Fast social-like learning of complex behaviors based on motor motifs

Carlos Calvo Tapia,[1] Ivan Y. Tyukin,[2] and Valeri A. Makarov[1,3,*]

[1]*Instituto de Matemática Interdisciplinar, Faculty of Mathematics, Universidad Complutense de Madrid, Plaza Ciencias 3, 28040 Madrid, Spain*

[2]*University of Leicester, Department of Mathematics, University Road, LE1 7RH, United Kingdom*

[3]*Lobachevsky State University of Nizhny Novgorod, Gagarin Ave. 23, 603950 Nizhny Novgorod, Russia*

(Received 22 January 2018; revised manuscript received 10 April 2018; published 24 May 2018)

Social learning is widely observed in many species. Less experienced agents copy successful behaviors exhibited by more experienced individuals. Nevertheless, the dynamical mechanisms behind this process remain largely unknown. Here we assume that a complex behavior can be decomposed into a sequence of n motor motifs. Then a neural network capable of activating motor motifs in a given sequence can drive an agent. To account for $(n-1)!$ possible sequences of motifs in a neural network, we employ the winnerless competition approach. We then consider a teacher-learner situation: one agent exhibits a complex movement, while another one aims at mimicking the teacher's behavior. Despite the huge variety of possible motif sequences we show that the learner, equipped with the provided learning model, can rewire "on the fly" its synaptic couplings in no more than $(n-1)$ learning cycles and converge exponentially to the durations of the teacher's motifs. We validate the learning model on mobile robots. Experimental results show that the learner is indeed capable of copying the teacher's behavior composed of six motor motifs in a few learning cycles. The reported mechanism of learning is general and can be used for replicating different functions, including, for example, sound patterns or speech.

DOI: 10.1103/PhysRevE.97.052308

I. INTRODUCTION

Social or imitation learning through transferring information from an experienced agent to a naive one is widely observed in many species, especially in primates and humans [1]. It relies on adopting successful actions of others and plays a fundamental role in development and communication. Electrophysiological studies suggest that the so-called mirror neurons take place in imitation learning [2,3]. Yet, the neuronal dynamical mechanisms underlying such a learning remain largely unknown.

Learning a task from scratch, i.e., without any prior knowledge, is a complicated problem [4]. Humans rarely attempt to do it. Instead, they usually extract chains of movement primitives from instructions and demonstrations by other humans. Such a paradigm also should be implemented in the future by completely autonomous robots [5], which will facilitate their social acceptance [6].

While learning, an agent (human or robot) first has to isolate essential *motor motifs* from the movement of another more experienced agent (e.g., "move right," "turn left," etc.). Then it must be able to compose a meaningful behavior from these motifs. Thus, the social learning is an example of a sequential solving of an inverse and then a forward problems. Given n motifs one can build up $(n-1)!$ different behaviors (e.g., for $n = 10$ we get 362 880). At first glance, such an explosive complexity may appear prohibitive for learning in small neural networks.

In this work we present an approach to fast social-like learning of complex motor behaviors from an arbitrary number of motor motifs. It allows synthesizing behaviors in a neural network of a learning agent by observation of the dynamics of a teacher in no more than $(n-1)$ learning cycles. Such a linear (versus factorial) growth of the learning time is reasonable for many applications.

The inverse problem, i.e., the segmentation of complex movements into a sequence of primitives, is similar to the problem of discovering network motifs in neural networks [7], which relies on statistical analysis of graphs. To segment movement trajectories Stulp and colleagues [8] proposed to use a clustering method with principal component analysis. However, even in common situations such an approach requires a huge number of demonstrations and may become unfeasible [9]. Another approach reduces the segmentation problem to a sequential recognition of movements and comparing them to a library of motor motifs [10]. Given that the library exists, this approach permits quite efficient sequencing. For the sake of simplicity, in our work we take a motif library for granted.

The forward problem can be conventionally subdivided into two tasks: (1) implementation of certain motifs (e.g., moving a hand from one point to another [11]) and (2) synthesis of complex behaviors from motor motifs. The first task can be approached by constructing dynamic movement primitives (DMPs) [12]. A DMP is given by a dynamical system with a single or multiple attractors and several parameters that are adjusted to account for the trajectory exhibited by an experienced human. The DMP approach is based on the optimal control and in the case of a relatively small number of degrees of freedom can be scaled to multiple but similar demonstrations, which produces more flexible solutions [13].

[*] vmakarov@ucm.es

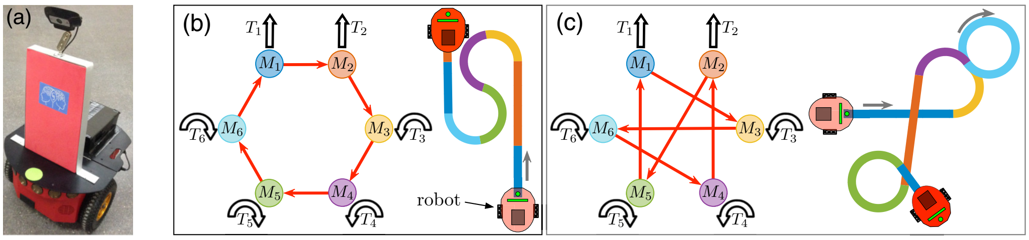


FIG. 1. Implementation of motor behaviors as sequences of motor motifs. (a) A Pioneer 3DX mobile robot used in experiments. (b), (c) Two examples of the robot's behaviors defined by graphs of six motor motifs (see main text). Depending on the order of the motifs and their durations the robot can move along different trajectories (color codifies the active motif).

The second task is usually approached from the paradigm of predictive control and optimization of trajectories [14,15]. We, however, explore the complementary but different question: How can a small neural network reproduce and learn one of $(n-1)!$ behaviors? From the dynamical systems viewpoint the social learning can be reduced to the problem of copying, in terms of synchronization, ordered in time patterns of neuronal activity from one experienced agent to another, novel one. Although the general "hardwired" architectures of the teacher's and learner's networks can be the same, their intrinsic dynamics defining the activity patterns can differ significantly. Then the learner should be able to adjust its network couplings by only observing the teacher dynamics, which leads to synchronization of the behaviors [16].

To tackle the problem here we adopt a use case approach. As a test bed we consider the generalized Lotka-Volterra model with global inhibitory couplings among neurons [16,17]. Biological neural networks usually exhibit high asymmetry in the coupling patterns [18,19], i.e., reciprocal connections between neurons i and j may differ significantly. Earlier it has been shown that this can be crucial for establishing temporal associations [20,21]. It also enables maintaining gaits in locomotion, retrieving ordered items from memory, and codification of information in population bursts [22,23]. An asymmetric Lotka-Volterra model can exhibit the so-called winnerless competition (WLC) behavior, which occurs in a vicinity of a stable heteroclinic loop connecting saddle equilibriums (see, e.g., Refs. [17,24,25]). Then, in the presence of a weak perturbation (e.g., noise; for details see Ref. [26] and references therein) a trajectory can wander in the phase space from one saddle to another, thus implementing a particular temporal pattern of neuronal excitation. This provides extremely rich behaviors even in small neural networks (see, e.g., Refs. [17,27–29]).

In this work we propose a learning algorithm capable of copying the behavior of one WLC neural network to another. Then we consider a teacher-learner social-like situation. A teacher exhibits a complex behavior consisting of a sequence of motor motifs [one of $(n-1)!$] with specific durations. Another agent, a learner, aims at mimicking the behavior. Despite the behavior complexity growing extremely rapidly with n, we show that the learning algorithm allows the learner to rewire "on the fly" the synaptic couplings in no more than $(n-1)$ learning cycles and converge exponentially fast to the durations of the teacher's motifs. We then validate the model on mobile robots.

II. MOTOR MOTIFS AND BEHAVIORS

Our first goal is to provide a neural network architecture that would enable a robot [Fig. 1(a); for details see Ref. [30]] to perform a series of simple motor actions or motifs. Then a complex motor behavior can be "programmed" as a sequence of motifs. For illustration, we select the following:

$\mathbf{M_1}$: Go straight during time interval T_1.
$\mathbf{M_2}$: Go straight during time interval T_2.
$\mathbf{M_3}$: Turn left during time interval T_3.
$\mathbf{M_4}$: Turn left during time interval T_4.
$\mathbf{M_5}$: Turn right during time interval T_5.
$\mathbf{M_6}$: Turn right during time interval T_6.

For the sake of simplicity, in the description of the motifs we did not specify the linear robot velocity, the radii of turns, etc. These parameters will be essential in Sec. V. Inclusion of similar motifs (e.g., M_1 and M_2) improves the robot flexibility and complexity of possible behaviors. Note also that the number and variety of motifs can be easily increased (with six motifs we can implement up to 120 behaviors).

We can now connect the selected motifs in a graph and thus implement a complex behavior. Figures 1(b) and 1(c) are two examples of graphs and the corresponding behaviors of the robot. In the first case, we selected the cyclic sequence [Fig. 1(b)]: $M_1 \to M_2 \to M_3 \to M_4 \to M_5 \to M_6 \to M_1 \to \cdots$, which generates a zigzag-like trajectory. In the second case the sequence was [Fig. 1(c)]: $M_1 \to M_3 \to M_6 \to M_4 \to M_2 \to M_5 \to M_1 \to \cdots$, which generates a more evolved trajectory.

Our second goal is to achieve a social-like learning of behaviors. We thus assign one "more experienced" robot as a teacher. The teacher knows how to reproduce some complex behavior given by a graph, similar to those shown in Figs. 1(b) and 1(c). Another robot, a learner, at the beginning cannot replicate the teacher's behavior. Its neural network is initialized randomly and hence generates an arbitrary behavior (e.g., go straight or make circles). Then the learner has to learn (1) the teacher's graph of motifs and (2) the time intervals $T_1, \ldots, T_n$ of all motifs.

III. NEURAL NETWORK DRIVING BEHAVIORS

Let us now provide a neural network model capable of driving a single robot, i.e., implementing all possible graphs

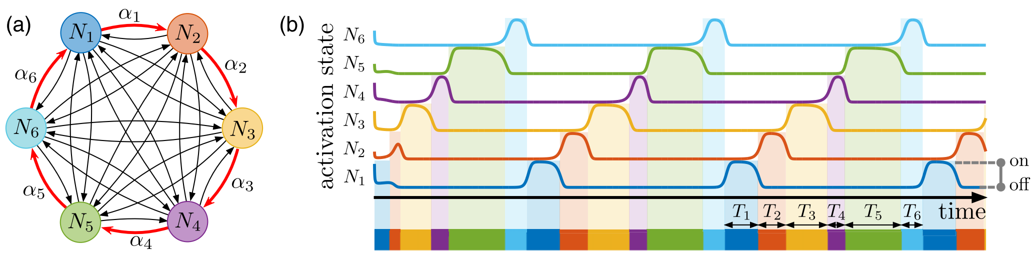


FIG. 2. Behavior driving neural network. (a) Schematic representation of the neural network (example for $n = 6$). All neurons (circles) are mutually coupled by inhibitory synapses (black and red links). Red links mark couplings $\alpha = (0.6,0.5,0.7,0.1,0.8,0.3)^T$ that define the activation sequence $N_1 \to N_2 \to N_3 \to \cdots$, corresponding to the graph of motor motifs shown in Fig. 1(b). The coupling strength of black links is equal to 2. (b) WLC dynamics generated by the network. Each neuron switches between on (active) and off (inactive) states. Bottom colored stripe shows the activation intervals $T_1,\dots,T_6$ of the corresponding neurons determined by α.

of motor motifs $M_1,\dots,M_n$ with the corresponding time intervals $T_1,\dots,T_n$.

We consider a neural network consisting of $n \geqslant 3$ globally coupled neurons [Fig. 2(a), $n = 6$]. Activation of neuron i during time interval T_i evokes the motor motif M_i, and hence the robot performs the corresponding motor action while the neuron stays active. Thus, we separate the physical implementation of each motor motif from the decision-making provided by the behavior-driving neural network. This enables employing the same dynamical principles for controlling different platforms and behaviors.

To describe a sequence of motor motifs [Figs. 1(b) and 1(c)] corresponding to a sequence of activations of neurons [Fig. 2(a)], it is convenient to introduce a directed cycle graph $G = (V,E)$ of length $n = |V|$, with the following vertices and edges (circles and red arrows in Figs. 1(b) and 1(c)]:

$$V = \{1,\dots,n\}, \quad E = \{(i,j) \in V \times V : i \to j\}. \tag{1}$$

Then every possible closed circuit of activation of neurons can be defined in terms of the adjacency or *pathway matrix*:

$$W = (w_{ij})_{i,j=1}^{n}, \quad w_{ij} = \begin{cases} 1 & \text{if } (j,i) \in E \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

In Eq. (2) $w_{ij} = 1$ corresponds to a situation when in the motor sequence after motif j goes motif i. Note that $\deg^+(v) = \deg^-(v) = 1$ for all $v \in V$, therefore W is an orthogonal permutation matrix.

There exist $(n-1)!$ different complete cyclic sequences represented by Hamiltonian cycles over V. Thus, even in a relatively small graph there can exist a huge number of possible cycles and hence motor behaviors.

To describe the network dynamics implementing sequences of motor motifs, we use the Lotka-Volterra equation generalized to an arbitrary number of neurons $n \geqslant 3$ [16,17]:

$$\dot{x} = x \odot (1_n - \rho_\alpha x) + \epsilon\, 1_n, \tag{3}$$

where $x(t) \in [0,1]^n$ is the activation state of the neurons resembling their firing rates, $\odot$ stands for the Hadamard product, $1_n = (1,1,\dots,1)^T$, ϵ is a small constant ($0 < \epsilon \ll 1$), and $\rho_\alpha \in \mathcal{M}_{n\times n}(\mathbb{R}_+)$ is the interneuron coupling matrix describing global (all-to-all) synaptic links among the neurons. Moreover, since all elements of ρ_α are positive, the couplings among neurons are inhibitory, i.e., activation of one neuron depresses the activity of the others.

Earlier it has been shown that if the matrix ρ_α satisfies appropriate conditions, then the network exhibits a WLC behavior [31]. In particular, to satisfy the conditions we can set the connectivity matrix to (see Ref. [16] for details):

$$\rho_{ij} = \begin{cases} 1 & \text{if } i = j \\ \alpha_j & \text{if } w_{ij} = 1 \\ 2 & \text{otherwise,} \end{cases} \tag{4}$$

where $\alpha_j \in (0,1)$. Then in Eq. (3) with $\epsilon = 0$, there exists an attracting heteroclinic circuit visiting in a cycle n saddles, each of which corresponds to activation of one neuron [31]. Such a cycle is defined by the pathway matrix W and corresponds to the graph G [Figs. 1(b) and 1(c)]. Since the heteroclinic circuit is structurally unstable, any small perturbation ($\epsilon > 0$) leads to an emergence of a stable limit cycle in the vicinity of the destroyed heteroclinic loop, which is the only attractor implementing a WLC behavior.

Figure 2(b) illustrates an example of the WLC dynamics. For the sake of visual clarity we have implemented the simplest excitation circuit: $N_1 \to N_2 \to N_3 \to N_4 \to N_5 \to N_6$ corresponding to the graph shown in Fig. 1(b). Loosely speaking, at each time instant only one neuron is on and it inhibits the activity of other neurons until another neuron "switches" on and the process repeats. The duration of the on state of neuron j is an increasing function of α_j, which tends to infinity if $\alpha_j \to 1$ [16]. Thus, we can individually control the time intervals of all motifs.

Therefore, playing with the pathway matrix W we can select one of the $(n-1)!$ sequences of activation of the neurons (and hence the graph of motor motifs), while choosing $\alpha = (\alpha_1,\dots,\alpha_n)^T$ we can set the temporal extension of activation for each neuron (and hence the time intervals $T_1,\dots,T_n$ for all motifs). Thus, the model (3) satisfies our needs and offers an extremely flexible way for generating a variety of motor behaviors.

IV. FAST LEARNING OF BEHAVIORS IN A TEACHER-LEARNER FRAMEWORK

Let us now consider a social-like situation with two agents: an experienced robot (a teacher) and a naive one (a learner). The teacher has a well-trained neural network (3), which allows reproducing some complex behavior [see, e.g., Figs. 1(b) and 1(c)]. As we have shown in Sec. III the teacher's behavior is defined by the pathway matrix W_x (subindex x refers to the teacher) and the vector α.

The dynamics of the neural network driving the learner is given by

$$\dot{y} = y \odot (1_n - \rho_\gamma y) + \epsilon\, 1_n, \qquad (5)$$

where $y(t) \in [0,1]^n$ is the activation state of the neurons and $\rho_\gamma(t)$ is the coupling matrix with entries $\gamma \in \mathbb{R}^n_+$ [counterpart of α in Eqs. (3) and (4)].

At the beginning the learner has an arbitrary pathway matrix W_y (counterpart of W_x) and the time interval vector γ. Thus, the dynamics of the learner can differ significantly from the dynamics of the teacher. Then the learner's goal is to learn the teacher's behavior and reproduce it.

A. Learning time intervals

Let us first assume that the learner "knows" the pathway matrix W_x, i.e., the graph of motor motifs of the teacher G. However, the time intervals of the motifs $(T_1,\dots,T_n)$ given by α in the teacher and by $\gamma(t)$ in the learner are unknown. Thus, we have

$$W_y(t) = W_x, \quad \gamma(0) \neq \alpha. \qquad (6)$$

Note that the initial values $\gamma(0)$ are taken from a random distribution, and we can even have a situation with $\gamma_j(0) \geqslant 1$ for some $j \in V$. In such a case the WLC conditions are not satisfied, and the network evolves to an equilibrium state, and hence the learner either stays still or performs a single motor motif.

The learner aims at replicating the teacher's behavior, i.e., $y(t)$ for $t > t' > 0$ must repeat in some sense the behavior of $x(t)$. To proceed further let us introduce two essential definitions.

Definition 1. Under the learning of a temporal pattern we understand a situation when a learner, sharing the same pathway matrix with the teacher ($W_y = W_x$), independently on the initial conditions $x(0)$, $y(0)$, $\gamma(0)$, can tune its synaptic couplings $\gamma(t;x)$ in such a way that

$$\lim_{t\to\infty} \gamma(t;x) = \alpha. \qquad (7)$$

We note that condition (7) implies that the learner will eventually succeed in reproducing the temporal sequence shown by the teacher:

$$\lim_{t\to\infty} (x(t) - y(t - t^*)) = 0_n, \qquad (8)$$

where t^* is some phase lag. This happens because for a fixed (stationary) $\gamma \in (0,1)^n$ the WLC model (5) has a single structurally stable attractor. Thus, given (7) at $t \to \infty$ the learner and the teacher will have the same stable limit cycle in their phase spaces. Then the phase lag t^* in (8) corresponds to a delay in movement along the limit cycle of the teacher and of the learner.

Definition 2. We say that the learning of a temporal pattern is exponentially fast if

$$\|\gamma(t) - \alpha\|_2 \leqslant M e^{-\kappa t}, \quad \forall t \geqslant 0$$

for some constants $M > 0$ and $\kappa > 0$.

We note that exponential convergence as opposed to a mere asymptotic one ensures robustness of the learning process.

1. Learning rule and exponential convergence

Following the adaptive system approach [32,33] we now postulate the learning rule:

$$\gamma(t) = \gamma_0 + W_y^T[\theta(t) - (x(t) - x_0)], \qquad (9)$$

where $\theta(t) \in \mathbb{R}^n$ is the control variable obeying

$$\dot{\theta} = x \odot (1_n - \rho_\gamma x) + \epsilon\, 1_n, \quad \theta(0) = 0. \qquad (10)$$

Note that learning rule [(9), (10)] does not depend on the direct knowledge of the teacher's internal parameters α and even of the own behavior of the learner $y(t)$, but it is fully based on the observation of the teacher's dynamics $x(t)$. Thus, the learning can be "mental", i.e., without motor execution.

For further calculations it is convenient to introduce the quadratic term:

$$p(t) = W_y^T x(t) \odot x(t). \qquad (11)$$

Then we can formulate the following result.

Theorem 1. Under the learning rule [(9), (10)] the learner learns exponentially fast any temporal pattern exhibited by the teacher.

Moreover, assuming that the teacher exhibits a stable T-periodic pattern [i.e., $x(t+T) = x(t)$], we have the following estimate for the convergence exponent:

$$\kappa = \min\{\langle p_j \rangle\}, \quad \langle p \rangle = \frac{1}{T}\int_{t_0}^{t_0+T} p(t)\,dt. \qquad (12)$$

The proof is provided in Appendix A.

2. Numerical simulations

Let us now illustrate Theorem 1 and the learning abilities of the proposed learning rule. For the sake of visual clarity we built three-neuron networks for the teacher and the learner. We then assigned three motor motifs: M_1, move right; M_2, turn left; and M_3, turn right. The teacher's sequence of motor motifs was set by selecting the pathway matrix:

$$W_x = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}, \qquad (13)$$

which imposes the motif sequence $M_1 \to M_3 \to M_2 \to \cdots$. The time interval vector was set to $\alpha = (0.2,0.6,0.8)^T$, i.e., $T_1 < T_2 < T_3$. Then we set the initial values of the learner's couplings $\gamma(0) = (1.6,0.1,2.3)^T$. Note that in this case $\gamma_1,\gamma_3 > 1$, and hence the learner cannot exhibit a WLC dynamics at the beginning. We now numerically integrate the model (3), (5), (9), and (10) and by using $x(t)$ estimate the convergence exponent (12): $\kappa = 0.0134$.

Figure 3 illustrates the learning process. As expected the couplings $\gamma(t)$ converge exponentially to α [Figs. 3(a) and 3(b)]. Steplike changes occur when the corresponding component of $p(t)$ goes through its maximum. This happens in time intervals when the activation passes from one neuron to another. At the beginning the learner does not follow a WLC dynamics, as expected, and neuron 3 (responsible for right turn) stays active for a long time interval [Fig. 3(c), $0.3T \lessapprox t \lessapprox 1.8T$, where T is the oscillation period]. Nevertheless, once all γ_j enter the region (0,1) [Fig. 3(a), $t \approx 1.5T$] the learner starts

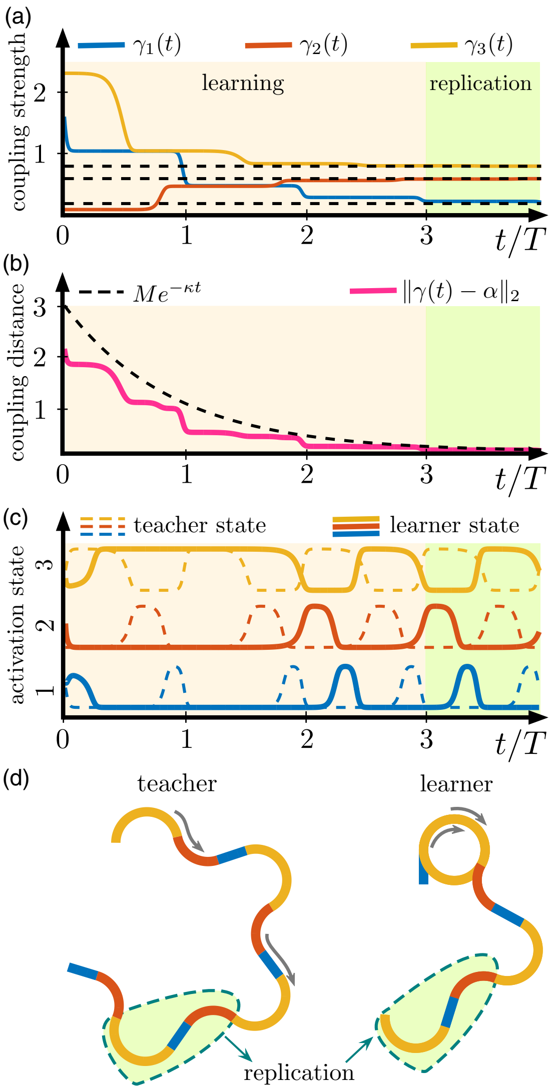


FIG. 3. Representative example of exponentially fast learning of temporal patterns. (a) Dynamics of the coupling strengths of the learner $\gamma(t)$ (solid curves) converging to the corresponding couplings of the teacher α (dashed lines). (b) Exponentially fast convergence of the couplings. (c) Synchronization (with a time shift) of oscillations in the learner with the teacher. (d) Simulation of the robots' movements. The teacher (left) performs zigzag movements, while the learner (right) at the beginning moves in a circle, but after two learning cycles it starts replicating the teacher's behavior (color coding is the same as in panel c).

generating a WLC behavior that finally synchronizes with the teacher's dynamics [Fig. 3(c), $t \gtrsim 2T$].

Figure 3(d) shows simulations of the robots' movements driven by the neural networks (see also Figs. 1(b) and 1(c)]. The teacher performs a zigzag movements. Since the time interval T_3 corresponding to motif M_3 (turn right) is the longest, the robot exhibits a tendency to turn right [Fig. 3(d), teacher]. At the beginning the learner has neuron 3 activated [Fig. 3(c), $0.3T \lessapprox t \lessapprox 1.8T$] and hence it performs circular movements [Fig. 3(d), learner]. Then, however, it synchronizes with the teacher and replicates its behavior.

B. Learning a sequence of motor motifs

Let us now consider the general case. At the beginning the learner starts from an arbitrary orthogonal matrix $W_y(0) \in \mathcal{M}_{n\times n}(\{0,1\})$, such that $W_y(0) \neq W_x$. Then $W_y(0)$ induces a wiring in the graph of motor motifs different from that of the teacher, i.e., $E_y \neq E_x$.

Without loss of generality we can assume that the teacher exhibits a T-periodic dynamics:

$$x(t) = x(t+T), \tag{14}$$

with a given pathway matrix W_x and a coupling vector α. Following the derivation similar to the proof of Theorem 1 (see Appendix A) we obtain the following equation describing the dynamics of the learner's couplings:

$$\dot{\gamma} = -p \odot \gamma + q, \tag{15}$$

where p is defined by (11) and

$$q = W_y^T x \odot \left[2x - \left(2 - W_y^T W_x \alpha\right) \odot W_y^T W_x x\right]. \tag{16}$$

Then from (15) we get

$$\gamma(t) = [\gamma_0 + g(t)] \odot f(t), \tag{17}$$

where

$$f(t) = e^{-\int_0^t p(\tau)d\tau}, \quad g(t) = \int_0^t q(\tau) \oslash f(\tau)\, d\tau, \tag{18}$$

where $\oslash$ stands for the Hadamard division. We note that $f(t)$ is a strictly positive function and $\gamma(t)$ does not now converge to α, but oscillates due to $W_y \neq W_x$.

1. Criterium of successful learning

For a successful learning (Def. 1), the pointwise limit of $\gamma(t+kT)$ [see also (B7)] for $k \to \infty$ must be equal to α, regardless of the value of t. However, in general it is an oscillating function given by (17). We then introduce the following mean-squared deviations $e_j : \mathbb{R}^2 \to \mathbb{R}$:

$$e_j(\delta) = \tfrac{1}{2}\langle[\delta_1 - (\gamma_j - \delta_2 f_j)]^2\rangle, \quad \forall j \in V, \tag{19}$$

where $\delta = (\delta_1, \delta_2)^T$ and $\langle\cdot\rangle$ denotes the time averaging operator over the period T. Note that for a given δ the learner can evaluate (19) along the trajectory, i.e., it is an observable variable.

We now can find the minimum of e_j by evaluating the gradient ∇e_j and the Hessian matrix:

$$\mathcal{H}(e_j) = \begin{pmatrix} 1 & \langle f_j\rangle \\ \langle f_j\rangle & \langle f_j^2\rangle \end{pmatrix}. \tag{20}$$

We observe that $\det(\mathcal{H}) = \mathrm{Var}[f_j] > 0$. Thus, e_j is convex and reaches the global minimum at $\nabla e_j = 0$. Solving this equation, we get the coordinates of the minimum:

$$\delta_j^* = \mathcal{H}^{-1}\begin{pmatrix} \langle\gamma_j\rangle \\ \langle\gamma_j f_j\rangle \end{pmatrix} = (\langle\gamma_j\rangle - \beta_j\langle f_j\rangle, \beta_j)^T, \tag{21}$$

where $\beta_j = \mathrm{Cov}[\gamma_j, f_j]/\mathrm{Var}[f_j]$. In other words, δ_j^* is composed of the intercept and slope values of linear regression of γ_j over f_j.

Given this observation we can formulate the following result.

Theorem 2. Let neuron $i \in V$ be the successor of neuron j in the learner's sequence of motor motifs, i.e., $(j,i) \in E_y$. Then

$$(j,i) \in E_x \Leftrightarrow e_j(\delta_j^*) = 0. \tag{22}$$

The proof is given in Appendix B. Equation (22) provides a criterium that enables indirect inferring on the structure of the pathway matrix in the teacher W_x by the learner by evaluating the error $e(\delta_j^*)$.

2. Iterative adjustment of W_y

Based on Theorem 2, we propose an iterative scheme to adjust the pathway matrix of the learner W_y. For further calculations, it is convenient to introduce the discrete time k, such that the timeline is divided into intervals:

$$I_k = [(k-1)T, kT), \quad k = 1, 2, \ldots. \tag{23}$$

We also denote

$$x_k = x(kT), \quad \theta_k = \theta(kT), \quad \gamma_k = \gamma(kT). \tag{24}$$

The learner starts with a random orthogonal matrix $W_y[0]$ and the time interval vector γ_0. Then in each interval I_k we have the matrix $W_y[k-1]$ that defines the structure of the coupling matrix $\rho_{k-1}(\gamma)$. The dynamics of the learner [compare to (5), (9), and (10)] is given by ($t \in I_k$)

$$\begin{cases} \dot{y} = y \odot [1 - \rho_{k-1}(\gamma) y] + \epsilon\, 1_n \\ \dot{\theta} = x \odot [1 - \rho_{k-1}(\gamma) x] + \epsilon\, 1_n \\ \gamma = \gamma_{k-1} + W_y^T[k-1](\theta - \theta_{k-1} - x + x_{k-1}). \end{cases} \tag{25}$$

We now find δ_j^* using (21) and then evaluate $e_j(\delta_j^*)$ from (19). By applying Theorem 2 [i.e., detecting j such that $e_j(\delta_j^*) \neq 0$], we find edges in the graph G_y that should be rewired (they do not appear in G_x). After rewiring we get new matrix $W_y[k]$. This procedure is repeated for $k = 1, 2, \ldots$. Thereby, we obtain a sequence of the matrices $\{W_y[k]\}$.

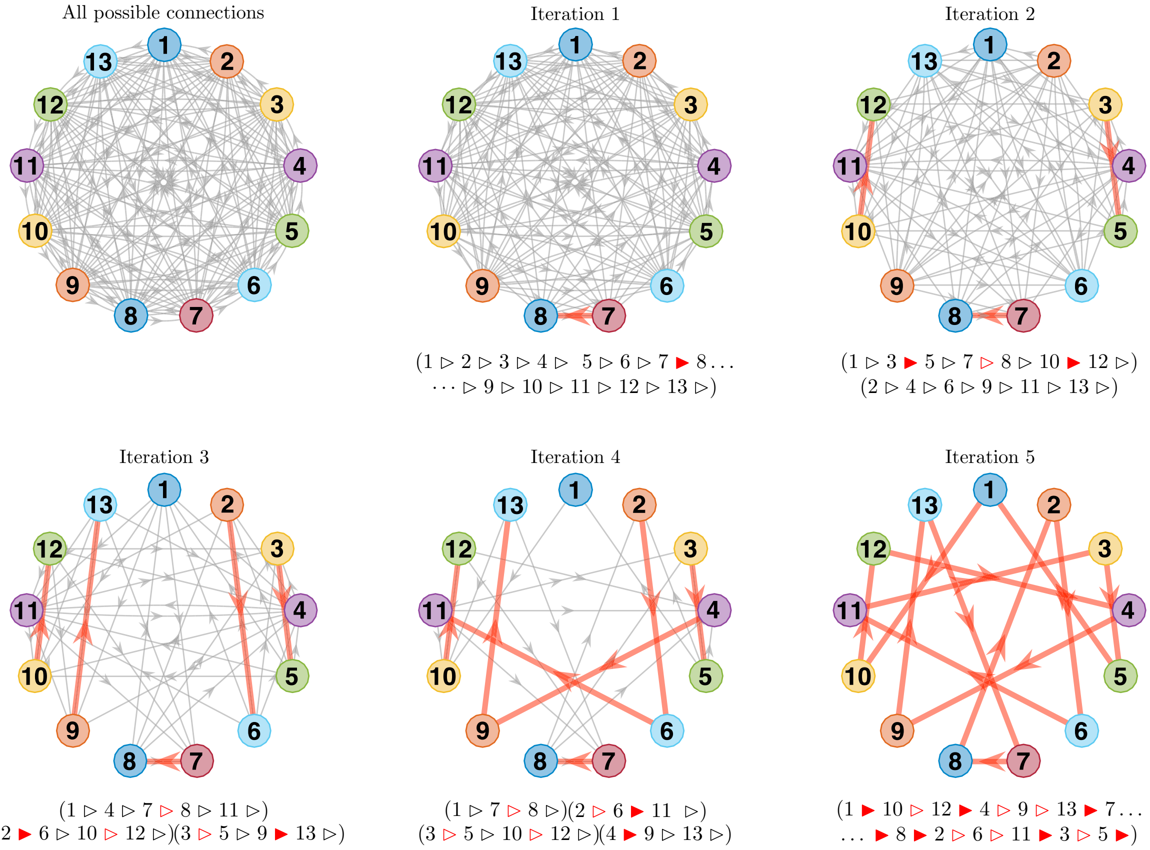
Below we will propose an algorithm of the network rewiring that ensures

$$W_y[k] = W_x, \quad \forall k \geqslant k^* > 0, \tag{26}$$

where $k^* < n$, i.e., the number of steps required to find W_x grows no more than linearly with the number of motor motifs n. Thus, the learner can learn rapidly the pathway matrix of the teacher. Then, in accordance with Theorem 1, the learner also learns the time interval vector α exponentially fast.

C. Learning algorithm

Summarizing the abovementioned results we propose the following algorithm of learning.



All possible connections

Iteration 1

$(1 \triangleright 2 \triangleright 3 \triangleright 4 \triangleright\ 5 \triangleright 6 \triangleright 7 \blacktriangleright 8 \ldots$
$\cdots \triangleright 9 \triangleright 10 \triangleright 11 \triangleright 12 \triangleright 13 \triangleright)$

Iteration 2

$(1 \triangleright 3 \blacktriangleright 5 \triangleright 7 \triangleright 8 \triangleright 10 \blacktriangleright 12 \triangleright)$
$(2 \triangleright 4 \triangleright 6 \triangleright 9 \triangleright 11 \triangleright 13 \triangleright)$

Iteration 3

$(1 \triangleright 4 \triangleright 7 \triangleright 8 \triangleright 11 \triangleright)$
$(2 \blacktriangleright 6 \triangleright 10 \triangleright 12 \triangleright)(3 \triangleright 5 \triangleright 9 \blacktriangleright 13 \triangleright)$

Iteration 4

$(1 \triangleright 7 \triangleright 8 \triangleright)(2 \triangleright 6 \blacktriangleright 11\ \triangleright)$
$(3 \triangleright 5 \triangleright 10 \triangleright 12 \triangleright)(4 \blacktriangleright 9 \triangleright 13 \triangleright)$

Iteration 5

$(1 \blacktriangleright 10 \triangleright 12 \blacktriangleright 4 \triangleright 9 \triangleright 13 \blacktriangleright 7 \ldots$
$\ldots \blacktriangleright 8 \blacktriangleright 2 \triangleright 6 \triangleright 11 \blacktriangleright 3 \triangleright 5 \blacktriangleright)$

FIG. 4. Representative example of dynamical learning of the teacher's graph $1 \to 10 \to 12 \to 4 \to 9 \to 13 \to 7 \to 8 \to 2 \to 6 \to 11 \to 3 \to 5 \to \cdots$ defining the sequence of motor motifs in the teacher's behavior. First panel shows in gray all possible connections. Iterations from 1 to 5 show correctly identified connections in red and the remaining ones in gray. A legend below each panel shows the learner's graph. Filled red triangles correspond to newly identified connections, open red triangles mark previously identified connections, and open black triangles label wrong connections.

$\mathbf{S_0}$: Initialize the parameters: 1) Select the initial permutation $\sigma{:}V \to V$, $\sigma = (1,2,\ldots,n)$; 2) Select random γ_0; 3) $\theta_0 \leftarrow 0$; 4) $k \leftarrow 0$; 5) $\Omega \leftarrow V$.

$\mathbf{S_1}$: While $\Omega \neq \emptyset$ repeat:

(1) $k \leftarrow k+1$.

(2) Set $W_y[k-1] = (w_{ij})_{i,j=1}^n$, where

$$w_{ij} = \begin{cases} 1 & \text{if } i = \sigma(j) \\ 0 & \text{otherwise.} \end{cases} \tag{27}$$

(3) Integrate (25) over I_k by using the coupling matrix ρ_{k-1} given by $W_y[k-1]$.

(4) Find δ_j^* and $e_j(\delta_j^*)$ from (19)–(21).

(5) Set $\Omega = \{j \in V : e_j(\delta_j^*) \neq 0\}$ and label its elements in ascending order $\Omega = \{\omega_1,\ldots,\omega_m\}$, with $\omega_k < \omega_{k+1}, \forall k < m$.

(6) For each $i \in \{1,\ldots,m\}$ update the permutation $\sigma(w_i) \leftarrow \sigma(w_{i^+})$, where

$$i^+ = \begin{cases} i+1 & \text{if } i < m \\ 1 & \text{if } i = m. \end{cases} \tag{28}$$

$\mathbf{S_2}$: Using the last W_y (note that now $W_y = W_x$) apply the learning rule (9) and (10) for $t \geqslant kT$.

The learning time of the proposed algorithm is given by the following theorem.

Theorem 3. The iterative learning process described at step $\mathbf{S}_1$ of the algorithm converges in $(n-1)$ iterations at most.

The proof is provided in Appendix C. Thus, the learning time grows linearly with the number of neurons, while the number of possible behaviors grows extremely fast, as factorial.

Let us now illustrate the algorithm numerically. We built two neural networks with 13 neurons each (a teacher and a learner) and selected a random pathway matrix W_x for the teacher. Then we initialized the learner (step $\mathbf{S}_0$) and started the algorithm. Figure 4 illustrates the process of learning of the connectivity matrix W_y (step $\mathbf{S}_1$).

At the beginning there exist 4.79×10^8 possible graphs of the motor motifs (panel "all possible connections"). At the first iteration (lasting one period of the teacher's behavior) the learning algorithm identifies that the link going from neuron 7 to neuron 8 exists in the teacher, while the others defined by $W_y[0]$ do not (Fig. 4, Iteration 1). Thus, we keep link 7 → 8 and change the others ($\mathbf{S}_1$, item 6 of the algorithm). Then a new wiring is set up and we repeat the calculation over the second period of the teacher's behavior. Within Iteration 2 the learner finds two more correct links: 10 → 12 and 3 → 5 (Fig. 4, Iteration 2). In the following three iterations, the learner gets to learn all connections $1 \to 10 \to 12 \to \cdots$. Thus, it was able to learn the teacher's sequence out of 4.79×10^8 possible cases in only five iterations or demonstrations of the teacher's behavior.

Then in step $\mathbf{S}_2$ the time intervals of the motor motifs (the values of γ) converge exponentially (Theorem 1). Note that this process does not depend on the number of motifs. Thus, the time interval required to learn a complex motor behavior grows linearly with the complexity of the behavior (i.e., with the number of motifs).

V. EXPERIMENTAL VALIDATION

Let us now test experimentally the proposed learning model. We equipped two robotic platforms [Pioneer 3DX, Adept Mobilerobotics, linear sizes l × w: 45.5 × 38.1 cm, Fig. 1(a)] with neural networks consisting of $n = 6$ neurons. The network dynamics had been implemented in Matlab (Mathworks) running on an on-board Intel NUC PC connected to the platform through a COM interface. One platform played the role of a teacher (with a predesigned complex motor behavior), while the other one was designated as a learner with the driving network randomly initialized.

In both platforms we implemented six motor motifs provided in Sec. II. The linear robot velocity was set to 10 cm/s and the radii of turns to 17 cm. These parameters were chosen to fit the robot's behavior within the available arena (8 × 3.5˜m). During all experiments the robots' behavior was recorded by a wide-angle zenithal USB camera. The geometric distortions of the actual position of the robots in the arena (due to projection) were corrected by applying an appropriate transformation.

The neural network dynamics of the teacher has been translated into motor commands by

$$i(t) = \operatorname{argmax}\{x_j(t)\}_{j=1}^6, \tag{29}$$

where $i(t) \in \{1,2,\ldots,6\}$ is the number of the motor motif performed by the robot at time instant t. The same mechanism has been used to drive the learner.

The pathway matrix W_x, defining the coupling matrix ρ_α, and hence the sequence of the motor motifs of the teacher, was selected such that it reproduced the graph shown in Fig. 1(c) (i.e., $M_1 \to M_3 \to M_6 \to M_4 \to M_2 \to M_5$). The values of α were set in such a way that the durations of the motor motifs were $T_1 = 7.0$ s, $T_2 = 7.1$ s, $T_3 = 4.1$ s, $T_4 = 4.1$ s, $T_5 = 9.4$ s, $T_6 = 11.0$ s. Then during one period $T = \sum T_i = 42.7$ s the teacher exhibited a behavior similar to that shown schematically in Fig. 1(c). Figure 5(a) shows the teacher's trajectory corresponding to five repetitions of the motor pattern. The robot repeats the behavior shown in Fig. 1(c) and produces a complex "flower"-like trajectory [Fig. 5(a)].

The neural network of the learner was initialized at random (W_y and γ were arbitrary chosen). Then the learning algorithm (Sect. IV C) has been activated and the robot started moving. Figure 5(b) shows the robot trajectory. At the beginning the robot performed quite chaotic movements, only partially reproducing the teacher's motor pattern. However, after three periods the learner was able to capture completely the pathway matrix W_x [i.e., $W_y(t) = W_x$ for $t > 4T$] and started repeating the teacher pattern quite reliably. Then the values of γ were also tuned (exponentially fast), and finally the learner was able to reproduce the teacher's behavior faithfully.

To quantify the difference between the trajectories performed by the robots we introduced the following metric:

$$D(t) = \min_{\tau \in [0,T]} \left[\frac{1}{T} \int_{t-T}^{t} |c_T(s) - c_L(s-\tau)|^2 \, ds \right]^{\frac{1}{2}}, \tag{30}$$

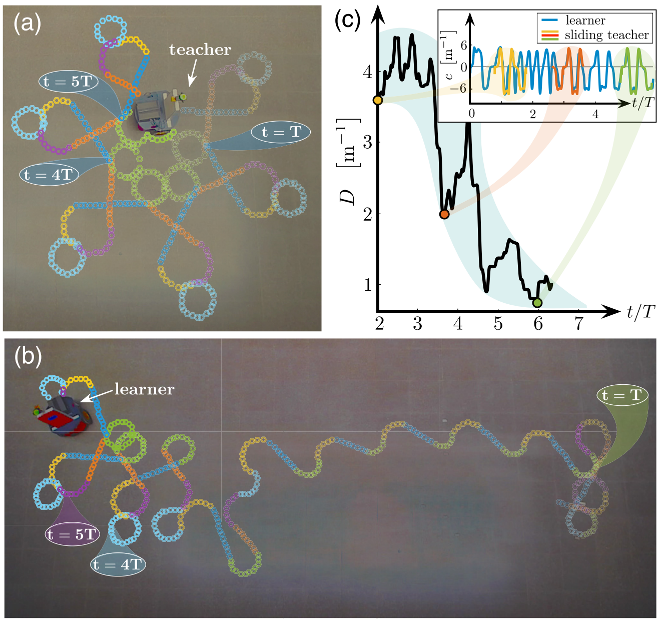


FIG. 5. Experimental validation of learning. (a) Robot teacher implements the behavior composed of six motor motifs shown in Fig. 1(c). The brightness of colors corresponds to time (the brighter, the closer to the present). (b) Trajectory of the robot learner. At the beginning it differs significant from the teacher's behavior. However, in only three cycles the robot does learn the sequence of turns, and then two more cycles are needed for final adjustment of the coupling strengths. Eventually the learner replicates almost exactly the behavior of the teacher. (c) Quantification of the learning error (30), defined as the distance between trajectories of the teacher and learner. Inset illustrates the trajectory curvature of the learner (blue curve) with superimposed curvatures of the teacher in three time windows marked by circles of different colors. The level of coincidence increases with time.

where c_T and c_L are the curvatures of the teacher's and learner's trajectories, respectively, evaluated by

$$c(t) = \frac{\tilde{x}'(t)\tilde{y}''(t) - \tilde{y}'(t)\tilde{x}''(t)}{(\tilde{x}'(t)^2 + \tilde{y}'(t)^2)^{3/2}}, \tag{31}$$

where $(\tilde{x}(t), \tilde{y}(t))$ is the parametric robot trajectory captured by the video camera. Since the linear robot velocity is constant, curvature (31) describes uniquely the robot trajectory on a two-dimensional plane. We also note that the metric (30) is invariant with respect to the translational and rotational symmetries and the phase lag in the learner [see Eq. (8)]. Thus, the introduced metric fulfills all requirements for a distance measure between two trajectories.

Experimentally acquired trajectories are not precise due to a number of reasons, starting from errors in the odometry [30], identification of the robot position, and ending by the distortions produced by the camera. Thus, to evaluate the metric (30) we upsampled the trajectories shown in Figs. 5(a) and 5(b) and denoised the data by a continuous wavelet approach [34]. Figure 5(c) shows the dynamics of the distance between the trajectories of the teacher and the learner. As expected the distance strongly oscillates at the beginning due to rewiring and changes of γ, but then decreases and approaches a stationary level, defined by measurement and motor noise. Thus, the learner robot can indeed adapt "on the fly" the wiring of its decision-making network and mimic the movements of the teacher.

VI. CONCLUSION

Cognitive learning of motor behaviors is a complicated problem even for humans. In this work, we have proposed an efficient mechanism for dynamical learning in neural networks. In a social-like situation, a teacher (mobile robot in experiments) can exhibit a complex behavior consisting of a sequence of n motor motifs (simple motor actions). In general, there exist $(n-1)!$ combinations of motifs. Moreover, all motifs have specific time durations that also play an important role in formation of the final motor behavior. Thus, the complexity and variety of the motor patterns available to an agent grow extremely rapidly with n. This poses two problems: (1) How can a small neural network implement such a variety of different behaviors? and (2) How can a learner (i.e., another neural network) copy the unique behavior of the teacher starting from random initial conditions?

To implement a motor behavior (e.g., to drive a robot teacher) we have proposed to link the output of n neurons with n motor motifs. Then an activation of a given neuron evokes the execution of a corresponding motor motif. This approach allows separating the "behavioral" neural network from the executive motor part. Thus, the network dynamics does not depend on specific motor motifs and physical properties of the agent, and hence can be easily transferred to other technical devices. To avoid contradicting commands in such an architecture, two neurons cannot be active simultaneously. This property has been achieved by employing the WLC paradigm that ensures sequential "switching" of neurons in the network and also enables simple independent control of the durations of "on" states of the neurons [16,17,24].

Using the motor motif approach, a learner has to solve two tasks: (1) Find the teacher's graph of motifs (i.e., the pathway matrix) and (2) Adjust the time durations of motifs (i.e., the coupling strengths). To deal with the latter task, earlier we provided a neural network model that enabled learning of the activation times [16,35]. However, to reproduce the teacher's behavior it was necessary to start simulation from appropriate initial conditions (although rather general). Besides, learning required a long time. Here we have introduced a learning rule and proved its exponentially fast (and thus robust) convergence independently of the initial conditions (Theorem 1). We note that the learning rule does not depend on the learner's state variable. Thus, during the learning an explicit motor implementation of the behavior is not necessary. Therefore, the agent can simply "observe" the teacher without moving itself. Such a silent or "mental" learning can be observed in humans and in artificial cognitive agents [36].

The learning of the teacher's graph (task 1) is based on the dynamic evaluation of the fitness of interneuron couplings over the learner's trajectory. We have provided a rigorous approach (Theorem 2) that enables network rewiring "on the fly." Such a rewiring requires at most $(n-1)$ iterations of the algorithm for its complete convergence (Theorem 3). Thus, the learning time grows only linearly with the number of motor motifs. In a numerical simulation with networks of $n = 13$ neurons we observed that the learner could find the teacher's pathway matrix (one out of 4.79×10^8 possible) in five steps only.

Thus, the provided approach enables fast learning in social-like situations. To validate it, we tested the algorithm on mobile robots. First, we implemented six motor motifs (go straight, turn left, etc.) as executable commands. These motifs can be considered as "genetically" programed primitives available to a roving robot. Then we implemented neural networks in two robots and designated one of them as a teacher and the other as a learner. The teacher has been programmed to reproduce a complex flower-like trajectory, while the learner was set at random. Then the learner started to learn the teacher's trajectory. At the beginning its movements were quite chaotic. However, after few cycles the learner has successfully "copied" the teacher's behavior. To confirm quantitatively the learning behavior, we have proposed a special metric, invariant with respect to translational and rotational symmetries and to a time lag in the execution of trajectories. We have shown that the distance between the teacher's trajectory and the learner's behavior decreases in time.

In conclusion, the reported mechanism of learning is quite general and can be used for replication of different behaviors on different platforms. Moreover, the behaviors to be replicated need not be motor. For instance, one can also think about replication of sound patterns or speech. It may also serve as a linker, connecting different scenarios and behaviors during cognitive navigation [37].

ACKNOWLEDGMENTS

This work has been supported by the Russian Science Foundation under Project No. 15-12-10018 (the problem statement and theoretical development) and by the Spanish Ministry of Economy and Competitiveness under Grant No. FIS2014-57090-P.

APPENDIX A: PROOF OF THEOREM 1

First, let us recall that the common operation of multiplication of matrices has a higher priority then the Hadamard operations. Now deriving (9) and using Eqs. (3) and (10) we obtain

$$\dot{\gamma} = -W^T(x \odot (\rho_\gamma - \rho_\alpha)x). \tag{A1}$$

Then we observe that $(\rho_\gamma - \rho_\alpha)x = W((\gamma - \alpha) \odot x)$. Substituting this to (A1) we get

$$\dot{\gamma} = -W^T x \odot (\gamma - \alpha) \odot x. \tag{A2}$$

Therefore,

$$\frac{d(\gamma - \alpha)}{dt} = -p \odot (\gamma - \alpha), \tag{A3}$$

where $p \in \mathcal{C}(\mathbb{R},[0,1]^n)$, given by (11), is a function of time completely defined by the dynamics of the teacher. Integrating (A3) we obtain

$$\gamma(t) - \alpha = (\gamma_0 - \alpha) \odot e^{-\int_0^t p(\tau)\,d\tau}. \tag{A4}$$

Since $x(t) = 0$ is not a solution of (3) and hence $x(t) > 0$ under a WLC dynamics, there exists $\kappa > 0$ such that $p(t) \geqslant \kappa$, which ensures the exponential convergence: $\|\gamma(t) - \alpha\| \leqslant \|\gamma_0 - \alpha\| e^{-\kappa t}$. If $x(t)$ is periodic, then a lower bound for κ is given by (12). ■

APPENDIX B: PROOF OF THEOREM 2

1. Preliminary results

For further calculations it is convenient to introduce two constants:

$$A = f(T) \in (0,1)^n, \quad B = g(T) \in \mathbb{R}^n, \tag{B1}$$

where f, g are given by (18), and T is the oscillation period of the teacher's pattern. These constants A and B are completely determined by the teacher's dynamics and the pathway matrix of the learner, W_y.

Lemma 1. Let T be the period of the teacher's pattern $x(t+T) = x(t)$. Then, under the learning rule (15) and (16), we have

$$\begin{aligned}\gamma(t+kT) = \gamma(t) - (1_n - A^{\circ k}) \odot (\gamma_0 \\ - A \odot B \oslash (1_n - A)) \odot f(t).\end{aligned} \tag{B2}$$

Note that $(\cdot)^{\circ k}$ in (B2) stands for the Hadamard kth power.

Proof. We notice that $p(t)$ [see Eq. (11)] is T-periodic and thus

$$\int_0^{t+kT} p(\tau)\,d\tau = k\int_0^T p(\tau)\,\mathrm{d}\tau + \int_0^t p(\tau)\,d\tau.$$

Therefore,

$$f(t+kT) = A^{\circ k} \odot f(t). \tag{B3}$$

Then $q(t)$ [see Eq. (16)] is also T-periodic, which together with (B3) yields

$$\begin{aligned} & g(t+kT) \\ & = \sum_{l=0}^{k-1} \int_{lT}^{(l+1)T} q(\tau) \oslash f(\tau)\,d\tau + \int_{kT}^{kT+t} q(\tau) \oslash f(\tau)\,d\tau \\ & = B \odot \sum_{l=0}^{k-1} A^{\circ -l} + A^{\circ -k} \odot \int_0^t q(\tau) \oslash f(\tau)\,d\tau. \end{aligned} \tag{B4}$$

Now by substituting (B3) and (B4) into (17) we get

$$\begin{aligned} \gamma(t+kT) &= \gamma_0 \odot A^{\circ k} \odot f(t) + B \odot \sum_{l=1}^{k} A^{\circ l} \odot f(t) + \gamma(t) \\ & \quad -\gamma_0 \odot f(t) = \gamma(t) \\ & \quad -\left[(1_n - A^{\circ k}) \odot \gamma_0 - B \odot \sum_{l=1}^{k} A^{\circ l}\right] \odot f(t). \end{aligned} \tag{B5}$$

Finally, since $A \in (0,1)^n$ we can evaluate the geometric series:

$$\sum_{l=1}^{k} A^{\circ l} = A \odot (1_n - A^{\circ k}) \oslash (1_n - A), \tag{B6}$$

which yields (B2). ■

Corollary 1. Given the sequence

$$\{\gamma(t+kT) : 0 \leqslant t < T\}_{k=1}^{\infty} \subset \mathcal{C}([0,T),\mathbb{R}^n),$$

there exists a constant vector $C \in \mathbb{R}^n$ such that

$$\lim_{k\to\infty} \gamma(t+kT) = \gamma(t) - C \odot f(t) \quad \forall t \in [0,T). \tag{B7}$$

Proof. Since $A \in (0,1)^n$, $\lim_{k\to\infty} A^{\circ k} = 0$. Then denoting $C = \gamma_0 - A \odot B \oslash (1_n - A)$ and applying Lemma 1 we arrive to (B7). ■

2. Proof of the theorem

On the one hand, given that $(j,i) \in E_x \cap E_y$ and taking into account Eq. (A4) we get

$$\gamma_j(t+kT) = \alpha_j + (\gamma_j(0) - \alpha_j) f_j(t+kT), \tag{B8}$$

and thus

$$\lim_{k\to\infty} \gamma_j(t+kT) = \alpha_j. \tag{B9}$$

Then by Corollary 1 there exists $C_j \in \mathbb{R}$ such that

$$\gamma_j(t) = \lim_{k\to\infty} \gamma_j(t+kT) + C_j f_j(t) = \alpha_j + C_j f_j(t). \tag{B10}$$

Thus, by choosing $\delta_j^* = (\alpha_j, C_j)^T$ we obtain $e_j(\delta_j^*) = 0$. This δ_j^* matches the obtained from (21).

On the other hand, if $e_j(\delta_j^*) = 0$ then

$$\gamma_j(t) = \delta_{j1}^* + \delta_{j2}^* f_j(t) \;\Rightarrow\; \dot{\gamma}_j(t) = \delta_{j2}\dot{f}_j(t). \tag{B11}$$

Now, we notice that $\delta_{j2}^* \dot{f}_j = -p_j \delta_{j2}^* f_j = -p_j(\gamma_j - \delta_{j1}^*)$ and thus

$$\dot{\gamma}_j(t) = -p_j(t)\gamma_j(t) + q_j(t), \tag{B12}$$

where $q_j(t) = p_j(t)\delta_{j1}^*$. Let $l \in V$ be the predecessor of $i \in V$, i.e., $(l,i) \in E_x$. From Eq. (16) we get

$$q_j(t) = 2p_j(t) + (\alpha_l - 2)x_i(t)x_l(t). \tag{B13}$$

Now comparing the expressions for q_j we get

$$x_l(t) = \frac{\delta_{j1}^* - 2}{\alpha_l - 2} x_j(t), \tag{B14}$$

which yields $l = j$, $\delta_{j1}^* = \alpha_j$ and thus the i-th rows of the pathway matrices W_x and W_y are the same. ■

APPENDIX C: PROOF OF THEOREM 3

Let us define, for each vertex $j \in V$, the cumulative set of tested successors up to step $k > 0$:

$$S_k(j) = \cup_{l=1}^{k} \{\sigma_l(j)\}. \tag{C1}$$

We note that $\sigma(j)$ changes (item 6 of the algorithm) ensuring an exhaustive sequential search due to the rule $\sigma(w_i) \leftarrow \sigma(w_{i+1})$ and the ascending order established for Ω (item 5 of the algorithm). If at step k the tested successor matches the teacher's network (item 4 of the algorithm), then the jth node is excluded from Ω at step $k+1$. Therefore, $S_{k+1}(j)$ stops increasing. Thus, we have

$$j \in \Omega_k \Leftrightarrow |S_{k+1}(j)| = k + 1. \tag{C2}$$

Now we notice that $S_k(j) \subset V \setminus \{j\}$ and hence the cardinality of the cumulative sets satisfies the inequality

$$|S_k(j)| \leqslant |V \setminus \{j\}| = n - 1, \quad k = 1,2,\ldots. \tag{C3}$$

Using (C2) and (C3) we thus obtain that $\forall j \in V$: $j \in \Omega_k$ iff $k < n - 1$. Therefore, $\Omega_{n-1} = \emptyset$, i.e., the iterative learning process converges at most in $n - 1$ steps. ■

[1] W. Hoppitt and K. N. Laland, *Social Learning: An Introduction to Mechanisms, Methods, and Models* (Princeton University Press, Princeton, 2013).

[2] V. Gallesea and A. Goldmanb, Mirror neurons and the simulation theory of mind-reading, Trend. Cogn. Sci. **4**, 252 (2000).

[3] R. Cook, G. Bird, C. Catmur, C. Press, and C. Heyes, Mirror neurons: From origin to function, Behav. Brain Sci. **37**, 177 (2014).

[4] S. Schaal, Is imitation learning the route to humanoid robots? Trend. Cogn. Sci. **3**, 233 (1999).

[5] C. Breazeal, D. Buchsbaum, J. Gray, D. Gatenby, and B. Blumberg, Learning from and about others: Towards using imitation to bootstrap the social understanding of others by robots, Artif. Life **11**, 31 (2005).

[6] E. Broadbent, R. Stafford, and B. MacDonald, Acceptance of healthcare robots for the older population: Review and future directions, Int. J. Soc. Robot. **1**, 319 (2009).

[7] U. Alon, Network motifs: theory and experimental approaches, Nat. Rev. Genet. **8**, 450 (2007).

[8] F. Stulp, E. Oztop, P. Pastor, M. Beetz, and S. Schaal, Compact models of motor primitive variations for predictable reaching and obstacle avoidance, in *Proc. Conf. IEEE-RAS on Humanoid Robots* (IEEE, 2009), pp. 589–595.

[9] A. Colome and C. Torras, Dimensionality reduction and motion coordination in learning trajectories with dynamic movement primitives, in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.* (IEEE, 2014), pp. 1414–1420.

[10] F. Meier, E. Theodorou, F. Stulp, and S. Schaal, Movement segmentation using a primitive library, in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.* (IEEE, 2011), pp. 3407–3412.

[11] J. A. Villacorta-Atienza, C. Calvo, S. Lobov, and V. A. Makarov, Limb movement in dynamic situations based on generalized cognitive maps, Math. Model. Nat. Phenom. **12**, 15 (2017).

[12] S. Schaal, P. Mohajerian, and A. Ijspeert, Dynamics systems vs. optimal control—A unifying view, Progr. Brain Res. **165**, 425 (2007).

[13] T. Matsubara, S. Hyon, and J. Morimoto, Learning parametric dynamic movement primitives from multiple demonstrations, Neur. Netw. **24**, 493 (2011).

[14] Y. Tassa, T. Erez, and E. Todorov, Synthesis and stabilization of complex behaviors through online trajectory optimization, in *IEEE/RSJ Int. Conf. Intell Robots Syst.* (IEEE, 2012), pp. 4906–4913.

[15] A. R. Ansari and T. D. Murphey, Sequential action control: Closed-form optimal control for nonlinear and nonsmooth systems, IEEE Trans. Robot. **32**, 1196 (2016).

[16] A. Selskii and V. Makarov, Synchronization of heteroclinic circuits through learning in coupled neural networks, Regular Chaotic Dyn. **21**, 97 (2016).

[17] M. Rabinovich, A. Volkovskii, P. Lecanda, R. Huerta, H. Abarbanel, and G. Laurent, Dynamical Encoding by Networks of Competing Neuron Groups: Winnerless Competition, Phys. Rev. Lett. **87**, 068102 (2001).

[18] N. Benito, G. Martin-Vazquez, J. Makarova, V. Makarov, and O. Herreras, The right hippocampus leads the bilateral integration of gamma-parsed lateralized information, Elife **5**, e16658 (2016).

[19] G. Martin-Vazquez, N. Benito, V. A. Makarov, O. Herreras, and J. Makarova, Diversity of LFPs activated in different target regions by a common CA3 input, Cereb. Cortex **26**, 4082 (2016).

[20] H. Sompolinsky and I. Kanter, Temporal Association in Asymmetric Neural Networks, Phys. Rev. Lett. **57**, 2861 (1986).

[21] I. Tyukin, A. Gorban, C. Calvo, J. Makarova, and V. A. Makarov, High-dimensional brain: A tool for encoding and rapid learning of memories by single neurons, Bull. Math. Biol. (2018), doi: 10.1007/s11538-018-0415-5.

[22] E. Del Rio, V. A. Makarov, M. G. Velarde, and W. Ebeling, Mode transitions and wave propagation in a driven-dissipative Toda-Rayleigh ring, Phys. Rev. E **67**, 056208 (2003).

[23] S. A. Lobov, M. O. Zhuravlev, V. A. Makarov, and V. B. Kazantsev, Noise enhanced signaling in STDP driven spiking-neuron network, Math. Model. Nat. Phenom. **12**, 109 (2017).

[24] M. A. Cohen and S. Grossberg, Absolute stability of global pattern formation and parallel memory storage by competitive neural networks, IEEE Tran. Syst. Man Cyber. **13**, 815 (1983).

[25] P. Varona, M. Rabinovich, A. Selverston, and Y. Arshavsky, Winnerless competition between sensory neurons generates chaos: A possible mechanism for molluscan hunting behavior, Chaos **12**, 672 (2002).

[26] V. S. Afraimovich, V. P. Zhigulin, and M. I. Rabinovich, On the origin of reproducible sequential activity in neural circuits, Chaos **14**, 1123 (2004).

[27] F. Hadaeghi, M. Reza, H. Golpayegani, and G. Murray, Towards a complex system understanding of bipolar disorder: A map based model of a complex winnerless competition, J. Theor. Biol. **376**, 74 (2015).

[28] T. Rost, M. Deger, and M. Nawrot, Winnerless competition in clustered balanced networks: Inhibitory assemblies do the trick, Biol. Cybern. **112**, 81 (2018).

[29] P. Arena, L. Fortuna, D. Lombardo, L. Patane, and M. G. Velarde, The winnerless competition paradigm in cellular nonlinear networks: Models and applications, Int. J. Circ. Theor. Appl. **37**, 505 (2009).

[30] C. Calvo, J. Villacorta-Atienza, V. I. Mironov, V. Gallego, and V. A. Makarov, Waves in isotropic totalistic cellular automata: Application to real-time robot navigation, Adv. Compl. Syst. **19**, 1650012 (2016).

[31] V. Afraimovich, M. Rabinovich, and P. Varona, Heteroclinic contours in neural ensembles and the winnerless competition principle, Int. J. Bifurc. Chaos **14**, 1195 (2004).

[32] I. Yu Tyukin, D. V. Prokhorov, and C. van Leeuwen, Adaptation and parameter estimation in systems with unstable target dynamics and nonlinear parametrization. IEEE Trans. Autom. Contr. **52**, 1543 (2007).

[33] I. Tyukin, *Adaptation in Dynamical Systems* (Cambridge University Press, Cambridge, 2011).

[34] A. E. Hramov, A. A. Koronovskii, V. A. Makarov, A. N. Pavlov, and E. Sitnikova, *Wavelets in Neuroscience* (Springer, Berlin, Heidelberg, 2015).

[35] V. A. Makarov, C. Calvo, V. Gallego, and A. Selskii, Synchronization of heteroclinic circuits through learning in chains of neural motifs, IFAC-PapersOnLine **49**, 80 (2016).

[36] J. Villacorta-Atienza and V. A. Makarov, Neural network architecture for cognitive navigation in dynamic environments, IEEE Trans. Neur. Netw. Learn. Syst. **24**, 2075 (2013).

[37] J. Villacorta-Atienza, C. Calvo, and V. A. Makarov, Prediction-for-CompAction: Navigation in social environments using generalized cognitive maps, Biol. Cybern. **109**, 307 (2015).