Statistical physics of learning in high-dimensional chaotic systems

Samantha Fournier, Pierfrancesco Urbani

Statistical physics of learning in high-dimensional chaotic systems

Samantha J. Fournier[1] and Pierfrancesco Urbani[1]

[1] *Université Paris-Saclay, CNRS, CEA, Institut de physique théorique, F-91191 Gif-sur-Yvette, France*

In many complex systems, elementary units live in a chaotic environment and need to adapt their strategies to perform a task, by extracting information from the environment and controlling the feedback loop on it. One of the main example of systems of this kind is provided by recurrent neural networks. In this case, recurrent connections between neurons drive chaotic behavior and when learning takes place, the response of the system to a perturbation should take into account also its feedback on the dynamics of the network itself. In this work, we consider an abstract model of a high-dimensional chaotic system as a paradigmatic model and study its dynamics. We study the model under two particular settings: Hebbian driving and FORCE training. In the first case, we show that Hebbian driving can be used to tune the level of chaos in the dynamics and this reproduces some results recently obtained in the study of more biologically realistic models of recurrent neural networks. In the latter case, we show that the dynamical system can be trained to reproduce simple periodic functions. To do this, we consider the FORCE algorithm –originally developed to train recurrent neural networks– and adapt it to our high-dimensional chaotic system. We show that this algorithm drives the dynamics close to an asymptotic attractor the larger the training time. All our results are valid in the thermodynamic limit thanks to an exact analysis of the dynamics through dynamical mean field theory.

I. INTRODUCTION

Biological neural networks can be described at a first approximation as elementary units, the neurons, which interact through synaptic connections. Neurons are non-linear response units in the sense that only if their incoming input current is larger than a threshold value, they spike an action potential which may trigger a spike train in other neurons [1, 2]. Such intermittent non-linear dynamics is at the fundamental basis of all high level brain activities and the way in which this micro-dynamics becomes the macro-response that triggers movements and actions in organisms is still not understood. However, it is believed that biological neural networks are not randomly connected. The synaptic connections between neurons are plastic and can be tuned (learned) to perform cognitive tasks. But the mechanism allowing such control and adaptation is still largely unknown [3].

This is at odds with artificial neural networks (ANNs) at the basis of the deep learning revolution. ANNs are high-dimensional networks typically trained to solve an optimization problem, be it to classify images [4], denoise them [5], or generate synthetic images [6, 7]. Generally, the feedforward structure of the architecture is very helpful since it allows the implementation of gradient based optimization algorithms, such as stochastic gradient descent through backpropagation.

Conversely in loopy networks, such training strategies are much more hard to implement. When the output of the neurons can be fed back into the neurons themselves, gradient based algorithms may become unstable because feedback signals may amplify or diminish, leading to diverging or vanishing gradients and non-converging dynamics. Since recurrent neural networks (RNNs) are closer (to some extent) to biological neural networks, the training problem in this case has become central also as a benchmark to propose biologically inspired learning strategies, which may be tested at the level of neurons' interactions and biology.

Models of RNNs have been studied for a long time [8]. In the simplest of settings, synaptic connections are random and no training is performed. In this case, one can observe that depending on the strength of the interactions between neurons, the dynamics can be either quiescent or chaotic. It has been shown through numerical simulations in the latter case that such RNNs can be successfully trained to perform a simple task [9]. This is done by considering a special subset of the network as a readout device whose output is fed back into the network to allow its control. Therefore, the training task aims at using the output device to suppress chaos and generate the desired response.

It is fair to say that this framework applies not only to recurrent neural networks. Biological systems as well as other complex systems (the financial market for example) typically live in chaotic environments and adaptation can be seen as a way to extract information from the (high-dimensional) chaos, and to adapt and control the feedback loop on the environment itself. Therefore, how to control and learn in chaotic environments is an ubiquitous problem. The purpose of this manuscript is to start the investigation of such problems in a simplified high-dimensional setting.

Instead of looking at specific models of RNNs or other complex chaotic systems, we consider an abstract high-dimensional chaotic system. There are several reasons to perform this abstraction step: on the one hand, we will show that the phenomenology found in specific realistic models can be found also in abstract ones, showing some degree of universality. On the other hand, the abstract models we present here have the advantage to be simpler to study from the statistical physics point of view. In particular, the dynamical mean field theory (DMFT) that we present will

provide a set of equations which describe the dynamics of the models in the thermodynamic limit. These equations can be integrated numerically more efficiently than in other systems. Since our primary goal is the application of these abstract models to RNNs, we analyze them in two steps. First, we show that the class of models that we consider share the same phenomenology as standard RNNs when untrained. In particular, we show that they can have a quiescent-to-chaotic transition as a function of the interaction strength between the degrees of freedom [8], and that the level of chaos in the chaotic phase can be tuned by Hebbian driving, analogously to what has been found in RNN models [10]. This implies that the models we consider are perfectly equivalent from the collective dynamics point of view to RNNs. Second, Sussillo and Abbott [9] have shown through numerical simulations that specific models of RNNs in their chaotic phase can be trained to perform a simple task. They developed an algorithm called FORCE to do this. We adapt their algorithm to our dynamical system and investigate the performances of this algorithm in the thermodynamic limit. We achieve this using DMFT. The main advantage of using our abstract models rather than standard RNNs (where the same analysis could be developed in principle) is that in order to study the learning dynamics, one needs to get access to long transient timescales, which is hard to do in standard RNNs where the DMFT analysis is much more complicated than in our case. The algebraic structure of our models is better suited for this task and therefore we manage to study learning in this case.

The plan of the paper is the following. In Sec.II, we will describe a simple set of high-dimensional chaotic dynamical systems which we will use as simplified abstract models. In Sec.III, we will discuss what happens when these dynamical systems are subjected to Hebbian driving, namely when the dynamics of the system itself shapes the synaptic interactions (with a simple form of the Hebb rule). In this case, we show that the abstract dynamical systems that we consider displays the same phenomenology that has been found in the context of a standard, more biologically inspired RNN under the same type of training. In particular, Clark and Abbott [10] have recently shown that Hebbian driving can shape chaos and suppress it, up to the point that the plastic synaptic couplings become so strong that chaos is completely frozen. We will review the phenomenology observed in [10] and develop a theory for it in the context of our simplified setting. In Sec.IV, we will instead consider a proper learning strategy. We will follow Ref.[9] and add to the dynamical system a readout unit which has to be trained such that its output matches a desired one. In order to perform this task, we will consider the FORCE algorithm [9] and adapt it to our dynamical systems. We DMFT analysis to show that the algorithm is effective in training the system also in the infinite size limit, and we track the behavior of the dynamical system during learning as a function of time. We will show that the learning dynamics bring the system closer to a dynamical attractor the longer the training time. Finally in Sec.V, we will discuss some perspectives on how to extend our framework.

II. A SIMPLE HIGH-DIMENSIONAL CHAOTIC SYSTEM

The simplest model of a recurrent neural network (RNN) is defined by a set of N neurons identified by an index $i = 1, \dots, N$. The state of each neuron is described by two variables, its membrane potential x_i and its firing rate r_i. The firing rate is in general a non-linear function of the membrane potential, typically $r_i = \tanh(x_i)$. The dynamics of the network is described by a set of ordinary differential equations

$$\dot{x}_i(t) = -x_i(t) + \frac{g}{\sqrt{N}} \sum_{j(\neq i)} J_i^j r_j(t) + H_i(t), \tag{1}$$

where the dot denotes the derivative with respect to time. Here, the matrix J_i^j describes the interactions between different neurons. Most importantly, this matrix is not supposed to be symmetric and therefore we will assume that $J_i^j \neq J_j^i$. Finally, $H_i(t)$ models some input current in neuron i. The model in Eq. (1) has been studied extensively in the past, especially when the synaptic coupling matrix J is thrown at random and fixed. In the simplest setting, one can assume that J_i^j are just independent Gaussian random variables with zero mean and unit variance. The control parameter g describes the strength of the random interactions between neurons. In the absence of the external current H_i and for $g = 0$, the dynamics of the network is described by a single stable attractor where $x_i = 0$ for all $i = 1, \dots, N$, meaning that all neurons are at rest. This attractor becomes unstable under linear perturbations as soon as $g > g_c$, where $g_c = 1$. In this case, the dynamics of the network is chaotic and the transition to chaos has been studied extensively in the past, see the pioneering work by Sompolinsky, Crisanti and Sommers [8] who developed the dynamical mean field theory for Eq. (1). In this chaotic phase, it has been shown in [9, 11] through numerical simulations on a finite system that the neural network can be efficiently trained. Therefore in the following, we will mainly focus on the properties of the chaotic phase.

The purpose of this work is to investigate up to which point Eq. (1) can be simplified, while retaining its main physical properties. For $g > g_c$, Eq. (1) represents a chaotic high-dimensional non-linear dynamical system. Therefore, we consider a different model still described by a set of N real dynamical variables $\underline{x} = \{x_i\}_{i=1,\dots,N} \in \mathbb{R}^N$, but we

avoid the introduction of the firing rates r_i which complicate the DMFT analysis, see [8]. In order to introduce the non-linearity in the equation we assume that

$$\dot{x}_i(t) = -\mu(t)x_i + \frac{\hat{g}}{N}\sum_{j,k} J_i^{jk} x_j x_k + H_i(t). \tag{2}$$

The matrices J_i are chosen to be GOE random matrices which means that

$$J_i^{jk} = J_i^{kj} \tag{3}$$

and

$$\overline{J_i^{jk}} = 0 \qquad \overline{\left(J_i^{j\neq k}\right)^2} = 1 \qquad \overline{\left(J_i^{jj}\right)^2} = 2\,. \tag{4}$$

We also assume that the matrices J_i and $J_{j(\neq i)}$ are independent and identically distributed. We emphasize that Eq. (2) has to be regarded as a non-linear, high-dimensional random dynamical system and the purpose of this paper is to investigate how much it resembles more standard models of RNNs. A similar dynamical system has been used to study driven glasses in [12], and the main difference with our current approach is that in [12] one adds to the lhs of Eq. (2) a conservative random force term which we completely avoid. Here, we would like to consider the model described by Eq. (2) as a simplified model of a RNN. Clearly, this model is not biologically plausible in the sense that the microscopic form of the dynamics is rather far from standard models such as Eq. (1), which try to model microscopic interactions between neurons. However, we will argue that the model has the same phenomenology as the more standard model of RNNs described by Eq. (1). The main reason to choose a dynamical system of the form of Eq. (2) is that it is simpler to study from the theoretical point of view. In particular, when we will come to study learning dynamics, we will need to develop the DMFT analysis at large timescales and this is very difficult for standard models of RNN such as Eq. (1).

We will study the behavior of the dynamical system described by Eq. (2) under different settings. First in Sec. III, we follow the recent work by Clark and Abbott [10] and introduce a Hebbian driving term in the dynamical system. We show that depending on the strength of the Hebbian couplings, one can either reduce the chaotic activity or freeze it completely to lead the network to a random fixed point attractor. Second in Sec. IV, we will discuss how Eq. (2) can be trained to reproduce a simple periodic function using the FORCE algorithm developed by Sussillo and Abbott in [9] and originally described to train the system in Eq. (1). We will also consider the discrete time algorithm defined by the Euler discretization of Eq. (2), defined as

$$x_i(t+\mathrm{d}t) = x_i(t) + \mathrm{d}t\left[-\mu(t)x_i + \frac{\hat{g}}{N}\sum_{j,k} J_i^{jk} x_j x_k + H_i(t)\right]. \tag{5}$$

At variance with the continuous time dynamics, such dynamical system depends also on the learning rate $\mathrm{d}t$. Both dynamical systems in Eq. (2) and (5) depend also on a confining potential term proportional to $\mu(t)$ which is enforced in order to avoid that the dynamics diverges to infinity.

In the following, we will develop a DMFT analysis which allows us to understand how the dynamical system behaves in the infinite size limit $N \to \infty$.

A. The statistical properties of the chaotic term

A crucial step to understand the behavior of Eq. (2) is to analyze the chaotic term defined by the random matrices J_i. It is useful to study the statistical properties of this term

$$\xi_i(t) = \frac{\hat{g}}{N}\sum_{j,k} J_i^{jk} x_j(t) x_k(t)\,. \tag{6}$$

It is clear that the average over the random matrix realization gives

$$\overline{\xi_i} = 0\,. \tag{7}$$

However ξ has an interesting dynamical two point correlation function

$$\overline{\xi_i(t)\xi_j(t')} = 2\hat{g}^2\delta_{ij}C^2(t,t'), \tag{8}$$

where the correlation function $C(t,t')$ is defined as

$$C(t,t') = \frac{1}{N}\sum_{i=1}^{N} x_i(t)x_i(t') \,. \tag{9}$$

Higher order correlation functions factorize and can be computed through Wick contractions due to the Gaussian nature of the matrices J_i.

Finally, we note that the form of the chaotic term is a particular case of a more general form. Indeed one can generalize

$$\xi_i(t) = \sum_{q=1}^{\infty} \frac{c_q}{N^{q/2}} J_i^{j_1\dots j_q} x_{j_1}(t)\dots x_{j_q}(t) \,. \tag{10}$$

By tuning carefully the coefficients c_q, one can get

$$\overline{\xi_i(t)\xi_j(t')} = \Xi(C(t,t')), \tag{11}$$

where $\Xi(z)$ is an arbitrary positive function for $z > 0$. In particular one can show that c_q^2 enters in the coefficient of the q-th term of the Taylor expansion of $\Xi(z)$. Note that both Eq. (6) and (10) describe a multibody interaction potential term. This is certainly not so natural from the biological perspective. However, in this particular work we use a multibody interaction because it is trivial to see that if Ξ is a linear function, the dynamical system becomes linear itself and therefore it is fully integrable if $\mu(t)$ does not depend on $\underline{x}$.

B. The confining potential term

Since the degrees of freedom in both Eq. (2) and (5) are continuous and real, one needs to enforce a confining mechanism to avoid that the system explores an infinite phase space. In the following, we choose two options.

- A standard way to impose a compact phase space is to bound the norm of the vector $\underline{x}$. Without losing generality, we enforce

$$\sum_{i=0}^{N} x_i(t)^2 = N \tag{12}$$

  and we dub the corresponding model as a *spherical model*. This implies that coupling $\mu(t)$ is self consistently determined to assure that at each infinitesimal time step the dynamical system never leaves the constraint in Eq. (12). We anticipate that in this case, the DMFT equations track the dynamics only in the continuous time limit, while the discrete time dynamics has a natural correction of order $\mathrm{d}t^2$ which is not properly taken into account by the Euler discretization of the DMFT equations [13, 14]. We also note that this form of the constraint is confining whatever the nature of the chaotic noise ξ and the corresponding form of its correlation functions Ξ.

- A different way to impose a confining potential is to consider a term that penalizes wild fluctuations of the norm of $\underline{x}$. A simple way to do that is to consider [13]

$$\mu(t) = f\left[\frac{1}{N}\sum_{i=1}^{N} x_i(t)^2\right], \tag{13}$$

  where the function $f(z)$ is positive and diverging function for $z \to \infty$. We dub the corresponding model a *confined model*. In this case, the DMFT dynamics can be tracked also in the discrete time step case [14]. However, the confining capability of the form in Eq. (13) depends strictly on the nature of the chaotic noise. In particular if we assume that both $\Xi(z)$ and $f(z)$ admit a polynomial expansion of finite degree, which degree we indicate respectively as d_Ξ and d_f, then the resulting dynamics is confined if

$$d_\Xi < d_f \,. \tag{14}$$

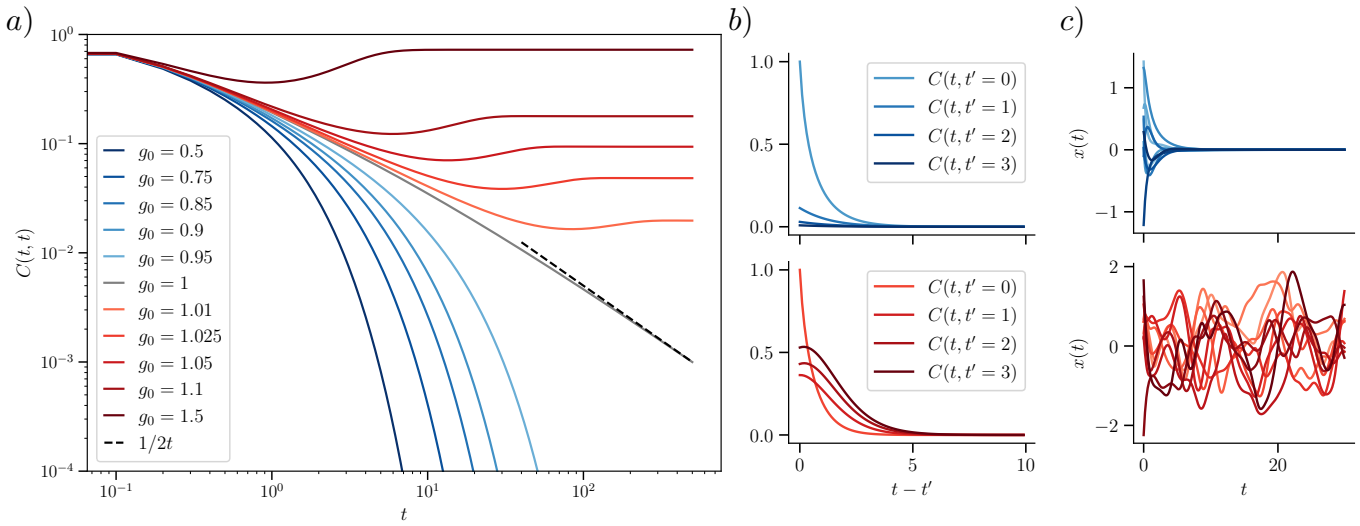

FIG. 1. Transition to chaos for the confined model defined by Eqs. (15) and (16) with $g = 1$ and $\mu(t) = 1 + C(t,t)$. *a*) Behavior of $C(t,t)$. The system is randomly initialized such that $C(0,0) = 1$. For $g_0 < g_0^c = 1$, the dynamics is attracted by the fixed point $\underline{x} = 0$. At the critical point $g_0 = 1$, the dynamics relaxes on the attractor with a power law decay. For $g_0 > g_0^c$, the dynamics stays chaotic and stabilizes on a region of phase space characterized by a limiting value of the norm $|\underline{x}(t)|^2 = \lim_{t\to\infty} C(t,t)$. *b*) Behavior of $C(t,t')$ for different values of t' as a function of $t - t'$, for $g_0 = 0.5$ (top) and $g_0 = 1.5$ (bottom). The chaotic regime (top) is characterized by fast decorrelation dynamics. *c*) Some traces of $\underline{x}$ obtained through numerical simulations for $g_0 = 0.5$ (top) and $g_0 = 1.5$ (bottom). When $g_0 = 0.5$, all the x_i go to 0; while they display chaotic dynamics for $g_0 = 1.5$.

III. TRANSITION TO CHAOS AND HEBBIAN DRIVING

We would now like to investigate whether the prototypical model of Eq. (2) is a qualitatively good model for RNNs. Specifically, we will focus on two aspects: first, we will show that the class of models in Eq. (2) can have a phase transition from a quiescent attractor phase to a chaotic activity phase, as the model in Eq. (1). Second, we will follow a recent work by Clark and Abbott [10] who showed that the level of chaos in a model of RNN described by Eq. (1) can be tuned by Hebbian driving of synapses. We will show that we can recover the same phenomenology as in [10] and we will analyze the corresponding dynamics in the thermodynamic limit through DMFT.

A. Transition to chaotic dynamics

In Sect. III B, we will consider the spherical model with $\Xi(z) = 2\hat{g}^2 z^2$. However in this case, given that the dynamics is constrained to be on the sphere and that there is no confining term pushing the system to a stable quiescent fixed point as in Eq. (1), one never encounters an attractor: the dynamics is always driven by the chaotic term whatever the strength of $\hat{g}$, as far as $\hat{g} > 0$. Therefore –at variance with the more standard model in Eq. (1)– the present model lacks a phase in which the dynamical system goes at rest to a stable attractor. In order to study this case, we consider a slightly different model, namely a confined model with

$$\mu(t) = 1 + C(t,t)\,. \tag{15}$$

Furthermore, we choose the following form for the correlation of the chaotic noise term

$$\Xi(z) = g_0^2 z + \frac{3g^2}{2} z^2 \tag{16}$$

and this corresponds to have a noise term of the form

$$\xi_i(t) = \frac{g_0}{\sqrt{N}} \sum_{j=1}^{N} J_i^j x_j(t) + \frac{\hat{g}}{N} \sum_{jk}^{N} J_i^{jk} x_j(t) x_k(t), \tag{17}$$

with $2\hat{g}^2 = 3g^2/2$. We are interested in considering what happens to the dynamical system as a function of g_0 at fixed g. We assume that the dynamics starts from an initial condition that is drawn from the flat measure over the sphere $C(t,t) = 1$. For $g_0 = 0$, the dynamical system has a fixed point at $\underline{x} = 0$ and a random initialization of the dynamics leads to this fixed point, see Fig.1. As for the neural network in Eq. (1), one can have a chaotic transition as a function of g_0. This happens when the fixed point at the origin looses linear stability. Indeed, by linearizing the dynamical system around $\underline{x} = 0$, one sees that the dynamics is described by $\delta\underline{\dot{x}}(t) = \mathcal{H}\delta\underline{x}(t)$, with the matrix $\mathcal{H}_{ij} = -\delta_{ij} + g_0 J_i^j/\sqrt{N}$ controlling the relaxation of the system. If $g_0 = 0$, the real part of the spectrum of $\mathcal{H}$ is negative and therefore the fixed point $\underline{x} = 0$ is attractive. Increasing g_0, the spectrum $\rho(\lambda)$ of $\mathcal{H}$ in the large N limit consists in a flat density of complex eigenvalues contained in a circle centered at $\lambda = -1$ in the complex plane. The circle invades the positive real axes at $g_0 = 1$ and therefore at this point the attractor $\underline{x} = 0$ looses stability. Beyond this point, the dynamics is found to be confined but chaotic. At the critical point, the approach to the marginally stable fixed point is algebraic and we show that $C(t,t) \simeq 1/(2t)$ when $t \to \infty$, see Fig.1. One can also show that for $g_0 < g_0^c$ and approaching the critical point, the dynamics relaxes exponentially to the fixed point $\underline{x} = 0$ with a characteristic time that diverges as $\tau \sim |g_0 - g_0^c|^{-1}$. The properties of the chaotic phase can be studied as well, following [8]. We use as diagnostic of chaos the fact that $C(t,t') \to 0$ for $t - t' \to \infty$ and $t' \to \infty$, as we show in Fig.1. In the same figure, we also show the behavior of some individual degrees of freedom as obtained from numerical simulations, where it is clear that the dynamics is chaotic.

B. Hebbian driving of synaptic plasticity

Eq. (2) describes the dynamics of a network where the interaction couplings are random and fixed in time. In [10], Clark and Abbott considered the case in which the activity of the neurons itself shapes the synaptic weights, which in turn control the interaction between neurons. In our model, this is equivalent to say that the dynamics of $\underline{x}$ re-shapes the interaction between degrees of freedom. In particular, following closely Clark and Abbott [10], we consider the case where in Eq. (2) the current $H_i(t)$ is a function of the state of the system through

$$H_i(t) = \sum_{j=1}^{N} A_i^j(t) x_j(t) \tag{18}$$

and the matrix $A_i^j(t)$ follows the dynamical equation

$$p\dot{A}_i^j(t) = -A_i^j(t) + \frac{k}{N} x_i(t) x_j(t)\,. \tag{19}$$

It is clear that the evolution of the plastic couplings A depends on the overall activity of the system and the strength of A depends on the coupling constant k, which is a control parameter. We note that the particular form chosen for the plastic term is not mandatory. One can easily generalize the setting to the case where

$$\begin{aligned} H_i(t) &= \sum_{j=1}^{N} A_i^{j_1 j_2 \dots j_q}(t) x_{j_1}(t) x_{j_2}(t) \dots x_{j_q}(t) \\ p\dot{A}_i^{j_1 j_2 \dots j_q}(t) &= -A_i^{j_1 j_2 \dots j_q}(t) + \frac{k}{N^q} x_i(t) x_{j_1}(t) \dots x_{j_q}(t)\,, \end{aligned} \tag{20}$$

and for $q = 1$ one gets back Eqs. (18) and (19)[1]. Eq. (20) can be rewitten as

$$A_i^{j_1 j_2 \dots j_q}(t) = A_i^{j_1 j_2 \dots j_q}(0) + \frac{k}{N^q p} \int_0^t \mathrm{d}s e^{-(t-s)/p} x_i(s) x_{j_1}(s) \dots x_{j_q}(s)\,. \tag{21}$$

In the following, we make the simplifying assumption that $A_i^{j_1 j_2 \dots j_q}(0) = 0$. Inserting this form into the dynamical equation for $\underline{x}$, we get

$$\dot{x}_i(t) = -\mu(t) x_i + \xi_i(t) + \frac{k}{p} \int_0^t \mathrm{d}s e^{-(t-s)/p} C^q(t,s) x_i(s)\,. \tag{22}$$

---

[1] One could also consider the case in which Eq. (21) is replaced by a sum of terms of different order in q. We will not discuss this case here but this generalization is straightforward.

The DMFT equations can be easily derived from Eq (22). Using the statistical properties of $\xi_i(t)$ one gets that the dynamical system is described by an effective process given by

$$\dot{x}(t) = -\mu(t)x(t) + \xi(t) + \frac{k}{p}\int_0^t \mathrm{d}s e^{-(t-s)/p} C^q(t,s)x(s), \tag{23}$$

where

$$\overline{\xi} = 0 \qquad \overline{\xi(t)\xi(t')} = \Xi\left[C(t,t')\right] \ . \tag{24}$$

Multiplying Eq. (23) and averaging over the effective noise $\xi(t)$, we get

$$\partial_t C(t,t') = -\mu(t)C(t,t') + \int_0^{t'} \mathrm{d}s\, \Xi\left[C(t,s)\right] R(t',s) + \frac{k}{p}\int_0^t \mathrm{d}s e^{-(t-s)/p} C^q(t,s)C(t',s)\,. \tag{25}$$

The response function $R(t,t')$ is defined as

$$R(t,t') = \left\langle \frac{\delta x(t)}{\delta \xi(t')} \right\rangle \tag{26}$$

and it obeys the following dynamical equation

$$\partial_t R(t,t') = -\mu(t)R(t,t') + \delta(t,t') + \frac{k}{p}\int_{t'}^t \mathrm{d}s e^{-(t-s)/p} C^q(t,s)R(s,t')\,. \tag{27}$$

At this point there are two options for the confining term $\mu(t)$. If we impose the spherical constraint of Eq. (12), this implies that $C(t,t) = 1$ at all times and one gets an equation for $\mu(t)$ directly by considering the equation for $C(t,t')$ and taking the limit $t' \to t$. In this way we get

$$\mu(t) = \int_0^t \mathrm{d}s \Xi\left[C(t,s)\right] R(t,s) + \frac{k}{p}\int_0^t \mathrm{d}s e^{-(t-s)/p} C^q(t,s)C(t,s)\,. \tag{28}$$

If the chaotic noise is not too wild and the constraint in Eq. (14) holds, then we can fix $\mu(t) = f[C(t,t)]$. In this case we need to provide a dynamical equation for $C(t,t)$ which is again easily derived from the one for $C(t,t')$. We get

$$\begin{aligned} \frac{\mathrm{d}C(t,t)}{\mathrm{d}t} &= 2\lim_{t'\to t}\partial_t C(t,t') \\ &= 2\left[\mu(t)C(t,t) + \int_0^t \mathrm{d}s\, \Xi\left[C(t,s)\right] R(t,s) + \frac{k}{p}\int_0^t \mathrm{d}s e^{-(t-s)/p} C^q(t,s)C(t,s)\right]\,. \end{aligned} \tag{29}$$

In this case, we also need to provide an initial condition for $C(0,0) = \tilde{C}$. It is easy to generalize the equations when the Hebbian driving is done up to a time t_h which we call the halting time, after which the coupling matrix A_{ij} is fixed[2]. Summarizing, we have the following equations for the correlation and response function

$$\begin{aligned} \partial_t C(t,t') &= -\mu(t)C(t,t') + \int_0^{t'} \mathrm{d}s\, \Xi\left[C(t,s)\right] R(t',s) + \frac{k}{p}\int_0^{\tilde{t}} \mathrm{d}s e^{-(\tilde{t}-s)/p} C^q(t,s)C(t',s) \\ \partial_t R(t,t') &= -\mu(t)R(t,t') + \delta(t,t') + \frac{k}{p}\int_{t'}^{\tilde{t}} \mathrm{d}s e^{-(\tilde{t}-s)/p} C^q(t,s)R(s,t') \end{aligned} \tag{30}$$

and depending on whether we have a spherical or confined model we have

$$\begin{cases} \mu(t) = \int_0^t \mathrm{d}s\, \Xi\left[C(t,s)\right] R(t,s) + \frac{k}{p}\int_0^{\tilde{t}} \mathrm{d}s e^{-(\tilde{t}-s)/p} C^q(t,s)C(t,s) & \text{spherical} \\ \frac{\mathrm{d}C(t,t)}{\mathrm{d}t} = 2\left[\mu(t)C(t,t) + \int_0^t \mathrm{d}s\, \Xi\left[C(t,s)\right] R(t,s) + \frac{k}{p}\int_0^{\tilde{t}} \mathrm{d}s e^{-(\tilde{t}-s)/p} C^q(t,s)C(t,s)\right] & \text{confined} \end{cases}\,. \tag{31}$$

The time $\tilde{t}$ is defined as $\tilde{t} = \min(t, t_h)$ and controls the dependence of the dynamics on t_h. The equations above can be easily integrated numerically. In the following, we will discuss the behavior of the solution for different values of Hebbian learning coupling k.

---

[2] One can also generalize the theory to more complex cases where the training is done with start and stop dynamics, namely when the plasticity is repeatedly switched on and off. However we do not treat this case within the DMFT but the extension is straightforward.

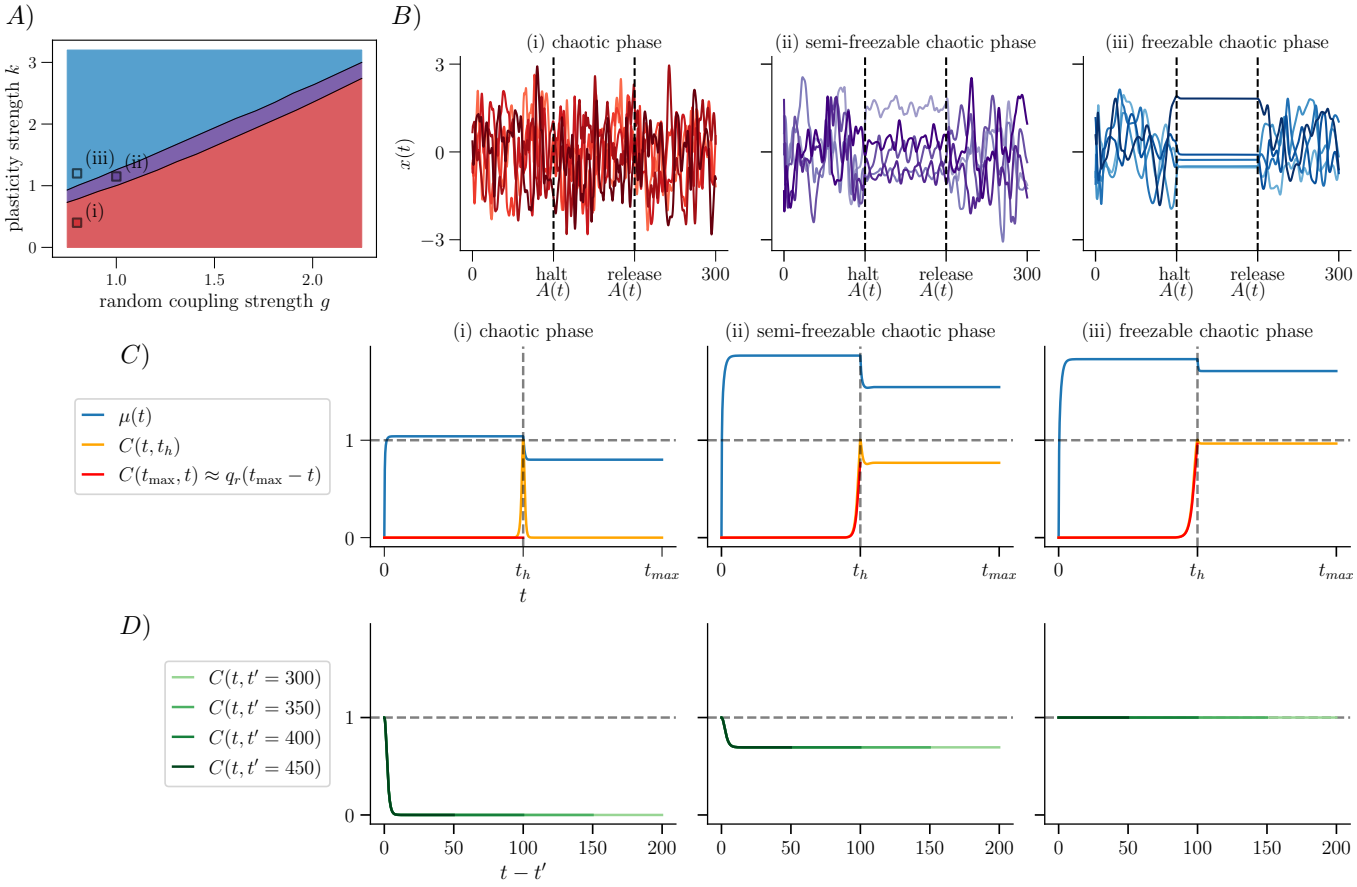

FIG. 2. *A*) The phase diagram of the model for $q = 1$ as a function of the strength of Hebbian plasticity k and of the strength of the couplings between degrees of freedom g. *B*) The traces of a set of randomly chosen $x_i(t)$ for a numerical simulation with $N = 100$ for the plots (i) and (iii), and $N = 200$ for plot (ii). The control parameters are tuned as in the points (i), (ii) and (iii) in the phase diagram of panel A). In the chaotic phase, halting the synaptic plasticity does not change the dynamics sufficiently enough to suppress chaos. In the semi-freezable chaotic phase instead, halting the synaptic plasticity leads the system to a chaotic attractor correlated with the configuration visited at the halting time. In the freezable chaotic phase, the dynamics converges to a fixed point attractor when the plasticity is halted. *C*) the behavior of a set of dynamical correlation functions as extracted from the numerical integration of the DMFT integrated up to time $t_{\max} = 500$ and fixing the halting time $t_h = 250$. In particular with the red line we plot $C(t_{\max}, t < t_h)$ as a proxy for $q_r(t_h - t)$ and *D*) we plot with green lines $C(t, t')$ for different $t' > t_h$ and as a function of $t - t'$. This shows that for t' sufficiently larger than t_h the dynamics reaches a TTI regime, and that the plateau for $t - t' \to \infty$ allows to distinguish between the SFCP and the FCP.

C. Freezable and semi-freezable chaos

We are now interested in the effect of plasticity on chaotic behavior. We will focus on the spherical model with

$$\Xi(z) = \frac{3g^2}{2} z^2, \tag{32}$$

which is always chaotic for $k = 0$. Furthermore, we will consider the $q = 1$ case in Eq. (21) to start with. Following Clark and Abbott, see [10], we consider the following protocol. Starting from a random initial condition on the sphere $C(0, 0) = 1$, we allow plastic behavior only for $t < t_h$. For $t \geq t_h$, the matrix $A_i^j(t)$ is fixed to its last value $A_i^j(t_h)$.

In [10], Clark and Abbott have identified three phases depending on the fate of the dynamical system after the halting time t_h. Depending on the strength of the Hebbian learning k, one can distinguish three phases:

- *Chaotic phase* (CP). At $k = 0$ the system is chaotic and the halting time does not have any effect. The chaotic phase survives also when k is small but finite. In this case, for $t \gg t_h$ the system completely decorrelates from the configuration at t_h.

- *Semi-freezable chaotic phase* (SFCP). For an intermediate range of k, one observes that the dynamics is still

chaotic but for $t \gg t_h$ the configurations explored are not completely decorrelated from the configuration of the system at t_h. Therefore, the dynamics lands on a chaotic attractor dynamically correlated with the configuration that the system had right before the halting time.

- *Freezable chaotic phase* (FCP). If k is sufficiently large, after the halting time, the dynamics settles to a point attractor and stops. The attractor point is correlated with the configuration visited at time t_h.

In order to carefully identify the tree phases, we need to consider a set of order parameters. The two phases SFCP and FCP can be identified by looking at

$$q_r(0) \equiv \lim_{t\to\infty} C(t, t_h) \, . \tag{33}$$

For both the SFCP and FCP we have that $q_r(0) > 0$, while when the system is in the CP, $q_r(0) = 0$. We can also introduce a generalization of Eq. (33). Indeed, we can consider

$$q_r(\Delta t) \equiv \lim_{t\to\infty} C(t, t_h - \Delta t) \, . \tag{34}$$

In the FCP and in the SFCP, $q_r(\Delta t)$ is a positive decreasing function of Δt while in the CP, we have $q_r(\Delta t) = 0$ for all intervals Δt. Therefore $q_r(\Delta t)$ allows to distinguish between the situation in which the system remains fully chaotic (CP) and when chaos is reduced, either completely (FCP) or not completely (SFCP).

In order to distinguish between the last two cases we need a different order parameter. We define

$$q_{EA} \equiv \lim_{t,t'\to\infty, t-t'\to\infty} C(t, t') \, . \tag{35}$$

In the FCP, we expect that $q_{EA} = 1$ while $0 < q_{EA} < 1$ in the SFCP. The location of the boundary between the different phases, depends on t_h. However, we will show that we can make some progress by looking at the asymptotic solution $t_h \to \infty$ (see Sect.III E).

The dynamical behavior in the three different phases can be visualized in the upper panel of Fig.2, where we show a few traces of $x_i(t)$ for numerical simulations. The corresponding phase diagram, as obtained from the DMFT analysis, is plotted in Fig.2, leftmost figure of the upper panel. All in all, the prototypical model of Eq. (2) under Hebbian driving displays the same phenomenology obtained in [10] with the more standard model of Eq. (1).

D. The case $q = 2$.

Before looking at the asymptotic solution of the DMFT equations, we would like also to investigate the behavior of the model with $q = 2$. In this case, we do not find evidence for a semi freezable chaotic phase: the system undergoes an abrupt transition from chaos to a fixed point. The corresponding phase diagram and qualitative behavior is shown in Fig.3.

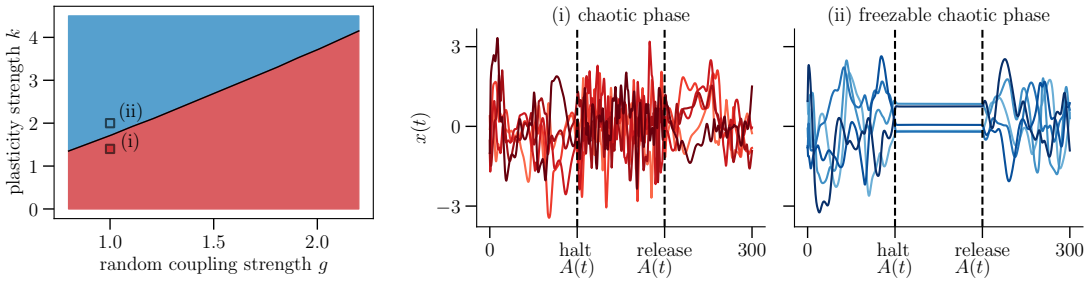

FIG. 3. *Left panel*: The phase diagram of the spherical model for $q = 2$. In this case there is no semi-freezable chaotic phase and the system undergoes a sharp transition from chaos to frozen chaos. *In the middle and right panel,* we show the traces of a set of randomly chosen x_i from a numerical simulation with $N = 100$, in the two phases at the control parameter points denoted by (i) and (ii) in the phase diagram. We note that the release of plasticity in the chaotic phase accelerates the dynamics of the variables x_i.

E. Asymptotic solution of the DMFT equations

While the DMFT allows to explore systematically the dynamics also away from stationarity, it is useful to provide an asymptotic solution of the equations when the halting time diverges at infinity, $t_h \to \infty$. It is clear that since the dynamics is either chaotic (fully chaotic or restricted to a sub-manifold) or it goes to a fixed point, in the asymptotic regime we expect that correlation functions become time translational invariant (TTI). Therefore we posit, with a slight abuse of notation, that

$$\begin{aligned} \lim_{t,t'\to\infty} C(t,t) &= C(t-t') \\ \lim_{t,t'\to\infty} R(t,t') &= R(t-t') \,. \end{aligned} \tag{36}$$

We now consider three asymptotic regimes for $t_h \to \infty$ and both $t, t' \to \infty$.

1. *Regime 1: $t, t' \ll t_h$ with $t - t' = \Delta t \sim \mathcal{O}(1)$.*

We first consider the regime in which t and t' are both diverging at infinity but they are smaller than the halting time (also diverging to infinity). In this regime we have that plasticity is never halted for the sake of t, t'. Using again TTI, we consider the asymptotic scaling functions defined as

$$\begin{aligned} C_1(\Delta t) &= \lim_{t,t'\to\infty, t-t'=\Delta t} C(t,t') \\ R_1(\Delta t) &= \lim_{t,t'\to\infty, t-t'=\Delta t} R(t,t') \\ \mu_\infty^{(1)} &= \lim_{t\to\infty, t<t_h} \mu(t) \,. \end{aligned} \tag{37}$$

Plugging this ansatz inside the dynamical equations we get

$$\begin{aligned} \partial_{\Delta t} C_1(\Delta t) &= -\mu_\infty^{(1)} C_1(\Delta t) + \frac{3g^2}{2}\int_0^\infty \mathrm{d}s C_1^2(s+\Delta t) R_1(s) + \frac{k}{p}\int_{-\Delta t}^\infty \mathrm{d}s\; e^{-(\Delta t+s)/p} C_1(s+\Delta t) C_1(s) \\ \partial_{\Delta t} R_1(\Delta t) &= -\mu_\infty^{(1)} R_1(\Delta t) + \frac{k}{p}\int_{-\Delta t}^\infty \mathrm{d}s\; e^{-(\Delta t+s)/p} C_1(s+\Delta t) R_1(s) \\ \mu_\infty^{(1)} &= \frac{3g^2}{2}\int_0^\infty \mathrm{d}s C_1^2(s) R_1(s) + \frac{k}{p}\int_{-\Delta t}^\infty \mathrm{d}s\; e^{-s/p} C_1^2(s) \,. \end{aligned} \tag{38}$$

These equations have not a causal structure since their rhs depends on times larger than Δt. However, they have a self-consistent structure and therefore can be solved by an iterative algorithm. One starts with a first guess of $\mu_\infty^{(1)}$, C_1 and R_1 and then uses these equations to produce an updated estimate of the same quantities. We verified that this numerical procedure converges fast and is compatible with the solution of the DMFT equations, which provides a first approximation of eqs. (38).

2. *Regime 2: $t' < t_h \ll t$ with $t_h - t' = \Delta t \sim \mathcal{O}(1)$.*

The second asymptotic regime is obtained by considering the situation in which one of the two times t and t' is smaller than the halting time, while the other is larger. This regime thus controls the connection between the two stationary regimes, before and after the halting time. Since we always consider $t' < t$, we have $t' < t_h$ and $t > t_h$. Furthermore, the dynamics for $t - t_h \sim \mathcal{O}(1)$ is not stationary. Therefore we consider the regime in which $t \to \infty$ and $t - t_h \to \infty$. Conversely, when $t_h - t' \sim \mathcal{O}(1)$, we are probing the asymptotic stationary regime of the dynamics before the plasticity is halted and we have access to this regime thanks to Eqs. (38). In this case, the only scaling function that we have to compute is therefore

$$q_r(\Delta t) = \lim_{t,t_h\to\infty, \Delta t\sim\mathcal{O}(1)} C(t, t_h - \Delta t) \,. \tag{39}$$

Furthermore, since $t \to \infty$ we have

$$\mu_\infty^{(2)} = \lim_{t\to\infty, t>t_h} \mu(t) \,. \tag{40}$$

The scaling equation for q_r is found just by looking at the equations in this regime. We get

$$\mu_\infty^{(2)} q_r(\Delta t) = \frac{3g^2}{2}\int_0^\infty \mathrm{d}s q_r^2(s+\Delta t)R_1(s) + \frac{k}{p}\int_{-\Delta t}^\infty \mathrm{d}s\ e^{-(\Delta t+s)/p} q_r(s+\Delta t)C_1(s) \tag{41}$$

This scaling equation is not autonomous since it depends on $\mu_\infty^{(2)}$ and C_1 and R_1. The equation for $\mu_\infty^{(2)}$ is found by looking at the third and last asymptotic regime.

3. *Regime 3: $t_h \ll t', t$ with $t - t' = \Delta t \sim \mathcal{O}(1)$.*

We finally consider the last asymptotic regime in which $t, t' > t_h$ and are infinitely far from t_h, namely $t - t_H \to \infty$ and $t' - t_h \to \infty$. In this case we need to consider the following scaling functions:

$$\begin{aligned} C_2(\Delta t) &= \lim_{t,t'\to\infty, t-t'=\Delta t} C(t,t') \\ R_2(\Delta t) &= \lim_{t,t'\to\infty, t-t'=\Delta t} R(t,t') \end{aligned} \tag{42}$$

which obey the following scaling equations

$$\begin{aligned} \partial_{\Delta t} C_2(\Delta t) &= -\mu_\infty^{(2)} C_2(\Delta t) + \frac{3g^2}{2}\int_0^\infty \mathrm{d}s C_2^2(s+\Delta t)R_2(s) + \frac{k}{p}\int_{-\Delta t}^\infty \mathrm{d}s\ e^{-(\Delta t+s)/p} q_r(s+\Delta t)q_r(s) \\ \partial_{\Delta t} R_2(\Delta t) &= -\mu_\infty^{(2)} R_2(\Delta t) \\ \mu_\infty^{(2)} &= \frac{3g^2}{2}\int_0^\infty \mathrm{d}s C_2^2(s)R_2(s) + \frac{k}{p}\int_{-\Delta t}^\infty \mathrm{d}s\ e^{-s/p} q_r^2(s)\,. \end{aligned} \tag{43}$$

The third scaling regime gives access to the order parameter which distinguishes between the SFCP and the FCP. Indeed we have

$$q_{EA} = \lim_{s\to\infty} C_2(s) \tag{44}$$

We also note that when one is in the FCP, we have $C_2(s) = 1 \ \ \forall s$.

4. *The overall structure of the asymptotic solution*

It is clear that regime 1 is fully autonomous and alone determines C_1, R_1 and $\mu_\infty^{(1)}$. Instead, we clearly see that regimes 2 and 3 are coupled by the scaling function $q_r(s)$ and by $\mu_\infty^{(2)}$. The way in which the third regime is coupled to the second is through the memory of all configurations visited for times close to t_h and this is encoded in the scaling function $q_r(\Delta t)$. We verified that these equations are satisfied by the approximate DMFT numerical solution. However, we have not been able to turn Eq. (41) into an algorithmic scheme to solve self-consistently the second and third regime. The naive iterative scheme suggested by the form of Eq. (41) seems not convergent to the right fixed point. Nevertheless, we have checked that equations (41) and (43) are coherent with the numerical solution of the DMFT equations. All in all, this analysis shows that Hebbian driving is a powerful way to control the level of chaos in the dynamical system, as much as this happens in standard RNNs (see [10]).

IV. FORCE TRAINING

Up to now, we have analyzed how a random high-dimensional chaotic system responds when Hebbian plasticity is switched on in the interactions between degrees of freedom. However for the moment, we did not treat the case in which the dynamical system is trained to perform a task. The purpose of this section is to extend the formalism developed before to address the question of how the dynamical system can be trained to produce a desired response.

It is well known that recurrent neural networks are difficult to train by energy minimization. Indeed, the recurrent structure of the interactions between the degrees of freedom implies that gradient signals can be indefinitely amplified due to feedback loops. Controlling this dynamics is therefore very complicated. Furthermore, it is fair to say that the extent to which one can think about biological neural network as devices that perform a gradient descent minimization

is unclear [15]. This is also because the computation of the gradient of a cost function is a complex operation that involves the so-called credit assignment problem, namely to select which control variables (or synapses) contribute the most to the error and therefore have the priority to be updated.

In order to overcome these difficulties, a number of strategies have been proposed to train recurrent neural networks. In the simplest setting, one would train a neural network such that a readout unit reproduces a complex periodic function. In other words, one sees the dynamical system as an out-of-equilibrium (chaotic) bath which generates some self-sustained dynamics and the main idea is to find a set of synaptic weights that connect the dynamical system to the readout unit so that its output is a desired one.

In this setting, one can distinguish two cases. If the readout unit is not fed back into the dynamical system, then the latter has a completely autonomous dynamics and therefore the problem of the explosion of gradients in a putative energy minimization training dynamics is mostly solved. This idea has been exploited enormously in the past and it is at the basis of Echo-state or Liquid-state networks [16–18].

A more complex setting consists in the situation where the output of the readout unit is re-injected into the dynamical system itself. This setting can be seen as a simplified version of training a single neuron and leaving the rest of the network unaltered. Given that the output of the readout neuron is fed back into the network, this setting suffers of the same instabilities of more general recurrent neural networks. In 2009, Sussillo and Abbott [9] have shown that one can efficiently train the readout unit coupled to the dynamical system in Eq. (1) via an algorithmic strategy called FORCE, which stands for First-Order Reduced and Controlled Error. The main idea of the algorithm is that the synaptic weights are updated always by keeping the error small along the whole training dynamics. The algorithm can be extended in many more complex situations, and more recently, it has been also shown that one can use it to train a set of Spiking Neural Networks (SNNs) [19] which differ from Eq. (1) because the dynamics of the membrane potential is resolved in time and the rates are computed microscopically as the number of times an action potential is fired.

It is fair to say that while numerical simulations have shown that FORCE can train recurrent neural networks with thousands of neurons, it is anyway unclear how the algorithm behaves on instances of infinite system size. This may be important for large scale neural networks and in particular for biological ones. The purpose of this section is to explore the performance of the FORCE algorithm in the context of the high-dimensional chaotic systems of the form represented in Eq. (2) and to construct a mean field theory analysis of such algorithms.

A. FORCE algorithm

We will first recall here the setting and the algorithm introduced in [9] and then we will adapt it to our setting. We first consider the Eq. (1) and introduce an input current of the form

$$H_i(t) = w_i^{(f)} z(t) \, . \tag{45}$$

The variable $z(t)$ is the output of the readout unit. In the simplest setting we consider

$$z(t) = \sum_{i=1}^{N} w_i^{(o)} r_i(t), \tag{46}$$

so that the output unit performs a linear readout of the state of the system. We have two sets of weights: $w_i^{(o)}$ are the synaptic weights connecting the dynamical system to the readout unit and these are the variables that we want to change in order to perform a task. The weights $w_i^{(f)}$ are instead the feedback weights and are supposed to be fixed. The taks we want the network to learn is to reproduce a function. Consider a periodic function $f(t)$ with period T. We would like to find that at the end of the training phase, the output of the readout neuron is $z(t) = f(t)$. In this way, learning will correspond to turn the chaotic noise of the dynamical activity of the untrained network to a more structured response. This task is the simplest one that cannot be performed without a feedback of the output neuron into the network itself[3]. In [9], Sussillo and Abbott have proposed the following training strategies to find a good set of weights $w_i^{(o)}$. In order to define them properly, we assume that the dynamical system in Eq. (1) is discretize with time step dt. Then we can define two algorithms:

[3] Simpler tasks like classification can instead be performed without feedback from the readout unit.

1. FORCE-I [9]: In this case we first define

$$z^+(t) = \underline{w}^{(o)}(t) \cdot \underline{r}(t + \mathrm{d}t) \tag{47}$$

and we update

$$\underline{w}^{(o)}(t + \mathrm{d}t) = \underline{w}^{(o)}(t) - \eta(t + \mathrm{d}t) \left(z^+(t) - f(t + \mathrm{d}t)\right) \underline{r}(t + \mathrm{d}t) \,. \tag{48}$$

Therefore in order to run the dynamics, in this case one first needs to update the dynamical variables $\underline{r}(t)$ and then the weights $\underline{w}(t)$. The learning rate $\eta(t)$ is a control parameter of the problem. It is known that this algorithm, while being more biologically plausible, suffers from instabilities and can learn only simple tasks [11]. These problems have been solved numerically by developing a different, more complex, and less biologically plausible algorithm, which is FORCE-II.

2. FORCE-II [9]: The update rule for the output weights is different. We define the error

$$e_-(t) = z^+(t - \mathrm{d}t) - f(t) \tag{49}$$

and update the weights with the following scheme

$$\underline{w}^{(o)}(t + \mathrm{d}t) = \underline{w}^{(o)}(t) - e_-(t + \mathrm{d}t) P(t + \mathrm{d}t) \underline{r}(t + \mathrm{d}t) \,. \tag{50}$$

The matrix $P(t)$ is an $N \times N$ matrix which follows a dynamical evolution given by the update rule

$$\begin{aligned} P(0) &= \frac{1}{\alpha} \mathbf{1} \\ P(t + \mathrm{d}t) &= P(t) - \frac{P(t) \underline{r}(t) \underline{r}(t)^T P(t)}{1 + \underline{r}(t + \mathrm{d}t)^T P(t) \underline{r}(t + \mathrm{d}t)} \end{aligned} \tag{51}$$

and we have indicated by $\mathbf{1}$ the identity matrix. The parameter α is a control parameter of the algorithm. This algorithm is naturally formulated in discrete time.

We now adapt both algorithms to train the dynamical system in Eq. (2). In order to simplify the formalism, we first consider $w_i^{(f)} = 1$ for all $i = 1, \ldots, N$. We underline that the formalism we are going to develop can be generalized to the case in which $w_i^{(f)}$ is taken to be random. In this way, we have only the set of weights that define the output unit and we call them $\underline{w}$. Therefore we define

$$z(t) = \frac{1}{N} \underline{w}(t) \cdot \underline{x}(t) \tag{52}$$

and we assume that the task of the learning protocol is to get $z(t) = f(t)$ at the end of learning. Both FORCE algorithms are formulated in terms of the variables $\underline{x}$ and $\underline{r}$. However, the dynamical system in Eq. (2) has only the $\underline{x}$ as degrees of freedom. In order to take into account this and the $N \to \infty$ limit, we consider a modified version of FORCE adapted to our setting.

- FORCE-I: we define

$$z^+(t) = \frac{1}{N} \underline{w}(t) \cdot \underline{x}(t + \mathrm{d}t) \tag{53}$$

and we update the weights according to

$$\underline{w}(t + \mathrm{d}t) = \underline{w}(t) - \eta(t + \mathrm{d}t) \left(z^+(t) - f(t + \mathrm{d}t)\right) \underline{x}(t + \mathrm{d}t) \tag{54}$$

It is very easy to show that at each time step, this algorithm is built in such a way that $f(t) = z(t)$ if $\eta(t)$ is carefully chosen (see below).

- FORCE-II: also in this case we define

$$e_-(t) = z^+(t - \mathrm{d}t) - f(t) \tag{55}$$

and we update the weights according to

$$\underline{w}(t + \mathrm{d}t) = \underline{w}(t) - e_-(t + \mathrm{d}t) P(t + \mathrm{d}t) \underline{x}(t + \mathrm{d}t) \,. \tag{56}$$

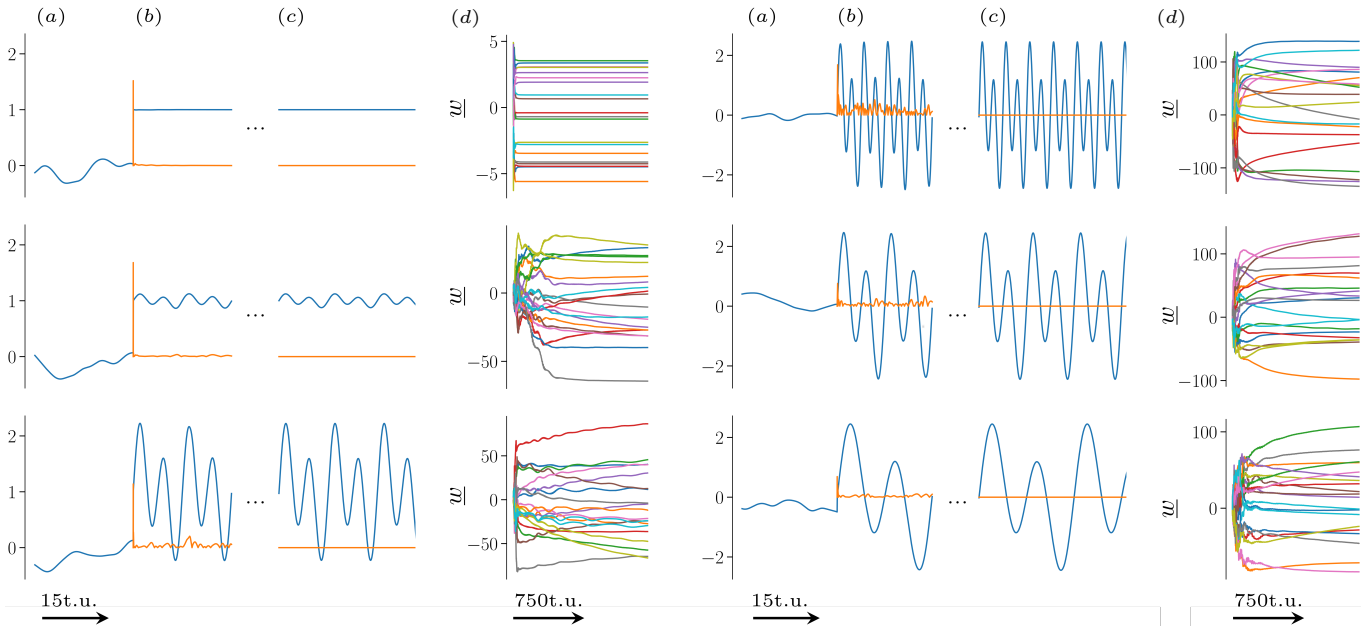

FIG. 4. The performance of FORCE-II to train the confined model. *Left panel*: Performances when training a periodic perturbation of varying amplitude around a constant. The *blue curve* is the network output $z(t)$, while the *orange curve* is the norm of the weight vector variation $\mathrm{d}\underline{w} = \underline{w}(t+\mathrm{d}t) - \underline{w}(t)$. *(a)*: Before training: Chaotic dynamics. The weights w_i are randomly uniformly initialised in $[-5, 5]$ so the network output $z(t)$ oscillates around 0. *(b)*: Training phase. FORCE-II drives quickly the output $z(t)$ to generate the target one so the weight update $\|\mathrm{d}\underline{w}\|$ is big initially and then decreases. In the left panel, $\|\mathrm{d}\underline{w}\|$ has been re-scaled by a factor 0.1 to allow the plot to fit better into the figure. Only the beginning of training is plotted. *(c)*: After training. Once $\|\mathrm{d}\underline{w}\|$ is sufficiently small (ideally of order 10^{-5}), training can be switched off. The output weight $\underline{w}$ is then fixed to its last value during training and $z(t)$ autonomously produces the target output (which is not plotted given that it superimposes to $z(t)$). Instead the plot of $\|\mathrm{d}\underline{w}\|$ (orange curve) is constant equal to 0 since $\underline{w}$ is fixed. *(d)*: Evolution of $\underline{w}$ during the training phase. Training lasted in total 1500 time units (t.u.) in all the plots to allow for comparison. *Right panel:* Same analysis as in the left panel but with the target being a periodic function with varying frequency.

The matrix $P(t)$ follows the dynamical evolution

$$\begin{aligned} P(0) &= \frac{1}{\alpha}\mathbf{1} \\ P(t+\mathrm{d}t) &= P(t) - \frac{1}{N}\frac{P(t)\underline{x}(t+\mathrm{d}t)\underline{x}(t+\mathrm{d}t)^T P(t)}{1+\frac{1}{N}\underline{x}(t+\mathrm{d}t)^T P(t)\underline{x}(t+\mathrm{d}t)} \end{aligned} \quad . \tag{57}$$

This algorithm works by keeping the error, namely $z(t) - f(t)$, small as time increases. In particular, we will study how the error decreases during learning.

B. Numerical simulations

In this section, we present a set of numerical simulations to show that the FORCE algorithm –as detailed in Sec. IV A and adapted to a random dynamical system– works to train it efficiently. We will focus on FORCE-II since FORCE-I can only be used to train simple functions [9]. We consider the confined model in Eq. (2) with $\mu(t) = C(t,t)$ and integrate numerically the dynamical equations at fixed learning rate equal to $\mathrm{d}t = 0.01$. In particular, we consider $N = 100$, $\alpha = 0.001$ and $\hat{g} = 0.7\sqrt{3/4}$ for all the data-set plotted in this section.

In Fig. 4 left panel, we consider the task of learning first a constant output $f(t) = 1$ and progressively add a small periodic perturbation around the constant value. Specifically, we choose $f(t) = A + Bg(t, \underline{\omega}^*)$ with $A = 1$, and B changes on each row of the figure. In the first row we have $B = 0$, while for the second row $B = 0.1$, and the last

$B = 1$. The function $g(t, \underline{\omega})$ is defined as

$$\begin{aligned} g(t, \underline{\omega}) &= \frac{1}{\sqrt{a^2 + b^2}} \left(a \sin(2\pi\omega_0 t) + b \sin(2\pi\omega_1 t)\right) \\ \underline{\omega}^* &= \{\omega_0 = 0.1,\ \omega_1 = 0.2\} \\ a &= 0.6 \\ b &= 1.2\,. \end{aligned} \tag{58}$$

In the right panel of Fig.4, we play the same game as learning a periodic function, this time changing the frequency and without any constant offset. In particular, we learn $f(t) = 2g(t, y\underline{\omega}^*)$ with $y = 0.5, 1, 2$ from bottom to top.

If the strength of the chaotic term is not too large (see Sect. IV C 5 for a precise way to quantify how strong it can be), we see that learning is possible. In this case when learning is switched on, the output of the network almost instantly matches the target function, which is a necessary condition for a successful FORCE-training [9]. As learning proceeds, the readout weights w_i should reach time-independent values. In practice however, we observe that reaching $\|\underline{w}(t + \mathrm{d}t) - \underline{w}(t)\| \approx 10^{-4}$ at the end of training gives satisfying performances in the testing phase after training. We also note that the amplitude and frequency of the target function influence the training process. In the left panel of Fig.4, the larger the amplitude B of the periodic perturbation, the slower learning takes place; while in the right panel of Fig.4, a periodic function with an intermediate frequency characterized by $y = 1$ is learned faster than one with $y = 0.5$ or 2.

Fig.5 instead sheds light on the small region of phase space reached by the dynamical system during training and shows how stable that region is after training, as a function of training time. We consider one learning episode during which the output $z(t)$ is trained to reproduce the target $f(t) = 3\sin(t/2)/2$. In the left panel, we plot the projection of $\underline{w}(t)$ during training on the first two principal components (PCs) of the auto-correlation matrix $\langle \underline{x}^T(t)\underline{x}(t)\rangle_t$, which is computed once $f(t)$ has been learned. The dots on this plot represent the position in the projected PC space of the dynamics after 8, 40, 60, 80 and 100 periods of $f(t)$. Thus, we see that the dynamics converges very fast to a small region of phase space where $z(t)$ matches $f(t)$, and then moves very slowly in that region. In the right panel of the same figure, we also plot the performance of the network if –during the same learning episode– we stop training after $n_h = 8$, 40, 60, 80 and 100 periods of $f(t)$. The performance of the network is measured with the error $\epsilon(n)$, defined as the temporal average of the squared difference between $f(t)$ and $z(t)$ evaluated along the n-the period of

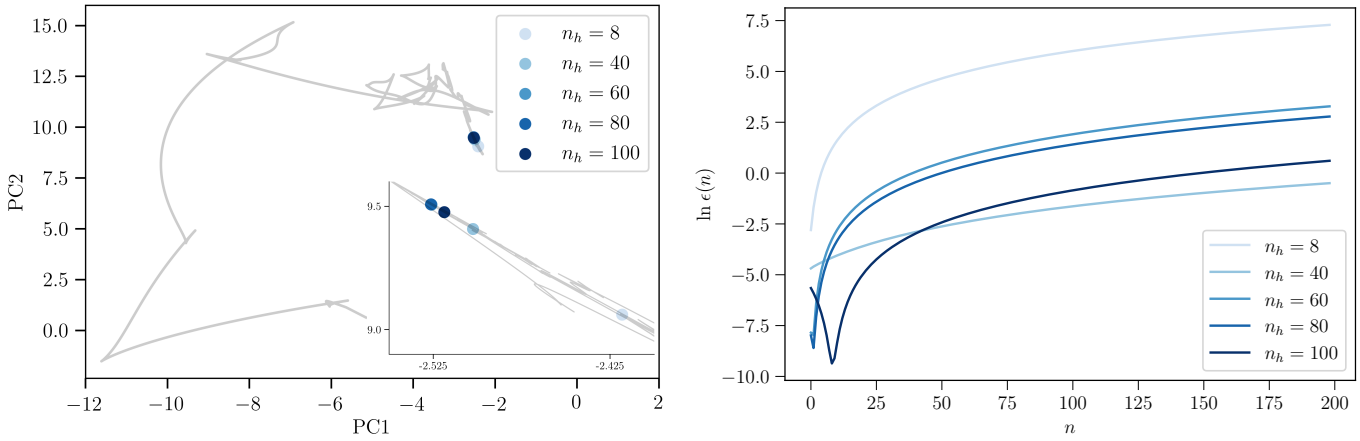

FIG. 5. *Left Panel*: Trajectory of the weight vector $\underline{w}$ (*gray* trace) during one learning episode, projected on the first two principal components (PCs) of the dynamically averaged correlation matrix $\langle \underline{x}^T(t)\underline{x}(t)\rangle_t$ (explained variance of the first two PCs, 0.978). Each component w_i is initialized randomly from a uniform distribution in $[-15, 15]$. As training proceeds, the dynamics reaches quickly a small region of phase space yielding good performances. The *blue* dots indicate the state of the system after learning for $n_h = 8$, 40, 60, 80, 100 periods of the target function $f(t)$. The *indent* figure is a zoom on the end of the trajectory, showing mild fluctuations even after learning has converged satisfyingly. Note that the dots for $n_h = 40$ and $n_h = 60$ are superimposed. *Right Panel*: The error after training as a function of time, measured in the number n of periods of $f(t)$. The *blue* curves show the errors after training for different periods of $f(t)$. If the training time is not large enough, soon after training the dynamics is not able to stay close to the desired output. Instead for larger training times, one reaches a configuration where the performance fluctuates also due to finite size.

the target function $f(t)$

$$\epsilon(n) = \int_{nT}^{(n+1)T} \mathrm{d}s |f(s) - z(s)|^2 \,. \tag{59}$$

In the right panel of Fig. 5, we see that the earlier we stop learning, the worst the performances. But after a while ($n_h \geq 40$), performance fluctuates as the dynamics wanders in the small region of phase space yielding small errors. All in all, Fig. 4 shows that the dynamical system in Eq. (2) can be trained with the FORCE-II algorithm that we have described in Sec. (IV A), see Eqs. (53)-(57).

C. Dynamical mean field theory of FORCE training

In recent years, there has been a growing interest in trying to apply DMFT to study learning in ANNs, especially in supervised learning settings with feed-forward networks [20–23]. In this section, we develop a DMFT analysis of both FORCE algorithms, which to the best of our knowledge has not been performed before. Since both algorithms are defined in the discrete time setting, we use the confined model for the dynamical system in order to follow its trajectory exactly in the large N limit.

1. The DMFT equations for the dynamical system

We first describe the DMFT for the dynamical system in Eq. (2) when the input current is given by $H_i(t) = z(t)$. We assume that time is discretized by a time step $\mathrm{d}t$. Using the same arguments as before, one can show that the DMFT equations are

$$\begin{aligned}
C(t+\mathrm{d}t, t') - C(t,t') &= \mathrm{d}t \left[-C(t,t)C(t,t') + \frac{3g^2}{2} \sum_{i=0}^{t'/\mathrm{d}t} C^2(t, i\mathrm{d}t) R(t', i\mathrm{d}t) + z(t)m(t') \right] \\
C(t+\mathrm{d}t, t+\mathrm{d}t) - C(t,t) &= 2\mathrm{d}t \left[-C(t,t)^2 + \frac{3g^2}{2} \sum_{i=0}^{t/\mathrm{d}t} C^2(t, i\mathrm{d}t) R(t, i\mathrm{d}t) + z(t)m(t) - \mathrm{d}t C(t,t) z(t) m(t) \right] \\
&\quad + \mathrm{d}t^2 \left[\frac{3g^2}{2} C^2(t,t) + C^3(t,t) + z^2(t) - 3g^2 C(t,t) \sum_{i=0}^{t/\mathrm{d}t} C^2(t, i\mathrm{d}t) R(t, i\mathrm{d}t) \right] \\
R(t+\mathrm{d}t, t') - R(t,t') &= -\mu(t) R(t,t') \mathrm{d}t + \delta_{t/\mathrm{d}t, t'/\mathrm{d}t} \\
m(t+\mathrm{d}t) - m(t) &= \mathrm{d}t \left[-\mu(t) m(t) + z(t) \right] \\
\text{with } \; C(0,0) &= \tilde{C} \\
R(0,0) &= m(0) = z(0) = 0 \,.
\end{aligned} \tag{60}$$

The function $m(t)$ controls the magnetization of the system and it corresponds to

$$m(t) = \frac{1}{N} \sum_{i=1}^{N} x_i(t) \,. \tag{61}$$

In the large N limit, $m(t)$ concentrates on its average (over the initial conditions of the dynamics and over the random realization of the chaotic noise term). Finally, the initial conditions for the dynamical correlators are due to the fact that we assume that the initial condition for $x_i(0)$ is extracted from a Gaussian measure with variance C_d and that $\underline{w}(0)$ is a vector with zero mean and uncorrelated with $\underline{x}(0)$. In all our numerical integration we considered $\tilde{C} = 1$.

From the point of view of the dynamical system of the $\underline{x}$ variables, the dynamics of the output unit is fully encoded in the variable $z(t)$. Therefore, the rest of the DMFT analysis concerns the characterization of the dynamical evolution of $z(t)$. Since we have two FORCE algorithms, we will now describe their corresponding DMFTs.

2. DMFT of FORCE-I

We need to consider both Eq. (53) and Eq. (54). Eq. (53) defines a scalar function, $z^+(t)$ which concentrates in the high dimensional limit. The goal of the DMFT analysis is to provide an equation for $z(t)$ and $z^+(t)$. Using Eq. (54) it is easy to show that

$$z(t+\mathrm{d}t) = z^+(t) - \eta(t+\mathrm{d}t)(z^+(t) - f(t+\mathrm{d}t))C(t+\mathrm{d}t, t+\mathrm{d}t) \tag{62}$$

In particular, this implies that if we choose $\eta(t+\mathrm{d}t) = 1/C(t+\mathrm{d}t, t+\mathrm{d}t)$ the dynamics during training runs on an error free trajectory since $z(t) = f(t)$ at all times. In order to close the DMFT analysis, we need to provide an equation for $z^+(t)$. This can be obtained by noting that the equation for $\underline{w}(t)$ can be rewritten as

$$\underline{w}(t) = \underline{w}(0) - \sum_{i=0}^{(t-\mathrm{d}t)/\mathrm{d}t} \eta((i+1)\mathrm{d}t)\left(z^+(i\mathrm{d}t) - f((i+1)\mathrm{d}t)\right)\underline{x}((i+1)\mathrm{d}t) \qquad t \geq \mathrm{d}t \tag{63}$$

Therefore, if we assume that $\underline{w}(0) = 0$, it is easy to show that

$$z^+(t) = -\sum_{i=0}^{(t-\mathrm{d}t)/\mathrm{d}t} \eta((i+1)\mathrm{d}t)\left(z^+(i\mathrm{d}t) - f((i+1)\mathrm{d}t)\right)C(t+\mathrm{d}t, (i+1)\mathrm{d}t) \qquad t \geq \mathrm{d}t\,. \tag{64}$$

So Eqs. (60), Eq. (62) and Eq. (64) define a causal system of equations that can be integrated numerically. They describe the behavior of the FORCE-I algorithm in the $N \to \infty$ limit. It is interesting to see that the behavior of the function $z^+(t)$ depends on $C(t,t')$ and therefore somehow has a memory of the system's history.

3. DMFT of FORCE-II

This case is more complicated due to the fact that the dynamics of the weights of the output unit depends on the dynamics of the matrix P which has a more complex flow equation. However, we will show that this change can be anyway taken into account in the high-dimensional limit. First of all, we consider Eq. (56) and multiply it by $\underline{x}(t+\mathrm{d}t)/N$. We get

$$z(t+\mathrm{d}t) = z^+(t) - e_-(t+\mathrm{d}t)\mathcal{P}(t+\mathrm{d}t, t+\mathrm{d}t, t+\mathrm{d}t), \tag{65}$$

where we have denoted

$$\mathcal{P}(t,t',t'') = \frac{1}{N}\underline{x}(t)^T P(t')\underline{x}(t'')\,. \tag{66}$$

Using the same argument as for FORCE-I, we can also write

$$z^+(t) = -\sum_{i=1}^{t/\mathrm{d}t} e_-(i\mathrm{d}t)\mathcal{P}(t+\mathrm{d}t, i\mathrm{d}t, i\mathrm{d}t) \qquad t \geq \mathrm{d}t\,. \tag{67}$$

It is clear that the exact solubility of the DMFT relies on the ability to find a recursion relation for the matrix elements of the operators $P(t)$. We will now show that such matrix elements can be obtained by recursive relations in terms of the correlation functions $C(t,t')$. First of all we have that

$$\mathcal{P}(t,0,t') = \frac{1}{\alpha}C(t,t')\,. \tag{68}$$

Furthermore, the dynamical equation for P gives

$$\mathcal{P}(t,s+\mathrm{d}t,t') = \mathcal{P}(t,s,t') - \frac{\mathcal{P}(t,s,s+\mathrm{d}t)\mathcal{P}(s+\mathrm{d}t,s,t')}{1+\mathcal{P}(s+\mathrm{d}t,s,s+\mathrm{d}t)}\,. \tag{69}$$

It is easy to convince oneself that this system of equations has a causal structure and therefore can be integrated numerically very easily. Therefore together with Eq. (60), we have the full DMFT equations given by

$$\begin{aligned}
&z(t+\mathrm{d}t) = z^+(t) - e_-(t+\mathrm{d}t)\mathcal{P}(t+\mathrm{d}t, t+\mathrm{d}t, t+\mathrm{d}t)\\
&z^+(t) = -\sum_{i=1}^{t/\mathrm{d}t} e_-(i\mathrm{d}t)\mathcal{P}(t+\mathrm{d}t, i\mathrm{d}t, i\mathrm{d}t)\\
&\begin{cases}\mathcal{P}(t,0,t') &= \frac{1}{\alpha}C(t,t')\\ \mathcal{P}(t,s+\mathrm{d}t,t') &= \mathcal{P}(t,s,t') - \frac{\mathcal{P}(t,s,s+\mathrm{d}t)\mathcal{P}(s+\mathrm{d}t,s,t')}{1+\mathcal{P}(s+\mathrm{d}t,s,s+\mathrm{d}t)}\,.\end{cases}
\end{aligned} \tag{70}$$

It is also clear that we can consider a continuous time limit leading to partial differential equations. We do not investigate this point in this work.

4. Generalizations

It is also useful to generalize the formalism presented above to the case in which there are k output neurons performing a linear readout of the system. In this case we consider that we have the input currents in the dynamical system given by

$$H_i(t) = \sum_{l=1}^{k} c_l z_l(t) \tag{71}$$

where c_l are constants that are fixed and of order one. We denote by $z_l(t)$ the output of the l unit and

$$z_l(t) = \frac{1}{N} \underline{w}_l \cdot \underline{x}(t) \tag{72}$$

where $\underline{w}_l$ are the weights of the l readout unit. We assume that there is no connection between the linear readout units and that they interact only via their feedback loops onto the dynamical system. In this case, the task would be that each readout unit produces a target function $f_l(t)$ for $l = 1, \dots k$. It is clear that the DMFT equations for the dynamical system can be straightforwardly generalized. We get

$$\begin{aligned}
C(t+\mathrm{d}t, t') - C(t,t') &= \mathrm{d}t \left[-C(t,t)C(t,t') + \frac{3g^2}{2} \sum_{i=0}^{t'/\mathrm{d}t} C^2(t, i\mathrm{d}t) R(t', i\mathrm{d}t) + m(t') \sum_{l=1}^{k} c_l z_l(t) \right] \\
C(t+\mathrm{d}t, t+\mathrm{d}t) - C(t,t) &= 2\mathrm{d}t \left[-C(t,t)^2 + \frac{3g^2}{2} \sum_{i=0}^{t/\mathrm{d}t} C^2(t, i\mathrm{d}t) R(t, i\mathrm{d}t) + m(t) \sum_{l=1}^{k} c_l z_l(t) \right] \\
&\quad + \mathrm{d}t^2 \left[\frac{3g^2}{2} C^2(t,t) + C^3(t,t) + \left(\sum_{l=1}^{k} z_l(t) \right)^2 - 3g^2 C(t,t) \sum_{i=0}^{t/\mathrm{d}t} C^2(t, i\mathrm{d}t) R(t, i\mathrm{d}t) \right] \\
R(t+\mathrm{d}t, t') - R(t,t') &= -\mu(t) R(t,t') \mathrm{d}t + \delta_{t/\mathrm{d}t, t'/\mathrm{d}t} \\
m(t+\mathrm{d}t) - m(t) &= \mathrm{d}t \left[-\mu(t) m(t) + \sum_{l=1}^{k} c_l z_l(t) \right] \\
\text{with } \; C(0,0) &= \tilde{C} \\
R(0,0) = m(0) &= z_l(0) = 0 \, .
\end{aligned} \tag{73}$$

Since we know that there is no direct interaction between the readout units, it is easy to perform the FORCE algorithm on all of them. We focus on FORCE-II. It is easy to show that for each $l = 1, \dots k$ we have a generalization of the DMFT equations for FORCE-II given by

$$\begin{aligned}
z_l(t+\mathrm{d}t) &= z_l^+(t) - e_-(t+\mathrm{d}t) \mathcal{P}_l(t+\mathrm{d}t, t+\mathrm{d}t, t+\mathrm{d}t) \\
z_l^+(t) &= -\sum_{i=1}^{t/\mathrm{d}t} e_-(i\mathrm{d}t) \mathcal{P}_l(t+\mathrm{d}t, i\mathrm{d}t, i\mathrm{d}t) \\
&\begin{cases} \mathcal{P}_l(t, 0, t') & = \frac{1}{\alpha} C(t,t') \\ \mathcal{P}_l(t, s+\mathrm{d}t, t') & = \mathcal{P}_l(t,s,t') - \frac{\mathcal{P}_l(t,s,s+\mathrm{d}t) \mathcal{P}_l(s+\mathrm{d}t, s, t')}{1 + \mathcal{P}_l(s+\mathrm{d}t, s, s+\mathrm{d}t)} \, . \end{cases}
\end{aligned} \tag{74}$$

It is clear that if $c_l = c$ and $f_l(t) = f(t)$ for all $l = 1, \dots, k$ the system has a mode collapse where all output neurons become the same. An interesting question would be how does the system behaves as soon as there is some small deviation from this rather symmetric situation. Can we understand the solution of the DMFT in terms of perturbation theory? This is left for future work. We note that the integration of the DMFT equations in this case is highly parallelizable. Indeed, each output neuron runs independently of the other and the only inputs needed are the dynamical correlation functions $C(t,t')$.

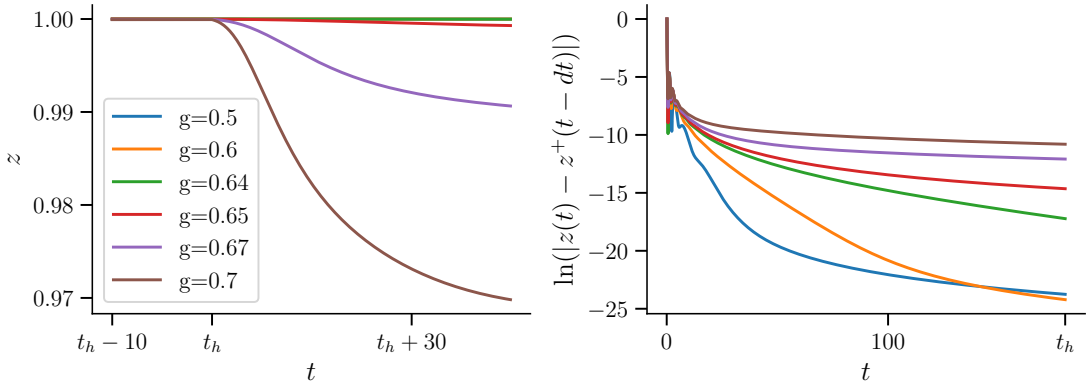

FIG. 6. *Left Panel*: the output of the network is trained to reproduce a constant function for times $t < t_h$. After t_h, training is stopped. If the chaos level is sufficiently small the network can be trained to stay close to the target output. *Right Panel*: the difference of $z(t) - z^+(t - \mathrm{d}t)$ as a function of time for different values of g. If g is larger than 0.64, the difference $z(t) - z^+(t - \mathrm{d}t)$ stays large and the network cannot be successfully trained.

5. *Numerical integration of the DMFT dynamics: performance of the algorithms*

In this section we show the results of the numerical integration of the DMFT equations describing FORCE-II, see Eqs. (60) and (65)-(70). We separate two cases, a simple case where the network needs to learn a constant function and the case in which it has to learn a periodic function. In all numerical integration we work with $dt = 0.1$ and $\alpha = 0.001$.

Learning a constant function – We consider the dynamical system trained with FORCE-II to reproduce a constant function $f(t) = 1$. In the left panel of Fig.6 we plot the output of the network $z(t)$ as a function of time across the end of the training phase and at the beginning of the post-training phase, for different values of the coupling constant g tuning the strength of the chaotic noise term. We clearly see that as soon as g is smaller than a critical value which is reasonably estimated between 0.64 and 0.65, the post-training phase is good and the system has been able to go to a fixed point. Conversely, if chaos is too strong the network is not able to stay close to the constant output. In the right panel of the same figure we plot the difference between the output $z(t)$ and $z^+(t - \mathrm{d}t)$. This difference is actually proportional to $\mathrm{d}\underline{w}(t) = \underline{w}(t) - \underline{w}(t - \mathrm{d}t)$ and therefore, if it decays to zero, it means that $\|\mathrm{d}\underline{w}\| \to 0$ and the output unit is reaching a fixed point. For small values of g it seems that this is the case, while for larger values of g, the output is not converging to a fixed point. In order to understand the critical value of g at which learning becomes possible, we can easily argue as follows. FORCE-II drives the dynamical system to $f(t) = 1 \equiv f_0$ across the training phase. If this drive is sufficient to let the dynamical system approach a fixed point, then the post-training phase will be such that the system stays at the attractor induced by the constant force $z(t) = f_0$. Therefore, the phase diagram can be drawn by looking at whether a constant force $z(t) = f(t) = f_0$ is sufficient to suppress chaos and induce an attractor in the dynamical system. This will be possible only if the level of chaos is sufficiently small with respect to f_0.

To understand the critical chaos strength, we assume that for $z(t) = f_0$ the dynamical system goes to a fixed point. The equations describing the fixed point are easily derived from the statistical properties of the chaotic noise term. Denoting

$$C_d = \lim_{t \to \infty} C(t, t) \tag{75}$$

we get that

$$C_d = \frac{1}{C_d^2} \left(\frac{3g^2}{2} C_d^2 + f_0^2 \right) . \tag{76}$$

In order to understand if this equation describes a fixed point, we need to compute its stability. Let us denote the coordinates of the fixed point as $\underline{x}^{(0)}$. Assuming that $\mathrm{d}t \to 0$ and expanding the dynamical system around this point, $x_i = x_i^{(0)} + \delta_i$ we get

$$\dot{\delta}_i = -\sum_{j=1}^{N} M_{ij} \delta_j \tag{77}$$

The stability of the fixed point is controlled by the real part of the spectrum of M. The matrix M is given by

$$M_{ij} = C_d \delta_{ij} + \frac{2}{N} x_i^{(0)} x_j^{(0)} + \frac{2\hat{g}}{N} \sum_{k=1}^{N} J_{jk}^i x_k^{(0)} \,. \tag{78}$$

It is easy to show that the real part of the spectrum of this random matrix touches zero when[4]

$$C_d = 2\hat{g}\sqrt{C_d} \tag{79}$$

and therefore learning can take place only for

$$g < g_c = \sqrt{\frac{C_d}{3}} \,. \tag{80}$$

Using Eq. (76) we get

$$g_c = \frac{1}{\sqrt{3}} \left(2 f_0^2\right)^{1/6} \,. \tag{81}$$

Therefore if $g < g_c(f_0)$, the dynamical system can learn a constant function f_0. If $f_0 = 1$ we get $g_c \simeq 0.648$ which agrees with the numerical integration of the DMFT equations (see Fig. 6).

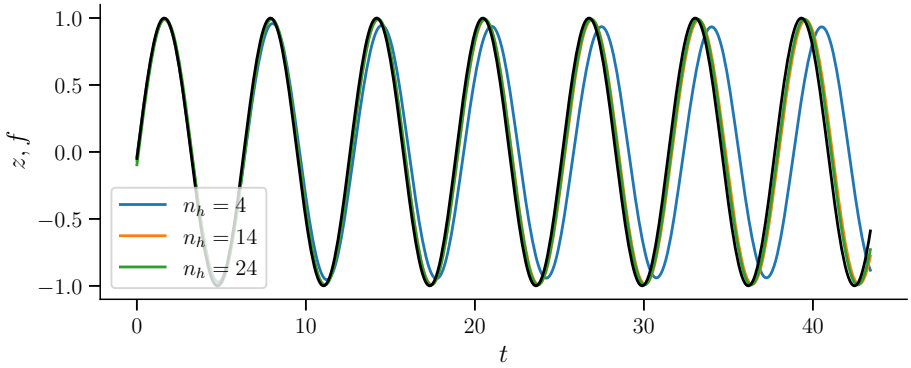

FIG. 7. The output $z(t)$ as obtained from the numerical integration of the DMFT equations, as a function of time in the post-training phase. The black line is the target output function $f(t)$. In lighter colors, the output for different values of the total learning time measured in number of periods of the function $f(t)$.

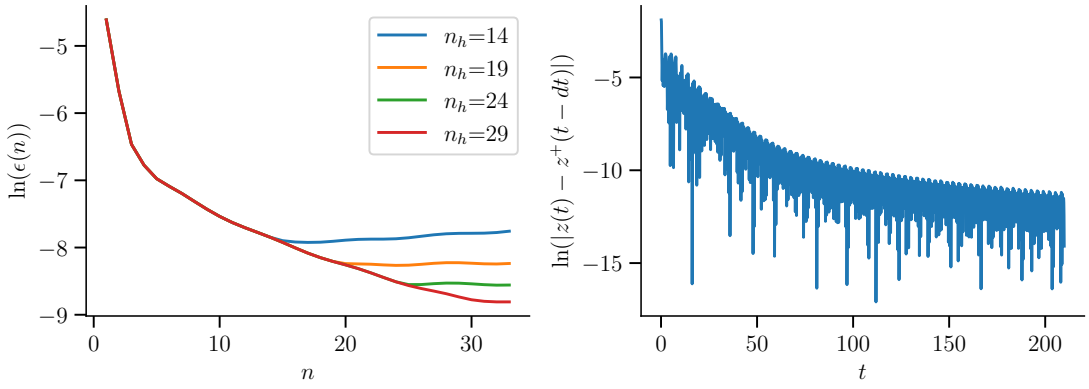

FIG. 8. *Right Panel*: the error during training and post training for different values of the training time measured in the number of periods of $f(t)$ for $f(t) = 3\sin(t)/2$. *Left Panel*: the difference between $z(t)$ and $z^+(t - \mathrm{d}t)$ which confirms that during training the dynamics is converging to a fixed point.

[4] Note that the matrix M_{ij} contains a low rank projector. However, depending on g this term may give rise to an isolated eigenvalue on the right of the bulk of the spectrum, and since here we are mostly focusing on the left side of the spectrum, this term is harmless.

Learning a periodic function – In Fig.7, we plot the output of the network in the post-training phase, as obtained by numerically integrating the DMFT equations, when the network is trained with different training times (measured in terms of periods of the periodic function $f(t)$). We choose to train the network on a simple sinusoidal function. We clearly see that the output stays closer to the black line (the function $f(t)$) the larger the number of training periods.

In order to better characterize this behavior, in the right panel of Fig.8 we plot the error $\epsilon(n)$ as a function of training periods, for different values of the training time measured in the number of periods of the function $f(t)$. We see that as soon as the training stops, the error increases exponentially, albeit with a rate that is smaller the larger the training time. Furthermore in the right panel, we plot the difference $z(t) - z^+(t - dt) \propto (\underline{w}(t) - \underline{w}(t - \mathrm{d}t))$ during training. We clearly see that FORCE-II is exponentially converging to an attractor and therefore this algorithm is effective in training the dynamical system in the infinite system size limit.

V. CONCLUSION AND PERSPECTIVES

We considered a simple set of high-dimensional chaotic systems and compared their dynamical behavior to standard RNNs under various driving forces and mechanisms. We showed in Sec. III that this class of models has chaotic properties and phases analogous to what was shown by Sompolinsky et al. [8] and Clark and Abbott [10] in more standard models of RNNs, thus establishing these models as good abstract models of more biologically grounded RNNs. We then showed in Sec. IV that the prototypical models we analyzed could also be trained via the FORCE algorithm to generate simple periodic patterns and we believe that this opens the way to study in detail the learning dynamics of more standard RNNs.

We now list a number of possible extensions of our approach, which can be studied using the methods developed in this work.

1. *The phase space of the readout weights $\underline{w}$.–* The DMFT analysis of FORCE can be simply closed on the dynamics of the scalars $z(t)$ and $z^+(t)$. However it would be very interesting to understand the dynamics of the weights $\underline{w}$. This is accessible from our formalism but we leave a detailed investigation for future work. Looking at this would clarify what is the feasible phase space of the linear readout vectors and how this space is explored by the learning algorithms. A complementary question is also related to the complexity of the function the system needs to learn. While for supervised learning tasks such as image classification it has been shown that a good measure of complexity is the intrinsic dimension of the manifold of the images of the dataset [24], here the situation is more unclear and a systematic study from DMFT seems possible.

2. *Possible interplay between Hebbian and FORCE training.–* It is well known that standard RNN can learn a task only if the level of chaos is within some working range (which may be dependent on the complexity of the task) [9, 11]. The same happens also if we use the dynamical system in Eq. (2). This is reasonable: if the level of chaos is too small, the endogenous dynamics is not sufficient to sustain the activity needed to produce a target function. Conversely, if the level of chaos is too strong, the system experiences wild fluctuations which prevent training. It would be very interesting if one could use Hebbian training as a way to tune the level of chaos during FORCE learning, in such a way that the learning task could be performed optimally.

3. *Hebbian learning: node perturbation and variants.–* FORCE learning, while being very effective, lacks of biological plausibility. For example, the algorithm relies on the computation of the matrices $P(t)$ which needs to be done off-line. It is clear that if one wants to use RNNs to model biological neural networks, it is crucial to engineer training strategies that are closer to be biologically plausible. In recent years, such line of research has been started and a few training strategies with varying degree of biological plausibility have been proposed, see [25–27]. A number of them is based on the use of an eligibility trace to solve the credit assignment problem. While in some cases there is a clear theoretical foundation for the working mechanisms of the algorithm [25], in others, the working principles are less understood and very limited [27]. A possible perspective is to try to adapt and use these training strategies in the context of the models we have been studying in this work.

4. *The high-dimensional competitive limit of linear readout units.–* We generalized our framework to the case in which there are many linear readout units. They are not directly interacting (there is no synaptic connection between them) but their interaction is mediated by the dynamical system itself. In this setting, there are two interesting perspectives to be investigated. On the one hand, it would be interesting to understand how two readout units can be trained to perform competitive tasks (which are tasks that are mutually exclusive to some degree) and what is the resulting dynamics. The other interesting limit to look at is when the number of the readout units is sent to infinity (but after the thermodynamic limit of the dynamical system itself). This would be an approximation for the situation in which the size of the central neural network is huge as compared to

the peripheric neural network (and it is the same setting that one encounters in low dimensional activities such as motor control).

5. *High-dimensional optimal control and generative modeling.–* In the previous sections, we have refereed to the endogenous drive term in Eq. (2) as a chaotic noise, see Eq. (6). An interesting perspective is to use this out-of-equilibrium noise as a bath to drive the readout units to explore target probability distributions. This would be the same strategy as in [7]. Given that the process of biasing a stochastic process to sample a given probability distribution can be recast into an optimal control problem [28], it is clear that this perspective is directly linked to high-dimensional version of optimal control [29] and the key point will be to control the statistics of the readout weights. It is also important to note that in this case the goal of the network is not to suppress chaos as in the learning tasks we have discussed in this work, but rather to control it.

6. *Spiking neural networks.–* This work has focused on a random high-dimensional chaotic system as a simplified and abstract model of a RNN. It would be interesting to investigate if this work can be generalized to spiking dynamics to model spiking neural networks [30].

Therefore, we believe that this work opens a set of interesting directions that we plan to explore in forthcoming works.

---

[1] E. R. Kandel, J. H. Schwartz, T. M. Jessell, S. Siegelbaum, A. J. Hudspeth, S. Mack, *et al.*, *Principles of neural science*, Vol. 4 (McGraw-hill New York, 2000).
[2] P. Dayan and L. F. Abbott, *Theoretical neuroscience: computational and mathematical modeling of neural systems* (MIT press, 2005).
[3] L. F. Abbott and S. B. Nelson, Nature neuroscience **3**, 1178 (2000).
[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, Communications of the ACM **60**, 84 (2017).
[5] M. Elad, B. Kawar, and G. Vaksman, arXiv preprint arXiv:2301.03362 (2023).
[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, Communications of the ACM **63**, 139 (2020).
[7] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, arXiv preprint arXiv:2011.13456 (2020).
[8] H. Sompolinsky, A. Crisanti, and H.-J. Sommers, Physical review letters **61**, 259 (1988).
[9] D. Sussillo and L. F. Abbott, Neuron **63**, 544 (2009).
[10] D. G. Clark and L. Abbott, arXiv preprint arXiv:2302.08985 (2023).
[11] D. C. Sussillo, *Learning in chaotic recurrent neural networks* (Columbia University, 2009).
[12] L. Berthier, J.-L. Barrat, and J. Kurchan, Physical Review E **61**, 5464 (2000).
[13] S. Sarao Mannelli and P. Urbani, Advances in Neural Information Processing Systems **34**, 187 (2021).
[14] F. Mignacco and P. Urbani, Journal of Statistical Mechanics: Theory and Experiment **2022**, 083405 (2022).
[15] J. C. Whittington and R. Bogacz, Trends in cognitive sciences **23**, 235 (2019).
[16] H. Jaeger, Bonn, Germany: German National Research Center for Information Technology GMD Technical Report **148**, 13 (2001).
[17] W. Maass, T. Natschläger, and H. Markram, Neural computation **14**, 2531 (2002).
[18] H. Jaeger and H. Haas, science **304**, 78 (2004).
[19] W. Nicola and C. Clopath, Nature communications **8**, 2208 (2017).
[20] F. Mignacco, F. Krzakala, P. Urbani, and L. Zdeborová, Advances in Neural Information Processing Systems **33**, 9540 (2020).
[21] F. Mignacco, P. Urbani, and L. Zdeborová, Machine Learning: Science and Technology **2**, 035029 (2021).
[22] B. Bordelon and C. Pehlevan, Advances in Neural Information Processing Systems **35**, 32240 (2022).
[23] P. J. Kamali and P. Urbani, arXiv preprint arXiv:2309.04788 (2023).
[24] A. Ansuini, A. Laio, J. H. Macke, and D. Zoccolan, Advances in Neural Information Processing Systems **32** (2019).
[25] I. R. Fiete and H. S. Seung, Physical review letters **97**, 048104 (2006).
[26] I. R. Fiete, M. S. Fee, and H. S. Seung, Journal of neurophysiology **98**, 2038 (2007).
[27] T. Miconi, Elife **6**, e20899 (2017).
[28] W. H. Fleming, Applied Mathematics and Optimization **4**, 329 (1977).
[29] P. Urbani, Journal of Physics A: Mathematical and Theoretical **54**, 324001 (2021).
[30] E. M. Izhikevich, *Dynamical systems in neuroscience* (MIT press, 2007).