# Burst-Dependent Plasticity and Dendritic Amplification Support Target-Based Learning and Hierarchical Imitation Learning

**Cristiano Capone** [*1] **Cosimo Lupo** [*1] **Paolo Muratore** [2] **Pier Stanislao Paolucci** [1]

## Abstract

The brain can learn to solve a wide range of tasks with high temporal and energetic efficiency. However, most biological models are composed of simple single-compartment neurons and cannot achieve the state-of-the-art performances of artificial intelligence. We propose a multi-compartment model of pyramidal neuron, in which bursts and dendritic input segregation give the possibility to plausibly support a biological target-based learning. In target-based learning, the internal solution of a problem (a spatio-temporal pattern of bursts in our case) is suggested to the network, bypassing the problems of error backpropagation and credit assignment. Finally, we show that this neuronal architecture naturally supports the orchestration of "hierarchical imitation learning", enabling the decomposition of challenging long-horizon decision-making tasks into simpler subtasks.

## 1. Introduction

The brain can learn a wide range of tasks very efficiently in terms of energy consumption and required evidences, motivating the search for biologically inspired learning rules for improving the efficiency of artificial intelligence. Most biologically plausible neural networks are composed so far of point neurons. Despite recent outstanding advances in this field (Nicola & Clopath, 2017; Bellec et al., 2020), biologically plausible neural networks cannot achieve the state-of-the-art performances of artificial intelligence (e.g., they struggle to solve the credit assignment problem (Payeur et al., 2021)).

Recent findings on dendritic computational properties (Poirazi & Papoutsi, 2020) and on the complexity of pyramidal neurons dynamics (Larkum, 2013) motivated the study of multicompartment neuron models in the development of new biologically plausible learning rules (Urbanczik & Senn, 2014; Guerguiev et al., 2017; Sacramento et al., 2018; Payeur et al., 2021).

In addition, it has been proposed that segregation of dendritic input (i.e. neurons receive sensory information and higher-order feedback in segregated compartments) (Guerguiev et al., 2017) and generation of high-frequency bursts of spikes (Payeur et al., 2021) would support backpropagation in biological neurons. However, these approaches require propagating errors with a fine spatio-temporal structure to all the neurons. It is not clear whether this is possible in biological networks. For this reason, in the last few years, target-based approaches (Lee et al., 2015; DePasquale et al., 2018; Manchev & Spratling, 2020; Meulemans et al., 2020; Muratore et al., 2021) started to gain more and more interest.

In a target-based learning framework, the targets, rather than the errors, are propagated through the network (Lee et al., 2015; Manchev & Spratling, 2020). In this framework, it is possible to directly suggest to the network the internal solution to a task (DePasquale et al., 2018; Muratore et al., 2021; Capone et al., 2022). However, target-based approaches require evaluating at the same time the spontaneous activity and the target activity of the network (DePasquale et al., 2018; Muratore et al., 2021). This is usually solved by evaluating the two activities in two different networks, which is not natural in terms of biological plausibility.

In the present work, we show that bursts and dendritic input segregation offer a natural solution to this dilemma. In our model, pyramidal neurons rely on two different apical dendritic compartments to simultaneously evaluate the target and the spontaneous activity. A coincidence mechanism between basal and apical inputs generating the burst (Larkum, 2013) eventually defines the (target or spontaneous) spatio-temporal bursting dynamics of the network.

We exploit dendritic computation in our model, to let arbitrary signals act as teaching signals which drive the learning procedure in a biologically plausible fashion. This allows to flexibly store and recall arbitrary trajectories.

[*]Equal contribution [1]INFN, Sezione di Roma, Rome, Italy [2]SISSA, International School for Advanced Studies, Trieste, Italy. Correspondence to: Cristiano Capone <cristiano0capone@gmail.com>.

*Proceedings of the $39^{th}$ International Conference on Machine Learning*, Baltimore, Maryland, USA, PMLR 162, 2022. 

Finally, we show that this neuronal architecture naturally allows for orchestrating *hierarchical imitation learning*, enabling the decomposition of challenging long-horizon decision-making tasks into simpler subtasks (Le et al., 2018; Pateria et al., 2021).

## 2. Results

### 2.1. Target-based learning with bursts

We define a model of pyramidal neuron (Figure 1A, bottom) composed of three separated compartments, the basal one (i.e. the soma, receiving the sensorial input), and two apical ones, the proximal apical compartment (receiving recurrent connections from the network) and the distal apical compartment (receiving the context/teaching signal from other areas of the cortex, with a higher level of abstraction).

The spike emitted by the soma of the $i$-th neuron is described by variable $z_i^t$, which is equal to 1 when the spike is emitted at time $t$ and 0 otherwise. The spikes emitted by the proximal and distal apical compartments are described by variables $a_i^t$ and $a_i^{\star,t}$, respectively. The underlying idea is that the distal compartment provides a target for the proximal one, motivating the use of the superscript symbol $\star$, which indicates the variables concerning the target.

Following (Larkum, 2013) a coincidence mechanism between the basal and the apical compartments has been implemented, yielding high-frequency bursts of spikes. In more detail, after a somatic spike, $z_i^t = 1$, a coincidence window is opened for a time interval $\Delta T$. This is described by the variable $\overline{z}_i^t$, the indicator function for $t' \in [t, t+\Delta T]$, which is 1 during this time window and 0 elsewhere. If a spike is generated by the distal or proximal apical compartment within such time window, $a_i^{t'} = 1$ or $a_i^{\star,t'} = 1$, with $t' \in [t, t+\Delta T]$, a high-frequency burst of spikes is then produced (Figure 1B). The proximal and distal burst variables can be respectively defined as

$$B_i^{t+1} = \overline{z}_i^t \, a_i^{t+1}$$
$$B_i^{\star,t+1} = \overline{z}_i^t \, a_i^{\star,t+1}$$

This architecture supports a burst-dependent learning rule (Figure 1A, top), enabling target-based learning. More specifically, the pattern of bursts defined by the proximal compartment (receiving the recurrent connections from the network) should mimic the one defined by the distal compartment (which receives the teaching signal). This is made possible by using the following plasticity rule for recurrent weights $J_{ij}^{b\to p}$ (which can be derived analytically through a likelihood maximization, see methods for details):

$$\Delta J_{ij}^{b\to p} = \eta \left[ a_i^{\star,t+1} - a_i^{t+1} \right] \overline{z}_i^t \, e_j^t \qquad (1)$$

where $e_j^t = \partial u_i^t / \partial J_{ij}^{b\to p}$ is referred to in the literature as the *spike response function* (Urbanczik & Senn, 2014).

Intuitively, such plasticity rule aims at aligning in time apical proximal spikes with apical distal ones when the somatic window $\overline{z}_i^t$ is open. We remark that such a learning rule can be computed online, and only requires observables which are locally accessible to the synapses in space and time.

As a first learning instance, we propose the store and recall of a 3D trajectory $y_k^{\star,t}$ ($k = 1, \dots, 3$; $t = 1, \dots, T$; $T = 1000$) in a network of $N = 500$ neurons (400 bursting neurons with the pyramidal architecture described above, plus 100 non-bursting point neurons). We chose $y_k^{\star,t}$ as a temporal pattern composed of 3 independent continuous signals, each of which specified as the superposition of the four frequencies $f \in \{1, 2, 3, 5\}$ Hz with uniformly extracted random amplitudes $A \in [0.5, 2.0]$, and phases $\phi \in [0, 2\pi]$:

$$y_k^{\star,t} = \sum_{n=1}^{4} A_{k,n} \cos\left(2\pi f_{k,n} t + \phi_{k,n}\right), \quad k = 1, 2, 3$$

This target trajectory is randomly projected through a Gaussian matrix with variance $\sigma_{\text{targ}}^2$ to the (distal) apical dendrites of the network as a teaching signal. This input shapes the spatio-temporal pattern of spikes $a_i^{\star,t}$ from the distal apical compartment (Figure 1C, bottom, orange points), as well as the related target spatio-temporal pattern of bursts $B_i^{\star,t}$ (Figure 1C, bottom, blue points) as described above.

A clock signal serving as a sensorial input is randomly projected (through a Gaussian matrix with variance $\sigma_{\text{in}}^2$) to the somatic dendrites. In more detail, the clock is here modeled as a sort of time step function with $I$ steps, such that at each time $t$ only component $i = \lfloor I \cdot t/T \rfloor$ is equal to one, while others are zero (see Table 1 for model parameters).

Learning is numerically implemented by several presentations of the same target trajectory $y^\star$ to the distal apical compartments, each time adjusting recurrent weights $J_{ij}^{b\to p}$ according to Eq. (1).

The internal bursting is translated into the output $y$ by means of a read-out matrix $J_{\text{out}}$, randomly initialized and to be trained following the rule derived by minimizing the mean squared error (mse) between the target output and the network output:

$$\Delta J_{ki}^{\text{out}} = \eta_{\text{out}} \left[ y_k^{\star,t} - \sum_h J_{kh}^{\text{out}} \hat{B}_h^t \right] \hat{B}_i^t \qquad (2)$$

where $\hat{B}_i^t$ is a time-smoothed version of burst variable $B_i^t$ (see methods for details).

At the end of the learning the plasticity of recurrent connections allows for a reliable reproduction of the target 3D trajectory (see Figure 1C, top, $\text{mse} = 0.01$), with an inter-

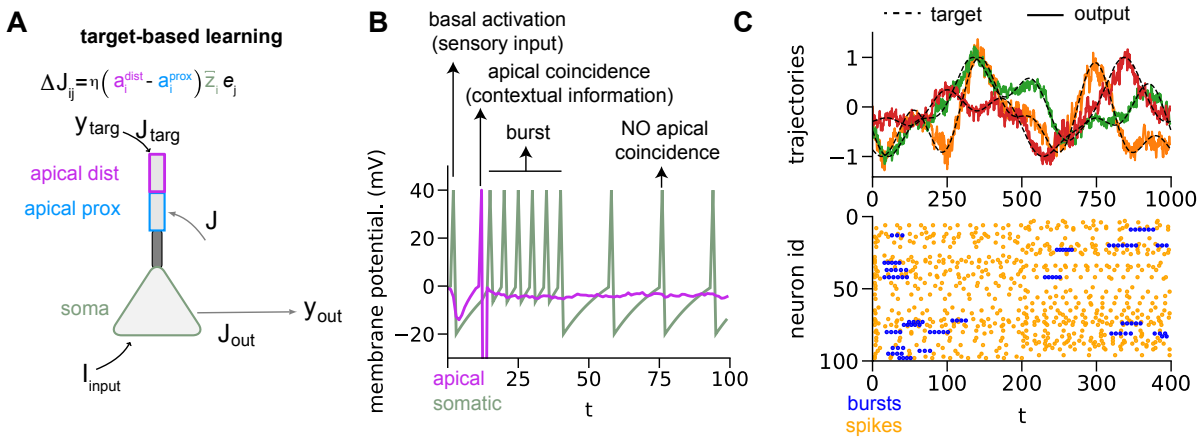


*Figure 1.* **Model structure. A**. The model of a pyramidal neuron, consisting of two separated compartments, the basal and the apical ones. The latter is further divided into two regions, proximal (receiving recurrent connections from the network) and distal (receiving teaching/context signals from other areas of the cortex). (top) Plasticity rule for the recurrent weights of the network. **B**. In addition to isolated spike signals emitted by the soma, a coincidence mechanism between basal and apical compartments allows for the generation of high-frequency bursts of spikes. **C**. Store and recall of a 3D trajectory. The target output is automatically encoded into a spatio-temporal pattern of bursts (bottom panel), learned online thanks to the plasticity of recurrent connections, allowing for reliable reproduction of the target trajectory (top panel).

nal bursting activity reproducing the target one (Figure 1C, bottom).

The same task is addressed in (Muratore et al., 2021) and (Bellec et al., 2020), obtaining mse values of $0.001$ and $0.01$, respectively. Though the latter result is very similar with the present one (approximately $0.01$, averaged over 5 realizations), a direct comparison is unfair, since the target is here encoded only through the bursts, that are way less than spikes, so providing a much sparser encoding. On the other hand, our model results in a remarkable improvement in terms of biological plausibility.

### 2.2. Apical signals as a flexible context selection

The distal apical compartment is designed not only to receive teaching signals, but also contextual information from other areas of the cortex, acting as a hint for the task to address. With this idea in mind, in this section we show that it is possible to exploit different context signals (projected through a Gaussian random matrix with variance $\sigma^2_{\text{cont}}$) to flexibly select and recall one of the trajectories previously stored in the network.

In the simplest configuration, two different contexts, A and B, can be modeled through 2D time-constant binary signals projected on the distal apical compartment, $\boldsymbol{\chi}_{(1)} = (1, 0)$ for A and $\boldsymbol{\chi}_{(2)} = (0, 1)$ for B (Figure 2A).

During the training, each context is associated with a well defined target to learn (again a 3D trajectory, as defined in the previous section). In Figure 2B, left side, they are reported in red and black, respectively (only one of the three trajectories for each target is reported, for simplicity); same color-coding is used for associated context signals. To stabilize the learning, we exploited the trick of halving the learning rates $\eta$ and $\eta_{\text{out}}$ every $100$ training iterations. The orthogonality of the contexts and related targets is further stressed by imposing a sparsification (of $75\%$ in the present case) in the random matrices we use to project the context and the target on the apical compartments of the network.

During the recall phase, the teacher signal is no longer present, while the context signal suggests to the network which of the learned trajectory to reproduce. We show that when the context is projected to the network, the desired output is correctly recalled (Figure 2B, left side). Moreover, if the context signal is turned off in the middle of the trajectory, the network is still able to self-sustain its inner dynamics, thanks to recurrent connections (Figure 2B, left side), and correctly replicate the selected trajectory.

Hence, the context works here as a "suggestion", so that once started the reproduction of the correct output trajectory, the context itself becomes useless.

To demonstrate the importance to project the context signal in the apical compartments, we compare these results with the case in which the context is projected in the basal ones (both during the training and the retrieval phases). In this case, the desired trajectory is correctly retrieved only when the context is on (Figure 2B, right side).

However, we observe that the basal context is interpreted as a necessary input, so that after the turnoff the network is no longer able to sustain bursts creation, in turn causing

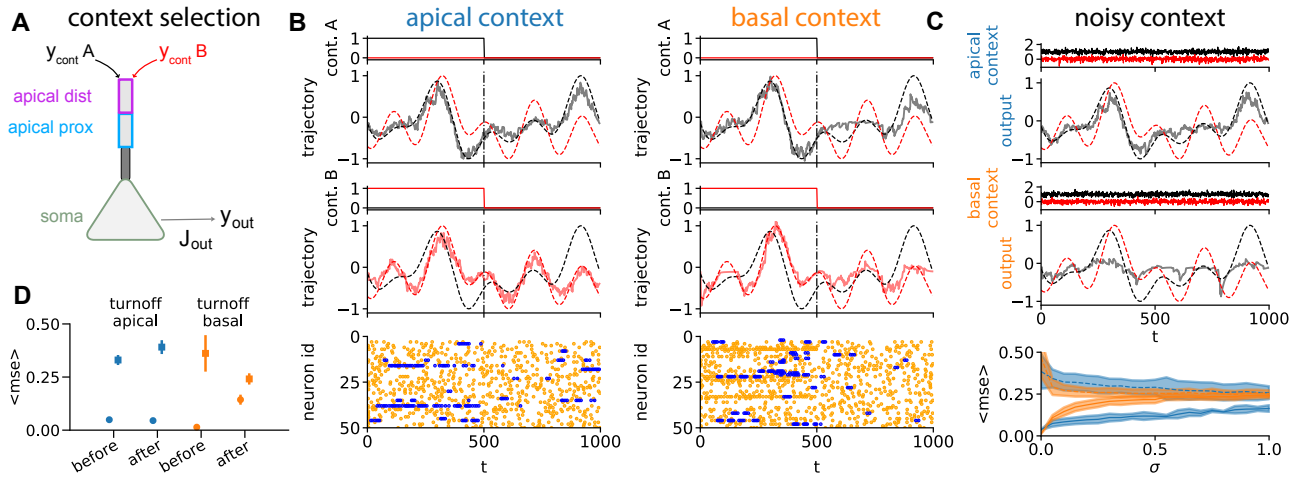


*Figure 2.* **Apical signals for dynamics selection. A**. Model of pyramidal neuron where a binary context signal (A or B) is projected on the apical distal compartment. The target to be reproduced by the network changes according to which context is active. **B**. (left side) The network is able to reproduce the correct output trajectory even if the context is provided only in the first time steps. (right side) An alternative model in which the context is projected on the basal compartment is no longer able to reproduce the correct output trajectory. **C**. (top) The trajectory produced by the network, in presence of noisy apical context A ($\sigma = 0.2$, black solid line), is similar to the trajectory targeted by the context A (black dashed line) and different from the trajectory targeted by the context B (red dashed line). Inset, the noisy context signal. (middle) The trajectory produced by the network, in presence of noisy basal context A ($\sigma = 0.2$, black solid line), is not similar to the trajectory targeted by the context A (black dashed line). Inset, the noisy context signal. (bottom) Average performances of the apical/basal (blue/orange) context as a function of the noise standard deviation $\sigma$. Solid lines: mse between the output and the target output. Dashed lines: mse between the output and the trajectory targeted by the other context. Averages and error bars are intended over 10 independent network/target realizations. **D**. Summary of performances of the two model versions (context projected on apical vs basal compartment) during "turnoff" test in the middle of the trajectory. Mean squared error in the second part of the trajectory (no context) is compared with respect to error in the first part (context still active); mean and variance are intended over 10 independent network/target realizations (round markers). mse between the output and the trajectory targeted by the other context is also reported, as a reference (square markers).

a dramatic drop in the test performances (Figure 2B, right side). Average mean squared errors, measured against both the correct target trajectory and the wrong one (i.e. the one corresponding to the other context signal), both before turnoff and after it, are provided in Figure 2D for the two neural architectures.

Furthermore, apical context architecture is also robust against corruption in the context signal, which may be the case when at higher cortical levels there is only a mild preference in favor of which strategy to adopt (in comparison with the training phase, where each target is clearly and univocally associated with a sharp context signal). Here a Gaussian white noise of variance $\sigma^2$ is added during test to context signals exploited in the training (Figure 2C, top panel, $\sigma = 0.2$). The produced trajectory is similar to the trajectory targeted by the context A (black dashed line) and different from the trajectory targeted by the context B (red dashed line). In Figure 2C, bottom panel (blue lines) it is reported the average mse (average over 10 independent realizations of the experiment) between the output and the target trajectory (solid blue line) as a function of $\sigma$. As a reference, we also report the mse between the output and the trajectory targeted by the other context signal (dashed blue line).

It is evident a resilience of the network with apical context, while the network with basal context suddenly loses the ability to reproduce the desired output already at low levels of noise (Figure 2C, middle panel and orange lines in bottom panel).

At higher levels of noise, basal-context network becomes in practice useless, while apical-context network is still able to reproduce the target trajectory with a remarkably small error (Figure 2C, bottom panel).

### 2.3. Hierarchical Imitation Learning

The proof that context can be used to flexibly choose which dynamics reproduce (and when), opens the pathway to more complicated neural architectures, naturally supporting *hierarchical imitation learning* (HIL). To our knowledge, no prior works are proposing biologically plausible implementations of hierarchical reinforcement or imitation learning.

We decomposed the network in two sub-networks, named

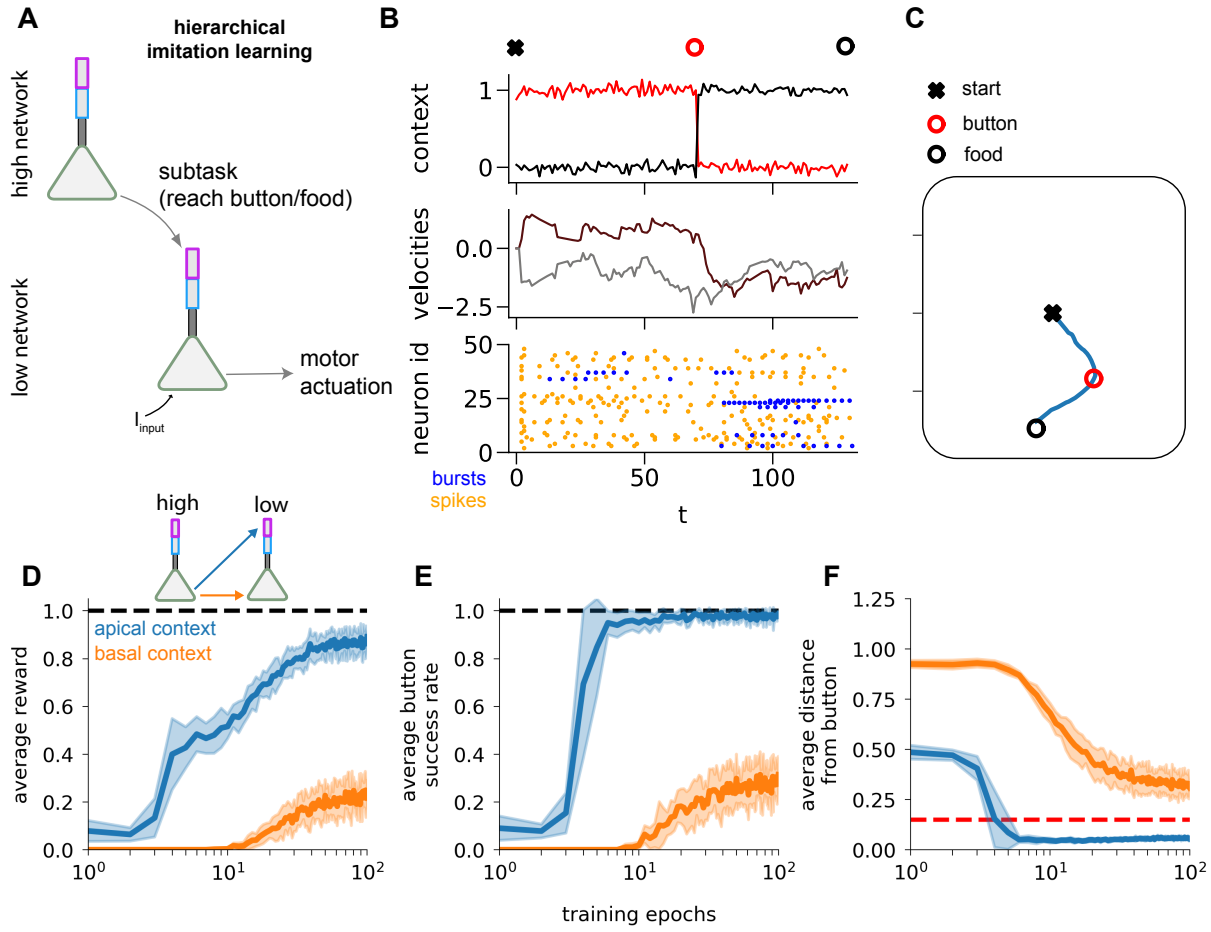


*Figure 3.* **Hierarchical Imitation Learning. A**. A two-level network, where high-level neurons produce a signal that serves as a context for the neurons in the low-level network. The two subnetworks received two different but synchronized teaching signals in the training phase. **B**. Button-and-food task, an agent placed at an initial position (black cross) in a 2D maze has to first reach a button (red circle) so to unlock the food (black circle) and then reach for it. The high-level network chooses the order of the two subtasks (reach_button and reach_food) and when to switch from one to the other. It projects the instruction as a contextual signal (top panel) to the apical compartments of the low-level network. The low-level network produces the output (velocities of the agent, center panel) necessary to solve the subtask as a read-out of its internal bursting activity (bottom panel, blue dots; orange dots represent the spiking activity). **C**. A sample spatial trajectory. Cross, red and black circles as in panel B. **D**. Reward as a function of training epochs (average and standard error over 10 realizations, in lines and shadings respectively). Blue and orange colors refer to the two different choices for the context projection on the low-network: apical or basal compartments, respectively (see also inset for a sketch of the model). Black dashed line at 1.0 indicates the maximum possible reward. **E**. Average success rate for pushing the button as a function of the training epochs. 1.0 is again the maximum possible value. Same color coding as in D. **F**. Average distance from the food at the end of the episode. The red dashed line represents the button size. Same color coding as in D.

*high-network* and *low-network* (Figure 3A). The high-network (the "manager") computes the optimal strategy to adopt in order to solve a task, and sends this information as a context signal to the low-network (the "worker"), which actually executes it.

We applied this strategy to the so-called *button & food* task. Here, an agent starts at the center of a square domain, which also features a button and an initially locked target (the "food"). The goal of the agent is to first press the button so to unlock the food, then reach for it. Both button and food positions are uniformly extracted in the domain $[0, 1] \times [0, 1]$.

The global task is naturally decomposed into two simpler sub-tasks (or goals): reach_button and reach_food. The high-network computes which goal to pursue and when, and the low-network implements the sub-policy to achieve the goal. Both the high- and the low-network share the same input ($I = 80$ input units), the vertical and horizontal differences of both button and food positions with respect to

the agent location ($\Delta^t = \{\Delta x_b^t, \Delta y_b^t, \Delta x_f^t, \Delta y_f^t\}$ respectively). Each of the $\Delta_i$ values is encoded by 20 input units with different Gaussian activation functions.

To perform learning, we consider a natural hierarchical extension of behavioral cloning. The expert provides a set of hierarchical demonstrations, each consisting of low-level trajectories (to be cloned by the low-network):

$$\left\{\left(\text{state}_{\text{L}}^t,\, \text{action}_{\text{L}}^t,\, \text{goal}_{\text{L}}^t\right)\right\}_{t=1}^{T},$$

as well as a high-level trajectory (to be cloned by the high-network):

$$\left\{\left(\text{state}_{\text{H}}^t,\, \text{action}_{\text{H}}^t\right)\right\}_{t=1}^{T}.$$

Both $\text{state}_{\text{L}}^t$ and $\text{state}_{\text{H}}^t$ are the input $\Delta^t$ described above. The $\text{action}_{\text{H}}^t$ is the target output of the high-network and the $\text{goal}_{\text{L}}^t$ of the low-network. It is projected to the low-network as a contextual signal in the distal apical compartment (Figure 3B, top) and is defined as a 2D binary signal:

$$\boldsymbol{y}_{\text{H}}^{\star,t} = \boldsymbol{\chi}_{(1)}\Theta\left(t < t_{\text{🔒}}\right) + \boldsymbol{\chi}_{(2)}\Theta\left(t > t_{\text{🔒}}\right),$$

where $\boldsymbol{\chi}_{(1)} = (1,0)$ and $\boldsymbol{\chi}_{(2)} = (0,1)$, and $t_{\text{🔒}}$ is the time when the button is reached. Intuitively, it selects the reach_button sub-policy for the first part of the task and then switches to reach_target.

Given the input $\text{state}_{\text{L}}^t$ and the context $\text{goal}_{\text{L}}^t$, the low-network is tasked to reproduce as output $\text{action}_{\text{L}}^t$ the velocity vector $\boldsymbol{y}_{\text{L}}^{\star,t} = \boldsymbol{v}^t = (v_x^t, v_y^t)$, where velocity components are computed so to reach the selected target in a straight line (Figure 3B, center and Figure 3C). Both high- and low-network outputs are computed as linear read-outs of their internal bursting activities, as described in Section 2.1 (Figure 3B, bottom, for the low-network).

The cloning procedure is implemented as a supervised learning, so to let the two networks reproduce the target outputs, given the input (and the context). The learning procedure is the same as the one described in Section 2.1.

Finally, the two-layer network is tested in closed-loop in the environment described above. The performances are measured via the following quantity:

$$\rho = \frac{\Xi_{\text{🔒}}\, r_0}{\min_{t>t_{\text{🔒}}} d\left(\boldsymbol{x}_{\text{agent}}^t, \boldsymbol{x}_{\text{food}}\right)},$$

where $r_0$ is the button and food size, $\Xi_{\text{🔒}}$ is the button-state indicator variable (zero when the button is locked and one otherwise), and finally $d\left(\boldsymbol{x}_{\text{agent}}^t, \boldsymbol{x}_{\text{food}}\right)$ is the Euclidean distance between the agent and food positions at time $t$. The condition for a successful button-press (a switch from locked to unlocked state) and target-reach is taken to be $d\left(\boldsymbol{x}_{\text{agent}}^t, \boldsymbol{x}_{\text{btn/food}}\right) \leq r_0$. Note how this choice effectively prevents the apparent divergence in the expression for $\rho$ as the episode is stopped when the target is reached, finally inducing a theoretical maximum achievable score of $\rho_{\max} = 1$.

After the presentation of many randomly positioned button-food pairs, we observe that such two-level network learns to correctly and efficiently solve the button & food task, with an average final score $\rho = 0.88 \pm 0.04$ and over 70% of success rate (i.e., both button-press and target-reach conditions were met). A sample spatial trajectory produced by the network is depicted in Figure 3C. In Figure 3D, blue line, we report the average reward (over 10 independent realizations) as a function of the training epochs. Similarly, in Figures 3E-F, blue curves, it is reported respectively the success rate in pushing the button, and the minimum distance from the button.

We run an additional experiment, where the high-network output is projected to the basal compartment of the low-network (rather than to the apical one, see Figure 3D, inset). The results are reported in Figures 3D-F, orange lines. This choice leads to poor performances of the hierarchical policy ($\rho = 0.24 \pm 0.07$), demonstrating the need for a contextual signal necessarily being of a different nature with respect to somatic input signals.

## 3. Methods

### 3.1. The model

Our model of pyramidal neuron considers three different compartments: a basal one ($b$) and two apical ones, named proximal ($p$) and distal ($d$), respectively (see Figure 1 for reference).

Consider a particular neuron $i$, with $i = 1, \ldots, N$; its real-valued membrane potential vector $\mathbf{v}_i^t = \left(v_i^t,\, u_i^t,\, u_i^{\star,t}\right)$ (the membrane potentials of basal, proximal apical, and distal apical compartments, respectively) follows a leaky-integrate-and-fire dynamics, which we can generically write as:

$$\mathbf{v}_i^{t+1} = \left[\left(1 - \frac{dt}{\tau_m}\right)\mathbf{v}_i^t + \frac{dt}{\tau_m}\mathbf{I}_i^{t+1}\right]\left(1 - \mathbf{s}_i^t\right) + \mathbf{v}^{\circlearrowleft}\mathbf{s}_i^t, \tag{3}$$

where the vector quantities $\mathbf{I}_i^t = (I_{(b),i}^t,\, I_{(p),i}^t,\, I_{(d),i}^t)$, $\mathbf{s}_i^t = (z_i^t,\, a_i^t,\, a_i^{\star,t})$ and $\mathbf{v}^{\circlearrowleft} = (\text{v}_{(b)}^{\circlearrowleft},\, \text{v}_{(p)}^{\circlearrowleft},\, \text{v}_{(d)}^{\circlearrowleft})$ represent the input current, the neuron spike and the reset potential, respectively, for each compartment (see following sections for their explicit definitions). In particular, the neural spike $\mathbf{s}_i^t$ is a stochastic variable determined by its sigmoid probability:

$$p\left(\mathbf{s}_i^{t+1} | \mathbf{v}_i^t\right) = \frac{\exp\left[\mathbf{s}_i^{t+1}\left(\frac{\mathbf{v}_i^t - v_{\text{thr}}}{\delta v}\right)\right]}{1 + \exp\left(\frac{\mathbf{v}_i^t - v_{\text{thr}}}{\delta v}\right)}, \tag{4}$$

with $v_{\text{thr}}$ being the firing threshold for the membrane potential and $\delta v$ a model parameter controlling the probabilistic

nature of the firing process. In the $\delta v \to 0$ limit, the spike-generation rule (4) becomes deterministic:

$$p(\mathbf{s}^{t+1}|\mathbf{v}^t) = \Theta[\mathbf{s}^{t+1}(\mathbf{v}^t - v_{\text{thr}})]\,.$$

We remark that we assume the deterministic limit to numerically implement the dynamics ($\delta v \to 0$).

#### 3.1.1. Temporal filtering and windows

We introduce the exponential filtering function $\mathsf{filter}\,(\xi^t, \tau)$, defined recursively as:

$$\mathsf{filter}\left(\xi^{t+1}, \tau\right) = \exp\left(-\frac{dt}{\tau}\right)\mathsf{filter}\left(\xi^t, \tau\right) + \\ + \left(1 - \exp\left(-\frac{dt}{\tau}\right)\right)\xi^{t+1}\,. \quad (5)$$

Basal spike signals are time-filtered through suitable time constants, depending on the direction they propagate. Using the previous definition, we introduce the following filtered quantities:

$$\hat{z}_i^{t+1} = \mathsf{filter}\left(z_i^{t+1}, \tau_s\right) \quad (6)$$

$$\hat{z}_{\text{ro},i}^{t+1} = \mathsf{filter}\left(z_i^{t+1}, \tau_{\text{ro}}\right) \quad (7)$$

$$\hat{z}_{\text{soma},i}^{t+1} = \mathsf{filter}\left(z_i^{t+1}, \tau_{\text{targ}}\right) \quad (8)$$

Such filtering is also applied to the adaptation term $\omega_i^t$, which is time-smoothed as:

$$\omega_i^{t+1} = \mathsf{filter}\left(z_i^{t+1}, \tau_\omega\right). \quad (9)$$

Coincidence between above-threshold somatic spikes $\hat{z}_{\text{soma},i}^t$ and apical proximal $a_i^t$ or apical distal $a_i^{\star,t}$ spikes opens a time window:

$$\overline{z}_i^t = \Theta[\hat{z}_{\text{soma},i}^t - \vartheta_{\text{soma}}]\,. \quad (10)$$

The onset of a burst in the proximal or distal compartments, can be expressed respectively as:

$$B_i^{t+1} = \overline{z}_i^t a_i^{t+1} \quad (11)$$

$$B_i^{\star,t+1} = \overline{z}_i^t a_i^{\star,t+1} \quad (12)$$

Aiming for a time-window variable that is active during burst activity, we can iterate the same construction developed for spikes and consider the filtered burst-onset $\hat{\mathbf{B}}_i^t$:

$$\hat{B}_i^{t+1} = \mathsf{filter}\left(B_i^{t+1}, \tau_{\text{targ}}\right) \quad (13)$$

$$\hat{B}_i^{\star,t+1} = \mathsf{filter}\left(B_i^{\star,t+1}, \tau_{\text{targ}}\right) \quad (14)$$

One can again use these filtered quantities to introduce proximal and distal burst windows as:

$$\overline{B}_i^{t+1} = \Theta[\hat{B}_i^{t+1} - \vartheta_{\text{burst}}] \quad (15)$$

$$\overline{B}_i^{\star,t+1} = \Theta[\hat{B}_i^{\star,t+1} - \vartheta_{\text{burst}}] \quad (16)$$

When at least one among proximal and distal bursts is above threshold, we finally have a neural burst activity window:

$$\overline{B}_{\vee,i}^{t+1} = \overline{B}_i^{t+1} \vee \overline{B}_i^{\star,t+1}\,, \quad (17)$$

which is the quantity that will feature in the dynamics of the compartments.

#### 3.1.2. Basal compartment

The membrane potential of the basal compartment evolves following the equations:

$$v_i^{t+1} = \left[\left(1 - \frac{dt}{\tau_m}\right)v_i^t + \frac{dt}{\tau_m}I_{(b),i}^{t+1}\right](1 - z_i^t) + \mathrm{v}_{(b)}^{\circlearrowleft} z_i^t$$

$$I_{(b),i}^t = \sum_{j=1}^{N} J_{ij}^{b\to b}\hat{z}_j^t + \sum_{k=1}^{n_{\text{inp}}} J_{ik}^{\text{inp}} I_k^{\text{inp},t} + \beta\overline{B}_{\vee,i}^t - b\hat{\omega}_i^t + v_0$$

with $J_{ik}^{\text{inp}}$ and $I_k^{\text{inp},t}$ respectively being the input projection matrix and the input current, while $v_0$ is a compartment-specific constant input. We introduced the basal reset potential:

$$\mathbf{v}_{(b)}^{\circlearrowleft} = \frac{v_{\text{reset},b}}{1 + \alpha\overline{B}_{\vee,i}^t}\,,$$

where $v_{\text{reset},b}$ is a compartment-specific scalar, $\alpha$ is a constant model parameter and $\overline{B}_{\vee,i}^t$ is the active burst-window variable (see Section TEMPORAL FILTERING AND WINDOWS for an explicit characterization). Notice how during the burst-window $\overline{B}_{\vee,i}^t$, the soma receives an extra input and the reset potential increases; we set $\alpha = 2$ and $\beta = 20$ to define the entity of such effects.

#### 3.1.3. Apical proximal compartment

The apical proximal compartment of each neuron is connected to basal compartments of all the neurons through recurrent connections $J_{ij}^{b\to p}$ (the ones to be trained to reproduce the desired target). The equation for this compartment dynamics are:

$$u_i^{t+1} = \left[\left(1 - \frac{dt}{\tau_m}\right)u_i^t + \frac{dt}{\tau_m}I_{(p),i}^{t+1}\right](1 - a_i^t) + \mathrm{v}_{(p)}^{\circlearrowleft} a_i^t$$

$$I_{(p),i}^t = \underbrace{\sum_{j=1}^{N} J_{ij}^{b\to p}\hat{z}_j^t(t)}_{\substack{\text{recurrent basal}-\text{proximal}\\ \text{connections}}} + u_0$$

The reset potential for the proximal apical compartment $\mathrm{v}_{(p)}^{\circlearrowleft} = v_{\text{reset},p}$ is a compartment-specific scalar, independent of burst activity, while $u_0$ is the compartment-specific constant input.

#### 3.1.4. Apical distal compartment

The signal to be learned (target) is considered as an input for the apical distal compartment: coefficient $f_{\text{apic}}$ is set

*Table 1.* **Parameter of numerical simulations**. Many parameters have the same value for all the simulations reported in the main text figures. When not the case, the different values used are clearly indicated. For Figure 3 two values for the low-network (L) and the high-network (H), respectively, have been reported, when different from each other. For $\eta$ and $\eta_{\text{out}}$ for Figure 2, we report the initial parameter values, as during learning they are discounted as discussed in Section 2.2.

| PARAMETER | FIG 1 | FIG 2 | FIG 3 [L – H] |
|---|---|---|---|
| $N$ | 500 | 1000 | 500 – 500 |
| $\sigma_{\text{targ}}$ | 20 | 30 | 0 – 100 |
| $\sigma_{\text{in}}$ | 12 | 12 | 20 |
| $\eta$ | 10 | 10 | 0 – 0.25 |
| $\eta_{\text{out}}$ | 0.01 | 0.01 | 0.03 |
| $I$ | 5 | 50 | N.D. |
| $\sigma_{\text{cont}}$ | 0 | 20 | 50 – 0 |
| $N_e$ | | 80% $N$ | |
| $N_i$ | | 20% $N$ | |
| $\tau_{\text{m}}$ | | 20 (ms) | |
| $\tau_{\text{s}}$ | | 2 (ms) | |
| $\tau_{\text{out}}$ | | 10 (ms) | |
| $\tau_{\text{targ}}$ | | 20 (ms) | |
| $\tau_{\omega}$ | | 200 (ms) | |
| $b$ | | 100 | |
| $v_{\text{reset},b}$ | | -20 (mV) | |
| $v_{\text{reset},d,p}$ | | -160 (mV) | |
| $v_0$ | | -1 (mV) | |
| $u_0$ | | -6 (mV) | |
| $u_0^\star$ | | -6 (mV) | |
| $v_{\text{thr}}$ | | 0 (mV) | |
| $\vartheta_{\text{soma}}$ | | $2.5 \times 10^{-2}$ | |
| $\vartheta_{\text{burst}}$ | | $1.25 \times 10^{-2}$ | |

to 1 during the learning stage, and then set to 0 to get rid of this term during spontaneous activity. Also, the input from the context (again randomly projected on the $N$ neurons) is given as input for the apical distal compartment. The equations for the apical distal compartment read:

$$u_i^{\star,t+1} = \left[\left(1 - \frac{dt}{\tau_m}\right) u_i^{\star,t} + \frac{dt}{\tau_m} I_{(d),i}^{t+1}\right](1 - a_i^{\star,t}) + \\ + \mathrm{v}_{(d)}^{\circlearrowleft} a_i^{\star,t}$$

$$I_{(d),i}^t = \underbrace{f_{\text{apic}} \sum_{k=1}^{n_{\text{output}}} J_{ik}^{\text{targ}} y_k^{\star,t}}_{\text{target/teach input}} + \underbrace{\sum_{k=1}^{n_{\text{cont}}} J_{ik}^{\text{cont}} C_k^t}_{\text{context}} + u_0^\star$$

where $y_k^{\star,t}$ is the target signal and $C_k^t$ the context signal, while $u_0^\star$ is the compartment-specific constant input. We report the model parameters, for the three figures, in Table 1.

### 3.2. Derivation of the learning rule

We derive the update rule for the recurrent weights of the network by maximizing the probability to reproduce the target spatio-tamporal pattern of bursts, extending previous approaches used for learning the target pattern of spikes (Pfister et al., 2006; Jimenez Rezende & Gerstner, 2014; Gardner & Grüning, 2016; Muratore et al., 2021). The first step is to write the probability to produce a burst in the neuron $i$ at time $t$, given the somatic window $\overline{z}_i^t$. We propose the following compact formulation:

$$p(B_i^{\star,t+1}|\overline{z}_i^t) = \frac{\exp\left[B_i^{\star,t+1}\Phi_i^t(\overline{z}_i^t)\right]}{1 + \exp\left[\Phi_i^t(\overline{z}_i^t)\right]}, \qquad (18)$$

where we have introduced $\Phi_i^t(\overline{z}_i^t) = a_i^{t+1}\overline{z}_i^t/\delta v - (1 - \overline{z}_i^t)\gamma$. By definition, a burst can only happen by means of a basal-apical spike coincidence, represented by the $a_i^{t+1}\overline{z}_i^t$ term. When the basal window is open ($\overline{z}_i^t = 1$) the burst probability reduces to the usual sigmoidal function. When the window is closed and $\overline{z}_i^t = 0$, we have $\Phi_i^t(\overline{z}_i^t) = -\gamma$, we can thus tune the $\gamma$ parameter to model the burst probability. In practice, we work in the $\gamma \to \infty$ limit where $\lim_{\gamma\to\infty} p(B_i^{\star,t+1}|\overline{z}_i^t = 0) = 0$, which agrees to the intuitive understanding that a closed basal window prevents any burst activity. We introduce the likelihood $\mathcal{L}$ of observing a given target burst activity $\mathbf{B}^\star$ given the basal-to-proximal connections $J_{ij}^{b\to p}$ as:

$$\mathcal{L}\left(\mathbf{B}^\star|J^{b\to p}\right) = \sum_{it}\Big[B_i^{\star,t+1}\Phi_i^t(\overline{z}_i^t) + \\ - \log\left(1 + \exp\left[\Phi_i^t(\overline{z}_i^t)\right]\right)\Big]. \qquad (19)$$

We can then maximize this likelihood by adjusting the synaptic connections so to achieve the target burst activity $\mathbf{B}^\star$. By differentiating with respect to the recurrent apical weights, we get:

$$\frac{\partial\mathcal{L}(\mathbf{B}^\star|J^{b\to p})}{\partial J_{ij}^{b\to p}} = \left[B_i^{\star,t+1} - p(B_i^{t+1} = 1)\right]\overline{z}_i^t e_j^t, \qquad (20)$$

where we have introduced the following two quantities:

$$p(B_i^{t+1} = 1) = \frac{\exp\left[\Phi_i^t(\overline{z}_i^t)\right]}{1 + \exp\left[\Phi_i^t(\overline{z}_i^t)\right]} \quad \text{and} \quad e_j^t = \frac{\partial u_i^t}{\partial J_{ij}^{b\to p}}.$$

Given the basal window $\overline{z}_i^t$ state, the target burst sequence is uniquely defined by the input projected to the apical distal compartment and can be written as $B_i^{\star,t+1} = \overline{z}_i^t a_i^{\star,t+1}$. If we take the model deterministic limit ($\delta v \to 0$, where $p(B_i^{t+1} = 1) = a_i^{t+1}\overline{z}_i^t$) and note that $\overline{z}_i^t\overline{z}_i^t = \overline{z}_i^t$, we can rewrite the previous expression in a cleaner form:

$$\frac{\partial\mathcal{L}(\mathbf{B}^\star|J^{b\to p})}{\partial J_{ij}^{b\to p}} = \left[a_i^{\star,t+1} - a_i^{t+1}\right]\overline{z}_i^t e_j^t. \qquad (21)$$

This means that the spikes in the proximal apical compartment $a_i^{t+1}$ should mimic the ones in the distal one $a_i^{\star,t+1}$, when the somatic window $\overline{z}_i^t$ is open. For simplicity, we discussed this version of the learning rule. However, in this work we exploited the non-deterministic version of the plasticity rule (finite $\delta v = 0.1$) that can be rewritten as:

$$\frac{\partial \mathcal{L}(\mathbf{B}^\star | J^{b\to p})}{\partial J_{ij}^{b\to p}} = \left[a_i^{\star,t+1} - p\left(\mathrm{a}_i^{t+1} = 1 | \mathrm{u}_i^t\right)\right] \overline{z}_i^t e_j^t \,, \tag{22}$$

where $p\left(\mathrm{a}_i^{t+1} = 1 | \mathrm{u}_i^t\right) = \frac{\exp\left(\frac{\mathrm{u}_i^t - v_{\mathrm{thr}}}{\delta v}\right)}{1+\exp\left(\frac{\mathrm{u}_i^t - v_{\mathrm{thr}}}{\delta v}\right)}$. We stress here how in the derivation we considered the basal-windows state $\overline{z}_i^t$ as given. Consequently, the target burst sequence $\mathbf{B}^\star$ is uniquely defined by the input projected to the apical distal compartment and the likelihood is well-defined. Though we are aware of the feedback influence of the burst activity on the basal-window configuration (bursts induce basal spikes, see the equation for basal current $I_{(b),i}^t$ in the BASAL COMPARTMENT section), we chose to neglect such contribution as it would have severely increased the difficulty of the derivation. The convergence to the chosen target thus cannot be granted.

However, the pattern of apical spikes $\{a^\star\}$ does not change during learning, being determined by the original teaching signal $y^\star$ and the variance $\sigma_{\mathrm{targ}}$ of its random projection to the network. As target bursts only occur after coincidence of an apical spike $a^\star$ and a basal spike z, pattern $\{B^\star\}$ is a subset of distal apical spikes $\{a^\star\}$. In principle, it is still possible that the target pattern oscillates between slightly different subsets of $\{a^\star\}$. Anyway, we provide a numerical demonstration that the target pattern of bursts converges to a well-defined pattern (see Appendix for details).

## 4. Discussion

Recently, more and more studies confirmed that dendrites are capable of producing spikes (Gasparini et al., 2004) and performing complex and non-linear computation (Poirazi & Papoutsi, 2020). Also, it has been observed that the initiation of broad calcium action potentials ("Ca2+ spikes") near the apical tuft of pyramidal layer-5 neurons, produces a long (up to 50 ms in vitro) plateau-type depolarization (Larkum, 2013). The coincidence between this phenomenon and a somatic spike induces high-frequency somatic bursts during such depolarization. In the present work, we model such mechanism through the variable $\overline{B^\star}$ (see Eq. (12)), that is 1 for a 30 ms time window, after the coincidence between the apical ($a^\star$) and the somatic spike ($z$).

We show that, thanks to such properties, pyramidal neurons can naturally support target-based learning, easily applicable to, e.g., store and recall tasks. Moreover, it becomes possible to use contextual signals to flexibly select the desired output from a repertoire of learned dynamics. These properties naturally combine to orchestrate a network with a hierarchical architecture, which in turn lends itself to *hierarchical imitation learning* (Le et al., 2018). HIL enables the decomposition of challenging long-horizon decision-making tasks into simpler sub-tasks, improving both learning speed and transfer learning, as skills learned by sub-modules can be re-used for different tasks. In our work, a high-level network (the manager) selects the correct policy for the task, suggesting it as a contextual signal to the low-level network (the worker), in charge of actually executing it. We also show how considering contextual information as an input for the apical compartment (instead of the basal one) is crucial for the correct decomposition and accomplishment of the task, in agreement with the biological interpretation of apical dendritic inputs as contextual signals from other cortical areas.

Though our hierarchical imitation learning approach requires devising handcrafted solutions for the different layers, our message is that the architecture we propose can efficiently support the implementation of hierarchical policies. In future works, we plan to replace behavioral cloning with more general learning schemes, such as feudal networks (Vezhnevets et al., 2017) where the "high network" (manager) moves in a latent space and the "low network" (worker) translates it into meaningful behavioral primitives.

To our knowledge, no other existing works propose a biologically plausible architecture to implement HIL. Furthermore, our model prepares the ground for further biological explorations. Tuning model parameters (e.g., the adaptation strength $b$) allows simulating the transition between different brain states (e.g., sleep and awake) (Wei et al., 2018; Goldman et al., 2020; Tort-Colet et al., 2021). Possible future investigation topics include the replay of patterns of bursts during sleep (Kaefer et al., 2020), and the effect of sleep on tasks performances (Wei et al., 2018; Capone et al., 2019; Golosio et al., 2021; Capone & Paolucci, 2022).

## Source code availability

The source code is available for download under CC-BY license in the https://github.com/cristianocapone/LTTB public repository.

## Acknowledgements

This work has been supported by the European Union Horizon 2020 Research and Innovation program under the FET Flagship Human Brain Project (grant agreement SGA3 n. 945539 and grant agreement SGA2 n. 785907) and by the INFN APE Parallel/Distributed Computing laboratory.

## References


Bellec, G., Scherr, F., Subramoney, A., Hajek, E., Salaj, D., Legenstein, R., and Maass, W. A solution to the learning dilemma for recurrent networks of spiking neurons. *Nature communications*, 11(1):1–15, 2020.

Capone, C. and Paolucci, P. S. Towards biologically plausible dreaming and planning. *arXiv preprint arXiv:2205.10044*, 2022.

Capone, C., Pastorelli, E., Golosio, B., and Paolucci, P. S. Sleep-like slow oscillations improve visual classification through synaptic homeostasis and memory association in a thalamo-cortical model. *Scientific reports*, 9(1):1–11, 2019.

Capone, C., Muratore, P., and Paolucci, P. S. Error-based or target-based? A unified framework for learning in recurrent spiking networks. *PLOS Computational Biology*, 2022. doi: 10.1371/journal.pcbi.1010221.

DePasquale, B., Cueva, C. J., Rajan, K., Escola, G. S., and Abbott, L. full-force: A target-based method for training recurrent networks. *PloS one*, 13(2):e0191527, 2018.

Gardner, B. and Grüning, A. Supervised learning in spiking neural networks for precise temporal encoding. *PloS one*, 11(8):e0161335, 2016.

Gasparini, S., Migliore, M., and Magee, J. C. On the initiation and propagation of dendritic spikes in ca1 pyramidal neurons. *Journal of Neuroscience*, 24(49):11046–11056, 2004.

Goldman, J., Kusch, L., Hazalyalcinkaya, B., Depannemaecker, D., Nghiem, T.-A., Jirsa, V., and Destexhe, A. Brain-scale emergence of slow-wave synchrony and highly responsive asynchronous states based on biologically realistic population models simulated in the virtual brain. *BioRxiv*, 2020.

Golosio, B., De Luca, C., Capone, C., Pastorelli, E., Stegel, G., Tiddia, G., De Bonis, G., and Paolucci, P. S. Thalamocortical spiking model of incremental learning combining perception, context and nrem-sleep. *PLoS Computational Biology*, 17(6):e1009045, 2021.

Guerguiev, J., Lillicrap, T. P., and Richards, B. A. Towards deep learning with segregated dendrites. *Elife*, 6:e22901, 2017.

Jimenez Rezende, D. and Gerstner, W. Stochastic variational learning in recurrent spiking networks. *Frontiers in Computational Neuroscience*, 8:38, 2014. ISSN 1662-5188. doi: 10.3389/fncom.2014.00038.

Kaefer, K., Nardin, M., Blahna, K., and Csicsvari, J. Replay of behavioral sequences in the medial prefrontal cortex during rule switching. *Neuron*, 106(1):154–165, 2020.

Larkum, M. A cellular mechanism for cortical associations: an organizing principle for the cerebral cortex. *Trends in neurosciences*, 36(3):141–151, 2013.

Le, H., Jiang, N., Agarwal, A., Dudik, M., Yue, Y., and Daumé III, H. Hierarchical imitation and reinforcement learning. In *International conference on machine learning*, pp. 2917–2926. PMLR, 2018.

Lee, D.-H., Zhang, S., Fischer, A., and Bengio, Y. Difference target propagation. In *Joint european conference on machine learning and knowledge discovery in databases*, pp. 498–515. Springer, 2015.

Manchev, N. and Spratling, M. W. Target propagation in recurrent neural networks. *J. Mach. Learn. Res.*, 21:7–1, 2020.

Meulemans, A., Carzaniga, F. S., Suykens, J. A., Sacramento, J., and Grewe, B. F. A theoretical framework for target propagation. *arXiv preprint arXiv:2006.14331*, 2020.

Muratore, P., Capone, C., and Paolucci, P. S. Target spike patterns enable efficient and biologically plausible learning for complex temporal tasks. *PloS one*, 16(2): e0247014, 2021.

Nicola, W. and Clopath, C. Supervised learning in spiking neural networks with force training. *Nature communications*, 8(1):2208, 2017.

Pateria, S., Subagdja, B., Tan, A.-h., and Quek, C. Hierarchical reinforcement learning: A comprehensive survey. *ACM Computing Surveys (CSUR)*, 54(5):1–35, 2021.

Payeur, A., Guerguiev, J., Zenke, F., Richards, B. A., and Naud, R. Burst-dependent synaptic plasticity can coordinate learning in hierarchical circuits. *Nature neuroscience*, pp. 1–10, 2021.

Pfister, J.-P., Toyoizumi, T., Barber, D., and Gerstner, W. Optimal spike-timing-dependent plasticity for precise action potential firing in supervised learning. *Neural computation*, 18(6):1318–1348, 2006.

Poirazi, P. and Papoutsi, A. Illuminating dendritic function with computational models. *Nature Reviews Neuroscience*, 21(6):303–321, 2020.

Sacramento, J. a., Ponte Costa, R., Bengio, Y., and Senn, W. Dendritic cortical microcircuits approximate the backpropagation algorithm. In Bengio, S., Wallach, H.,

Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31*, pp. 8721–8732. Curran Associates, Inc., 2018.

Tort-Colet, N., Capone, C., Sanchez-Vives, M. V., and Mattia, M. Attractor competition enriches cortical dynamics during awakening from anesthesia. *Cell Reports*, 35(12): 109270, 2021.

Urbanczik, R. and Senn, W. Learning by the dendritic prediction of somatic spiking. *Neuron*, 81(3):521–528, 2014.

Vezhnevets, A. S., Osindero, S., Schaul, T., Heess, N., Jaderberg, M., Silver, D., and Kavukcuoglu, K. Feudal networks for hierarchical reinforcement learning. In *International Conference on Machine Learning*, pp. 3540–3549. PMLR, 2017.

Wei, Y., Krishnan, G. P., Komarov, M., and Bazhenov, M. Differential roles of sleep spindles and sleep slow oscillations in memory consolidation. *PLoS computational biology*, 14(7):e1006322, 2018.

# Appendix: Burst-Dependent Plasticity and Dendritic Amplification Support Target-Based Learning and Hierarchical Imitation Learning

## Numerical evidence of convergence

As mentioned in the main text, we can not provide a mathematical proof of the convergence toward the chosen target of burst activity by means of the learning rule proposed here. However, strong evidences in this direction can be found numerically.

We run several independent realizations of the same task of Figure 1, i.e., the store and recall of a 3D trajectory. We look at the distance between the target and the spontaneous spatio-temporal pattern of bursts during the training, and also at the self-distance in the pattern of spontaneous bursts across consecutive training iterations.

The parameters used for these simulations (when different from those used for Figure 1) are: $\eta = 2.5$, $\eta_{\text{out}} = 2.5 \times 10^{-3}$, $\sigma_{\text{targ}}$ variable from 10 (black) to 1000 (yellow). Data averaged over 10 independent network/target realizations. The distance between two patterns of bursts $A = \{A_i^t\}$ and $B = \{B_i^t\}$ is defined as:

$$\mathcal{D}(A,B) \equiv \sqrt{\frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T}(A_i^t - B_i^t)^2}\,.$$

For small values of $\sigma_{\text{targ}}$, comparable to the ones used for main text figures, target bursts rapidly settle after some hundreds of training iterations (Figure S1, left); within the same training scale, also spontaneous burst activity matches the target one, with a negligible error (Figure S1, middle).

We prove that in a broad range of $\sigma_{\text{targ}}$ values (roughly up to $\sigma_{\text{targ}} = 100$), the target pattern of bursts converges to a well-defined one (Figure S1, right, blue dots), while for higher values of $\sigma_{\text{targ}}$ the convergence slows down. This is related to the increase of the number of bursts for high values of $\sigma_{\text{targ}}$ (Figure S1, right, red dots).

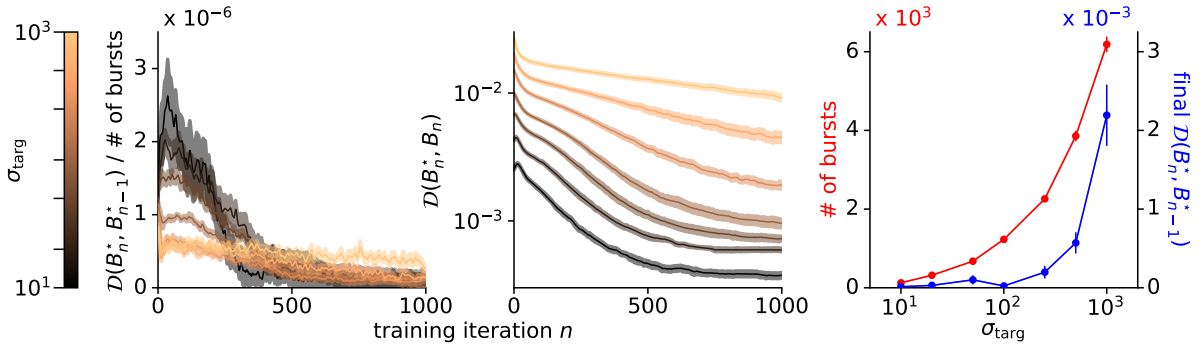


*Figure S1.* **Convergence of the target pattern of bursts vs $\sigma_{\text{targ}}$. (left)** $\mathcal{D}(B_n^\star, B_{n-1}^\star)/(\text{number of bursts})$ as a function of the number $n$ of learning iterations, for different $\sigma_{\text{targ}}$ values (lower to higher values, from dark to light). **(middle)** Distance between the target and spontaneous pattern of bursts $\mathcal{D}(B_n^\star, B_n)$ after $n$ learning iterations. **(right)** Blue: average final $\mathcal{D}(B_n^\star, B_{n-1}^\star)$ value as a function of $\sigma_{\text{targ}}$. Red: average number of bursts as a function of $\sigma_{\text{targ}}$.

We made further simulations, for different network sizes ($N = 125,\ 250,\ 500,\ 1000,\ 2000$) at a given value $\sigma_{\text{targ}} = 10$. We always observe convergence, which is even faster for larger networks.

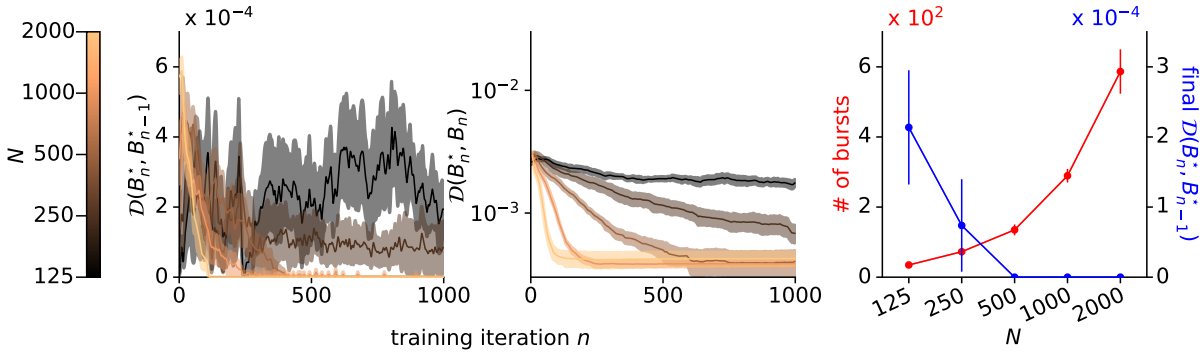


*Figure S2.* **Convergence of the target pattern of bursts vs $N$. (left)** $\mathcal{D}(B_n^\star, B_{n-1}^\star)$ as a function of the number $n$ of learning iterations, for different network sizes $N$ (lower to higher values, from dark to light). **(middle)** Distance between the target and spontaneous pattern of bursts $\mathcal{D}(B_n^\star, B_n)$ after $n$ learning iterations. **(right)** Blue: average final $\mathcal{D}(B_n^\star, B_{n-1}^\star)$ value as a function of $N$. Red: average number of bursts as a function of $N$.