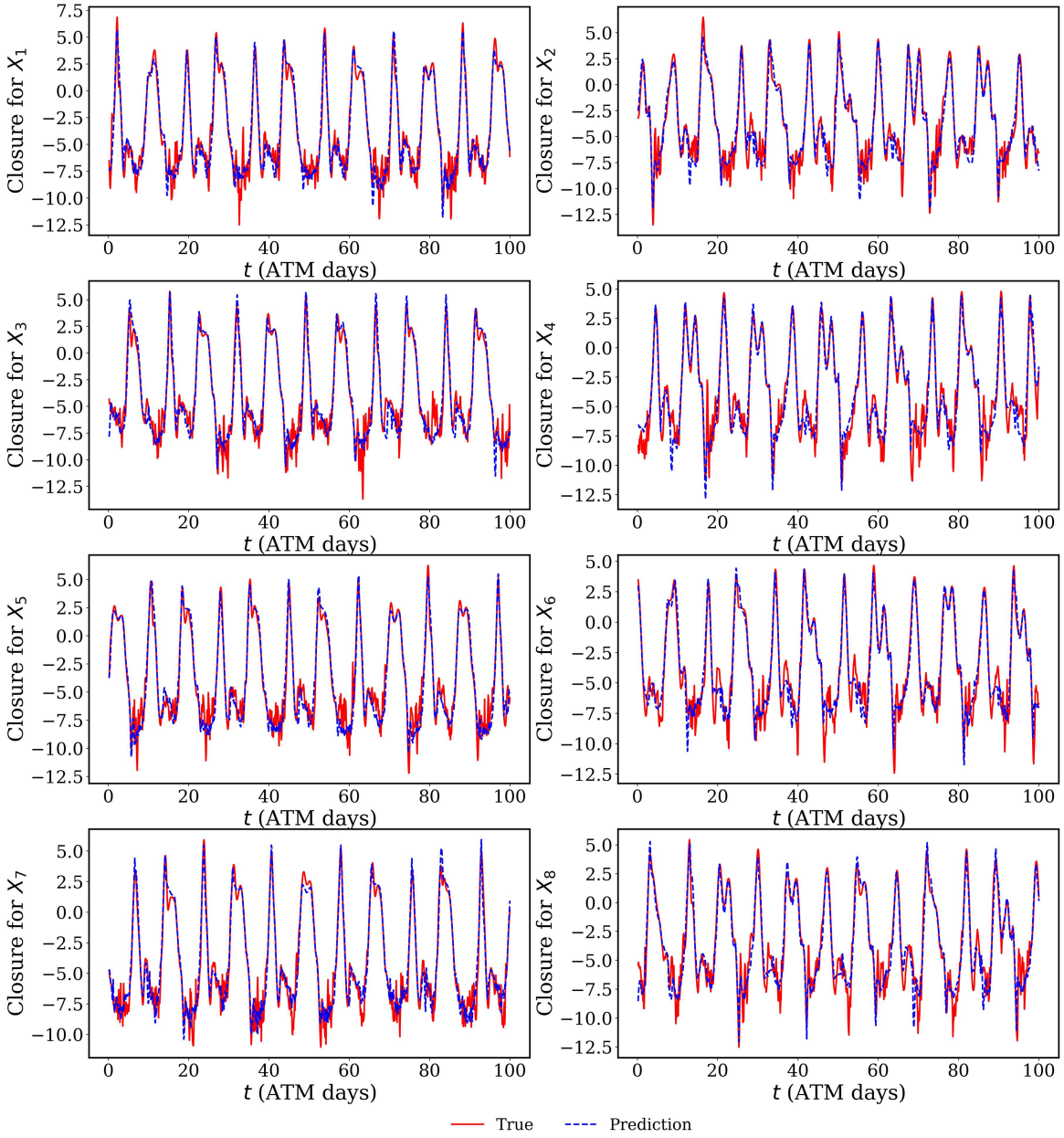

FIG. 5. Online predictions for the closure terms using the memory-based parameterization starting from the last training point for $F = 15$: Predictions correspond to the $P(\cdot)$ terms of the parameterized DDE (4), while true trajectories correspond to the coupling term in (1).

variables. On the other hand, the inferred closure terms show a lower frequency that is closer to the frequency of the slow-varying variables than to the fast-varying variables' one. This result is coherent with the fact that the model is trained to match and predict the slow-varying variables, while no data or information on the closure terms or the fast-varying (high-frequency) variables are available. Although some high frequencies of the closure terms are filtered out by the parameterization given the inherent problem setup considered (based on slow variables), the parameterized model is still able to provide stable and accurate online temporal predictions for the resolved variables.

D. Numerical error

An interesting question that can be asked for any parameterization is: "what is the parameterization actually

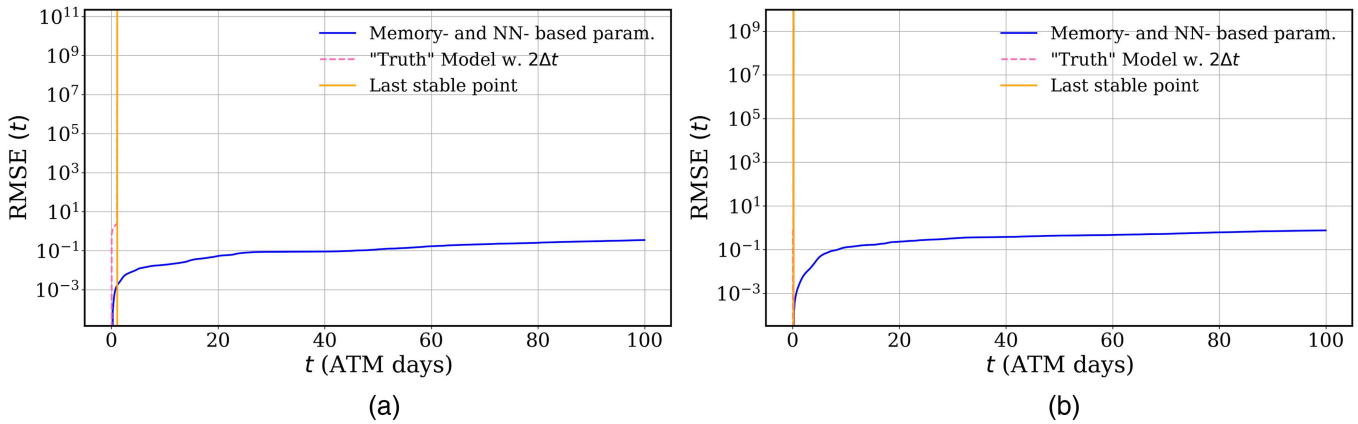

FIG. 6. Temporal evolution of the resolved variables' RMSE starting from the first training point for the memory-based parameterized model (4) and the "true" model (1) and (2), both being solved with a time step equal to $2\Delta t = 0.02$. (a) $F = 15$ and (b) $F = 18$.

TABLE III. Parameterizations performance in the online/coupled task: final RMSE and correlation coefficients for the resolved variables when integrating the parameterized models using 100 random initial conditions. Best parameterization performances for each task are highlighted with bold for clarity.

Parameterization	RMSE for $\boldsymbol{X}$ with $F = 15$	Correlation coefficient for $\boldsymbol{X}$ with $F = 15$	RMSE for $\boldsymbol{X}$ with $F = 18$	Correlation coefficient for $\boldsymbol{X}$ with $F = 18$
Memory-based	**0.385**	**0.841**	**0.819**	**0.384**
Instantaneous NN-based	0.535	0.710	0.898	0.315
Wilks	0.469	0.778	0.956	0.224

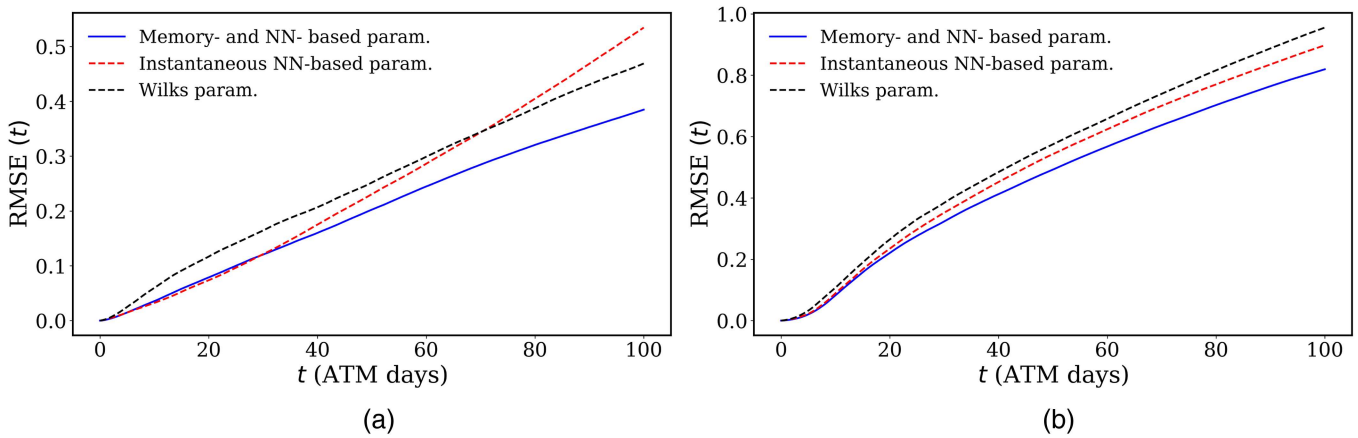

FIG. 7. Temporal evolution of the resolved variables' RMSE in an online task for different parameterization schemes when integrating the parameterized models using 100 random initial conditions. (a) $F = 15$ and (b) $F = 18$.

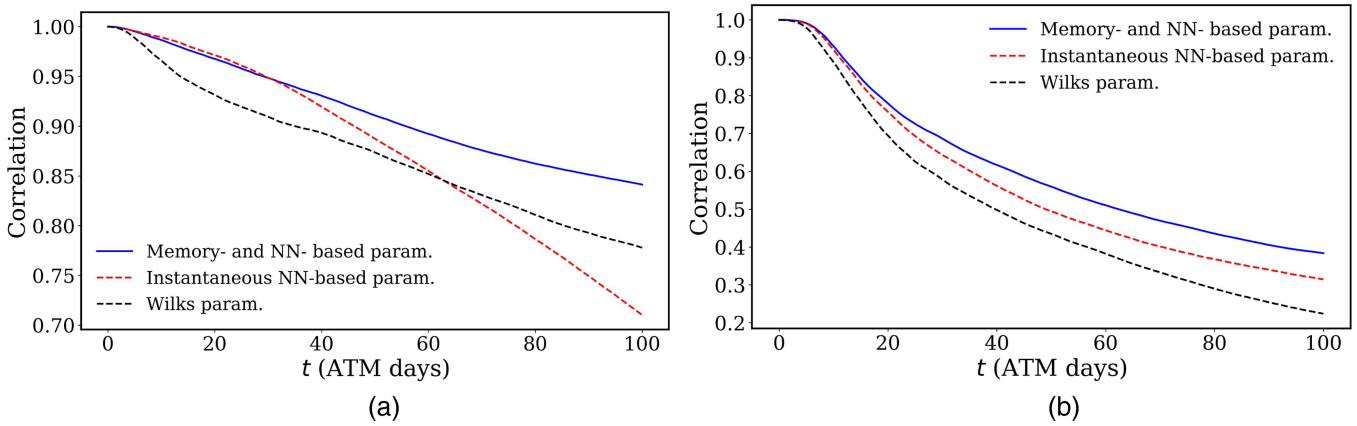

FIG. 8. Temporal evolution of the resolved variables' correlation coefficient in an online task for different parameterization schemes when integrating the parameterized models using 100 random initial conditions. (a) $F = 15$ and (b) $F = 18$.

learning?." Although the parameterization is designed to learn some specific terms (e.g., the coupling term of Lorenz '96 in this work), there is no constraint to guarantee that the learning process is limited to these quantities.

In the problem setup considered in this work, coarse temporal resolution (temporal coarse-graining) is an important source of error that is accounted for and partially resolved by the memory-based parameterization, as shown in the results presented in Sec. III C. Indeed, the "true" model (1) and (2) is integrated using RK4 with a time step equal to $dt = 0.005$, while the memory-based parameterized model is trained on data with a coarser time step equal to $\Delta t = 2dt = 0.01$. This means that the memory-based parameterized model is integrated with a time step equal to $2\Delta t = 0.02$, which is 4 times the time step used to integrate the "true" model.

A simple exercise to verify the impact of the parameterization on correcting the numerical error introduced by the finer temporal resolution considered for the parameterized model is to solve the "true" model with a time step equal to $2\Delta t = 0.02$ and compare its RMSE with the error obtained for the parameterized model. For this task, the solution of the "true" model with a time step equal to $dt = 0.005$ is taken as reference.

Solving the "true" model with a time step equal to $2\Delta t = 0.02$ returns a solution that diverges at $t = 1.1$ATM days and $t = 0.15$ATM days for $F = 15$ and 18, respectively. We show the corresponding RMSEs in Figs. 6(a) and 6(b), which also includes the RMSEs of the memory-based parameterized model (4) solved with the same time step equal to $2\Delta t = 0.02$ with $F = 15$ and 18, respectively. This result means that the memory-based parameterization does not only learn the coupling term but also corrects some of

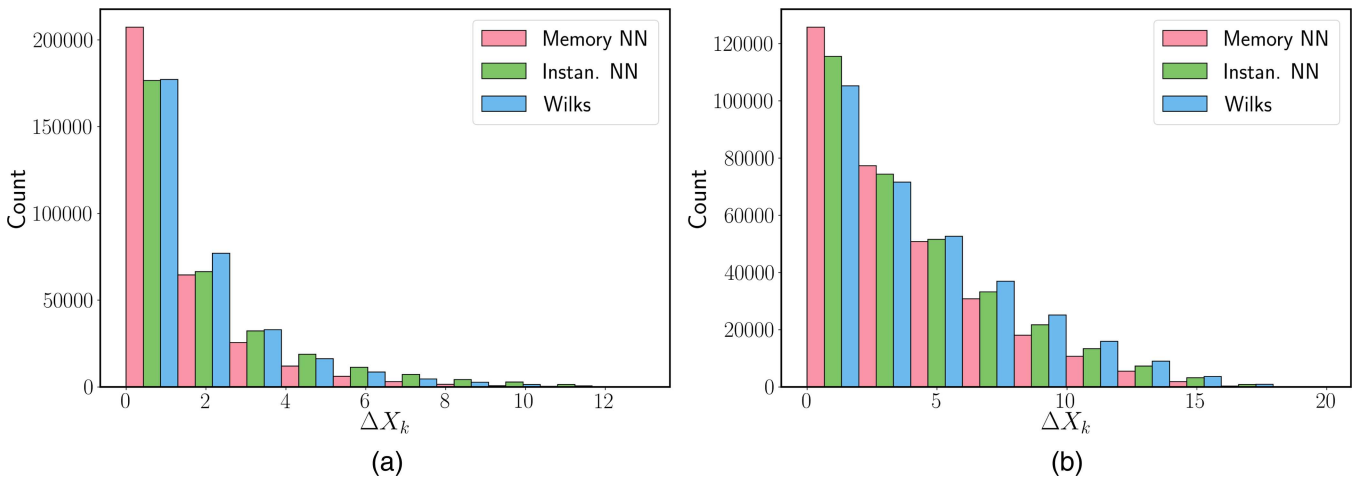

FIG. 9. Histograms of the errors ΔX_k, $k = 1, \ldots, K$ for different parameterization schemes: errors considered correspond to the predictions made in an online task for the 100 random initial conditions. (a) $F = 15$ and (b) $F = 18$.

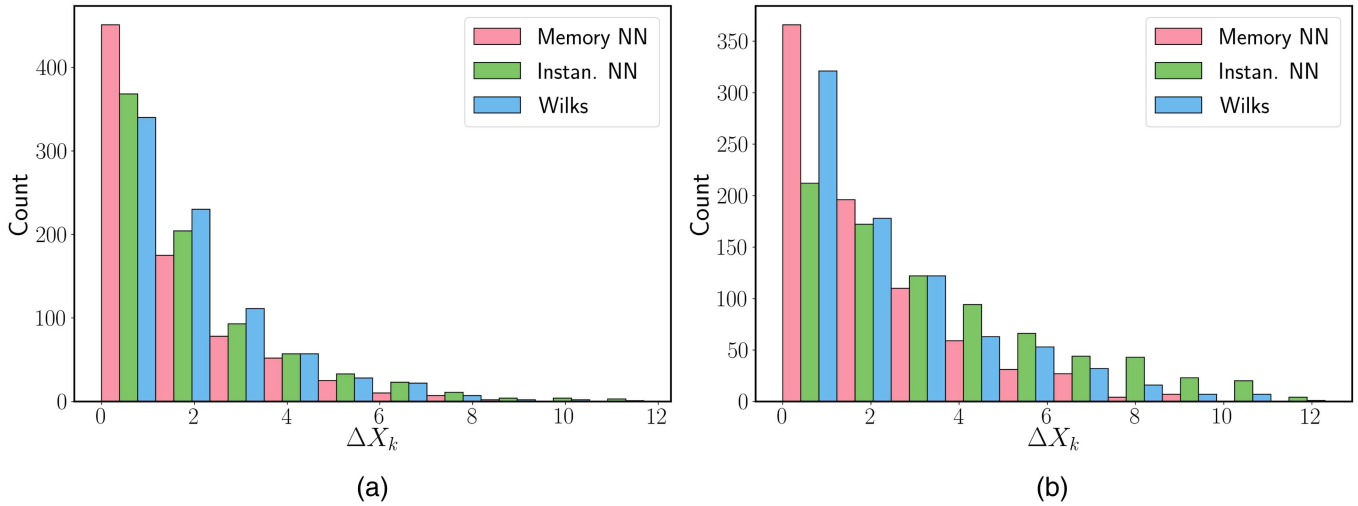

FIG. 10. Histograms of the errors ΔX_k, $k = 1, \ldots, K$ for different parameterization schemes and $F = 15$ at different time horizons: errors considered correspond to the predictions made in an online task for the 100 random initial conditions. (a) 50 ATM days horizon and (b) 100 ATM days horizon.

the numerical integration error introduced due to the assumed data sparsity/coarse time step. Indeed, the data are generated by solving the "true" model with a temporal discretization equal to $dt = 0.005$, while training data observations are only considered with a time step $\Delta t = 2dt = 0.01$, showing that the differentiable solver approach allows accounting for at least part of the numerical discretization error.

E. Generalization to unseen initial conditions

The epistemic (structural) variance of the different parameterization schemes is now evaluated by considering 100 random initial conditions on which the parameterized models were not trained. Similarly to the study detailed in Sec. III C, the parameterization performance is compared in an online setting (i.e., in which the parameterization is coupled to the slow variables equations and integrated forward in time) by solving the parameterized differential equations with the inferred closure parameters. The parameterized models are integrated in time until $t = 20$ MTU $= 100$ ATM days.

This test is considered to assess the generalization performance of the parameterization schemes beyond the data on which they were trained, similarly to the temporal integration of the parameterized models starting from the last training point that was considered in Sec. III C.

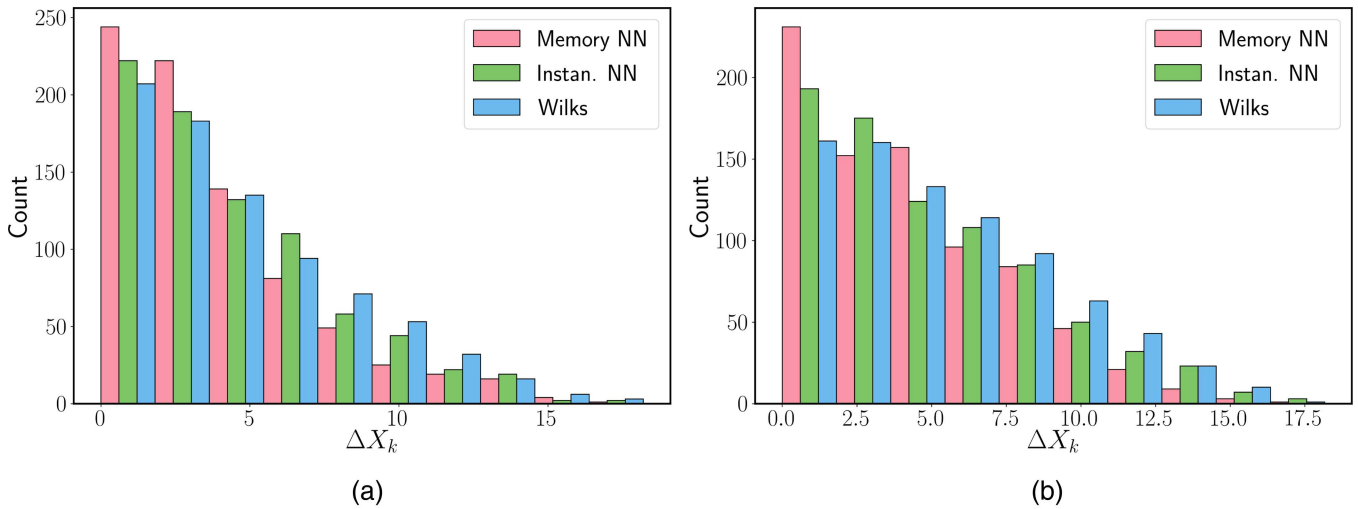

FIG. 11. Histograms of the errors ΔX_k, $k = 1, \ldots, K$ for different parameterization schemes and $F = 18$ at different time horizons: errors considered correspond to the predictions made in an online task for the 100 random initial conditions. (a) 50 ATM days horizon and (b) 100 ATM days horizon.

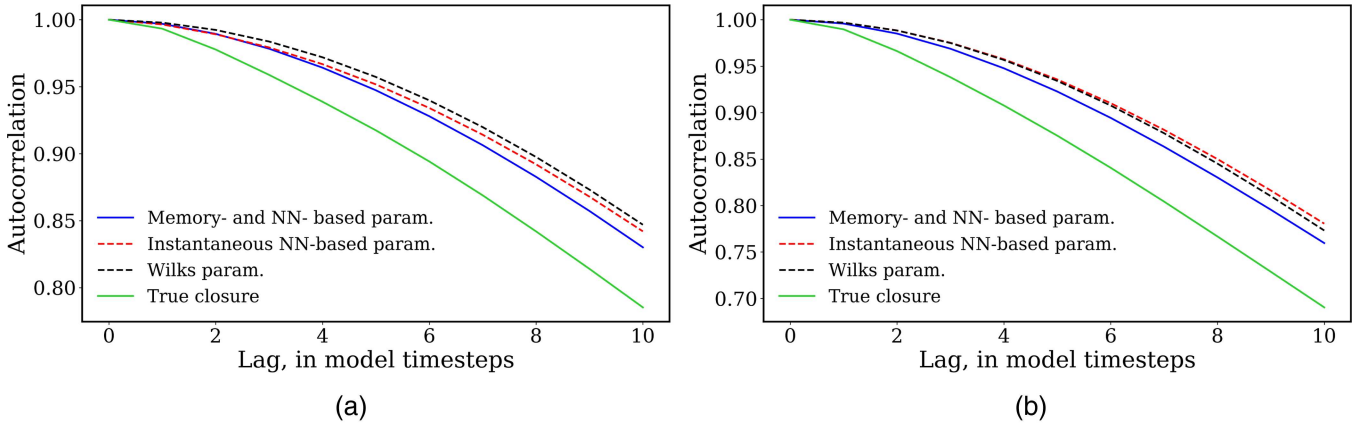

FIG. 12. Temporal autocorrelation function of the inferred closure term for different parameterization schemes and forcing values when integrating the parameterized models using 100 random initial conditions. (a) $F = 15$ and (b) $F = 18$.

Table III gathers the final model performance consisting of RMSEs and correlation coefficients for the resolved slow-varying variables and forcing $F = 15$ and 18 when integrating the parameterized models using the 100 random initial conditions. Figures 7 and 8(b) show the temporal evolution of the resolved variables' RMSE and correlation coefficient when integrating the parameterized models using the 100 random initial conditions for forcing $F = 15$ and 18. For any forcing value, the memory-based parameterization clearly outperforms both instantaneous schemes by returning predictions with lower RMSE and higher correlation coefficient over the entire time integration interval. These results for the random and unseen initial conditions confirm the improved generalization capabilities of the parameterized model when considering the memory-based scheme as suggested in Sec. III C.

Given the number of predictions ($\sim 2 \times 10^5$) performed with the 100 random initial conditions and temporal integration until $t = 20$ MTU $= 100$ ATM days, we can infer a statistical characterization of the error for the inferred resolved variables (i.e., uncertainty quantification) by approximation the probability of the discrepancy $\Delta X_k = \sqrt{(\boldsymbol{X}_k - \boldsymbol{X}_k^*)^2}$, $k = 1, \ldots, K$ evaluated at the inference time instances t_i, $i = 1, \ldots, N$ given the initial condition $\boldsymbol{X}(0)$. These errors correspond to the resolved variables discrepancies obtained with the online task of solving the parameterized dynamical systems for the 100 random initial conditions. Since consecutive predictions can be strongly correlated, we uniformly sub-sample the temporal errors with a factor of 5 in order to remedy the data "redundancy," resulting in an effective sample size of $\sim 4 \times 10^4$. Besides, since the errors for the different variables X_k, $k = 1, \ldots, K$ vary in a very similar range and for the sake of clarity, we decided to group all errors for different variables together. In the Appendix, we provide separate histograms for the error of each variable X_k, $i = k, \ldots, 8$.

Figures 9(a) and 9(b) show the histograms of the errors ΔX_k, $k = 1, \ldots, K$ for different parameterization schemes and $F = 15$ and 18, respectively. For both forcing values, the memory-based parameterization's bins are higher for the lower error values compared to both instantaneous schemes (Wilks and NN-based parameterization) and lower for the larger error values, confirming the improved online future states predictions when accounting for the memory effect on the closure term.

In addition to the probability distribution of errors at all time instances, it is interesting to investigate the evolution of the error distribution at given time instances for different horizons. Figures 10(a) and 10(b) show the histograms of the errors ΔX_k, $k = 1, \ldots, K$ for different parameterization schemes and $F = 15$ at the time instances $t = 50$ ATM days and $t = 100$ ATM days, respectively. Similarly, Figs. 11(a) and 11(b) show the errors histograms for $F = 18$. These results show that the difference between the error distributions of different parameterized models tend to diminish for longer time horizons. However, even up to $t = 100$ ATM days, the memory-based parameterization still shows an error distribution that is clearly more centered around 0 compared to the instantaneous parameterization schemes, confirming again the improved online future states predictions, when accounting for the memory effect on the closure term, at different time horizons.

As an additional test, we consider a Generative Adversarial Networks (GAN) parameterization of Lorenz '96,[29] since it is also a machine learning approach and it was tested for the same set of Lorenz '96 parameters. It is worth mentioning that such a GAN parameterization scheme[29] requires training data on the resolved variables and on the closure terms, while the proposed online memory-based parameterization does not use any information on the closure terms, which may favor the GAN parameterization approach.[29] We consider the non-normalized RMSE defined in Eq. (10) of Ref. 29 and evaluated at $t = 2$ MTU which is the final time considered in Fig. 5 of Ref. 29. We evaluate the RMSE using the 100 random and unseen initial conditions. From Fig. 5 of Ref. 29, the GAN-based parameterizations and the polynomial one (based on a third-order polynomial and a first-order auto-regressive stochastic process) return predictions with RMSE between roughly

3.8 and 5.5. The Wilks parameterization implemented in this study based on a fourth-order polynomial and a first-order auto-regressive stochastic process) gives predictions with a RMSE of 4.60, while the memory-based parameterization predictions have a RMSE of 2.86 at $t = 2$ MTU when tested on the 100 random unseen initial conditions. These results confirm the improved performance when accounting for the memory effect on the closure term even when comparing with the GAN-based parameterization.[29]

Finally, we consider a memory metric in order to quantify the inference accuracy of the closure term.[59] In particular, Figs. 12(a) and 12(b) show the temporal autocorrelation of the inferred closure term as a function of model time steps lag for different parameterization schemes and forcing values when integrating the parameterized models using 100 random initial conditions with $F = 15$ and 18, respectively. In coherence with the results observed so far regarding the accuracy of the online inference of the resolved variables, the memory-based parameterization shows the best results in terms of the inference accuracy of the closure term among all schemes considered. Indeed, its corresponding temporal autocorrelation function of the inferred closure term is the closest to the one of the "true" closure for any model time steps lag considered. This additional metric justifies again that the memory-based scheme is able to better infer the temporal evolution of the parameterization term by learning the effect of the resolved variables' previous states on the closure term evolution.

IV. CONCLUSION

In this study, we proposed a memory-based parameterization scheme for dynamical systems, which is evaluated online (i.e., coupled to the resolved variables' dynamical system). Using a differentiable ODE solver for training allows inferring a parameterized dynamical system that is numerically stable since the closure parameters are inferred while solving the corresponding dynamical system. It also allows learning the parameterization implicitly without requiring data of the closure terms. Since many physical processes have substantial memory, excluding memory from the parameterization can lead to further uncertainties for instantaneous models relying only the current time state. The time-lags defining the memory terms of the proposed parameterization and the time step used to solve the parameterized model are chosen in order to avoid data interpolation during temporal integration of the differentiable parameterized model.

We tested the proposed memory-based parameterization on the chaotic Lorenz '96 system with a coarse temporal resolution. We considered different error metrics that take into account memory effect for error accumulation and exponential growth for chaotic systems. The results proved that relying on a memory-based parameterization allows a better inference of the closure and resolved variables' future states across time scales compared to instantaneous schemes (whether with polynomial representation and a first-order auto-regressive stochastic process or neural network). We also showed that the proposed parameterization is capable of not only learning the coupling terms but also of accounting and correcting for the numerical error introduced by the temporal discretization, which enhances stability and accuracy of the parameterized system's resolution. Finally, the generalization capability of the proposed memory-based parameterization scheme was tested by considering random and unseen initial conditions. This test's results showed its capability to produce more accurate temporal forecasts for the resolved variables compared to instantaneous schemes, confirming the improved generalization performance when accounting for the memory effect on the closure term.

The proposed memory-based parameterization could be improved by finding a more rigorous approach to find the optimal number of previous resolved variables' states considered as inputs to the closure. In addition, other neural network architectures or different machine learning surrogate models could be investigated with the proposed memory-based parameterization, such as long short-term memory neural networks. Such a modification may be critical if the proposed parameterization scheme is tested on more complex problems, while it will be more challenging to implement in large-scale numerical codes often based on Fortran or C, such as for ocean turbulence, atmospheric convection, and clouds formation. Finally, the proposed memory-based parameterization can be extended into a stochastic formulation by relying on a Bayesian formalism, such as using Markov Chain sampling in order to accurately quantify different uncertainty sources. We leave this for future work.

ACKNOWLEDGMENTS

M.A.B. and P.G. would like to thank funding from the Department of Energy DE-SC0022323, National Science Foundation Science and Technology Center Award No. 2019625 STC: Center for Learning the Earth with Artificial Intelligence and Physics (LEAP) and NSF AGS-PRF Fellowship Award AGS-2218197.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Mohamed Aziz Bhouri: Conceptualization (lead); Data curation (lead); Formal analysis (equal); Methodology (lead); Visualization (lead); Writing – original draft (lead); Writing – review & editing (equal). **Pierre Gentine:** Conceptualization (supporting); Formal analysis (equal); Funding acquisition (lead); Methodology (supporting); Supervision (lead); Writing – review & editing (equal).

DATA AVAILABILITY

The data that support the findings of this study are openly available in Github at https://github.com/bhouri0412/Hist_Bayesian_Closure, Ref. 60.

APPENDIX A: ERRORS PROBABILITY DISTRIBUTION FUNCTION

Figures 13 and 14 show the histograms of the error for each variable ΔX_k, $k = 1, \ldots, K$ for different parameterization schemes and $F = 15$ and 18, respectively.

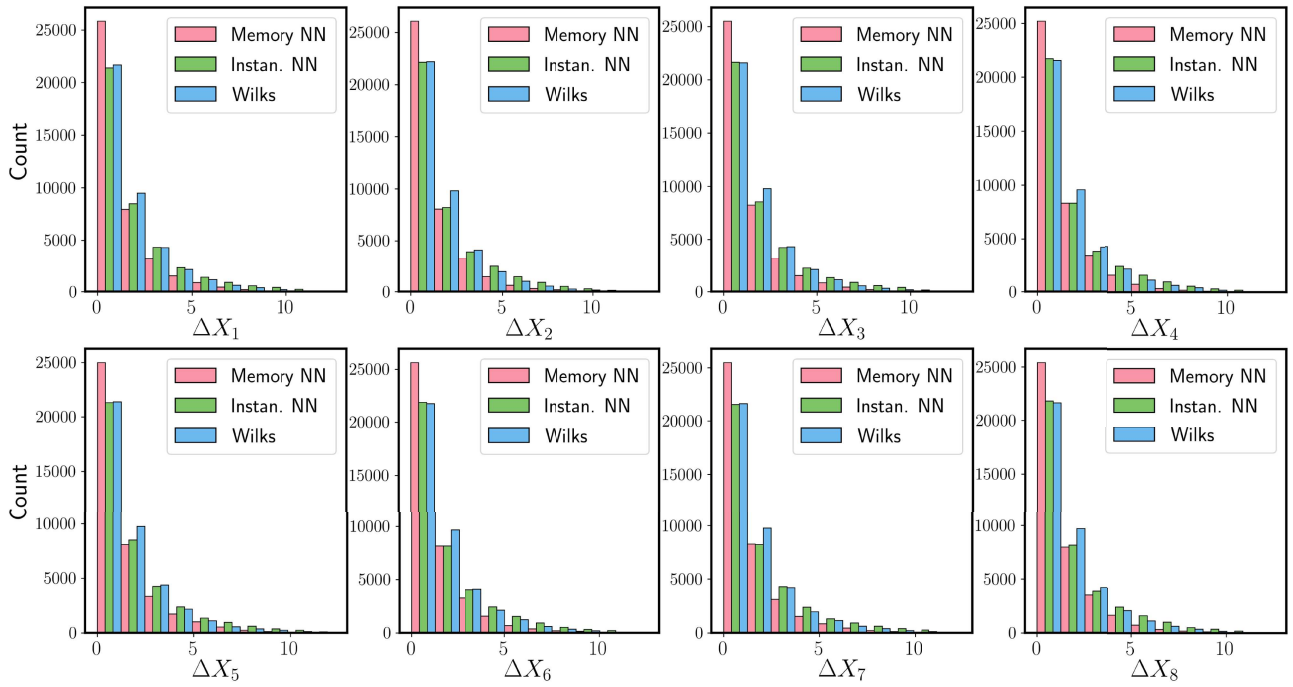

FIG. 13. Histograms of the errors $\Delta X_k, k = 1, \ldots, K$ for different parameterization schemes and $F = 15$: errors considered correspond to the predictions made in an online task for the 100 random initial conditions.

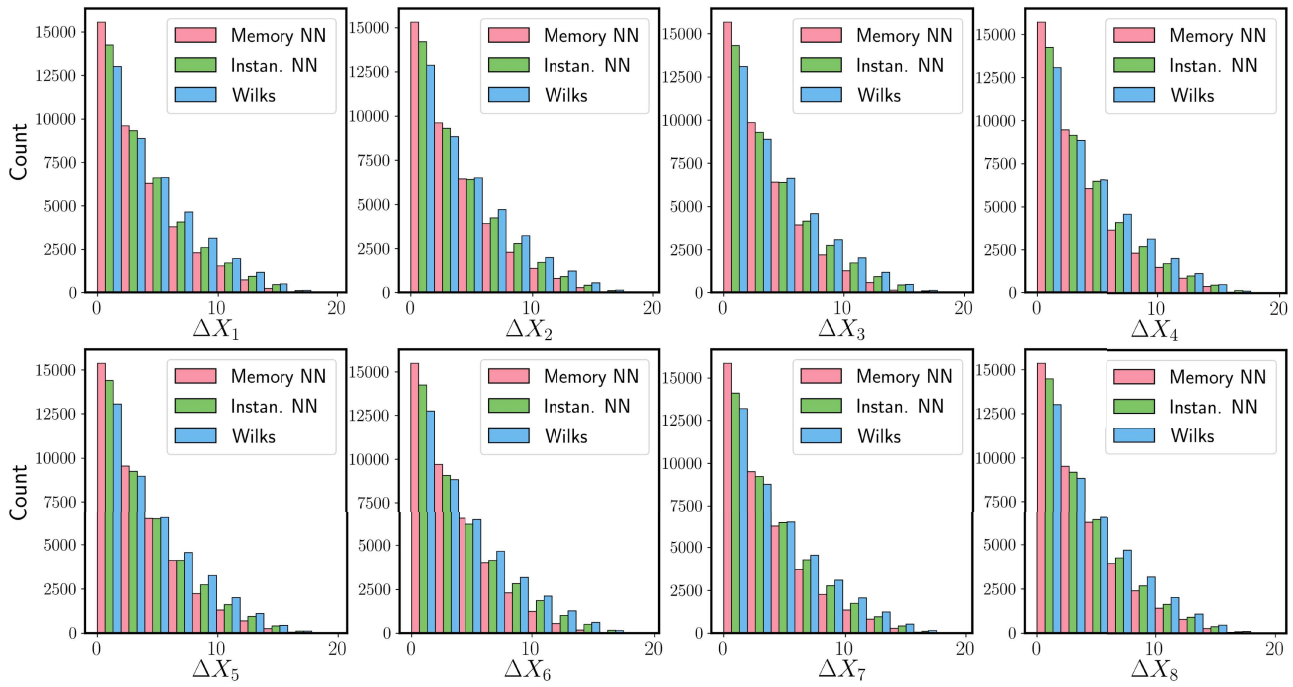

FIG. 14. Histograms of the errors $\Delta X_k, k = 1, \ldots, K$ for different parameterization schemes and $F = 18$: errors considered correspond to the predictions made in an online task for the 100 random initial conditions.

REFERENCES

[1]N. D. Brenowitz and C. S. Bretherton, "Prognostic validation of a neural network unified physics parameterization," Geophys. Res. Lett. **45**, 6289–6298, https://doi.org/10.1029/2018GL078510 (2018).
[2]S. Rasp, M. S. Pritchard, and P. Gentine, "Deep learning to represent subgrid processes in climate models," Proc. Natl. Acad. Sci. U.S.A. **115**, 9684–9689 (2018).
[3]P. Gentine, M. Pritchard, S. Rasp, G. Reinaudi, and G. Yacalis, "Could machine learning break the convection parameterization deadlock?," Geophys. Res. Lett. **45**, 5742–5751, https://doi.org/10.1029/2018GL078202 (2018).
[4]P. A. O'Gorman and J. G. Dwyer, "Using machine learning to parameterize moist convection: Potential for modeling of climate, climate change, and extreme events," J. Adv. Model. Earth Syst. **10**, 2548–2563 (2018).
[5]T. Bolton and L. Zanna, "Applications of deep learning to ocean data inference and subgrid parameterization," J. Adv. Model. Earth Syst. **11**, 376–399 (2019).
[6]E. Lorenz, "Predictability: A problem partly solved," in *Seminar on Predictability, 4–8 September 1995*, Vol. 1, ECMWF (ECMWF, Shinfield Park, Reading, 1995), pp. 1–18.
[7]T. N. Palmer, "A nonlinear dynamical perspective on model error: A proposal for non-local stochastic-dynamic parametrization in weather and climate prediction models," Q. J. R. Meteorol. Soc. **127**, 279–304 (2001).
[8]J. J. Tribbia and D. P. Baumhefner, "Scale interactions and atmospheric predictability: An updated perspective," Mon. Weather Rev. **132**, 703–713 (2004).
[9]A. Karimi and M. Paul, "Extensive chaos in the Lorenz-96 model," Chaos **20**, 043105 (2010).
[10]R. Buizza, M. Milleer, and T. N. Palmer, "Stochastic representation of model uncertainties in the ECMWF ensemble prediction system," Q. J. R. Meteorol. Soc. **125**, 2887–2908 (1999).
[11]C. Sanchez, K. D. Williams, and M. Collins, "Improved stochastic physics schemes for global weather and climate models," Q. J. R. Meteorol. Soc. **142**, 147–159 (2016).
[12]H. M. Christensen, J. Berner, D. R. B. Coleman, and T. N. Palmer, "Stochastic parameterization and EL Niño-southern oscillation," J. Clim. **30**, 17–38 (2017).
[13]M. Leutbecher, S.-J. Lock, P. Ollinaho, S. T. K. Lang, G. Balsamo, P. Bechtold, M. Bonavita, H. M. Christensen, M. Diamantakis, E. Dutra, S. English, M. Fisher, R. M. Forbes, J. Goddard, T. Haiden, R. J. Hogan, S. Juricke, H. Lawrence, D. MacLeod, L. Magnusson, S. Malardel, S. Massart, I. Sandu, P. K. Smolarkiewicz, A. Subramanian, F. Vitart, N. Wedi, and A. Weisheimer, "Stochastic representations of model uncertainties at ECMWF: State of the art and future vision," Q. J. R. Meteorol. Soc. **143**, 2315–2339 (2017).
[14]G. C. Craig and B. G. Cohen, "Fluctuations in an equilibrium convective ensemble. Part I: Theoretical formulation," J. Atmos. Sci. **63**, 1996–2004 (2006).
[15]B. Khouider, J. Biello, and A. J. Majda, "A stochastic multicloud model for tropical convection," Commun. Math. Sci. **8**, 187–216 (2010).
[16]G. Vissio and V. Lucarini, "A proof of concept for scale-adaptive parametrizations: The case of the Lorenz '96 model," Q. J. R. Meteorol. Soc. **144**, 63–75 (2018).
[17]G. Vissio and V. Lucarini, "Evaluating a stochastic parametrization for a fast–slow system using the Wasserstein distance," Nonlinear Process. Geophys. **25**, 413–427 (2018).
[18]M. Sakradzija and D. Klocke, "Physically constrained stochastic shallow convection in realistic kilometer-scale simulations," J. Adv. Model. Earth Syst. **10**, 2755–2776 (2018).
[19]L. Bengtsson, J.-W. Bao, P. Pegion, C. Penland, S. Michelson, and J. Whitaker, "A model framework for stochastic representation of uncertainties associated with physical processes in NOAA's next generation global prediction system (NGGPS)," Mon. Weather Rev. **147**, 893–911 (2019).
[20]M. Santos Gutiérrez, V. Lucarini, M. D. Chekroun, and M. Ghil, "Reduced-order models for coupled dynamical systems: Data-driven methods and the Koopman operator," Chaos **31**, 053116 (2021).
[21]G. J. Shutts and T. N. Palmer, "Convective forcing fluctuations in a cloud-resolving model: Relevance to the stochastic parameterization problem," J. Clim. **20**, 187–202 (2007).
[22]G. Shutts and A. C. Pallarès, "Assessing parametrization uncertainty associated with horizontal resolution in numerical weather prediction models," Philos. Trans. R. Soc. A: Math., Phys. Eng. Sci. **372**, 20130284 (2014).
[23]J. Dorrestijn, D. T. Crommelin, A. P. Siebesma, H. J. J. Jonker, and C. Jakob, "Stochastic parameterization of convective area fractions with a multicloud model inferred from observational data," J. Atmos. Sci. **72**, 854–869 (2015).
[24]H. M. Christensen, I. M. Moroz, and T. N. Palmer, "Stochastic and perturbed parameter representations of model uncertainty in convection parameterization," J. Atmos. Sci. **72**, 2525–2544 (2015).
[25]J. Bessac, A. Monahan, H. Christensen, and N. Weitzel, "Stochastic parameterization of subgrid-scale velocity enhancement of sea surface fluxes," Mon. Weather Rev. **147**, 1447–1469 (2019).
[26]H. M. Christensen, "Constraining stochastic parametrisation schemes using high-resolution simulations," Q. J. R. Meteorol. Soc. **146**, 938–962 (2020).
[27]V. M. Krasnopolsky, M. S. Fox-Rabinovitz, and D. V. Chalikov, "New approach to calculation of atmospheric model physics: Accurate and fast neural network emulation of longwave radiation in a climate model," Mon. Weather Rev. **133**, 1370–1383 (2005).
[28]T. Schneider, S. Lan, A. Stuart, and J. A. Teixeira, "Earth system modeling 2.0: A blueprint for models that learn from observations and targeted high-resolution simulations," Geophys. Res. Lett. **44**, 12396–12417, https://doi.org/10.1002/2017GL076101 (2017).
[29]D. J. Gagne II, H. M. Christensen, A. C. Subramanian, and A. H. Monahan, "Machine learning for stochastic parameterization: Generative adversarial networks in the Lorenz '96 model," J. Adv. Model. Earth Syst. **12**, e2019MS001896 (2020).
[30]A. P. Guillaumin and L. Zanna, "Stochastic-deep learning parameterization of ocean momentum forcing," J. Adv. Model. Earth Syst. **13**, e2021MS002534 (2021).
[31]J. Yuval and P. A. O'Gorman, "Stable machine-learning parameterization of subgrid processes for climate modeling at a range of resolutions," Nat. Commun. **11**, 3295 (2020).
[32]C. Ma, J. Wang, and E. Weinan, "Model reduction with memory and the machine learning of dynamical systems," Commun. Comput. Phys. **25**, 947–962 (2018).
[33]S. Pan and K. Duraisamy, "Data-driven discovery of closure models," SIAM J. Appl. Dyn. Syst. **17**, 2381–2413 (2018).
[34]S. Pan and K. Duraisamy, "On the structure of time-delay embedding in linear models of non-linear dynamical systems," Chaos **30**, 073135 (2020).
[35]R. Maulik, A. Mohan, B. Lusch, S. Madireddy, P. Balaprakash, and D. Livescu, "Time-series learning of latent-space dynamics for reduced-order model closure," Phys. D: Nonlinear Phenom. **405**, 132368 (2020).
[36]N. Agarwal, D. Kondrashov, P. Dueben, E. Ryzhov, and P. Berloff, "A comparison of data-driven approaches to build low-dimensional ocean models," J. Adv. Model. Earth Syst. **13**, e2021MS002537 (2021).
[37]A. J. Chorin and F. Lu, "Discrete approach to stochastic parametrization and dimension reduction in nonlinear dynamics," Proc. Natl. Acad. Sci. U.S.A. **112**, 9804–9809 (2015).
[38]Y. Yang, M. A. Bhouri, and P. Perdikaris, "Bayesian differential programming for robust systems identification under uncertainty," Proc. R. Soc. A **476**, 20200290 (2020).
[39]M. A. Bhouri and P. Perdikaris, "Gaussian processes meet neuralodes: A Bayesian framework for learning the dynamics of partially observed systems from scarce and noisy data," Proc. R. Soc. A **380**(2229), 20210201 (2022).
[40]D. S. Wilks, "Effects of stochastic parametrizations in the Lorenz '96 system," Q. J. R. Meteorol. Soc. **131**, 389–407 (2005).
[41]D. Crommelin and E. Vanden-Eijnden, "Subgrid-scale parameterization with conditional Markov chains," J. Atmos. Sci. **65**, 2661–2675 (2008).
[42]J. Dorrestijn, D. T. Crommelin, J. A. Biello, and S. J. Böing, "A data-driven multi-cloud model for stochastic parametrization of deep convection," Philos. Trans. R. Soc. A: Math., Phys. Eng. Sci. **371**, 20120374 (2013).
[43]K. Law, D. Sanz-Alonso, A. Shukla, and A. Stuart, "Filter accuracy for the Lorenz 96 model: Fixed versus adaptive observation operators," Physica D **325**, 1–13 (2016).
[44]S. Hatfield, A. Subramanian, T. Palmer, and P. Düben, "Improving weather forecast skill through reduced-precision data assimilation," Mon. Weather Rev. **146**, 49–62 (2018).

[45]F. Kwasniok, "Data-based stochastic subgrid-scale parametrization: An approach using cluster-weighted modelling," Philos. Trans. R. Soc. A: Math., Phys. Eng. Sci. **370**, 1061–1086 (2012).
[46]H. M. Arnold, I. M. Moroz, and T. N. Palmer, "Stochastic parametrizations and model uncertainty in the Lorenz 96 system," Philos. Trans. R. Soc. A: Math., Phys. Eng. Sci. **371**, 20110479 (2013).
[47]P. D. Dueben and P. Bauer, "Challenges and design choices for global weather and climate models based on machine learning," Geosci. Model Dev. **11**, 3999–4009 (2018).
[48]P. A. G. Watson, "Applying machine learning to improve simulations of a chaotic dynamical system using empirical error correction," J. Adv. Model. Earth Syst. **11**, 1402–1417 (2019).
[49]D. Coppin and S. Bony, "On the interplay between convective aggregation, surface temperature gradients, and climate sensitivity," J. Adv. Model. Earth Syst. **10**, 3123–3138 (2018).
[50]M. Colin, S. Sherwood, O. Geoffroy, S. Bony, and D. Fuchs, "Identifying the sources of convective memory in cloud-resolving simulations," J. Atmos. Sci. **76**, 947–962 (2019).
[51]S. Shamekh, K. D. Lamb, Y. Huang, and P. Gentine, "Implicit learning of convective organization explains precipitation stochasticity," Proc. Natl. Acad. Sci. U.S.A. **120**, e2216158120 (2023).
[52]W. H. Enright and M. Hu, "Interpolating Runge-Kutta methods for vanishing delay differential equations," Computing **55**, 223–236 (1995).
[53]F. Ismail, R. A. Al-khasawneh, A. S. Lwin, and M. Suleiman, "Numerical treatment of delay differential equations by Runge-Kutta method using hermite interpolation," J. Indust. Appl. Math. **18**, 79–90 (2002).
[54]A. Iserles, "A first course in the numerical analysis of differential equations," in *Cambridge Texts in Applied Mathematics*, 2nd ed. (Cambridge University Press, 2008).
[55]V. Le Guen and N. Thome, "Shape and time distortion loss for training deep time series forecasting models," in *Advances in Neural Information Processing Systems*, Vol. 32, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Curran Associates, Inc., 2019).
[56]D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv:1412.6980 (2014).
[57]A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," Physica D **404**, 132306 (2020).
[58]S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Comput. **9**, 1735–1780 (1997).
[59]S. M. Iacus, *Simulation and Inference for Stochastic Differential Equations: With R Examples (Springer Series in Statistics)*, 1st ed. (Springer Publishing Company, Incorporated, 2008).
[60]Dataset: M. A. Bhour and P. Gentine (2022). "Code and data accompanying the manuscript titled 'History-based, Bayesian, closure for stochastic parameterization: Application to Lorenz '96," Github, https://github.com/bhouri0412/Hist_Bayesian_Closure.