# Theoretical framework for learning through structural plasticity

Gianmarco Tiddia,* Luca Sergi, and Bruno Golosio
*Department of Physics, University of Cagliari, 09042 Monserrato, Italy*
*and Istituto Nazionale di Fisica Nucleare (INFN), Sezione di Cagliari, 09042 Monserrato, Italy*

(Received 28 August 2023; accepted 19 June 2024; published 22 October 2024)

A growing body of research indicates that structural plasticity mechanisms are crucial for learning and memory consolidation. Starting from a simple phenomenological model, we exploit a mean-field approach to develop a theoretical framework of learning through this kind of plasticity, capable of taking into account several features of the connectivity and pattern of activity of biological neural networks, including probability distributions of neuron firing rates, selectivity of the responses of single neurons to multiple stimuli, probabilistic connection rules, and noisy stimuli. More importantly, it describes the effects of stabilization, pruning, and reorganization of synaptic connections. This framework is used to compute the values of some relevant quantities used to characterize the learning and memory capabilities of the neuronal network in training and testing procedures as the number of training patterns and other model parameters vary. The results are then compared with those obtained through simulations with firing-rate-based neuronal network models.

DOI: 10.1103/PhysRevE.110.044311

## I. INTRODUCTION

Together with temporary and reversible changes of synaptic efficacy such as short and long-term plasticity mechanisms, structural changes in the synaptic morphology of the network are fundamental mechanisms that take place in healthy brains. These changes occur at longer timescales than the short- or long-term mechanisms mentioned above and consist in the stabilization, creation of new synapses, or erasure of synapses that have not been stabilized [1,2]. This type of synaptic plasticity, called *structural plasticity*, can be spontaneous but also experience-based [3], and it has a key role in the stabilization of new concepts that need to be kept in memory after learning [4].

Indeed, it is known that neurotransmitters can be neurotrophic factors, i.e., participate in the growth or suppression of dendritic spines, synapses, axons, and dendrites [1,5,6]. Thus, structural plasticity is a neural-activity-driven mechanism, which can increase or decrease the number of synapses. Such modifications are flanked by a homeostatic kind of structural plasticity, which has a balancing effect achieved by adding or removing synapses, as described in Ref. [7].

Moreover, the number of synapses in the brain can change over time. In Ref. [8] it is shown that synaptic density in the human cortex reaches the highest values at 1–2 years age, it drops during adolescence and stabilizes between ages 16 and 72, followed by a slight decline. However, although synaptic density remains approximately stable during adulthood, rewiring of network connections occurs as well to efficiently store new memories [9,10]. The activity-dependent connectivity changes, together with the rearrangement of synapses lead to a fine-tuning of the brain's circuits [11]. Indeed, some synapses can be strengthened through long-term potentiation (LTP) and new connections can be formed next to the already potentiated ones to further enhance synaptic transmission. However, when the presynaptic and postsynaptic neuron activities have a low correlation, their connection is more likely to be removed. The latter process is called synaptic pruning and it is considered essential for optimizing activity propagation and memory capacity [12–14]. Furthermore, it is commonly believed that synaptic pruning and rewiring dysfunction are neural correlates of developmental disorders such as autism or schizophrenia [15,16], leading to, respectively, a higher or lower synaptic density with respect to neurotypical subjects [17–19].

In the last decades, computational neuroscience has investigated brain dynamics at different scales, from cellular [20] to mesoscopic and macroscopic through mean-field approaches [21–28]. Regarding synaptic plasticity, computational models were mostly focused on plasticity mechanisms that involve strengthening or weakening of existing synapses, like short-term plasticity (STP) [29] or spike-timing-dependent plasticity (STDP) [30] and on their role in short-term, long-term, working memory, and learning [31–36]. Only in recent times, computational models of structural plasticity and connectivity rearrangements during learning were developed, showing intriguing results. [13] and [14] describe a model of structural plasticity based on ”effectual connectivity”, defined in these works as the fraction of synapses able to represent a memory stored in a network. By structural plasticity, effectual connectivity is improved, since synapses that do not code for the memory are moved to optimize network's connectivity.

[*] Contact author: gianmarco.tiddia@dsf.unica.it

Their model defines synapses using a Markov model of three states: potential (i.e., not instantiated), instantiated but silent or instantiated and stabilized. Structural plasticity is thus related to the passage of the synapses from a potential state to an instantiated state (and vice versa), whereas changes only related to the synaptic weight are described by the potentiation of the instantiated synapses. With such a model, it is possible to show that networks with structural plasticity have higher or comparable memory capacity to networks with dense connectivity and it is possible to explain some cognitive mechanisms such as the spacing effect [13].

[37] simulated a spiking neural network with structural plasticity and STDP, showing that structural plasticity reduces the amount of noise of the network after a learning process, thus making the network able to have a clearer output. Furthermore, such a network with structural plasticity shows higher learning speed than the same network with only STDP implemented.

Some new insights about the importance of synaptic pruning are also shown in Ref. [9], in which different pruning rates were studied suggesting that a slowly decreasing rate of pruning over time leads to more efficient network architectures.

The model proposed in this work considers two populations of neurons, $\mathcal{P}_1$ and $\mathcal{P}_2$, with synaptic connections directed from the first to the second population (i.e., in a feed-forward fashion). During the training phase, $\mathcal{P}_1$ receives an input stimulus, while $\mathcal{P}_2$ receives a contextual stimulus. The model assumes that synaptic stabilization is a probabilistic process driven by pre- and postsynaptic spiking activity. For each pattern given in input to the model, only a small fraction of neurons contribute to this process. Connection stabilization is complemented by a rewiring process: connection pruning, which eliminates unstabilized connections, and creation of new connections, which restores network balance. This framework does not specifically refer to a particular region of the brain; rather, it is built on characteristics that are ubiquitous for several brain areas.

As discussed above, the biochemical and biophysical mechanisms underlying structural plasticity are extremely complex and only partially understood to date. For this reason, rather than attempting to build a biologically detailed model, this work exploits a relatively simple phenomenological model, including both the activity-driven and the homeostatic contributions; despite the lower complexity, this model accounts for the effects of structural plasticity in terms of the stabilization of synaptic connections between neurons with a high activity correlation as well as those of pruning and rewiring the connections for which this correlation is lower. This approach is also justified by the requirement for a simple and effective computational model suitable for simulating networks with a relatively large number of neurons and connections and for representing learning processes with sizable numbers of training and testing patterns. The model will then serve as the foundation for the creation of a mean-field-based theoretical framework for learning through synaptic plasticity capable of accounting for a variety of biological network properties. This framework is used in a training and testing procedure to characterize learning and memory capacity of plastic neuronal networks as the number of training patterns and other model parameters vary. The results are then compared with those obtained through simulations based on feed-forward firing-rate-based neuronal networks.

The proposed approach is capable of accounting for different probabilistic connection rules, firing rate probability distributions, presence of noise in stimuli, thus providing a general framework to study the impact of structural plasticity on learning in large-scale neuronal network models.

## II. MODEL DESCRIPTION

This section describes the model proposed in this work. A pseudo-code of the algorithm used to model the structural plasticity mechanism is provided in Algorithm 1. The neuronal network consists of two neuron populations, $\mathcal{P}_1$ and $\mathcal{P}_2$, with $10^5$ neurons each. The exchange of information between the two populations takes place through the feed-forward connections from the population $\mathcal{P}_1$ to the population $\mathcal{P}_2$, which in the model are on average $5 \times 10^3$ per neuron of $\mathcal{P}_2$ for a total of $5 \times 10^8$ connections. Each connection has an initial baseline synaptic weight $\mathcal{W}_\text{b}$ (*initialization* in Algorithm 1). To mimic the activity of $\mathcal{P}_1$ in response to an external input signal (e.g., a visual input), the model assigns to each neuron of this population a value of firing rate derived from a predefined firing rate distribution. This way, an input is modeled as a firing rate pattern of $\mathcal{P}_1$. During the training stage, using the same approach previously described for $\mathcal{P}_1$, the population $\mathcal{P}_2$ is injected with a stimulus (e.g., auditory), that we identify as a *contextual stimulus* (as proposed in Ref. [33]). In this phase we assume that the activity of $\mathcal{P}_2$ is entirely derived by the contextual stimulus, neglecting the contribution of the connectivity between $\mathcal{P}_1$ and $\mathcal{P}_2$ (*training* in Algorithm 1).

The structural plasticity model follows the categories proposed by Ref. [7], i.e., activity-dependent and homeostatic. The firing rate patterns of the two populations have a role in the activity-dependent structural plasticity. The potentiation and stabilization of a synaptic connection occurs when the firing rates of both the presynaptic and the postsynaptic neurons are concurrently above a certain threshold (the definition of which varies depending on the firing rate distribution). In our model, the synaptic weight of a stabilized connection increases from $\mathcal{W}_\text{b}$ to a value $\mathcal{W}_\text{s} > \mathcal{W}_\text{b}$ (*synapses potentiation and stabilization* in Algorithm 1). This is a computationally effective way of taking into account the several biological mechanisms that concur in the stabilization of the connection between two neurons. We flank this mechanism with synaptic rewiring, which consists of the mechanism of pruning of the connections that have not been stabilized yet together with the creation of new connections handled by homeostatic structural plasticity (*synapse pruning* and *synapse creation* in Algorithm 1). We apply synaptic rewiring on the simulation periodically after a certain number of simulation steps. Once a synaptic connection has been stabilized, it will maintain the synaptic weight $\mathcal{W}_\text{s}$, without the possibility of returning to the initial state $\mathcal{W}_\text{b}$. Thus, these connections are prevented from being pruned in further simulation steps.

The sets of input and contextual patterns used for network training are independent firing-rate patterns of the two

ALGORITHM 1. Pseudo-code of structural plasticity during training.

```
INITIALIZATION
# creating initial connectivity
for $i$ from 1 to $\mathcal{N}_2$ do
    $\mathcal{C}_i$ = extractNumberOfConnections($\mathcal{C}$, connection_rule)
    for $c$ from 1 to $\mathcal{C}_i$ do
        # presynaptic neuron index is extracted randomly from $\mathcal{P}_1$
        $j$ = generateRandomInteger(1, $\mathcal{N}_1$)
        createConnection($i$, $j$)
        $\mathcal{W}_{i,j} = \mathcal{W}_\mathrm{b}$
    end for
end for
TRAINING
# training on $\mathcal{T}$ patterns
for $t$ from 1 to $\mathcal{T}$ do
    # providing to the network a firing rate pattern stimulusInjection()
    for $i$ from 1 to $\mathcal{N}_2$ do
        for $c$ from 1 to $\mathcal{C}_i$ do
            # retrieve index of the presynaptic neuron for connection $c$
            $j$ = sourceNodeIndex($i$, $c$)
            SYNAPSES POTENTIATION AND STABILIZATION
            # synapse potentiation and stabilization if both neurons have high rate
            if $\nu_i \geqslant \nu_{th}$ and $\nu_j \geqslant \nu_{th}$ then
                $\mathcal{W}_{i,j} = \mathcal{W}_\mathrm{s}$
            end if
            SYNAPSES PRUNING
            # connection pruning after every rewiring_step
            if isMultipleInteger($t$, rewiring_step) then
                if $\mathcal{W}_{i,j} = \mathcal{W}_\mathrm{b}$ then
                    pruneConnection($i$, $j$)
                end if
            end if
        end for
        SYNAPSES CREATION
        # creation of new connections after every rewiring step
        if isMultipleInteger($t$, rewiring_step) then
            $k$ = numberOfStabilizedConnections($i$)
            $\mathcal{C}_i$ = extractNumberOfConnections($\mathcal{C}$, conn_rule)
            for $c$ from $k$ to $\mathcal{C}_i$ do
                $j$ = chooseRandomNeuronFromPopulation($\mathcal{P}_1$)
                createConnection($i$, $j$)
                $\mathcal{W}_{i,j} = \mathcal{W}_\mathrm{b}$
            end for
        end if
    end for
end for
```

populations randomly generated from predefined firing-rate probability distributions. The training process is performed using $\mathcal{T}$ independent input patterns, together with the corresponding contextual stimuli.

During training, when both input and contextual stimulus are used, a fraction of the neurons of the population $\mathcal{P}_2$ assumes a high value of firing rate (i.e., above threshold), becoming thus representative of that input. These neurons, called *coding*, or *selective* neurons, play a vital role in input coding. The average input signal to these neurons will be called $\langle\mathcal{S}_\mathrm{c}\rangle$. The nonselective neurons of $\mathcal{P}_2$ will instead be called *noncoding* or *background* neurons, and their average input signal will be indicated with $\langle\mathcal{S}_\mathrm{b}\rangle$. The average incoming signals to background and coding neurons of $\mathcal{P}_2$ (i.e., $\langle\mathcal{S}_\mathrm{b}\rangle$ and $\langle\mathcal{S}_\mathrm{c}\rangle$) is evaluated in the test phase, during which an input pattern is provided to $\mathcal{P}_1$ and the signal incoming to neurons of $\mathcal{P}_2$ is entirely derived from the connectivity between the two populations optimized during training thanks to the structural plasticity mechanism. Input patterns in the test phase are derived from the training patterns with the addition of noise. The proposed model accounts for the ability of the network to learn the association between input patterns and the corresponding contextual stimuli.

A diagram of the training and testing processes is shown in Fig. 1. The biological motivations for the choices we adopted for the model design are reported in the following section.

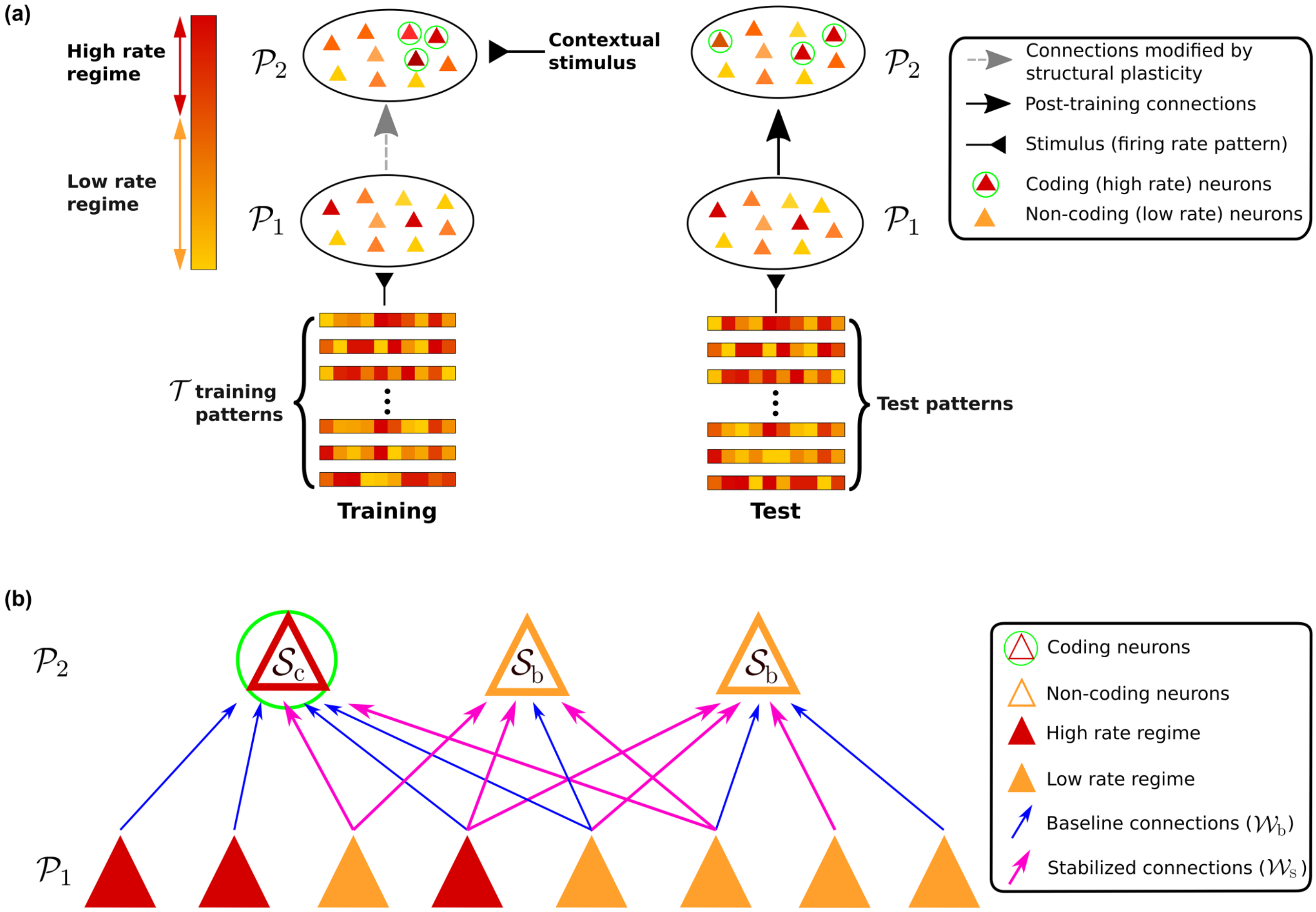
(a)
High rate regime
Low rate regime
$\mathcal{P}_2$
$\mathcal{P}_1$
Contextual stimulus
$\mathcal{T}$ training patterns
Training
Test patterns
Test
Connections modified by structural plasticity
Post-training connections
Stimulus (firing rate pattern)
Coding (high rate) neurons
Non-coding (low rate) neurons
(b)
$\mathcal{S}_c$
$\mathcal{S}_b$
$\mathcal{S}_b$
Coding neurons
Non-coding neurons
High rate regime
Low rate regime
Baseline connections ($\mathcal{W}_b$)
Stabilized connections ($\mathcal{W}_s$)

FIG. 1. Schematic representation of the network model. The triangles represent neurons of the network, with their rate indicated by the color (from yellow to red). (a) During training (left), a stimulus is injected into $\mathcal{P}_1$, and a contextual stimulus is injected into $\mathcal{P}_2$. In this phase, structural plasticity occurs by refining and reorganizing the connectivity between the two neuron populations (dashed arrow). Moreover, during training we assume that the activity of $\mathcal{P}_2$ is only derived from the contextual stimulus, neglecting the contribution of the connections between the populations. The stimuli are randomly generated from a predefined firing rate probability distribution. In the test phase (right), a pattern is injected without the corresponding contextual stimulus and the connections between the two populations project the input to the neurons of $\mathcal{P}_2$. The test patterns are derived from the training patterns with the addition of noise. (b) Detail of the test phase, in which is shown a subset of neurons of $\mathcal{P}_1$ and $\mathcal{P}_2$ (triangles) connected through connections (arrows). The triangles are colored so that amber triangles fall in the low rate regime and red ones in the high rate regime, and the connections are distinguished between baseline (blue, with synaptic weight $\mathcal{W}_b$) and stabilized (fuchsia, with synaptic weight $\mathcal{W}_s$). The neuron of $\mathcal{P}_2$ on the left is a coding neuron, i.e., it is representative of the input injected, and thus receives an input signal $\mathcal{S}_c$. The other neurons of $\mathcal{P}_2$, being noncoding neurons, receive a background signal $\mathcal{S}_b$.

### A. Biological justification of the model choices

The model described above has several biologically motivated features. First, the average number of connections per neuron has been chosen in agreement with experimental estimation [38]. Training and test patterns are generated so that a fraction of the neurons targeted by the external stimuli show a high-rate regime in response to the input.

During training, when input and contextual stimuli are used, a fraction of the neurons of $\mathcal{P}_2$ become representative of that input. Indeed, the existence of neurons showing selective firing rates in response to specific stimuli is largely confirmed by experimental results. Additionally, the neuron populations assume a continuous firing rate distribution when targeted by an external stimulus, with the firing rate of each neuron derived by the distribution. In agreement with the experimental observations, we employ a lognormal distribution of the firing rate [39].

Regarding the structural plasticity mechanism, we followed the description of Ref. [7], thus dividing the structural mechanism into two categories: activity-dependent and homeostatic. Activity-dependent mechanisms lead to stabilization and potentiation of a synapse when the pre- and postsynaptic neurons assume a high-rate regime. Indeed, it is believed that the potentiation of the synapses that connect neurons with strongly correlated activity is accompanied by greater stability over time [40]. As a result, the synapse increases the synaptic efficacy to a value $\mathcal{W}_s$ than the baseline value $\mathcal{W}_b$. This can be seen as an effect of LTP, according to which we have the formation of new synapses, but also the increase of synaptic efficacy of the existing ones. Thus, the mechanism of stabilization proposed here embraces both structural and functional synaptic modifications, both mediated by LTP [41].

Homeostatic effects of structural plasticity take into account the mechanisms of synaptic pruning of the connections that have not been stabilized and the creation of novel synaptic contacts. We modeled these effects through synaptic rewiring, according to which nonstabilized connections are periodically pruned and new connections are created. This mechanism is not performed after every simulation step since it is known that the homeostatic effects take place on a longer timescale than the activity-dependent ones [7]. The mechanism of synaptic rewiring will be discussed in detail later in this work. Stabilized connections cannot be affected, by definition, by synaptic pruning, and will keep their synaptic weight to $\mathcal{W}_{\mathrm{s}}$, without the possibility of returning to the initial state $\mathcal{W}_{\mathrm{b}}$. This approach is a simplified way of representing the biological structural changes that make a stabilized connection strong and durable.

In the next section, we derive the mean-field equations of the model, whose description and parameters are summarized in the Tables I and II in Appendix A.

## III. THEORETICAL DERIVATIONS

Here we introduce the theoretical framework of the model. As mentioned earlier, in this model the injection of an external input to a neuron population is represented by neurons showing firing rate patterns generated from a probability distribution $\rho(\nu)$. The distinction between high-rate and low-rate neurons is based on two rate thresholds, $\nu_{\mathrm{t},1}$ for the population $\mathcal{P}_1$ and $\nu_{\mathrm{t},2}$ for the population $\mathcal{P}_2$. The value of these thresholds is related to the fraction of neurons above the threshold (i.e., high-rate) for the two populations, $\alpha_1$ and $\alpha_2$, respectively, by the equations

$$\begin{aligned}\alpha_1 &= \int_{\nu_{\mathrm{t},1}}^{\infty} \rho(\nu) d\nu,\\ \alpha_2 &= \int_{\nu_{\mathrm{t},2}}^{\infty} \rho(\nu) d\nu.\end{aligned} \tag{1}$$

Thus, the average number of high-rate neurons when an external stimulus is provided is

$$\begin{aligned}N_{h,1} &= \mathcal{N}_1\alpha_1,\\ N_{h,2} &= \mathcal{N}_2\alpha_2,\end{aligned} \tag{2}$$

where $\mathcal{N}_1$ and $\mathcal{N}_2$ are the number of neurons of $\mathcal{P}_1$ and $\mathcal{P}_2$, respectively. The average rates of the neurons of $\mathcal{P}_1$ below and above threshold, $\langle\nu_{\ell,1}\rangle$ and $\langle\nu_{h,1}\rangle$, can be computed from the firing rate distribution as

$$\begin{aligned}\langle\nu_{\ell,1}\rangle &= \int_0^{\nu_{\mathrm{t},1}} \nu\rho(\nu)d\nu \Big/ \int_0^{\nu_{\mathrm{t},1}} \rho(\nu)d\nu = \frac{1}{\beta_1}\int_0^{\nu_{\mathrm{t},1}} \nu\rho(\nu)d\nu,\\ \langle\nu_{h,1}\rangle &= \int_{\nu_{\mathrm{t},1}}^{\infty} \nu\rho(\nu)d\nu \Big/ \int_{\nu_{\mathrm{t},1}}^{\infty} \rho(\nu)d\nu = \frac{1}{\alpha_1}\int_{\nu_{\mathrm{t},1}}^{\infty} \nu\rho(\nu)d\nu,\end{aligned} \tag{3}$$

where $\beta_1 = 1 - \alpha_1$. Similar equations can be used to compute $\langle\nu_{\ell,2}\rangle$ and $\langle\nu_{h,2}\rangle$. In general, the thresholds $\nu_{\mathrm{t},1}$ and $\nu_{\mathrm{t},2}$ can be different; however, in this work, we assume that the thresholds are the same. From these equations, the average firing rate can be expressed as

$$\begin{aligned}\langle\nu\rangle = \int_0^{\infty} \nu\rho(\nu)d\nu &= \beta_1\langle\nu_{\ell,1}\rangle + \alpha_1\langle\nu_{h,1}\rangle\\ &= \beta_2\langle\nu_{\ell,2}\rangle + \alpha_2\langle\nu_{h,2}\rangle.\end{aligned} \tag{4}$$

The theoretical framework aims to provide an estimation of how well such a feed-forward network is able, after a training process, to store the learned patterns only through the connections changed by the structural plasticity mechanism. As a result of training, a subset of neurons of $\mathcal{P}_2$, called *coding neurons*, should receive during the test phase a higher input than the rest of the neurons of $\mathcal{P}_2$, which we call *background neurons*. In such a framework, an effective approach for estimating the capacity of the network to recognize the pattern would be to evaluate, during the test phase, the signal-difference-to-noise-ratio (SDNR) using the formula

$$\mathrm{SDNR} = \frac{|\langle\mathcal{S}_{\mathrm{c}}\rangle - \langle\mathcal{S}_{\mathrm{b}}\rangle|}{\sigma_b}, \tag{5}$$

where $\langle\mathcal{S}_{\mathrm{c}}\rangle$ and $\langle\mathcal{S}_{\mathrm{b}}\rangle$ are, respectively, the average input signal to coding and background neurons of the $\mathcal{P}_2$ population due to the connections coming from the $\mathcal{P}_1$ population and $\sigma_b$ is the standard deviation of the signal received by the background neurons. This choice is justified by the need to evaluate the memory capacity associated with the plasticity of the connections from $\mathcal{P}_1$ to $\mathcal{P}_2$. In Appendix B we derive the relation between SDNR and the probability of correctly recalling a pattern after training:

$$P_{\mathrm{C}} \simeq \frac{1}{2}\left[1 + \mathrm{erf}\left(\frac{\mathrm{SDNR}}{\sqrt{8}}\right)\right]. \tag{6}$$

This equation can be used to set a lower limit to the SDNR ensuring a probability of correct recall greater than a predetermined threshold $P_{\mathrm{C}}$ (e.g., $P_{\mathrm{C}} = 0.95$). Since the SDNR depends on the number of training patterns $\mathcal{T}$, it can be used to estimate the highest number of patterns viable for training which ensures retrieval during testing with a probability $P_{\mathrm{C}}$ (i.e., $\mathcal{T}_{\max}$). Training the network with an excessive number of patterns surpassing this maximum value makes the SDNR insufficient for distinguishing coding from background signals, increasing the likelihood of incorrect pattern recall.

During training, when input and contextual stimuli are provided to the network, a connection is stabilized if both the presynaptic and the postsynaptic neurons assume a firing rate above the threshold.

In this work, we use a lognormal distribution of the firing rates for the continuous model. Indeed, it is known that rate distribution in the cortex is long-tailed and skewed with a lognormal shape [39]. More details on the implementation of the lognormal distribution and the choice of the average low and high rates can be found in Appendix C. Nevertheless, the following derivations are valid for a generic probability distribution $\rho(\nu)$.

The test set consists of $V$ firing-rate patterns of the neurons of $\mathcal{P}_1$, randomly extracted from the $\mathcal{T}$ input patterns of the train set. Here we consider the case where the patterns are unchanged, thus each input pattern of the test set is identical to an input pattern of the train set. In a later section, we will discuss the effect of altering these patterns by adding noise. To estimate the values of $\langle\mathcal{S}_\mathrm{c}\rangle$, $\langle\mathcal{S}_\mathrm{b}\rangle$, and $\sigma_\mathrm{b}^2$ we can start by computing the average number of incoming stabilized connections for every neuron of $\mathcal{P}_2$ after a training process in which $\mathcal{T}$ independent patterns are given. This is useful since the input to a neuron of $\mathcal{P}_2$ driven by the connections from $\mathcal{P}_1$ is determined by the product between the synaptic weights of the connections and the rate of the respective presynaptic neurons.

The probability that a generic connection is stabilized in a single example is $\alpha_1\alpha_2$, and thus the probability that it is not stabilized after $\mathcal{T}$ training examples is $(1-\alpha_1\alpha_2)^\mathcal{T}$. The probability $p_\mathcal{T}$ that a connection is stabilized in at least one of the $\mathcal{T}$ training examples is given by the complement of the previous expression:

$$p_\mathcal{T} = 1-(1-\alpha_1\alpha_2)^\mathcal{T}. \tag{7}$$

As $k$ is the number of stabilized connections for a neuron, the average number of stabilized connections per neuron of $\mathcal{P}_2$ is defined as the product of the expression above by the number of incoming connections per neuron $\mathcal{C}$. This way, the average number of stabilized connections per neuron is

$$\langle k\rangle = \mathcal{C}[1-(1-\alpha_1\alpha_2)^\mathcal{T}] = \mathcal{C}p_\mathcal{T}. \tag{8}$$

For each neuron we have on average $\langle k\rangle$ stabilized connections with synaptic weight $\mathcal{W}_\mathrm{s}$ and $\mathcal{C}-\langle k\rangle$ nonstabilized connections with the baseline synaptic weight $\mathcal{W}_\mathrm{b}$. For simplicity, the previous equation and the following derivations assume that the number of incoming connections per neuron is constant. However, in the most general case, this number can be driven from a distribution. In this regard, Sec. III A describes the changes in the theoretical framework in the case of Poisson-distributed in-degrees.

First of all, we calculate the input signal to a background neuron of $\mathcal{P}_2$. Let $\mathcal{C}$ be the number of incoming connections to this neuron, $k$ the number of stabilized connections, $\nu_1,\cdots,\nu_k$ the firing rates of the presynaptic neurons of the stabilized connections, and $\xi_1,\cdots,\xi_{\mathcal{C}-k}$ the firing rates of the presynaptic neurons of the nonstabilized connections. The input signal is then

$$\mathcal{S}_\mathrm{b} = \mathcal{W}_\mathrm{s}\sum_{i=1}^{k}\nu_i + \mathcal{W}_\mathrm{b}\sum_{i=1}^{\mathcal{C}-k}\xi_i. \tag{9}$$

To calculate the average background signal we should average the expression given by Eq. (9) over all the possible values of $k$ and of the firing rates. As $P(k)$ is the probability that $k$ of these connections are stabilized, the probability of having $k$ stabilized connections and rates in the range $(\nu_1,\nu_1+d\nu_1),\ldots,(\nu_k,\nu_k+d\nu_k)$, $(\xi_1,\xi_1+d\xi_1),\ldots,(\xi_{\mathcal{C}-k},\xi_{\mathcal{C}-k}+d\xi_{\mathcal{C}-k})$ is $P(k)\rho(\nu_1)\cdots\rho(\nu_k)\rho(\xi_1)\cdots\rho(\xi_{\mathcal{C}-k})d\nu_1\cdots d\nu_k d\xi_1\cdots d\xi_{\mathcal{C}-k}$. Thus,

$$\begin{aligned}\langle\mathcal{S}_\mathrm{b}\rangle &= \sum_k P(k)\int d\nu_1\cdots\int d\nu_k\int d\xi_1\cdots\int d\xi_{\mathcal{C}-k}\rho(\nu_1)\cdots\rho(\nu_k)\rho(\xi_1)\cdots\rho(\xi_{\mathcal{C}-k})\\ &\quad\times[\mathcal{W}_\mathrm{s}(\nu_1+\cdots+\nu_k)+\mathcal{W}_\mathrm{b}(\xi_1+\cdots+\xi_{\mathcal{C}-k})]\\ &= \sum_k P(k)[\mathcal{W}_\mathrm{s}k\langle\nu\rangle+\mathcal{W}_\mathrm{b}(\mathcal{C}-k)\langle\nu\rangle] = [\mathcal{W}_\mathrm{s}\langle k\rangle+\mathcal{W}_\mathrm{b}(\mathcal{C}-\langle k\rangle)]\langle\nu\rangle,\end{aligned} \tag{10}$$

where we used the fact that $\int\nu\rho(\nu)d\nu = \int\xi\rho(\xi)d\xi = \langle\nu\rangle$. In this equation, we can clearly observe two distinct contributions: one related to stabilized connections, which depends on the mean value $\langle k\rangle$, and the other related to baseline connections, which depends on $\mathcal{C}-\langle k\rangle$. Both these contributions are multiplied by the average firing rate of $\mathcal{P}_1$ neurons, which is related to the nonselectivity of the incoming signal.

The variance of the background signal can be derived by applying its definition:

$$\begin{aligned}\sigma_\mathrm{b}^2 = \langle(\mathcal{S}_\mathrm{b}-\langle\mathcal{S}_\mathrm{b}\rangle)^2\rangle &= \sum_k P(k)\int d\nu_1\cdots\int d\nu_k\int d\xi_1\cdots\int d\xi_{\mathcal{C}-k}\rho(\nu_1)\cdots\rho(\xi_{\mathcal{C}-k})\\ &\quad\times\left[\mathcal{W}_\mathrm{s}\sum_{i=1}^{k}\nu_i+\mathcal{W}_\mathrm{b}\sum_{i=1}^{\mathcal{C}-k}\xi_i-[\mathcal{W}_\mathrm{s}\langle k\rangle+\mathcal{W}_\mathrm{b}(\mathcal{C}-\langle k\rangle)]\langle\nu\rangle\right]^2.\end{aligned} \tag{11}$$

Taking advantage of the equality $\langle k\rangle = k+(\langle k\rangle-k)$, we can rewrite

$$\begin{aligned}\mathcal{W}_\mathrm{s}\langle k\rangle+\mathcal{W}_\mathrm{b}(\mathcal{C}-\langle k\rangle) &= \mathcal{W}_\mathrm{s}k+\mathcal{W}_\mathrm{s}(\langle k\rangle-k)+\mathcal{W}_\mathrm{b}[(\mathcal{C}-k)+(k-\langle k\rangle)]\\ &= \mathcal{W}_\mathrm{s}k+\mathcal{W}_\mathrm{b}(\mathcal{C}-k)+\mathcal{W}_\mathrm{s}(\langle k\rangle-k)+\mathcal{W}_\mathrm{b}(k-\langle k\rangle).\end{aligned} \tag{12}$$

Inserting this last expression in Eq. (11) and rewriting the terms with the multiplicative factors $k$ and $\mathcal{C}-k$ with summations, such as for example $\mathcal{W}_\mathrm{s}k\langle\nu\rangle = \mathcal{W}_\mathrm{s}\sum_{i=1}^{k}\langle\nu\rangle$, we obtain

$$\sigma_\mathrm{b}^2 = \sum_k P(k)\int d\nu_1\cdots\int d\xi_{\mathcal{C}-k}\rho(\nu_1)\cdots\rho(\xi_{\mathcal{C}-k})\left[\mathcal{W}_\mathrm{s}\sum_{i=1}^{k}(\nu_i-\langle\nu\rangle)+\mathcal{W}_\mathrm{b}\sum_{i=1}^{\mathcal{C}-k}(\xi_i-\langle\nu\rangle)+(k-\langle k\rangle)(\mathcal{W}_\mathrm{s}-\mathcal{W}_\mathrm{b})\langle\nu\rangle\right]^2. \tag{13}$$

The mixed terms of the equation above are null because $\int \rho(x)(x-\langle x\rangle)dx = 0$, ergo we can write the variance of the background signal as follows:

$$\begin{aligned}\sigma_{\mathrm{b}}^2 &= \sum_k P(k)\int d\nu_1\cdots\int d\xi_{\mathcal{C}-k}\rho(\nu_1)\cdots\rho(\xi_{\mathcal{C}-k})\\ &\quad\times\left[\mathcal{W}_{\mathrm{s}}^2 k\langle(\nu-\langle\nu\rangle)^2\rangle+\mathcal{W}_{\mathrm{b}}^2(\mathcal{C}-k)\langle(\nu-\langle\nu\rangle)^2\rangle+(\mathcal{W}_{\mathrm{s}}-\mathcal{W}_{\mathrm{b}})^2(k-\langle k\rangle)^2\langle\nu\rangle^2\right]\\ &= \left[\mathcal{W}_{\mathrm{s}}^2\langle k\rangle+\mathcal{W}_{\mathrm{b}}^2(\mathcal{C}-\langle k\rangle)\right]\sigma_\nu^2+(\mathcal{W}_{\mathrm{s}}-\mathcal{W}_{\mathrm{b}})^2\sigma_k^2\langle\nu\rangle^2,\end{aligned} \tag{14}$$

where $\sigma_k^2 = \langle(k-\langle k\rangle)\rangle$. In the previous formula, we note two contributions depending, respectively, on the variance of the firing rate and the variance of the number of stabilized connections. The value of the variance of $k$ is not shown here, but is derived in Appendix D, whereas the variance of the rate is, by definition, $\sigma_\nu^2 = \langle\nu^2\rangle - \langle\nu\rangle^2$.

Now we estimate the average input to a coding neuron of $\mathcal{P}_2$. The neuron receives signals from neurons of $\mathcal{P}_1$ coming from both stabilized and baseline connections. If $\mathcal{C}$ is the number of incoming connections, then the average number of high-rate presynaptic neurons will be $\alpha_1\mathcal{C}$, while those with low rate will be, on average, $\mathcal{C}' = \mathcal{C}(1-\alpha_1)$. Since the input pattern used here for testing is identical to the corresponding training pattern, the $\alpha_1\mathcal{C}$ connections coming from high-rate neurons will certainly be stabilized. The remaining $\mathcal{C}'$ connections come from neurons of $\mathcal{P}_1$ at a low rate, however, they may have been stabilized in one of the other $\mathcal{T}-1$ training patterns. The average number of stabilized connections from low-rate neurons can be calculated using Eq. (8):

$$\langle k'\rangle = \mathcal{C}'p_{\mathcal{T}-1} = \mathcal{C}(1-\alpha_1)p_{\mathcal{T}-1}, \tag{15}$$

where $p_{\mathcal{T}-1}$ represents the probability shown in Eq. (7) but calculated for $\mathcal{T}-1$ examples. For $\alpha_1\alpha_2 \ll 1$, we can observe from Eq. (7) than $p_{\mathcal{T}-1} \simeq p_{\mathcal{T}}$ and thus

$$\langle k'\rangle \simeq \mathcal{C}(1-\alpha_1)p_{\mathcal{T}} = \langle k\rangle(1-\alpha_1). \tag{16}$$

Formalizing what we just discussed and using the definition of $\langle\nu_\ell\rangle$ and $\langle\nu_h\rangle$ given by Eq. (3) we can write

$$\begin{aligned}\langle\mathcal{S}_{\mathrm{c}}\rangle &= \mathcal{W}_{\mathrm{s}}\alpha_1\mathcal{C}\langle\nu_h\rangle+\mathcal{W}_{\mathrm{s}}\langle k'\rangle\langle\nu_\ell\rangle+\mathcal{W}_{\mathrm{b}}(\mathcal{C}'-\langle k'\rangle)\langle\nu_\ell\rangle\\ &= \mathcal{W}_{\mathrm{s}}\alpha_1\mathcal{C}\langle\nu_h\rangle+\mathcal{W}_{\mathrm{s}}\langle k\rangle(1-\alpha_1)\langle\nu_\ell\rangle\\ &\quad+\mathcal{W}_{\mathrm{b}}(\mathcal{C}-\langle k\rangle)(1-\alpha_1)\langle\nu_\ell\rangle\\ &= \mathcal{W}_{\mathrm{s}}\alpha_1\mathcal{C}\langle\nu_h\rangle+[(\mathcal{W}_{\mathrm{s}}-\mathcal{W}_{\mathrm{b}})\langle k\rangle+\mathcal{C}\mathcal{W}_{\mathrm{b}}](1-\alpha_1)\langle\nu_\ell\rangle,\end{aligned} \tag{17}$$

where we used the expression of $\langle k'\rangle$ and the same approximation $p_{\mathcal{T}-1} \simeq p_{\mathcal{T}}$ shown in Eq. (16). Indeed, this equation does not take into account the rewiring process, but only the effect of stabilization. Please see Sec. III B for a derivation of $\langle\mathcal{S}_{\mathrm{c}}\rangle$ which takes into account both the effects of structural plasticity. We identify this case as "with rewiring" to distinguish it from the case in which nonstabilized connections are not pruned and rewired. Indeed, this distinction is useful to estimate the contribution of this mechanism on $\langle\mathcal{S}_{\mathrm{c}}\rangle$.

### A. Poisson distribution of incoming connections per neuron

Hitherto we considered a model in which each neuron of $\mathcal{P}_2$ has a fixed number of incoming connections, i.e., a fixed in-degree, $\mathcal{C}$. However, a more general and more realistic approach would consider $\mathcal{C}$ as a variable across the neurons of $\mathcal{P}_2$ according to an appropriate probability distribution $P(\mathcal{C})$. Here we focus on the case where the number of incoming connections follows a Poisson distribution (i.e., a Poisson-indegree connection rule), although the approach we present here can be easily extended to other distributions. The values of $\langle\mathcal{S}_{\mathrm{c}}\rangle$ and $\langle\mathcal{S}_{\mathrm{b}}\rangle$, previously averaged over the rate $\nu$ and the number of stabilized connections $k$, should be also averaged over the number of incoming connections, so that

$$\begin{aligned}\langle\langle\mathcal{S}_{\mathrm{b}}\rangle_{\nu,k}\rangle_{\mathcal{C}} &= \sum_{\mathcal{C}} P(\mathcal{C})\langle\mathcal{S}_{\mathrm{b}}\rangle_{\nu,k},\\ \langle\langle\mathcal{S}_{\mathrm{c}}\rangle_{\nu,k}\rangle_{\mathcal{C}} &= \sum_{\mathcal{C}} P(\mathcal{C})\langle\mathcal{S}_{\mathrm{c}}\rangle_{\nu,k},\end{aligned} \tag{18}$$

where $\langle\mathcal{S}_{\mathrm{b}}\rangle_{\nu,k}$ is given by Eq. (10) and $\langle\mathcal{S}_{\mathrm{c}}\rangle_{\nu,k}$ is given by Eq. (17). Since these equations are linear in $\mathcal{C}$ and since $\sum_{\mathcal{C}}\mathcal{C}P(\mathcal{C}) = \langle\mathcal{C}\rangle$, Eqs. (10) and (17) would show $\langle\mathcal{C}\rangle$ instead of $\mathcal{C}$ when averaged over the number of incoming connections per neuron.

The variance can be obtained from the equation

$$\mathrm{Var}(\langle\mathcal{S}_{\mathrm{b}}\rangle_{\nu,k,\mathcal{C}}) = \sigma_{\nu,k,\mathcal{C}}^2 = \left\langle\mathcal{S}_{\mathrm{b}}^2\right\rangle_{\nu,k,\mathcal{C}} - (\langle\mathcal{S}_{\mathrm{b}}\rangle_{\nu,k,\mathcal{C}})^2. \tag{19}$$

Knowing that $\langle\sigma_{\mathrm{b}}^2\rangle_{\mathcal{C}} = \langle\langle\mathcal{S}_{\mathrm{b}}^2\rangle_{\nu,k}-\langle\mathcal{S}_{\mathrm{b}}\rangle_{\nu,k}^2\rangle_{\mathcal{C}} = \langle\mathcal{S}_{\mathrm{b}}^2\rangle_{\nu,k,\mathcal{C}} - \langle\langle\mathcal{S}_{\mathrm{b}}\rangle_{\nu,k}^2\rangle_{\mathcal{C}}$ and that $\langle k\rangle = p\mathcal{C}$ we can write

$$\begin{aligned}\sigma_{\nu,k,\mathcal{C}}^2 &= \left\langle\sigma_{\mathrm{b}}^2\right\rangle_{\mathcal{C}}+\left\langle\langle\mathcal{S}_{\mathrm{b}}\rangle_{\nu,k}^2\right\rangle_{\mathcal{C}}-(\langle\mathcal{S}_{\mathrm{b}}\rangle_{\nu,k,\mathcal{C}})^2\\ &= \left\langle\sigma_{\mathrm{b}}^2\right\rangle_{\mathcal{C}}+\{\langle\nu\rangle[\mathcal{W}_{\mathrm{b}}+p(\mathcal{W}_{\mathrm{s}}-\mathcal{W}_{\mathrm{b}})]\}^2[\langle\mathcal{C}^2\rangle-\langle\mathcal{C}\rangle^2]\\ &= \left\langle\sigma_{\mathrm{b}}^2\right\rangle_{\mathcal{C}}+\langle\nu\rangle^2[\mathcal{W}_{\mathrm{b}}+p(\mathcal{W}_{\mathrm{s}}-\mathcal{W}_{\mathrm{b}})]^2\sigma_{\mathcal{C}}^2.\end{aligned} \tag{20}$$

### B. Connection rewiring

In the proposed approach, rewiring is implemented by periodically pruning unstabilized connections and creating new ones. These procedures are performed with a fixed step on the number of training examples, which we will call *rewiring step*, denoted by the letter $r$. The creation of the new connections is made in such a way as to keep the distribution of the number of incoming connections per neuron unchanged. If $k$ is the number of stabilized incoming connections of a neuron of $\mathcal{P}_2$, then after pruning all the nonstabilized connections, $\mathcal{C}-k$ new connections will be created. $\mathcal{C}$ is a fixed number if the fixed-indegree connection rule is used, while it is extracted from a Poisson distribution if the Poisson-indegree rule is selected; in both cases, the presynaptic neurons are randomly extracted from $\mathcal{P}_1$. For this reason, rewiring leaves the expressions of the background signal and of the variance on this signal unchanged, while, as we will see, it modifies the input signal to coding neurons. A diagram of the rewiring process

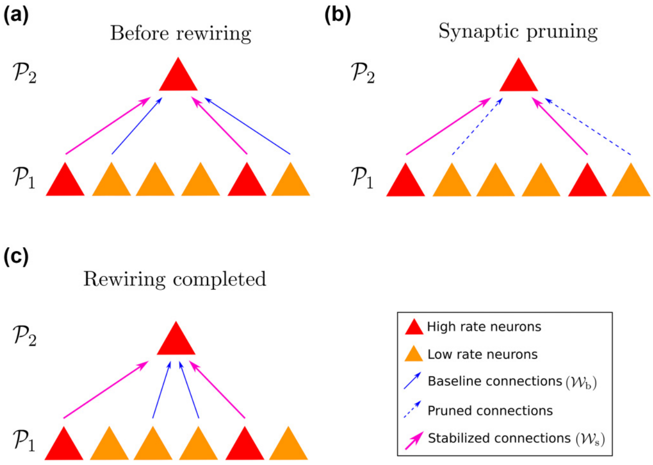

FIG. 2. Diagram of the rewiring process. Red and amber triangles represent high and low rate neurons with average rate $\langle \nu_h \rangle$ and $\langle \nu_\ell \rangle$, respectively. Blue and fuchsia connections represent baseline connections with a weight of $\mathcal{W}_b$ and stabilized connections with a weight of $\mathcal{W}_s$, respectively. (a) Diagram representing the connectivity between a subset of $\mathcal{P}_1$ neurons and a $\mathcal{P}_2$ neuron after synaptic stabilization and before rewiring. (b) Synaptic pruning removes non-stabilized connections every $r$ steps. Pruned connections are depicted as dashed arrows. (c) After pruning, new connections are created, having a randomly chosen presynaptic neuron.

is shown in Fig. 2, which illustrates the activity of a high-rate neuron of $\mathcal{P}_2$ and of the presynaptic neurons of its incoming connections in a training example, and the effect of connection rewiring.

The average number of incoming connections that are stabilized in the current example is equal to the average number of high-rate presynaptic neurons, $\alpha_1 \mathcal{C}$. The average number of incoming connections that are stabilized in other examples after the entire training, $\langle k' \rangle$, is given by Eq. (15):

$$\langle k' \rangle \simeq p_{\mathcal{T}} \mathcal{C}(1-\alpha_1). \tag{21}$$

Let $t$ be the next training index for which rewiring will be applied, and $k'_t$ the number of connections from low-rate neurons that are stabilized before $t$. These connections will not be affected by rewiring, so even in the test phase with the same input pattern they will have low-rate presynaptic neurons. The average value of $k'_t$ is

$$\langle k'_t \rangle = p_t \mathcal{C}(1-\alpha_1), \tag{22}$$

where $p_t$ is given by an expression analogous to the one obtained for $p_{\mathcal{T}}$ [Eq. (7)]

$$p_t = 1 - (1-\alpha_1\alpha_2)^t. \tag{23}$$

However, there will be $k' - k'_t$ connections displaced by rewiring and stabilized in training examples of index greater than $t$. Putting all the contributions together, we obtain the following expression for $\mathcal{S}_c$:

$$\begin{aligned}\langle \mathcal{S}_c \rangle &= \alpha_1 \mathcal{C} \mathcal{W}_s \langle \nu_h \rangle + (\langle k' \rangle - \langle k'_t \rangle)\mathcal{W}_s \langle \nu \rangle + \langle k'_t \rangle \mathcal{W}_s \langle \nu_\ell \rangle \\ &\quad + \mathcal{W}_b(\mathcal{C} - \alpha_1 \mathcal{C} - \langle k' \rangle)\langle \nu \rangle \\ &= \alpha_1 \mathcal{C} \mathcal{W}_s \langle \nu_h \rangle + \langle k' \rangle \mathcal{W}_s \langle \nu \rangle + \mathcal{W}_b[\mathcal{C}(1-\alpha_1) - \langle k' \rangle]\langle \nu \rangle \\ &\quad - \langle k'_t \rangle \mathcal{W}_s(\langle \nu \rangle - \langle \nu_\ell \rangle).\end{aligned} \tag{24}$$

To obtain the average value of $\mathcal{S}_c$ over all examples, $\langle k'_t \rangle$ must be averaged over all values of the index $t$ for which rewiring is done, i.e.,

$$t = ri, \quad i = 0, \ldots, \frac{\mathcal{T}}{r}, \tag{25}$$

where $r$ is the rewiring step and for simplicity we assume that $\mathcal{T}$ is a multiple of $r$ and that there is a final rewiring after the last training step. The average of $p_t$ over the rewiring values of $t$ is

$$\langle p_t \rangle = 1 - \frac{br}{\mathcal{T} + r}, \tag{26}$$

where we introduced a parameter $b$ defined as

$$b = \frac{1 - (1-\alpha_1\alpha_2)^{\mathcal{T}+r}}{1 - (1-\alpha_1\alpha_2)^r}. \tag{27}$$

The complete derivation is shown in Appendix E.

### C. Introduction of noise into input patterns

In a realistic learning model, the test patterns will never be exactly the same as the training ones. The ability of a learning model to generalize is linked to the ability to recognize which training pattern or patterns are most similar to a given test pattern, according to appropriate metrics. To study the generalization capability of the model proposed in this work, the test input patterns were generated starting from the corresponding training input patterns by adding noise, which is represented by a deviation extracted from a given probability distribution with assigned standard deviation. In Appendix F we describe the effect that noise with a truncated Gaussian distribution has on the firing rates and on the variables $\mathcal{S}_b$, $\mathcal{S}_c$, $\sigma_b^2$, and SDNR, and we derive the modified equations.

## IV. COMPUTATIONAL SIMULATIONS OF THE MODEL

The validation of the equations derived in the previous sections was done through simulations with firing-rate-based neuronal network models. The code of the simulator was written in C++ programming language compiled using the GCC compiler ([42]) (version 10.2.0) and with the GSL ([43]) (version 2.7) scientific libraries. The simulations have been performed using the supercomputers Galileo 100 and JUSUF [44]. The networks used for the simulations are generated according to the selected connection rule. In particular, in the case of the fixed-indegree rule, $\mathcal{C}$ incoming connections are created for each neuron of the $\mathcal{P}_2$ population, where $\mathcal{C}$ has a fixed value. In the case of the Poisson-indegree rule, for each neuron of the population $\mathcal{P}_2$ the number of incoming connections $C$ is extracted from a Poisson distribution with mean $\langle C \rangle$. In both cases, the indexes of the presynaptic neurons are randomly extracted on the $\mathcal{P}_1$ population. The connection weights are initially set to the baseline value, $\mathcal{W}_b$. Each training input pattern of the model is generated by extracting, for each neuron of $\mathcal{P}_1$, a random number $\nu$ from a lognormal distribution; if $\nu < \nu_{t,1}$ the rate of the neuron is considered high-rate, otherwise it falls in the low-rate regime. An analogous procedure is used to generate the corresponding contextual stimulus pattern on the neurons of the population

$\mathcal{P}_2$ (using a rate threshold $\nu_{t,2}$). A connection is stabilized in a training example if both the presynaptic and the postsynaptic neuron are in the high-rate regime. Connection rewiring is performed every $r$ training steps, as described in Sec. III B. The test set is generated by randomly extracting $V$ input patterns from the train set. The patterns of the test set are altered by adding noise extracted from a truncated Gaussian distribution with zero mean.

To estimate $\langle \mathcal{S}_b \rangle$ and $\langle \mathcal{S}_c \rangle$ we compute the input of each $\mathcal{P}_2$ neuron as the sum of the rate of the presynaptic neurons of its incoming connections multiplied by the synaptic weights (i.e., $\mathcal{W}_s$ or $\mathcal{W}_b$). The variance $\sigma_b^2$ is evaluated by the formula

$$\sigma_b^2 = \langle \mathcal{S}_b^2 \rangle - \langle \mathcal{S}_b \rangle^2, \tag{28}$$

where the mean values are calculated over the input signals to all the background neurons of $\mathcal{P}_2$. This way, it is possible to obtain the SDNR according to Eq. (5).

## V. RESULTS

This section compares the results of the simulations of the firing rate model with the theoretical predictions described in Sec. III. We present the comparison between the theoretical values of the average input signal to background neurons $\langle \mathcal{S}_b \rangle$, the average input signal to coding neurons $\langle \mathcal{S}_c \rangle$, the variance of the background input signal $\sigma_b^2$ and the signal-difference-to-noise ratio with the values obtained from the simulations. This way, we are able to assess the capability of the population $\mathcal{P}_2$, and thus of the network, to recognize a pattern memorized during the training phase by computing the SDNR using the quantities described above. Last, by setting a threshold to the SDNR, we can extract the maximum amount of patterns the network can correctly recall, i.e., the memory capacity.

Here, we present simulation results with a Poisson-driven number of incoming connections, with $\langle \mathcal{C} \rangle = 5000$. We opted for such an approach since it is more realistic than adopting a fixed amount of connections per neuron. Additionally, the rewiring mechanism is always performed with a rewiring step $r = 100$, when not explicitly specified. Each simulation is repeated 10 times using a different seed for random number generation to ensure the robustness of the simulation results. The values shown in the plots are a result of averaging over the different seeds.

### A. Comparison between theoretical predictions and simulation results

To provide a quantitative estimation of the discrepancy between the theoretical predictions and the simulations, we evaluate their relative error, using the theoretical values as a reference.

The first study we present is oriented towards the estimation of these parameters as a function of the number of training patterns $\mathcal{T}$. As the number of training patterns increases, so does the number of patterns encoded by each neuron. Since $\alpha_2$ is the probability that a neuron of $\mathcal{P}_2$ is in a high-rate level for a single training pattern, on average such neuron will encode $\alpha_2 \mathcal{T}$ patterns of the entire training set. This multiple selectivity of individual neurons is also present in biological neural networks, in which the same neuron can be selective for several stimuli [45].

The test set consists of $V = 1000$ input patterns, generated as described in Sec. III. Thus, the simulation outcome used for our analysis is an average over the entire test set of the $\mathcal{S}_b$, $\mathcal{S}_c$, $\sigma_b^2$ and SDNR values obtained for each test pattern. As described previously, the test input patterns are altered from the corresponding training input patterns by adding noise extracted from a truncated Gaussian distribution, with assigned standard deviation. In this section, we present simulation results and comparisons with theoretical predictions for standard deviation values ranging from 0.2 to 2 Hz. The choice of these values is related to the average of the firing rate distribution of the neuron populations when a pattern is provided to the network. As the firing rate is distributed as a lognormal, we set the highest value of standard deviation to the same value of $\langle \nu_\ell \rangle$ (i.e., 2 Hz), which is near the average firing rate of the whole neuron population. A larger noise would result in fluctuations significantly larger than the average rate of $\mathcal{P}_1$ when the pattern is injected, thus critically altering the rate distribution.

Figure 3 shows the curves obtained using different values for the standard deviation of the noise, together with the relative error with respect to theoretical predictions. Moreover, it also shows a set of results in which the test patterns are not altered by noise to notice the difference that the noise addition makes in the model results. It can be observed that the curves obtained from the simulations are compatible with the theoretical ones for all the noise levels.

Regarding $\langle \mathcal{S}_b \rangle$ and $\langle \mathcal{S}_c \rangle$, the curves corresponding to different noise levels appear perfectly superimposed. This is due to the fact that the noise is driven by a distribution with zero mean, and thus the addition of noise to the quantities represented in the curves does not alter their average (see Appendix F for the details). Regarding $\sigma_b^2$, the values corresponding to different noise levels differ from each other and increase with the standard deviation of the noise, in agreement with the theoretical model.

The relative error between simulation results and theoretical prediction is quite small: for $\langle \mathcal{S}_b \rangle$ and $\langle \mathcal{S}_c \rangle$ the errors span between 0.01% and 0.05%, whereas $\sigma_b^2$ shows a relative error of around 1% for all the simulations performed with a different number of training patterns. The orange horizontal line in Fig. 3(d) represents $\text{SDNR}_{\text{thr}}$, i.e., the minimum value of SDNR that is needed for the network for reliably recalling a pattern, and it is derived using Eq. (6) (see Appendix B for additional details). From that, it is possible to derive the network memory capacity $\mathcal{T}_{\text{max}}$. We can notice that depending on the noise level, $\mathcal{T}_{\text{max}}$ spans from around 28 000 when the noise has a standard deviation of 2 Hz to more than 30 000 when no noise is applied.

However, the addition of noise with fluctuations greater than or comparable to the average firing rate can produce negative rate values for a fraction of the neurons. Considering that negative rate values are not physically possible, this behavior can be corrected in the simulations by simply replacing negative values of the firing rates with zero, i.e., saturating negative rates to zero. This correction is equivalent to having a signal-dependent noise distribution, with an average value greater than zero. This negative rate correction has been applied in

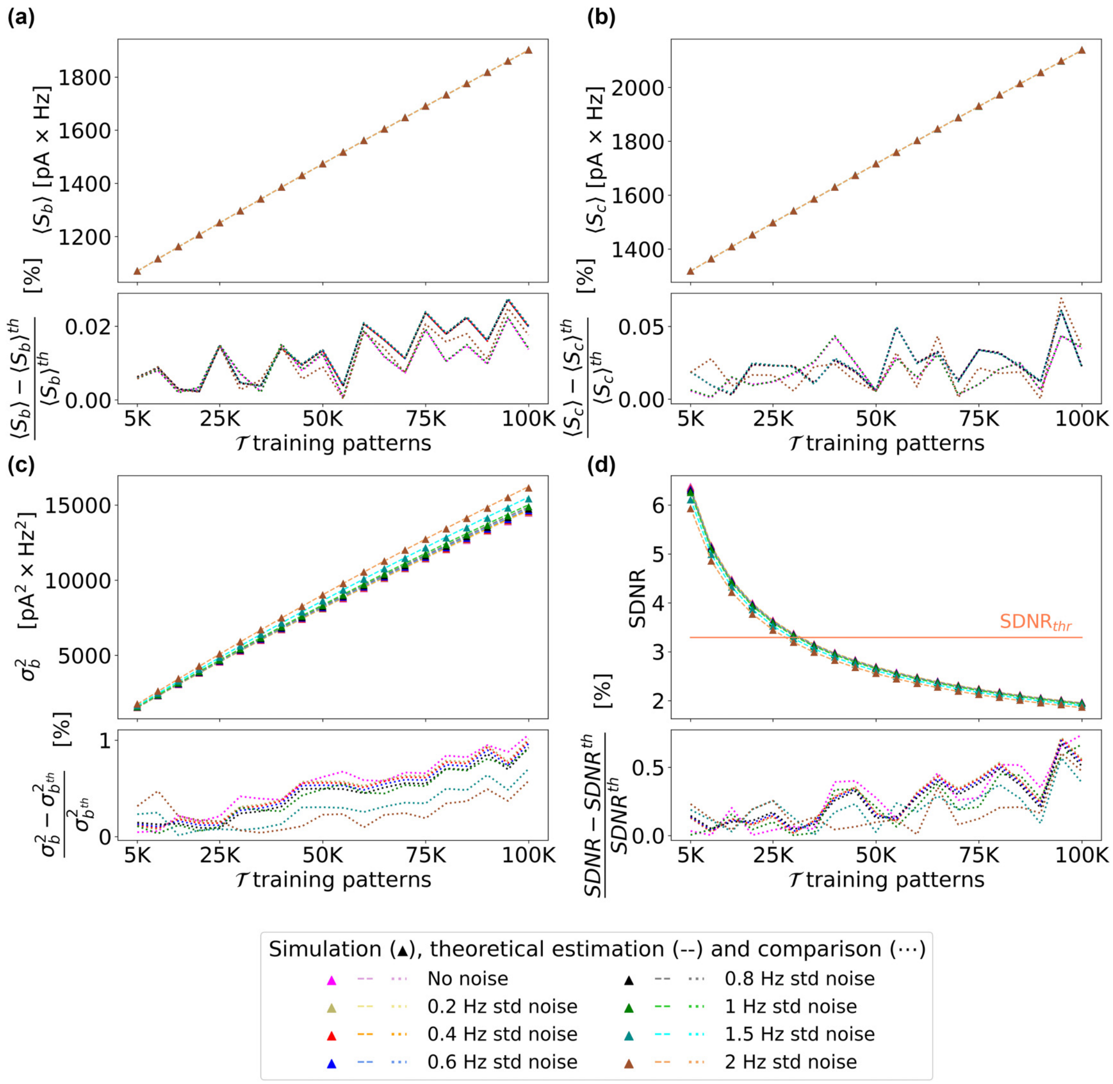

FIG. 3. Values of $\langle \mathcal{S}_\mathrm{b} \rangle$ (a), $\langle \mathcal{S}_\mathrm{c} \rangle$ (b), $\sigma_\mathrm{b}^2$ (c), SDNR (d) and percent errors with respect to the theoretical prediction, as a function of the number of training patterns $\mathcal{T}$. Bigger subplots represent the values of the quantities considered as a function of $\mathcal{T}$ for different noise levels, whereas each smaller subplot represents the percentage error of the values shown in the upper subplot. The different color families identify the simulation and theory results when no noise is provided (magenta-pink), or having a noise standard deviation of 0.2 Hz (dark khaki-khaki), 0.4 Hz (red-orange), 0.6 Hz (blue-light blue), 0.8 Hz (black-gray), 1 Hz (green-light green), 1.5 Hz (dark cyan-cyan), and 2 Hz (sienna-sand). Because the noise has zero mean, the lines corresponding to the theoretical values of $\langle \mathcal{S}_\mathrm{b} \rangle$ and $\langle \mathcal{S}_\mathrm{c} \rangle$ coincide, with noise having an impact only in the evaluation of the variance of the background signal. The orange horizontal line in panel (d) represents the minimum SDNR for the network to be able to correctly recall the patterns during test.

the simulation, however, the current theoretical model is not able to take this effect into account. Since negative values are replaced by zeros, we would expect the average values of $\mathcal{S}_\mathrm{b}$ and $\mathcal{S}_\mathrm{c}$ evaluated by the simulations that exploit saturation to be greater than the values predicted by the theoretical model. Figure 4 shows the behavior of the model with this correction on the neurons firing rate.

As can be seen from the figure, the discrepancies between simulations and theoretical predictions are much higher and can arrive at 10%. We also notice that the level of noise with a standard deviation smaller than the average rate does not give rise to significant discrepancies between simulations and theoretical estimations, also because the number of neurons whose firing rate is saturated to zero is smaller. Moreover, it can be seen that a higher noise level yields a smaller SDNR and, consequently, a lower memory capacity. For instance, the memory capacity with 2 Hz of noise standard deviation is around 25 000 with saturation enabled, whereas without saturation is around 28 000.

Last, in Fig. 3 the relative error of $\sigma_\mathrm{b}^2$ is greater than that shown for $\langle \mathcal{S}_\mathrm{b} \rangle$ and $\langle \mathcal{S}_\mathrm{c} \rangle$. This is due to a simplification used in theoretical derivation to derive the expression of the variance. The values of $\mathcal{S}_\mathrm{b}$ from which we compute the variance are obtained by incoming connections from neurons of $\mathcal{P}_1$, but since connections are created randomly, different neurons of the $\mathcal{P}_2$ population may have presynaptic neurons in common, and therefore their input signals are correlated. The theoretical model does not take this correlation into account. However, in Appendix G we show that the values adopted here make the bias due to this simplification negligible.

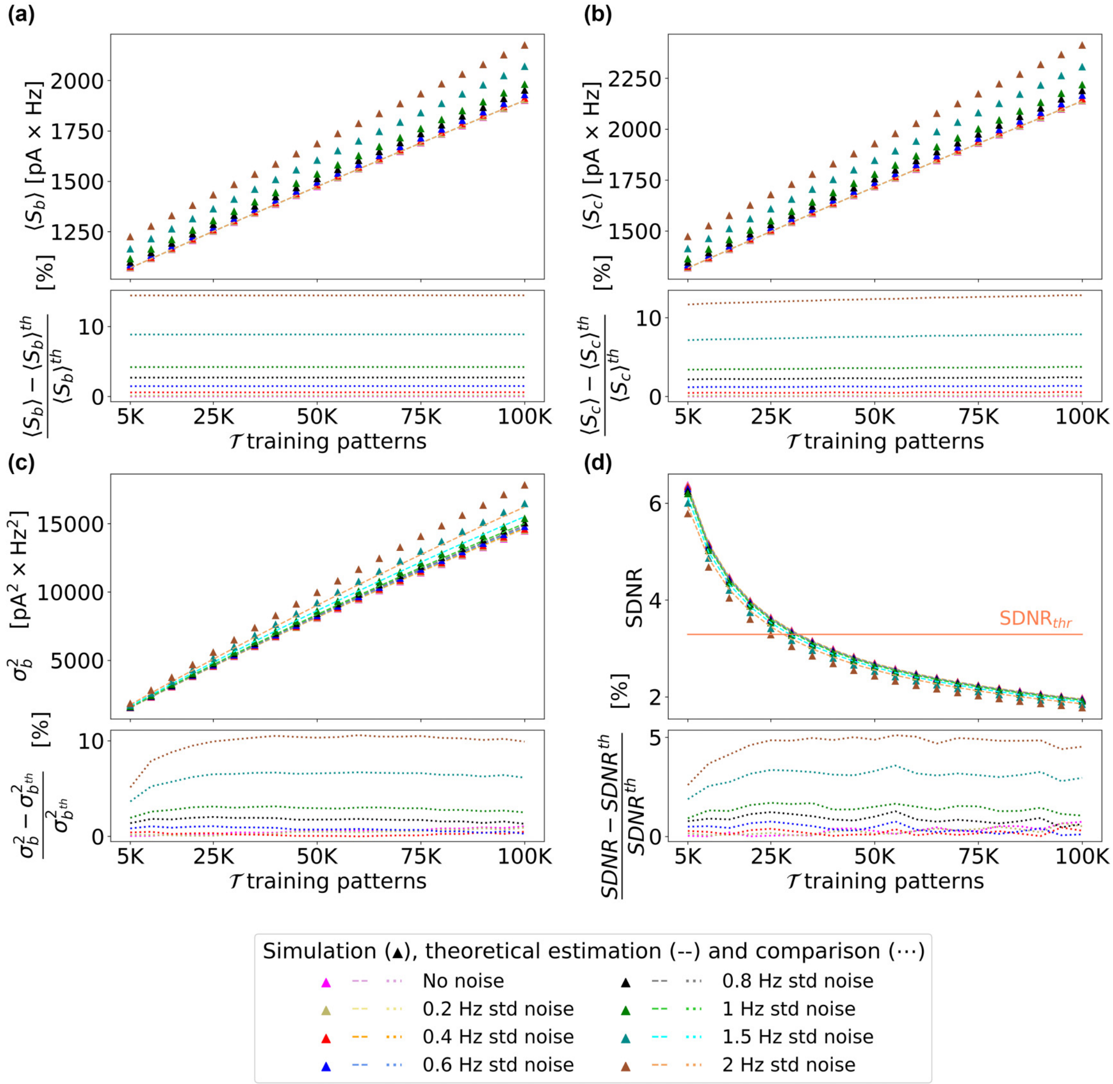

FIG. 4. Values of $\langle\mathcal{S}_\mathrm{b}\rangle$ (a), $\langle\mathcal{S}_\mathrm{c}\rangle$ (b), $\sigma_\mathrm{b}^2$ (c), SDNR (d) and percent errors with respect to the theoretical prediction, as a function of the number of training patterns $\mathcal{T}$ when negative rates due to noise addition are saturated to zero. The different color families identify the simulation and theory results when no noise is provided (magenta-pink), or having a noise standard deviation of 0.2 Hz (dark khaki-khaki), 0.4 Hz (red-orange), 0.6 Hz (blue-light blue), 0.8 Hz (black-gray), 1 Hz (green-light green), 1.5 Hz (dark cyan-cyan), and 2 Hz (sienna-sand). The orange horizontal line in panel (d) represents the minimum SDNR for the network to be able to correctly recall the patterns during test.

### B. Impact of synaptic rewiring

In the simulations discussed so far, the rewiring mechanism was always performed with a rewiring step $r = 100$. This means that every 100 training patterns, all the unstabilized connections are removed, and new connections are created. This operation represents the effect of homeostatic structural plasticity, which aims at keeping the network balanced by reorganizing connections, while activity-dependent structural plasticity focuses on the stabilization of connections.

To motivate the choice of this step for connection rewiring, we show here the results for networks trained for $\mathcal{T} = 10\,000$ patterns with a different rewiring step $r$. We also show the results of a simulation that does not perform rewiring, to highlight the different behavior of a network that combines connection stabilization with periodic rewiring and that of a network that exploits only connection stabilization. Figure 5 shows the results obtained by these simulations using different rewiring intervals.

As can be noticed, the values of $\mathcal{S}_\mathrm{b}$, $\mathcal{S}_\mathrm{c}$, $\sigma_\mathrm{b}^2$, and SDNR do not change significantly as the rewiring step varies. This means that the value of the step $r$ chosen for the connection rewiring has no substantial impact on the results of the simulations. However, some differences emerge when comparing the results of simulations with or without connection rewiring; it can be observed that the signal-difference-to-noise ratio has a lower value when connection rewiring is disabled. This confirms that connection rewiring grants a higher capability of recognizing an input pattern among the several patterns for which the network was trained. Figure 5(c) shows also a discrepancy between the simulation without rewiring and the theoretical estimation. Indeed, the theoretical prediction does not show a dependence from the value of $r$, but the

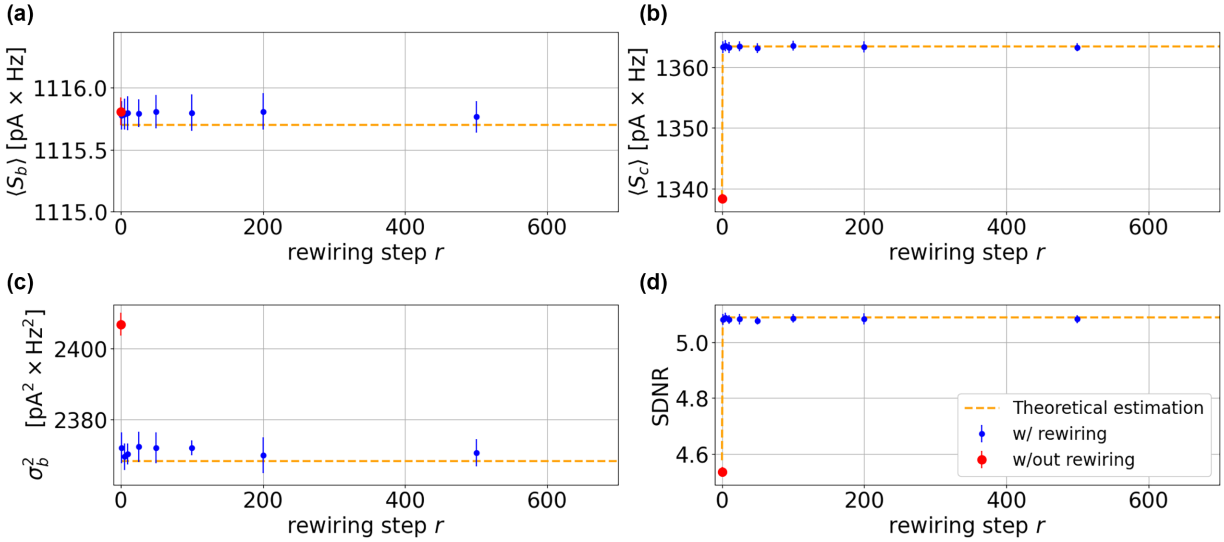

FIG. 5. Values of $\langle \mathcal{S}_\mathrm{b} \rangle$ (a), $\langle \mathcal{S}_\mathrm{c} \rangle$ (b), $\sigma_\mathrm{b}^2$ (c), and SDNR (d) for a network trained with 10000 patterns as a function of the rewiring step $r$. The simulations used a continuous rate distribution and a noisy input driven by truncated Gaussian distribution with a standard deviation of 1 Hz. Red dots indicate the simulation outcome when connection rewiring is disabled, whereas the blue dots show the simulation results with connection rewiring, using different values of $r$. Error bars represent the standard deviation of the mean obtained from 10 simulations using different seeds for random number generation, and the orange dotted lines represent the theoretical estimations.

simulation results show that when rewiring is not performed, the variance of the background values is higher than expected. This is due to the presence of the bias discussed previously (see also Fig. 11 in Appendix G). Indeed, synaptic rewiring has a similar effect as random repositioning of nonstabilized connections, which reduces such a bias.

We also applied a similar protocol for simulations enabling or disabling connection rewiring as a function of the number of training patterns $\mathcal{T}$. The results are shown in Fig. 6.

We can say that the performance of the model is improved when connection rewiring is enabled, and the relative difference between a rewired or just stabilized connectivity increases when increasing the number of training patterns. Regarding memory capacity, we can notice that the simulation with rewiring reaches the threshold value of SDNR for $\mathcal{T}_\mathrm{max} \simeq 30\,000$ (indeed, it is the same result as the one shown in Fig. 3 for 1 Hz noise), whereas it decreases to less than 25 000 when rewiring is disabled, resulting in 20% difference in memory capacity.

The effect of rewiring can become more relevant when a greater number of connections is stabilized at every step (i.e., with greater values of $\alpha_1$ and $\alpha_2$). Furthermore, the importance of the rewiring mechanisms can significantly change when the average number of connections is not constant, but increases or decreases as a result of rewiring itself. This aspect will be explored in future work.

## VI. DISCUSSION

In the previous section, the predictions of the theoretical framework have been compared with the results of simulations performed using feed-forward firing-rate-based neuronal networks. This comparison shows that the proposed framework can accurately predict the values of various relevant quantities for assessing learning and memory capacity in the presence of structural plasticity mechanisms, with differences between theoretical framework and simulation in the order of 1–2%.

The proposed model offers a detailed framework that incorporates various features observed in biological neural networks. In comparison with the classical Hopfield model, our model considers a more realistic scenario. It accounts for the lognormal distribution of neuron firing rates, random connectivity, connection pruning and rewiring, as well as the potentiation and stabilization of connections between neurons with highly correlated activity.

Since the biochemical and biophysical mechanisms underlying structural plasticity are multiple and extremely complex, we opted for a phenomenological approach to capture their main aspects: a simple model of structural plasticity has been exploited, able to represent plasticity processes driven by neuronal activity as well as mechanisms which leads to homeostasis, in agreement with the work of Ref. [7] which divides structural plasticity mechanisms into these two categories. Structural plasticity driven by neuronal activity is achieved through the potentiation and stabilization of synapses connecting neurons that are concurrently at a high-rate level. This process can be triggered by other forms of plasticity that modify synaptic efficacy, such as STDP, followed by mechanisms involving cytoarchitectural changes, such as the creation of novel connections next to the already existing ones.

The homeostatic form of structural plasticity involves a balance between pruning connections that are not utilized over time and the creation of novel connections. This is achieved in the simulations through periodic connection rewiring, which consists of the removal of nonstabilized connections followed by the creation of new connections. Indeed, potentiation and stabilization not only increase the synaptic efficacy but also prevent connections from being pruned in a subsequent rewiring process. This mechanism can reduce the

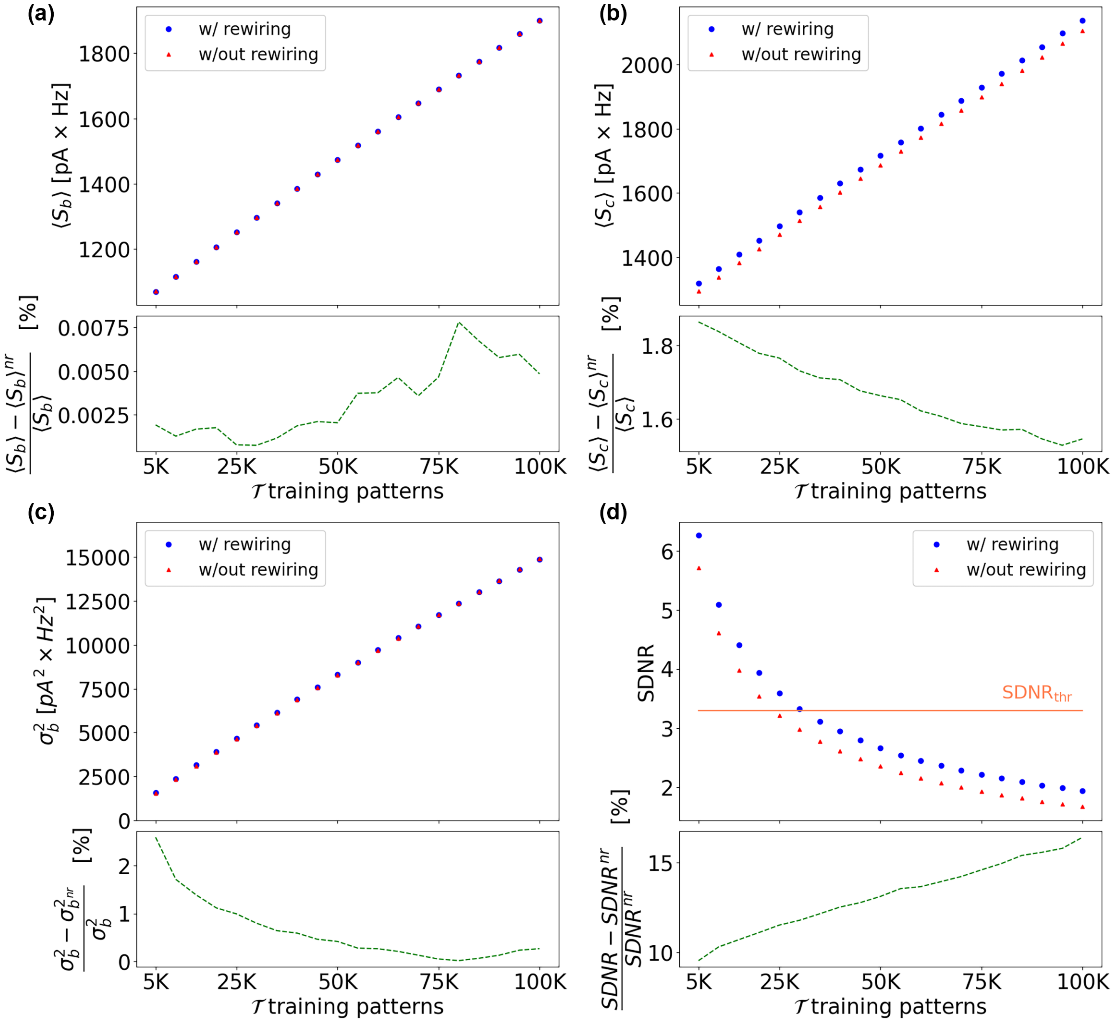

FIG. 6. Values of $\langle \mathcal{S}_b \rangle$ (a), $\langle \mathcal{S}_c \rangle$ (b), $\sigma_b^2$ (c), and SDNR (d) as a function of the number of training patterns $\mathcal{T}$ when rewiring is performed every 100 training patterns (blue dots) and when rewiring is disabled (red triangles). Green lines in the subpanels show the difference between the values obtained with connections rewired or not rewired (indicated with *nr*) as percentage error. The orange horizontal line in panel (d) represents the minimum SDNR for the network to be able to correctly recall the patterns during test.

phenomenon known as catastrophic forgetting, since prevents the deterioration of synaptic contacts already potentiated and stabilized, as also discussed in Ref. [13]. This way, previously learned patterns cannot be forgotten because of further learning and synaptic refinement.

This is a simplified approach to model the structural changes that make a connection stable and ensure long-term memory maintenance. Indeed, structural plasticity mechanisms are also present in brain areas that do not have a significant portion of stable synapses. For instance, in regions of the brain such as the hippocampus, long-term memory storage and maintenance occurs despite the absence of a significant fraction of stable connections. This phenomenon is attributed to the remarkable capacity for synaptic turnover and structural plasticity observed in this region [46]. Studies have shown that new dendritic spines tend to form close to viable presynaptic terminals, with a higher probability of spine pruning when postsynaptic activity is not correlated with presynaptic activity [47]. Therefore, while individual synapses may not persist over time, postsynaptic activity can induce the formation of new dendritic spines near the same site. These newly formed dendritic spines contribute to strengthening the connection between two neurons, facilitating memory retention. In our model, we adopt a simplified but computationally efficient approach to represent these complex dynamics. A single connection can represent multiple synapses between the same pair of neurons, and in the presence of correlated presynaptic and postsynaptic activity, the proliferation of dendritic spines around the same site is depicted by the stabilization and potentiation of this connection.

Moreover, we were able to estimate the memory capacity through the evaluation of the SDNR. As shown in Fig. 3, having a feed-forward network of $\mathcal{N}_1 = \mathcal{N}_2 = 100\,000$ neurons, the network can correctly retrieve between 30 000 and 28 000 patterns, depending on the noise level of the patterns provided during test.

In particular, higher noise provided to the input corresponds to lower values of SDNR, and thus a lower memory capacity. This behavior is also expected from the theoretical model, since noise addition makes the correct pattern retrieval more difficult. However, fluctuations in the same order, or larger, than the average firing rate can significantly alter the distribution. For this reason, we limited the level of noise to a maximum of 2 Hz of standard deviation (i.e., near the average firing rate of the population, which is closer to $\langle \nu_\ell \rangle$).

We also studied the behavior of the network in a more realistic condition related to the neurons firing rate as a result of noise addition. Indeed, the noise can lead to neurons showing negative firing rates, which are not biologically possible. Correcting this issue by saturating the negative firing rates to zero, we noticed in Fig. 4 that the SDNR results are slightly lower to the same simulation with rate correction disabled: for instance, when the noise has a standard deviation of 2 Hz, the network can correctly retrieve around 25 000 patterns, which is about 12% less than in the case without rate correction. This correction is not implemented in the theoretical framework, and thus we notice higher discrepancies in this case, as also shown in Fig. 10 of Appendix F, in which we further increase the noise level to highlight the differences between theory and simulation due to the saturation of negative firing rates. Indeed, a different choice for the values of $\langle \nu_\ell \rangle$ and $\langle \nu_h \rangle$ (and thus a different average rate of the whole distribution) would have an impact on the discrepancies discussed here. In particular, a higher average rate would strongly diminish the amount of neurons having negative firing rate as a result of the noise addition, which saturation is not implemented in the theoretical framework.

Whereas Figs. 3 and 4 show the difference in memory capacity due to noise level, Fig. 6 shows the impact of rewiring on the estimation of the memory capacity: this mechanism is able to increase the SDNR between 10% and 15%, resulting in the increase of memory capacity by around 20% with respect to the same simulation in which rewiring was disabled. Indeed, synaptic rewiring plays an important role in the refinement of the network, together with the fact that stabilized connections are prevented from being pruned.

The framework proposed in this work can be surely extended. It can potentially provide a tool to describe the impact of structural plasticity in cognitive processes such as learning in a large-scale model of the cortex with natural density and plausible characteristics. For instance, the stabilization mechanism can be probability-driven, with a probability depending on the rate of pre- and postsynaptic neurons. This would replace the current deterministic mechanism that requires a firing rate threshold to be exceeded by both neurons to have synaptic stabilization. Moreover, the probability could depend on other variables not necessarily related to the firing rate. In particular, it has been hypothesized that plasticity mechanisms may also depend on the bursting activity of neurons [48,49]. The stabilization probability of a connection could therefore depend, in addition to the firing rate of the presynaptic and postsynaptic neurons, also on their bursting activity.

While our current model focuses on feed-forward connections between two neuron populations, it does not include self-connections within the target population necessary to describe recurrent network dynamics. Consequently, our model serves as a foundational framework rather than a comprehensive model. Future extensions of our work could involve incorporating self-connections and exploring recurrent network dynamics. Additionally, introducing an inhibitory population could enable the modeling of mechanisms like soft winner-take-all, where competition among neuron groups coding different patterns is mediated by inhibitory signals. Indeed, it is known that the mechanisms of competition through lateral inhibition play a key role in biological learning [50]. These enhancements would enrich our model's capabilities and align it more closely with the complexities of biological neural networks.

In this extended model, the theoretical framework can allow to obtain the differential equations governing the dynamics of the activity of the population $\mathcal{P}_2$ and the dependence of the coefficients of these equations on the number of training patterns and on the other model parameters [51]. Such an extension is currently under development and it will be the subject of future work.

Another extension of the model could describe more in detail the mechanisms of synaptic pruning and rewiring. Indeed, connection rewiring as intended in the current model preserves the total number of connections over time, which is a typical behavior of a healthy adult brain [8]. However, to shed light on the importance of these mechanisms in neurological disorders, or to perform studies focused on this mechanism in different life stages, this mechanism should be extended to enable different "speed" for the processes embedded in structural plasticity.

Moreover, it would be interesting to expand this work through simulations of spiking neural networks, to study learning through structural plasticity in more detail. Indeed, simulators such as NEST [52] and its GPU implementation [53] can lead to fast and efficient simulations of large-scale models on supercomputer clusters.

In conclusion, this work intends to propose a theoretical framework for learning through structural plasticity. This framework can describe synaptic potentiation, stabilization, pruning, and rewiring, and includes several features that can be added in a modular fashion. The validation has been performed through simulations with firing-rate-based neuronal network models, showing remarkable compatibility between the results of the simulations and theoretical predictions.

All the simulation code needed to reproduce the results reported in this work, along with the related documentation, is publicly available at Zenodo: [54].

## ACKNOWLEDGMENTS

We thank Prof. Dr. Paolo Ruggerone and Dr. Ke Zuo for the fruitful discussion on structural plasticity mechanisms. This study was supported by the European Union's Horizon 2020 Framework Programme for Research and Innovation under Specific Grant Agreements No. 945539 (Human Brain Project SGA3), No. 785907 (Human Brain Project SGA2), and by the Italian Ministry of University and Research (MUR) Piano Nazionale di Ripresa e Resilienza (PNRR), Project e.INS Ecosystem of Innovation for Next Generation Sardinia—spoke 10-CUP F53C22000430001—MUR Code No. ECS00000038. We acknowledge the use of Fenix Infrastructure resources, which are partially funded from the European Union's Horizon 2020 research and innovation programme through the ICEI project under the Grant Agreement No. 800858.

The paper's conceptualization, data curation, formal analysis, investigation, methodology, software development, validation, visualization, original draft writing, and review/editing were collaboratively conducted by B.G., G.T. and L.S., with B.G. also overseeing funding acquisition, project administration, resources, and supervision.

## APPENDIX A: TABLES AND NEURON DYNAMICS

### 1. Model description

Table I reports a detailed description of the network model.

### 2. Model parameters

Table II reports the model parameters and the default values employed in the simulations.

### 3. Neuron Dynamics

The dynamics of the neurons evolve according to the following differential equation:

$$\tau_i \frac{\nu_i}{dt} = -\nu_i + \Phi\left(\sum_j \mathcal{W}_{ij}\nu_j + b_i\right), \tag{A1}$$

where $\nu_i$ is the firing rate of a single neuron $i$, $\tau_i$ is a time constant expressing the time needed for the neuron to reach a steady-state firing rate when a constant input is given, $\mathcal{W}_{ij}$ is a matrix containing the synaptic weights, $b_i$ is the activation threshold, and $\Phi$ represents the activation function. In the asymptotic regime, in which the average firing rates no longer change over time, the previous equation reduces to

$$\nu_i = \Phi\left(\sum_j \mathcal{W}_{ij}\nu_j + b_i\right). \tag{A2}$$

To determine the neuron output rate we have to choose an activation function.

Assuming that the signal is well below saturation, for simplicity, the neuron response can be modeled by a threshold-linear (or ReLU) function

$$\Phi(x) = \mathcal{A}\max\{0, x\}, \tag{A3}$$

where $\mathcal{A}$ is a multiplicative coefficient. With this choice, the average rates of coding and noncoding neurons of $\mathcal{P}_2$ can be written as

$$\begin{aligned}\langle \nu_{\mathrm{c}} \rangle &= \langle \Phi(\mathcal{S}_{\mathrm{c}} + S_{\mathrm{o}} - S_{\mathrm{thresh}}) \rangle,\\ \langle \nu_{\mathrm{nc}} \rangle &= \langle \Phi(\mathcal{S}_{\mathrm{b}} + S_{\mathrm{o}} - S_{\mathrm{thresh}}) \rangle,\end{aligned} \tag{A4}$$

where $S_{\mathrm{o}}$ is the input signal from (excitatory and/or inhibitory) neuron populations different from $\mathcal{P}_1$ and $S_{\mathrm{thresh}}$ is the activation threshold. Assuming that the total input signal is above the threshold for both coding and background neurons, the average rates will be linear functions of the input signals:

$$\begin{aligned}\langle \nu_{\mathrm{c}} \rangle &= \mathcal{A}(\langle \mathcal{S}_{\mathrm{c}} \rangle + \langle S_{\mathrm{o}} \rangle - S_{\mathrm{thresh}}),\\ \langle \nu_{\mathrm{nc}} \rangle &= \mathcal{A}(\langle \mathcal{S}_{\mathrm{b}} \rangle + \langle S_{\mathrm{o}} \rangle - S_{\mathrm{thresh}}),\end{aligned} \tag{A5}$$

while the variance of the noncoding neuron rate will be

$$\sigma_{\mathrm{nc}}^2 = \mathcal{A}^2(\sigma_{\mathrm{b}}^2 + \sigma_{\mathrm{o}}^2). \tag{A6}$$

TABLE I. Description of the network model (continued).

Summary

| | |
|---|---|
| Populations | $\mathcal{P}_1$, $\mathcal{P}_2$ |
| Connectivity | sparse random connectivity |
| Neurons | firing-rate-based models of point-like neurons |
| Synapses | structural plasticity |
| Input | firing rate pattern extracted from a probability distribution |

Populations

| Name | Elements | Size |
|---|---|---|
| $\mathcal{P}_1$ | point-like neurons | $\mathcal{N}_1$ |
| $\mathcal{P}_2$ | point-like neurons | $\mathcal{N}_2$ |

Neuron

| | |
|---|---|
| Type | firing-rate-based neuron model with linear activation function |
| Description | the state of each neuron is entirely described by the continuous variable $\nu$, which represents its firing rate. We assume that, for a given input pattern to a neuron population, the external stimulus targeting each neuron is stationary. Therefore, the neuron firing rate rapidly converges to a steady value. The firing rate of $\mathcal{P}_1$ neurons is derived from a continuous distribution. During training, the neuron activity of $\mathcal{P}_2$ is entirely dependent on the contextual signal (i.e., the contribution of the input signal from $\mathcal{P}_1$ is neglected). During test, neurons of $\mathcal{P}_2$ show a response that depends only on the signal projected by the connections from $\mathcal{P}_1$. In this phase, $\mathcal{P}_2$ neurons show a linear response (see next section for more details). Depending on the firing rate, a neuron can be considered at a low or high rate regime. We define $\nu_{\mathrm{t},1}$ and $\nu_{\mathrm{t},2}$ the thresholds that distinguish high and low rate neurons in $\mathcal{P}_1$ and $\mathcal{P}_2$. |

Connectivity

| Source | Target | Pattern |
|---|---|---|
| $\mathcal{P}_1$ | $\mathcal{P}_2$ | the incoming connections are generated by randomly extracting the source neurons from $\mathcal{P}_1$; the in-degree (i.e., the number of incoming connections) can be homogeneous, with a fixed number of $\mathcal{C}$ connections per neuron of $\mathcal{P}_2$, or driven by a Poisson distribution;<br>synaptic weights are $\mathcal{W}_{\mathrm{b}}$ for nonstabilized connections and $\mathcal{W}_{\mathrm{s}}$ for stabilized ones, with $\mathcal{W}_{\mathrm{s}} > \mathcal{W}_{\mathrm{b}}$;<br>multiple connections between the same couple of presynaptic and postsynaptic neurons ("multapses") are allowed by default, but they can be disabled. |

TABLE I. *(continued).*

| | |
|---|---|
| Synapse | |
| Type | structural plasticity |
| Description | initial synaptic weights are set to $\mathcal{W}_b$ for all the instantiated connections; when a training pattern is used, considering a connection between a $\mathcal{P}_1$ neuron $i$ and a $\mathcal{P}_2$ neuron $j$:<br>$\mathcal{W}_b \rightarrow \mathcal{W}_s$ if $\nu_i > \nu_{t,1}$ and $\nu_j > \nu_{t,2}$, i.e., when both presynaptic and postsynaptic neurons are at the high rate regime, the connection is stabilized. Once a connection is stabilized, it cannot return to the initial weight. |
| Connection rewiring | |
| Description | after every $r$ training patterns, nonstabilized connections are pruned, and new connections are created: if $k$ is the number of incoming stabilized connections of a neuron of $\mathcal{P}_2$, $\mathcal{C} - k$ new connections will be created, where $\mathcal{C}$ is a fixed number if the fixed-indegree connection rule is used, while it is extracted from a Poisson distribution if the Poisson-indegree rule is selected; in both cases, the presynaptic neurons are randomly extracted from $\mathcal{P}_1$. |
| Input stimulus | |
| Description | firing rate pattern of the neurons of $\mathcal{P}_1$ selected from the training or from the test set. |
| Contextual stimulus | |
| Description | firing rate pattern of the neurons of $\mathcal{P}_2$ selected from the training set; used only in the training phase. |
| Train set | |
| Type | set of $\mathcal{T}$ independent firing-rate patterns of the neurons of $\mathcal{P}_1$ (input stimulus) and $\mathcal{P}_2$ (contextual stimulus) |
| Description | each pattern is randomly generated from predefined firing rate probability distributions (a lognormal probability distribution, in this work). The thresholds $\nu_{t,1}$ and $\nu_{t,2}$ are chosen so that the average rate of the neurons belonging to the low or high rate regime corresponds to $\nu_\ell$ or $\nu_h$, respectively, defined as model parameters. The contribution of the signals projected through the connections from $\mathcal{P}_1$ to $\mathcal{P}_2$ is considered negligible in this case. |
| Test set | |
| Type | set of $V$ firing-rate patterns of the neurons of $\mathcal{P}_1$ (input stimulus) |
| Description | each pattern is randomly extracted from the train set and altered by a noise, which is modeled by a random deviation extracted from a predefined probability distribution, and added to the firing rate of each neuron. In this work, we model the noise using a truncated Gaussian distribution. The response of the neurons of $\mathcal{P}_2$ is entirely dependent on the signals projected through the connections from $\mathcal{P}_1$. |

TABLE II. Model parameters.

| | Network and connectivity | |
|---|---|---|
| Name | Value | Description |
| $\mathcal{N}_1$ | 100 000 | number of neurons of $\mathcal{P}_1$ |
| $\mathcal{N}_2$ | 100 000 | number of neurons of $\mathcal{P}_2$ |
| $\mathcal{C}$ | 5000 | number of connection in-degrees per neuron of $\mathcal{P}_2$. In the case of Poisson-driven in-degree, this parameter represents the average of the Poisson distribution. |
| $\mathcal{T}$ | variable | number of training patterns |
| $r$ | 100 | number of training patterns between two consecutive connection rewiring |
| | Neuron | |
| Name | Value | Description |
| $\nu_\ell$ | 2.0 spikes/s | average firing rate for low rate neurons |
| $\nu_h$ | 50 spikes/s | average firing rate for high rate neurons |
| | Synapse | |
| Name | Value | Description |
| $\mathcal{W}_b$ | 0.1 pA | baseline synaptic weight |
| $\mathcal{W}_s$ | 1 pA | stabilized synaptic weight |
| | Stimulus | |
| Name | Value | Description |
| $\alpha_1$ | 0.001 | probability for a neuron of $\mathcal{P}_1$ of falling in the high rate regime when an input stimulus is injected |
| $\alpha_2$ | 0.001 | probability for a neuron of $\mathcal{P}_2$ of falling in the high rate regime when a contextual stimulus is injected |

The SDNR calculated on the rate will therefore be

$$\mathrm{SDNR}_\nu = \frac{|\langle \nu_c \rangle - \langle \nu_{nc} \rangle|}{\sigma_{nc}} = \frac{|\langle \mathcal{S}_c \rangle - \langle \mathcal{S}_b \rangle|}{\sqrt{\sigma_b^2 + \sigma_o^2}}, \tag{A7}$$

which has an expression similar to that reported in Eq. (5), with the only difference that there is an additional contribution to the noise due to the signal coming from other populations.

It should also be noted that the definitions of SDNR reported in Eqs. (5) and (A7) refer to the mean signal difference between single coding and background neurons. Now, we can evaluate the SDNR on the total input signal to coding neurons and an equivalent number of background neurons. Calling $N_{h,2}$ the mean number of coding neurons in the population

$\mathcal{P}_2$, we can define

$$\mathrm{SDNR_{pop}} = \frac{|N_{h,2}\langle \mathcal{S}_\mathrm{c}\rangle - N_{h,2}\langle \mathcal{S}_\mathrm{b}\rangle|}{\sqrt{N_{h,2}}\sigma_\mathrm{b}} = \frac{\sqrt{\alpha_2 \mathcal{N}_2}|\langle \mathcal{S}_\mathrm{c}\rangle - \langle \mathcal{S}_\mathrm{b}\rangle|}{\sigma_\mathrm{b}}, \tag{A8}$$

where we used Eq. (2) and $\sqrt{N_{h,2}}\sigma_\mathrm{b}$ is the standard deviation of the total input signal to $N_{h,2}$ noncoding neurons. Thus, $\mathrm{SDNR_{pop}}$ scales with the square root of $\alpha_2 N_2$.

## APPENDIX B: SDNR AND MEMORY CAPACITY

In this Appendix, we adopt a methodology similar to the one presented in Ref. [55]. However, in our model, each neuron's probability of being classified as coding or background is distinct ($\alpha_2$ and $\beta_2$, respectively). Additionally, the variance of the signal in input on coding neurons may differ from that observed in background neurons. It is also important to note that in Ref. [55], the SDNR is called the discriminability of the signals and referred to by the symbol $d'$.

For simplicity, we assume that the coding and the background neuron signals can be approximated by Gaussian distributions, as shown in Fig. 7, and we set a threshold halfway between the averages of the two signals.

By setting a threshold, we can notice that a certain fraction of background neurons receive a signal that can be identified as coding. This fraction, identified by the yellow shaded area, represents the false-positive rate (FPR) and can be described by the equation

$$\mathrm{FPR} = \frac{1}{\sqrt{2\pi\sigma_\mathrm{b}^2}} \int_{\frac{\langle \mathcal{S}_\mathrm{c}\rangle + \langle \mathcal{S}_\mathrm{b}\rangle}{2}}^{\infty} e^{\frac{(x-\langle \mathcal{S}_\mathrm{b}\rangle)^2}{2\sigma_\mathrm{b}^2}}\, dx. \tag{B1}$$

Reversely, coding neurons that receive a relatively low signal may be identified as background neurons. This fraction, identified by the red shaded area, represents the false-negative rate (FNR) and can be described by the equation

$$\mathrm{FNR} = \frac{1}{\sqrt{2\pi\sigma_\mathrm{c}^2}} \int_{-\infty}^{\frac{\langle \mathcal{S}_\mathrm{c}\rangle + \langle \mathcal{S}_\mathrm{c}\rangle}{2}} e^{\frac{(x-\langle \mathcal{S}_\mathrm{c}\rangle)^2}{2\sigma_\mathrm{c}^2}}\, dx. \tag{B2}$$

The probability of correct recall $P_\mathrm{C}$ is given by the sum of the rate of correct detections and the rate of correct rejections; equivalently, it is one minus the sum of the rates of false negatives and false positives:

$$P_\mathrm{C} = 1 - \alpha_2 \mathrm{FNR} - \beta_2 \mathrm{FPR}. \tag{B3}$$

The prefactors $\alpha_2$ and $\beta_2$ in Eq. (B3) account for the different probability of having coding neurons and background neurons. So $P_\mathrm{C}$ can be expressed as

$$\begin{aligned} P_\mathrm{C} &= 1 - \alpha_2 \frac{1}{\sqrt{2\pi\sigma_\mathrm{c}^2}} \int_{-\infty}^{\frac{\langle \mathcal{S}_\mathrm{c}\rangle + \langle \mathcal{S}_\mathrm{b}\rangle}{2}} e^{\frac{(x-\langle \mathcal{S}_\mathrm{c}\rangle)^2}{2\sigma_\mathrm{c}^2}}\, dx \\ &\quad - \beta_2 \frac{1}{\sqrt{2\pi\sigma_\mathrm{b}^2}} \int_{\frac{\langle \mathcal{S}_\mathrm{c}\rangle + \langle \mathcal{S}_\mathrm{b}\rangle}{2}}^{\infty} e^{\frac{(x-\langle \mathcal{S}_\mathrm{b}\rangle)^2}{2\sigma_\mathrm{b}^2}}\, dx \\ &= 1 - \alpha_2 \frac{1}{\sqrt{2\pi\sigma_\mathrm{c}^2}} \int_{-\infty}^{\langle \mathcal{S}_\mathrm{c}\rangle} e^{\frac{(x-\langle \mathcal{S}_\mathrm{c}\rangle)^2}{2\sigma_\mathrm{c}^2}}\, dx \\ &\quad + \alpha_2 \frac{1}{\sqrt{2\pi\sigma_\mathrm{c}^2}} \int_{\frac{\langle \mathcal{S}_\mathrm{c}\rangle + \langle \mathcal{S}_\mathrm{b}\rangle}{2}}^{\langle \mathcal{S}_\mathrm{c}\rangle} e^{\frac{(x-\langle \mathcal{S}_\mathrm{c}\rangle)^2}{2\sigma_\mathrm{c}^2}}\, dx \\ &\quad - \beta_2 \frac{1}{\sqrt{2\pi\sigma_\mathrm{b}^2}} \int_{\langle \mathcal{S}_\mathrm{b}\rangle}^{\infty} e^{\frac{(x-\langle \mathcal{S}_\mathrm{b}\rangle)^2}{2\sigma_\mathrm{b}^2}}\, dx \\ &\quad + \beta_2 \frac{1}{\sqrt{2\pi\sigma_\mathrm{b}^2}} \int_{\langle \mathcal{S}_\mathrm{b}\rangle}^{\frac{\langle \mathcal{S}_\mathrm{c}\rangle + \langle \mathcal{S}_\mathrm{b}\rangle}{2}} e^{\frac{(x-\langle \mathcal{S}_\mathrm{b}\rangle)^2}{2\sigma_\mathrm{b}^2}}\, dx. \end{aligned} \tag{B4}$$

So, using the definition of the error function erf, we obtain

$$\begin{aligned} P_\mathrm{C} &= 1 - \frac{1}{2}(\alpha_2 + \beta_2) \\ &\quad + \frac{1}{2}\left[\alpha_2 \mathrm{erf}\left(\frac{\mathrm{SDNR}}{\sqrt{8}}\phi\right) + \beta_2 \mathrm{erf}\left(\frac{\mathrm{SDNR}}{\sqrt{8}}\right)\right] \\ &= \frac{1}{2} + \frac{1}{2}\left[\alpha_2 \mathrm{erf}\left(\frac{\mathrm{SDNR}}{\sqrt{8}}\phi\right) + \beta_2 \mathrm{erf}\left(\frac{\mathrm{SDNR}}{\sqrt{8}}\right)\right], \end{aligned} \tag{B5}$$

where $\phi$ is defined as the ratio between the standard deviation on the input signal of background neurons and the standard deviation on the input signal of coding neurons:

$$\phi = \frac{\sigma_\mathrm{b}}{\sigma_\mathrm{c}}. \tag{B6}$$

Simulations show that this value is approximately $\phi \simeq \frac{1}{3}$ (data not shown). Equation (B5) show also that the probability of correct recall depends on the number of neurons through $\alpha_2$ and $\beta_2$, which take into account the size of the populations $\mathcal{P}_2$. In Eq. (B5), the term with $\alpha_2 = 10^{-3}$ is negligible, leaving only the term with $\beta_2 = (1 - 10^{-3}) \simeq 1$:

$$P_\mathrm{C} \simeq \frac{1}{2}\left[1 + \mathrm{erf}\left(\frac{\mathrm{SDNR}}{\sqrt{8}}\right)\right]. \tag{B7}$$

In the graph below we show a plot of the probability of correct recall as a function of the SDNR using the previous formula.

By setting a recall probability of 95%, as shown in Fig. 8, we obtain a threshold equal to $\mathrm{SDNR_{thr}} \simeq 3.3$. This threshold allows us to calculate the maximum number of patterns stored by our network, as shown in Sec. V, thereby quantifying the memory capacity of our network.

## APPENDIX C: LOGNORMAL DISTRIBUTION OF THE FIRING RATE

The theoretical framework proposed in this work is valid for a generic firing rate probability distribution. However, the model validation presented in the result section is focused on a lognormal distribution, which is a continuous probability distribution of a random variable $\nu$ whose logarithm $\ln(\nu)$ is normally distributed. The probability density function of this distribution is

$$\rho_\mathrm{LN}(\nu) = \frac{1}{\sqrt{2\pi}\sigma\nu} \cdot \exp\left(-\frac{(\ln(\nu) - \mu)^2}{2\sigma^2}\right), \tag{C1}$$

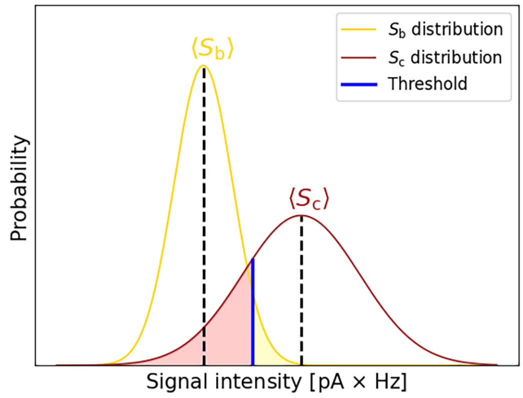

FIG. 7. Schematic representation of signal distributions for coding neurons (dark red line) and background neurons (gold line). The blue line shows the midpoint of the means of both distributions, used to distinguish coding neurons from background neurons. The shaded red area represents the fraction of neurons mistakenly identified as background, while the yellow area represents the fraction of neurons mistakenly identified as coding.

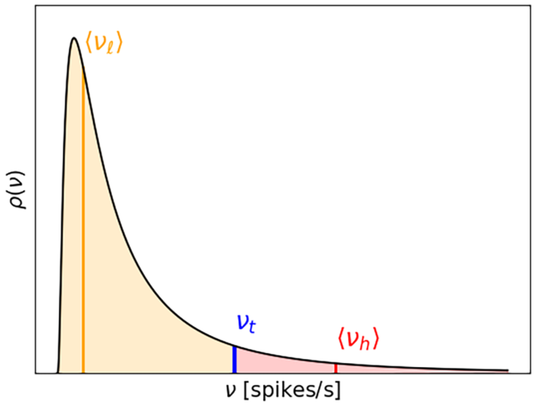

FIG. 9. Lognormal distribution of firing rate. The black solid line indicates the probability distribution, which is divided into two sections by the rate threshold $\nu_t$ (blue, vertical line). The amber band represents the distribution of rate below the threshold, whose mean is $\langle \nu_\ell\rangle$ (amber, vertical line). The red band represents the distribution of neurons whose rate is above the threshold. Here the average of this section is $\langle \nu_h\rangle$ (red, vertical line).

where $\mu$ and $\sigma$ are the mean and standard deviation of $\ln(\nu)$. Expanding Eq. (3) using Eq. (C1) we have

$$\begin{aligned}\langle \nu_\ell\rangle &= \frac{1}{\beta_1}\int_{-\infty}^{y_t} \nu(y)G_{\sigma,\mu}(y)dy\\ &= \frac{1}{\beta_1}\int_{-\infty}^{y_t} e^y\frac{1}{\sqrt{2\pi\sigma^2}}e^{\frac{-(y-\mu)^2}{2\sigma^2}}dy,\\ \langle \nu_h\rangle &= \frac{1}{\alpha_1}\int_{y_t}^{\infty} \nu(y)G_{\sigma,\mu}(y)dy\\ &= \frac{1}{\alpha_1}\int_{y_t}^{\infty} e^y\frac{1}{\sqrt{2\pi\sigma^2}}e^{\frac{-(y-\mu)^2}{2\sigma^2}}dy, \qquad \text{(C2)}\end{aligned}$$

where $y$ is a variable representing the logarithm of the firing rate, $y = \ln(\nu)$, and follows a normal distribution $G_{\sigma,\mu}(y)$, while $y_t$ represents the value linked to the threshold value on the rate $\nu_t$ ($y_t = \ln(\nu_t)$). Figure 9 depicts an example of firing rate distribution, with the threshold and the average values of low and high firing rates.

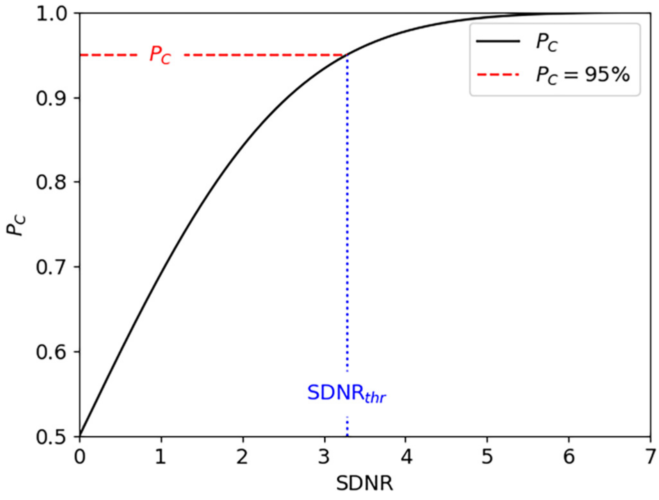

FIG. 8. Probability of correct recall as a function of the signal-difference-to-noise ratio on the single neuron. The blue dotted line represents the threshold $\mathrm{SDNR_{thr}} \simeq 3,3$ corresponding to a 95% probability (red dotted line).

In the logarithmic representation the area of the portion of the Gaussian $G_{\sigma,\mu}(y)$ having $y < y_t$ corresponds to the probability that a neuron has a low rate, $\beta_1$. Therefore, we can write

$$\beta_1 = \int_{-\infty}^{y_t} G_{\sigma,\mu}(y)dy = \frac{1}{2} + \int_{\mu}^{y_t}\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(y-\mu)^2}{2\sigma^2}}dy. \qquad \text{(C3)}$$

Substituting $x = \frac{y-\mu}{\sqrt{2}\sigma}$ we obtain

$$\beta_1 = \frac{1}{2} + \int_0^{\frac{y_t-\mu}{\sqrt{2}\sigma}}\frac{1}{\sqrt{\pi}}e^{-x^2}dx = \frac{1}{2} + \frac{1}{2}\mathrm{erf}\left(\frac{y_t-\mu}{\sqrt{2}\sigma}\right), \qquad \text{(C4)}$$

where with $\mathrm{erf}(x)$ we indicate the error function, defined as

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}dt, \qquad \text{(C5)}$$

and then

$$y_t = \mu + \sqrt{2}\sigma\,\mathrm{erf}^{-1}(2\beta_1 - 1), \qquad \text{(C6)}$$

where $\mathrm{erf}^{-1}$ is the inverse of the erf function. By substituting $z = y - \mu$ we can rewrite $\nu_h$ from Eq. (C2) as

$$\begin{aligned}\langle \nu_h\rangle &= \frac{1}{\alpha_1}\int_{y_t-\mu}^{\infty}\frac{1}{\sqrt{2\pi\sigma^2}}e^{z+\mu-\frac{z^2}{2\sigma^2}}dz\\ &= \frac{1}{\alpha_1}\frac{e^{\mu}}{\sqrt{2\pi\sigma^2}}\int_{y_t-\mu}^{\infty}e^{-\frac{z^2-2\sigma^2 z}{2\sigma^2}}dz\\ &= \frac{1}{\alpha_1}\frac{e^{\mu}}{\sqrt{2\pi\sigma^2}}\int_{y_t-\mu}^{\infty}e^{-\frac{(z-\sigma^2)^2-\sigma^4}{2\sigma^2}}dz\\ &= \frac{1}{\sqrt{2\pi}}\frac{e^{\mu+\frac{\sigma^2}{2}}}{\sigma\alpha_1}\int_{y_t-\mu}^{\infty}e^{\frac{-(z-\sigma^2)^2}{2\sigma^2}}dz. \qquad \text{(C7)}\end{aligned}$$

Making a further substitution $\xi = \frac{z-\sigma^2}{\sqrt{2}\sigma}$ finally we find

$$\begin{aligned}
\langle \nu_h \rangle &= \frac{1}{\sqrt{\pi}} \frac{e^{\mu+\frac{1}{2}\sigma^2}}{\alpha_1} \int_{\frac{y_t-\mu-\sigma^2}{\sqrt{2}\sigma}}^{\infty} e^{-\xi^2} d\xi \\
&= \frac{1}{\sqrt{\pi}} \frac{e^{\mu+\frac{1}{2}\sigma^2}}{\alpha_1} \left( \int_0^{\infty} e^{-\xi^2} d\xi - \int_0^{\frac{y_t-\mu-\sigma^2}{\sqrt{2}\sigma}} e^{-\xi^2} d\xi \right) \\
&= \frac{e^{\mu+\frac{1}{2}\sigma^2}}{\alpha_1} \left[ \frac{1}{2} - \frac{1}{2} \mathrm{erf}\left( \frac{y_t - \mu - \sigma^2}{\sqrt{2}\sigma} \right) \right] \\
&= \frac{\langle \nu \rangle}{2\alpha_1} \left[ 1 - \mathrm{erf}\left( \mathrm{erf}^{-1}(2\beta_1 - 1) - \frac{\sigma}{\sqrt{2}} \right) \right], \qquad \text{(C8)}
\end{aligned}$$

where with $\langle \nu \rangle$ we indicate the average rate, which for the lognormal distribution is given by the known expression

$$\langle \nu \rangle = e^{\mu+\frac{1}{2}\sigma^2}. \qquad \text{(C9)}$$

With similar steps we obtain the expression of $\langle \nu_l \rangle$:

$$\langle \nu_l \rangle = \frac{\langle \nu \rangle}{2\beta_1} \left[ 1 - \mathrm{erf}\left( \mathrm{erf}^{-1}(2\alpha_1 - 1) - \frac{\sigma}{\sqrt{2}} \right) \right]. \qquad \text{(C10)}$$

From these two equations we can finally derive the relationships between $\sigma$, $\beta_1$, $\nu$ and $\nu_h$ or $\nu_l$, respectively:

$$\sigma = \sqrt{2} \left[ \mathrm{erf}^{-1}(2\beta_1 - 1) - \mathrm{erf}^{-1}\left( 1 - \frac{2\alpha_1 \langle \nu_h \rangle}{\langle \nu \rangle} \right) \right], \qquad \text{(C11)}$$

$$\sigma = \sqrt{2} \left[ \mathrm{erf}^{-1}(2\alpha_1 - 1) - \mathrm{erf}^{-1}\left( 1 - \frac{2\beta_1 \langle \nu_l \rangle}{\langle \nu \rangle} \right) \right]. \qquad \text{(C12)}$$

Using Eq. (C9) we can rewrite $\mu$ as

$$\mu = \ln(\langle \nu \rangle) - \frac{\sigma^2}{2}. \qquad \text{(C13)}$$

The average rate $\langle \nu \rangle$ can also be expressed as a function of $\langle \nu_h \rangle$ and $\langle \nu_\ell \rangle$:

$$\langle \nu \rangle = \alpha_1 \langle \nu_h \rangle + \beta_1 \langle \nu_\ell \rangle. \qquad \text{(C14)}$$

The latter equations allow us to express the parameters of the lognormal distribution $\sigma$ and $\mu$ as a function of the parameters of the model, $\alpha_1$, $\langle \nu_h \rangle$ and $\langle \nu_\ell \rangle$.

## APPENDIX D: ESTIMATION OF THE VARIANCE OF $k$

In this Appendix, we will compute the variance on the number of stabilized connections in input to a neuron of $\mathcal{P}_2$ (i.e., $\sigma_k^2$) which, as we have seen previously, enters the formula for the variance on the background signal. For the calculation, we will use the table below which represents the two states, high rate (1) or low rate (0), for a single neuron of the population $\mathcal{P}_2$ and the presynaptic neurons of its input connections in a complete simulation over $\mathcal{T}$ patterns.

TABLE III. Table representing the two states high rate (1) or low rate (0) for a single neuron of the population $\mathcal{P}_2$ and for the presynaptic neurons of its input connections in a complete simulation. Each row represents a training pattern, with index ranging from 0 to $\mathcal{T} - 1$. The first two columns represent the training pattern index $t$ and the rate level $\mathcal{O}$, high or low, of the $\mathcal{P}_2$ neuron. The other columns $\mathcal{I}_j$ represent the rate level, high or low, of the presynaptic neurons connected to the neuron of $\mathcal{P}_2$ through its $C$ incoming connections. The entries for rate levels can be 0 or 1 for low rate and for high rate, respectively; in case of continuous distribution of the rate, the two levels correspond to a rate over or under the threshold $\nu_{\mathrm{t}}$. The table shows the case in which the $\mathrm{P}_2$ neuron is in the high-rate level for the first $m$ examples and in the low-rate level for $\mathcal{T} - m$ examples, while the last $\mathcal{C} - k$ presynaptic neurons are in the low-rate level for the first $m$ examples.

| t | $\mathcal{O}$ | $\mathcal{I}_0$ | $\mathcal{I}_1$ | .... | $\mathcal{I}_{k-1}$ | $\mathcal{I}_k$ | .... | $\mathcal{I}_{\mathcal{C}-1}$ |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | $x_{00}$ | | | $x_{k-1,0}$ | 0 | | 0 |
| 1 | 1 | .. | | | .. | 0 | | 0 |
| 2 | 1 | .. | | | .. | 0 | | 0 |
| .. | .. | .. | | | .. | .. | | .. |
| m-1 | 1 | $x_{0m}$ | | | $x_{k-1,m-1}$ | 0 | | 0 |
| .. | 0 | | | | | | | |
| .. | .. | | | | | | | |
| $\mathcal{T} - 1$ | 0 | | | | | | | |

Given the scheme of Table III, we call:

$\alpha_1$: probability that a neuron of $\mathcal{P}_1$ is in the high-rate level, i.e., probability that a cell of a column $\mathcal{I}_j$ is equal to one;

$\alpha_2$: probability that the neuron of $\mathcal{P}_2$ is in the high-rate level for a given example, i.e., probability that a cell of the column $\mathcal{O}$ is equal to one;

$\alpha_2^m$: probability that the neuron of $\mathcal{P}_2$ is in the high-rate level for the first $m$ patterns;

$(1 - \alpha_2)^{\mathcal{T}-m}$: probability that the neuron of $\mathcal{P}_2$ is in the low-rate level for the remaining $\mathcal{T} - m$ patterns;

$(1 - \alpha_1)^m$: probability that a neuron of $\mathcal{P}_1$ is in the low-rate level for the first $m$ patterns;

$1 - (1 - \alpha_1)^m$: probability that a neuron of $\mathcal{P}_1$ is in the high-rate level for at least one pattern out of the first $m$;

$[1 - (1 - \alpha_1)^m]^k$: probability that every neuron of $\mathcal{P}_1$ of the columns $\mathcal{I}_0, \ldots., \mathcal{I}_{k-1}$ is above threshold for at least one pattern among the first $m$;

$(1 - \alpha_1)^{m(\mathcal{C}-k)}$: probability that every neuron of $\mathcal{P}_1$ of the last $\mathcal{C} - k$ columns is below threshold for the first $m$ patterns.

Now we can combine all these results to calculate the probability that one neuron of $\mathcal{P}_2$ and $k$ presynaptic neurons of its input connections are at the high level for $m$ generic patterns (i.e., not necessarily the first $m$). To do this we have to take into account that the neuron of $\mathcal{P}_2$ will not necessarily be at the high level in the first $m$ examples and that the neurons of $\mathcal{P}_1$ at the high level will not necessarily be the first $k$ (as in the case shown in the table). For this, we have to use binomial coefficients that will take into account all possible combinations in the choice of $m$ patterns out of all possible $\mathcal{T}$ patterns and in the choice of $k$ presynaptic neurons out of a total of $\mathcal{C}$ connections:

$$\begin{aligned}
Q(m, k) = &\binom{\mathcal{T}}{m} p_2^m (1 - \alpha_2)^{\mathcal{T}-m} \binom{\mathcal{C}}{k} \\
&\times [1 - (1 - \alpha_1)^m]^k \; (1 - \alpha_1)^{m(\mathcal{C}-k)}. \qquad \text{(D1)}
\end{aligned}$$

The probability that $k$ connections of a generic neuron of $\mathcal{P}_2$ are stabilized can be calculated by adding $Q(m,k)$ over all possible values of $m$:

$$P(k) = \sum_{m=0}^{\mathcal{T}} Q(m,k), \tag{D2}$$

and the average number of stabilized connections can be calculated as

$$\begin{aligned}\langle k\rangle &= \sum_{m,k} kQ(m,k) = \sum_m \binom{\mathcal{T}}{m} p_2^m (1-\alpha_2)^{\mathcal{T}-m} \sum_{k=0}^{\mathcal{C}} k\binom{\mathcal{C}}{k}\left(1-\beta_1^m\right)^k \beta_1^{m(\mathcal{C}-k)}\\ &= \sum_m \binom{\mathcal{T}}{m} p_2^m (1-\alpha_2)^{\mathcal{T}-m}\mathcal{C}\left(1-\beta_1^m\right)\\ &= \mathcal{C}\left[\sum_m \binom{\mathcal{T}}{m} p_2^m (1-\alpha_2)^{\mathcal{T}-m} - \sum_m \binom{\mathcal{T}}{m}(\alpha_2\beta_1)^m (1-\alpha_2)^{\mathcal{T}-m}\right]\\ &= \mathcal{C}\left[1 - \sum_m \binom{\mathcal{T}}{m}(\alpha_2-\alpha_1\alpha_2)^m (1-\alpha_2)^{\mathcal{T}-m}\right],\end{aligned} \tag{D3}$$

where $\beta_1 = 1-\alpha_1$ and we have used the formula for the mean value of a binomial distribution:

$$\sum_{k=0}^{n} k\binom{n}{k} p^k (1-p)^{n-k} = np. \tag{D4}$$

Using the relationship

$$(a+b)^n = \sum_{k=0}^{n}\binom{n}{k} a^k b^{n-k}, \tag{D5}$$

we can get the expression of $\langle k\rangle$:

$$\langle k\rangle = \mathcal{C}[1-(1-\alpha_1\alpha_2)^{\mathcal{T}}]. \tag{D6}$$

To calculate $\sigma_k^2$ we must also calculate $\langle k^2\rangle$:

$$\begin{aligned}\langle k^2\rangle &= \sum_{m,n} Q(m,k)k^2,\\ \langle k^2\rangle &= \sum_m \binom{\mathcal{T}}{m} p_2^m (1-\alpha_2)^{\mathcal{T}-m}\\ &\quad\cdot \sum_{k=0}^{C}\binom{\mathcal{C}}{k}\left(1-\beta_1^m\right)^k \beta_1^{m(C-k)} k^2.\end{aligned} \tag{D7}$$

After some calculations, analogous to the case of $\langle k\rangle$, we obtain the following formula:

$$\begin{aligned}\langle k^2\rangle =& C(C-1)[1+\alpha_1\alpha_2(\alpha_1-2)]^{\mathcal{T}}\\ &- C(2C-1)(1-\alpha_1\alpha_2)^{\mathcal{T}} + C^2.\end{aligned} \tag{D8}$$

Finally, the variance can be calculated from Eqs. (D6) and (D8) as

$$\sigma_k^2 = \langle k^2\rangle - \langle k\rangle^2. \tag{D9}$$

The values of $\langle k\rangle$ and $\langle k^2\rangle$ derived in the above formulas represent the expected values of $k$ and $k^2$ over all neurons of $\mathcal{P}_2$ for a random input pattern. The average number of stabilized incoming connections of a background neuron in a given example can be computed by observing that, by definition, such neuron cannot have stabilized connections for the considered example, while the average number of connections stabilized in different examples can be computed by replacing $\mathcal{T}$ with $\mathcal{T}-1$ in Eq. (D6). Same applies to $\langle k^2\rangle$. In this work, we assume that $\alpha_1\alpha_2 \ll 1$, and thus $(1-\alpha_1\alpha_2)^{\mathcal{T}} \simeq (1-\alpha_1\alpha_2)^{\mathcal{T}-1}$ and $[1+\alpha_1\alpha_2(\alpha_1-2)]^{\mathcal{T}} \simeq [1+\alpha_1\alpha_2(\alpha_1-2)]^{\mathcal{T}-1}$, and thus the equations above are approximately valid also for computing the average number of stabilized connections of a noncoding neuron and its variance.

## APPENDIX E: CALCULATION OF THE MEAN VALUE OF $k_t'$ OVER THE REWIRING STEPS

In Sec. III B we obtained the expression of $\mathcal{S}_\mathrm{c}$ in the presence of rewiring and we observed that this depends on the parameter $k_t'$, given by [Eq. (22)]:

$$\langle k_t'\rangle = p_t\mathcal{C}(1-\alpha_1), \tag{E1}$$

where, according to Eq. (23), $p_t$ is given by

$$p_t = 1-(1-\alpha_1\alpha_2)^t. \tag{E2}$$

To calculate the mean value of $\mathcal{S}_\mathrm{c}$ for all patterns, $k_t'$ should be averaged over all the values of the training index $t$ for which the rewiring is performed, i.e.,

$$t = ri \quad i = 0,\ldots,\frac{\mathcal{T}}{r}, \tag{E3}$$

where $r$ is the rewiring step and for simplicity we assume that $\mathcal{T}$ is a multiple of $r$ and that there is a final rewiring after the last training step. The average of $p_t$ over the rewiring values of $t$ is

$$\begin{aligned}\langle p_t\rangle &= \frac{\sum_{i=0}^{\mathcal{T}/r} 1-[(1-\alpha_1\alpha_2)^r]^i}{\frac{\mathcal{T}}{r}+1}\\ &= 1-\frac{\sum_{i=0}^{\mathcal{T}/r}[(1-\alpha_1\alpha_2)^r]^i}{\frac{\mathcal{T}}{r}+1} = 1-\frac{br}{\mathcal{T}+r},\end{aligned} \tag{E4}$$

where we introduced a parameter $b$ defined as

$$b = \sum_{i=0}^{\mathcal{T}/r} [(1-\alpha_1\alpha_2)^r]^i = \frac{1-[(1-\alpha_1\alpha_2)^r]^{\mathcal{T}/r+1}}{1-(1-\alpha_1\alpha_2)^r} = \frac{1-(1-\alpha_1\alpha_2)^{\mathcal{T}+r}}{1-(1-\alpha_1\alpha_2)^r}. \tag{E5}$$

## APPENDIX F: NOISE ADDITION DURING TEST PHASE

As anticipated in Sec. III C, to assess the generalization capacity of the model proposed in this work, the test input patterns were generated starting from the corresponding training input patterns by adding noise with a given probability distribution. More specifically, each test pattern is generated by adding to the rate of the corresponding training pattern the contribution of a further extraction from a truncated Gaussian distribution $G(\eta)_{\mu_T,\sigma_T}$. Therefore, the single neuron rate in a test pattern will be given by the following formula:

$$\nu_{\text{tot}} = \nu + \eta, \tag{F1}$$

where $\eta$ is a rate driven by the distribution $G(\eta)_{\mu_T,\sigma_T}$. The input signal to a neuron of the population $\mathcal{P}_2$ can be expressed as the scalar product between the vector $\vec{\mathcal{W}}$ of the weights and the vector $\vec{\nu}_{\text{tot}}$ of the rates of the presynaptic neurons:

$$\vec{\mathcal{W}} \cdot \vec{\nu}_{\text{tot}} = \vec{\mathcal{W}} \cdot \vec{\nu} + \vec{\mathcal{W}} \cdot \vec{\eta}. \tag{F2}$$

Since the noise distribution has zero mean, its contribution to the average values of the signals in input to the coding and background neurons, $\langle \mathcal{S}_c \rangle$ and $\langle \mathcal{S}_b \rangle$, will be zero. However, it will affect the variance of the background signal, $\sigma_b^2$. Since $\nu$ and $\eta$ are independent and random variables, the overall variance will be equal to the sum of the variance in the absence of noise $\sigma_b^2$ [see Eq. (14)] plus the variance due to noise. Thus,

$$\sigma_b^{*2} = \sigma_b^2 + \langle k \rangle (\mathcal{W}_s \sigma_\eta)^2 + (C - \langle k \rangle)(\mathcal{W}_b \sigma_\eta)^2 = \sigma_b^2 + \sigma_\eta^2 \mathcal{C}\big[p\mathcal{W}_s^2 + (1-p)\mathcal{W}_b^2\big], \tag{F3}$$

where $\sigma_\eta^2 = \sigma_T^2$ is the variance of $G(\eta)_{\mu_T,\sigma_T}$. Truncating the Gaussian distribution in the symmetric interval $[-2\sigma, 2\sigma]$, the mean is zero, whereas the variance is

$$\sigma_T^2 = \sigma^2 \left[1 - \frac{4 \cdot e^{-2}}{\sqrt{2\pi}\,\text{erf}(\sqrt{2})}\right]. \tag{F4}$$

As mentioned in the text, adding noise with fluctuations greater than or comparable to the average firing rate can produce negative rate values for a fraction of the neurons. Since negative rate values are not physically possible, this behavior can be corrected in the simulations by simply replacing negative values of the firing rates with zero, i.e., saturating negative rates to zero. In Fig. 4, we noticed that when the standard deviation of the noise is equal to $\langle \nu_\ell \rangle = 2$ Hz the differences between simulation and theoretical framework increase up to 5% for the SDNR and more than 10% for $\langle \mathcal{S}_c \rangle$, $\langle \mathcal{S}_b \rangle$, and $\sigma_b^2$, meaning that the theoretical framework is currently not able to take this effect into account. In particular, the SDNR resulting from simulations is smaller than the theoretical estimation. For better visibility, we show in Fig. 10 the comparison of the SDNR with nonsaturated and saturated negative rates, respectively, using values of noise standard deviation up to 5 Hz.

As can be seen, while the addition of noise without negative rate correction leads to discrepancies between simulations and theory in the order of 0.5% in the case of a noise standard deviation of 5 Hz, the same discrepancy with negative rates saturated to zero is around 10%. The reason behind this discrepancy is related to the fraction of neurons of $\mathcal{P}_1$ that, because of the noise, can result having a negative firing rate. Indeed, the firing rate distribution of the neurons is lognormal with an average near to $\langle \nu_\ell \rangle$, so in case of addition of noise with standard deviation compatible or even greater than this value a large fraction of the neurons can have their firing rate set to zero and thus do not project any input to neurons of $\mathcal{P}_2$ when saturation is enabled. Nevertheless, it should be considered that the noise levels shown in Fig. 10 are relatively high when compared to the average rate used in these simulations, and thus may lead to significant changes in the rate distributions. Indeed, a different choice for the values of $\langle \nu_\ell \rangle$ and $\langle \nu_h \rangle$ (and thus a different average rate of the whole distribution) would have an impact on the discrepancies shown here. In particular, a higher average rate would strongly diminish the amount of neurons having a negative firing rate as a result of the noise addition.

## APPENDIX G: ESTIMATION OF THE BIAS ON $\sigma_b^2$

As mentioned in the main text, Fig. 3 show that the relative error of $\sigma_b^2$ is greater than that shown for $\langle \mathcal{S}_b \rangle$ and $\langle \mathcal{S}_c \rangle$. This is due to the possibility of input correlation related to random connectivity, which is not taken into account by the theoretical framework. In particular, the average number of presynaptic neurons in common to two arbitrary neurons of $\mathcal{P}_2$ depends on the total number of neurons of $\mathcal{P}_1$ and on the number of incoming connections per neuron of $\mathcal{P}_2$. Calling $\mathcal{N}_1 = \mathcal{N}$, we can state that the bias due to this simplification becomes more and more relevant when the ratio $\mathcal{C}/\mathcal{N}$ increases. To estimate this bias as a function of the $\mathcal{C}/\mathcal{N}$ ratio, we performed a series of simulations with a fixed number of training patterns, $\mathcal{T} = 1000$, changing the $\mathcal{C}/\mathcal{N}$ ratio. Figure 11 shows the results of this analysis.

As can be seen in the right panel of Fig. 11, a greater value of $\mathcal{C}/\mathcal{N}$ leads to a higher discrepancy between theoretical prediction and simulation. However, such a ratio, for natural density circuits in the brain, is very far from values of $\mathcal{C}/\mathcal{N}$ near unity. Indeed, a plausible value of the ratio would be less than 0.1, resulting in negligible relative errors.

## APPENDIX H: DISCRETE RATE MODEL

The framework described in this work adopts a general approach for the firing rate distribution when an external stimulus is injected into a neuron population. Indeed, it would be possible to simplify this assumption by considering a discrete distribution, so that neurons can assume, when a stimulus is provided, only two possible values: $\nu_h$ (high rate) or $\nu_\ell$ (low rate). In this Appendix, we show that this approach

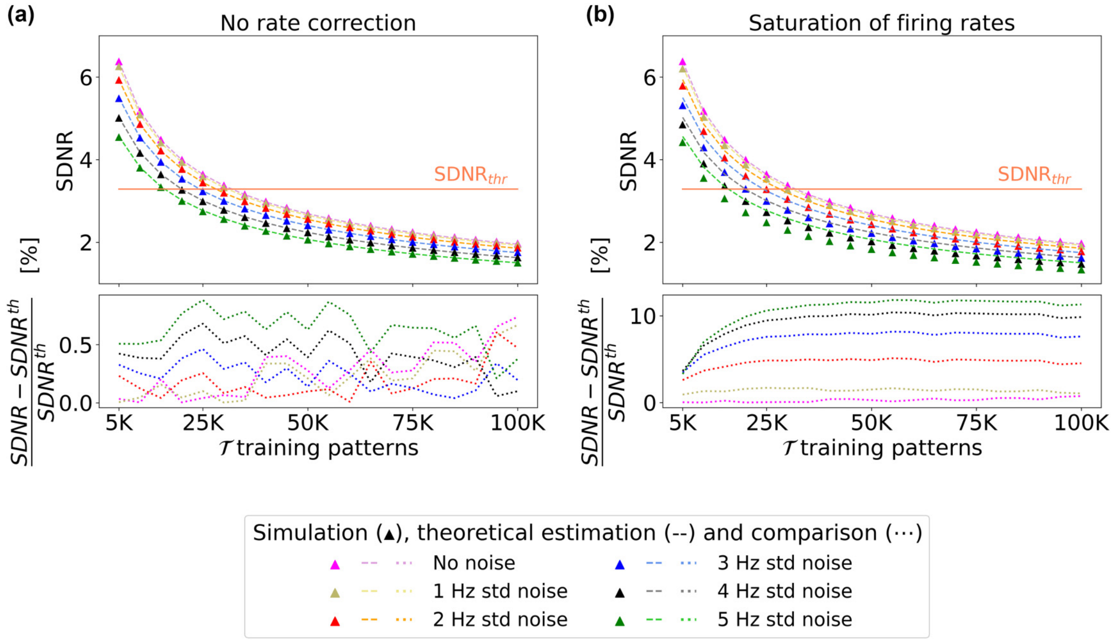

FIG. 10. Values SDNR and percent error with respect to the theoretical predictions, as a function of the number of training patterns $\mathcal{T}$ when firing rates values during test phase are not corrected if negative (a) and when they are saturated to zero (b). The different color families identify the simulation and theory results when no noise is provided (magenta-pink), or having a noise standard deviation of 1 Hz (dark khaki-khaki), 2 Hz (red-orange), 3 Hz (blue-light blue), 4 Hz (black-gray), and 5 Hz (green-light green). The orange horizontal line represents the minimum SDNR for the network to recall the patterns during test correctly.

can provide identical results with respect to the most general derivation for a continuous distribution.

During training, population $\mathcal{P}_1$ is targeted by an external input, so the neuron can have a high or low rate in agreement with the probabilities $\alpha_1$ and $\beta_1 = 1 - -\alpha_1$, of falling in the high or low activity regime. The corresponding pattern for the contextual stimulus is generated similarly, extracting the values of the firing rates of the $\mathcal{P}_2$ neurons, $\nu_h$ or $\nu_\ell$, with probabilities $\alpha_2$ and $\beta_2 = 1 - -\alpha_2$, respectively.

A connection will be stabilized in a training example if both the presynaptic and the postsynaptic neurons assume a high firing rate $\nu_h$, and the probability that a connection is stabilized in at least one of the $\mathcal{T}$ training examples can be derived at the same way as shown in Sec. III, so that

$$p_{\mathcal{T}} = 1 - (1 - \alpha_1\alpha_2)^{\mathcal{T}}. \tag{H1}$$

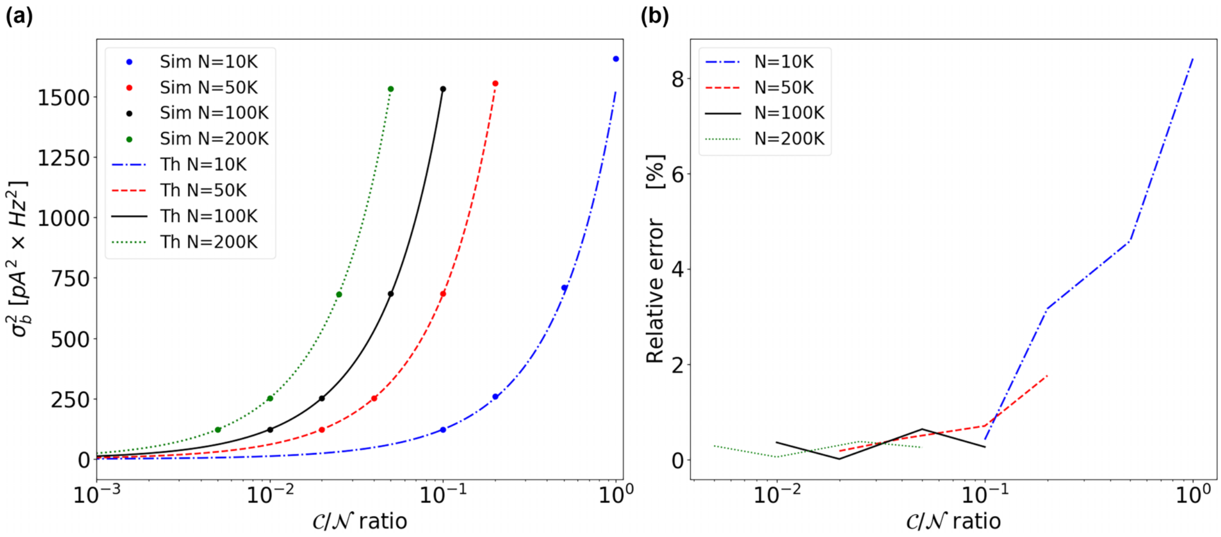

FIG. 11. (a) Comparison between theoretical and experimental values of $\sigma_b^2$ as a function of the $\mathcal{C}/\mathcal{N}$ ratio for different values of $\mathcal{N}$. Lines represent the theoretical prediction (Th), whereas dots represent the values obtained from the simulation (Sim). (b) Relative error between simulation results and theoretical prediction.

The average number of stabilized connections is the same as shown in Eq. (8), so

$$\langle k\rangle = \mathcal{C}[1-(1-\alpha_1\alpha_2)^{\mathcal{T}}] = \mathcal{C}p_{\mathcal{T}}. \tag{H2}$$

The test set consists of $V$ firing-rate patterns of the neurons of $\mathcal{P}_1$, randomly extracted from the $\mathcal{T}$ input patterns of the train set. In the following, the patterns are unaltered, thus each input pattern of the test set is identical to an input pattern of the train set. The contextual stimuli are not used in the testing phase.

The average rate of the neurons in the population $\mathcal{P}_1$ is

$$\langle \nu\rangle = \alpha_1\nu_h + (1-\alpha_1)\nu_\ell. \tag{H3}$$

The input signal targeting a background neuron of $\mathcal{P}_2$ is equal to the weighted sum of the signals coming from the $\mathcal{C}$ connections:

$$\mathcal{S}_\text{b} = \mathcal{W}_\text{s}\sum_{i=1}^{k}\nu_i + \mathcal{W}_\text{b}\sum_{i=1}^{\mathcal{C}-k}\xi_i, \tag{H4}$$

where $\mathcal{C}$ is the number of incoming connections, $k$ is the number of stabilized connections, $\nu_i$ are the firing rates of the neurons connected to the stabilized connections and $\xi_i$ are the firing rates of the neurons connected to the unstabilized connections. From the linearity of $\mathcal{S}_\text{b}$ with respect to $\nu_i$ and $\xi_i$ and from the fact that the rates of presynaptic neurons have the same mean value $\langle\nu\rangle$, it follows that

$$\langle \mathcal{S}_\text{b}\rangle = [\mathcal{W}_\text{s}\langle k\rangle + \mathcal{W}_\text{b}(\mathcal{C}-\langle k\rangle)]\langle\nu\rangle. \tag{H5}$$

From these results, we can now calculate the variance on the background signal, which is defined as

$$\sigma_b^2 = \langle(\mathcal{S}_\text{b} - \langle\mathcal{S}_\text{b}\rangle)^2\rangle. \tag{H6}$$

Using Eqs. (H4) and (H5), we can compute the variance as

$$\sigma_b^2 = \left\langle\left[\mathcal{W}_\text{s}\sum_{i=1}^{k}\nu_i + \mathcal{W}_\text{b}\sum_{i=1}^{\mathcal{C}-k}\xi_i - [\mathcal{W}_\text{s}\langle k\rangle + \mathcal{W}_\text{b}(\mathcal{C}-\langle k\rangle)]\langle\nu\rangle\right]^2\right\rangle. \tag{H7}$$

Taking advantage of the equality $\langle k\rangle = k + (\langle k\rangle - k)$, we can rewrite

$$\begin{aligned}\mathcal{W}_\text{s}\langle k\rangle &+ \mathcal{W}_\text{b}(\mathcal{C}-\langle k\rangle)\\ &= \mathcal{W}_\text{s}k + \mathcal{W}_\text{s}(\langle k\rangle - k) + \mathcal{W}_\text{b}[(\mathcal{C}-k) + (k-\langle k\rangle)]\\ &= \mathcal{W}_\text{s}k + \mathcal{W}_\text{b}(\mathcal{C}-k) + \mathcal{W}_\text{s}(\langle k\rangle - k) + \mathcal{W}_\text{b}(k-\langle k\rangle).\end{aligned} \tag{H8}$$

Inserting this last expression in Eq. (H7) and rewriting the terms with the multiplicative factors $k$ and $\mathcal{C}-k$ with summations, such as, for example, $\mathcal{W}_\text{s}k\langle\nu\rangle = \mathcal{W}_\text{s}\sum_{i=1}^{k}\langle\nu\rangle$, we obtain

$$\sigma_b^2 = \left\langle\left[\mathcal{W}_\text{s}\sum_{i=1}^{k}(\nu_i - \langle\nu\rangle) + \mathcal{W}_\text{b}\sum_{i=1}^{\mathcal{C}-k}(\xi_i - \langle\nu\rangle) + (k-\langle k\rangle)(\mathcal{W}_\text{s}-\mathcal{W}_\text{b})\langle\nu\rangle\right]^2\right\rangle. \tag{H9}$$

Taking into account that the mixed terms go to zero since $\sum_i\langle(x_i - \langle x\rangle)\rangle = 0$, setting $\sum_i(x_i-\langle x\rangle)^2 = \sigma_x^2$, we will have that

$$\sigma_b^2 = [\mathcal{W}_\text{s}^2\langle k\rangle + \mathcal{W}_\text{b}^2(\mathcal{C}-\langle k\rangle)]\sigma_\nu^2 + (\mathcal{W}_\text{s}-\mathcal{W}_\text{b})^2\sigma_k^2\langle\nu\rangle^2, \tag{H10}$$

where $\sigma_k^2 = \langle(k-\langle k\rangle)^2\rangle$. In the previous formula we note two contributions depending, respectively, on the variance of the firing rate and on the variance of the number of stabilized connections. The value of the variance of $k$ is derived in Appendix D, whereas the variance of the rate is $\sigma_\nu^2 = \langle\nu^2\rangle - \langle\nu\rangle^2$. Please note that the variance of the rate differs from the one adopted in the case of continuous rate distribution.

To estimate the average input to a coding neuron of $\mathcal{P}_2$, we can follow the same derivation shown in Sec. III, so that

$$\begin{aligned}\langle\mathcal{S}_\text{c}\rangle &= \mathcal{W}_\text{s}\mathcal{C}\alpha_1\nu_h + \mathcal{W}_\text{s}\langle k'\rangle\nu_\ell + \mathcal{W}_\text{b}(\mathcal{C}'-\langle k'\rangle)\nu_\ell\\ &= \mathcal{W}_\text{s}\mathcal{C}\alpha_1\nu_h + \mathcal{W}_\text{s}\langle k\rangle(1-\alpha_1)\nu_\ell\\ &\quad + \mathcal{W}_\text{b}(\mathcal{C}-\langle k\rangle)(1-\alpha_1)\nu_\ell\\ &= \mathcal{W}_\text{s}\mathcal{C}\alpha_1\nu_h + [(\mathcal{W}_\text{s}-\mathcal{W}_\text{b})\langle k\rangle + \mathcal{C}\mathcal{W}_\text{b}](1-\alpha_1)\nu_\ell.\end{aligned} \tag{H11}$$

The previous formula does not consider the rewiring of the connections; the effect of rewiring is taken into account in Eq. (24). In the discrete rate approximation, the equation changes so that instead of having the average high or low rate we simply have the values $\nu_h$ and $\nu_\ell$. Now, it would be possible to obtain the signal-difference-to-noise-ratio (SDNR) using Eq. (5).

Having derived the values of $\langle\mathcal{S}_\text{b}\rangle$, $\sigma_\text{b}^2$, and $\langle\mathcal{S}_\text{c}\rangle$ in the case of lognormal and discrete firing rate distribution, it would be interesting to compare the results of the simulations employing the two approaches. As we discussed, the main difference in calculating $\langle\mathcal{S}_\text{b}\rangle$ and $\langle\mathcal{S}_\text{c}\rangle$ with a continuous rate model versus the discrete model is that the discrete values of $\nu_l$ and $\nu_h$ are replaced, respectively, by the average values of the rate below and above threshold, calculated on the continuous probability distribution. However, the variance of the background signal differs in the two models, because it depends on the variance of the rate, $\sigma_\nu^2$, which depends on the firing rate distribution adopted.

Figure 12 shows the comparison of the simulation outcomes using discrete and lognormal rate values, using the same parameters needed to produce the data shown in Fig. 3.

We can see that the curves of $\langle\mathcal{S}_\text{b}\rangle$ and $\langle\mathcal{S}_\text{c}\rangle$ obtained from the simulations using the continuous firing rate distribution are superimposed on those obtained using the discrete model; this is because the choice of the threshold on the lognormal distribution is done so that the average values for low and high rate, $\langle\nu_l\rangle$ and $\langle\nu_h\rangle$, correspond to the values adopted in the discrete rate model. However, the variance of the background signal $\sigma_\text{b}^2$ differs in the two models, because it depends on the variance of the firing rate, $\sigma_\nu$, which is different in the two cases. This leads also to the different behavior of the SDNR.

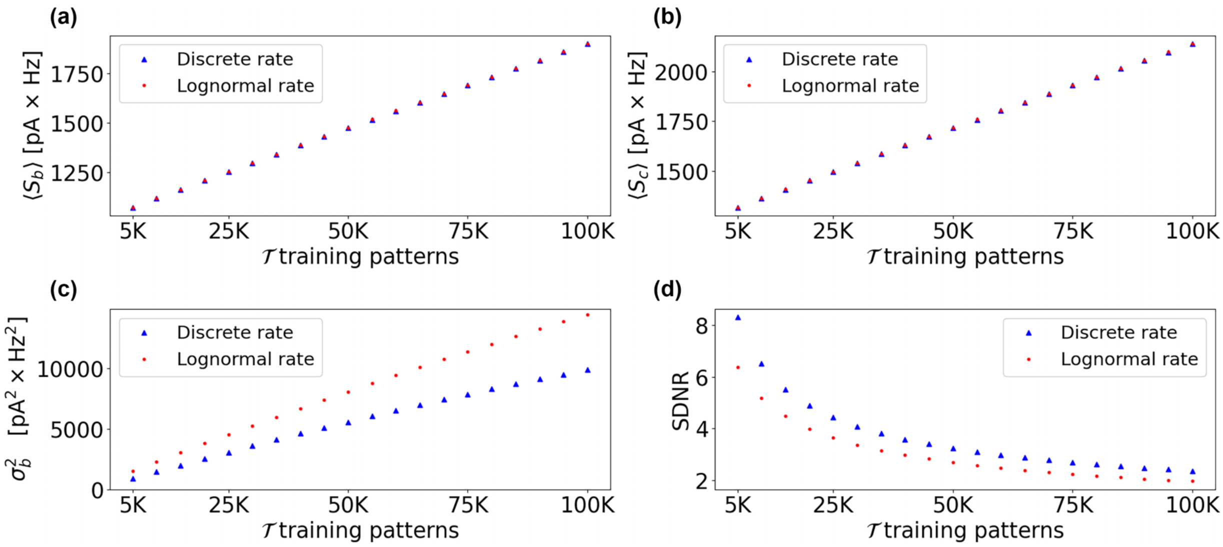

FIG. 12. Comparison between $\langle \mathcal{S}_\mathrm{b} \rangle$ (a), $\langle \mathcal{S}_\mathrm{c} \rangle$ (b), $\sigma_\mathrm{b}^2$ (c), and SDNR (d) obtained from simulations using discrete (blue line) or lognormal (red dotted line) firing rate distribution. The values are given as a function of the number of training patterns $\mathcal{T}$, and no noise is applied during test. The lognormal rate simulation results are the same as the ones of Fig. 3 labeled as "no noise."

[1] R. Lamprecht and J. LeDoux, Structural plasticity and memory, Nat. Rev. Neurosci. **5**, 45 (2004).

[2] C. Tetzlaff, C. Kolodziejski, I. Markelic, and F. Wörgötter, Timescales of memory, learning, and plasticity, Biol. Cybern. **106**, 715 (2012).

[3] M. Butz, F. Wörgötter, and A. van Ooyen, Activity-dependent structural plasticity, Brain Res. Rev. **60**, 287 (2009).

[4] M. Fu and Y. Zuo, Experience-dependent structural plasticity in the cortex, Trends Neurosci. **34**, 177 (2011).

[5] M. P. Mattson, Neurotransmitters in the regulation of neuronal cytoarchitecture, Brain Res. Rev. **13**, 179 (1988).

[6] D. A. Richards, J. M. Mateos, S. Hugel, V. de Paola, P. Caroni, B. H.Gähwiler, and R. A. McKinney, Glutamate induces the rapid formation of spine head protrusions in hippocampal slice cultures, Proc. Natl. Acad. Sci. USA **102**, 6166 (2005).

[7] M. Fauth and C. Tetzlaff, Opposing effects of neuronal activity on structural plasticity, Front. Neuroanat. **10**, 75 (2016).

[8] P. R. Huttenlocher, Synaptic density in human frontal cortex—Developmental changes and effects of aging, Brain Res. **163**, 195 (1979).

[9] S. Navlakha, A. L. Barth, and Z. Bar-Joseph, Decreasing-rate pruning optimizes the construction of efficient and robust distributed networks, PLoS Comput. Biol. **11**, 1 (2015).

[10] K. Zito and K. Svoboda, Activity-dependent synaptogenesis in the adult mammalian cortex, Neuron **35**, 1015 (2002).

[11] J. Sakai, How synaptic pruning shapes neural wiring during development and, possibly, in disease, Proc. Natl. Acad. Sci. USA **117**, 16096 (2020).

[12] D. B. Chklovskii, B. W. Mel, and K. Svoboda, Cortical rewiring and information storage, Nature (London) **431**, 782 (2004).

[13] A. Knoblauch, E. Körner, U. Körner, and F. T. Sommer, Structural synaptic plasticity has high memory capacity and can explain graded amnesia, catastrophic forgetting, and the spacing effect, PLoS One **9**, 1 (2014).

[14] A. Knoblauch and F. T. Sommer, Structural plasticity, effectual connectivity, and memory in cortex, Front. Neuroanat. **10**, 63 (2016).

[15] T. Bourgeron, A synaptic trek to autism, Curr. Opin. Neurobiol. **19**, 231 (2009).

[16] C. E. Moyer, M. A. Shelton, and R. A. Sweet, Dendritic spine alterations in schizophrenia, Neurosci. Lett. **601**, 46 (2015).

[17] L. A. Glantz and D. A. Lewis, Decreased dendritic spine density on prefrontal cortical pyramidal neurons in schizophrenia, Arch. Gen. Psych. **57**, 65 (2000).

[18] J. J. Hutsler and H. Zhang, Increased dendritic spine densities on cortical projection neurons in autism spectrum disorders, Brain Res. **1309**, 83 (2010).

[19] M. Pagani, N. Barsotti, A. Bertero, S. Trakoshis, L. Ulysse, A. Locarno, I. Miseviciute, A. D. Felice, C. Canella, K. Supekar, A. Galbusera, V. Menon, R. Tonini, G. Deco, M. V. Lombardo, M. Pasqualetti, and A. Gozzi, mTOR-related synaptic pathology causes autism spectrum disorder-associated functional hyperconnectivity, Nat. Commun. **12**, 6084 (2021).

[20] H. Markram, E. Muller, S. Ramaswamy, M. W. Reimann, M. Abdellah, C. A. Sanchez, A. Ailamaki, L. Alonso-Nanclares, N. Antille, S. Arsever, G. A. A. Kahou, T. K. Berger, A. Bilgili, N. Buncic, A. Chalimourda, G. Chindemi, J.-D. Courcol, F. Delalondre, V. Delattre, S. Druckmann, R. Dumusc, J. Dynes, S. Eilemann, E. Gal, M. E. Gevaert, J.-P. Ghobril, A. Gidon, J. W. Graham, A. Gupta, V. Haenel, E. Hay, T. Heinis, J. B. Hernando, M. Hines, L. Kanari, D. Keller, J. Kenyon, G. Khazen, Y. Kim, J. G. King, Z. Kisvarday, P. Kumbhar, S. Lasserre, J.-V. L. Bé, B. R. Magalhães, A. Merchán-Pérez, J. Meystre, B. R. Morrice, J. Muller, A. Muñoz-Céspedes, S. Muralidhar, K. Muthurasa, D. Nachbaur, T. H. Newton, M. Nolte, A. Ovcharenko, J. Palacios, L. Pastor, R. Perin, R. Ranjan, I. Riachi, J.-R. Rodríguez, J. L. Riquelme, C. Rössert, K. Sfyrakis, Y. Shi, J. C. Shillcock, G. Silberberg, R. Silva, F.

Tauheed, M. Telefont, M. Toledo-Rodriguez, T. Tränkler, W. V. Geit, J. V. Díaz, R. Walker, Y. Wang, S. M. Zaninetta, J. DeFelipe, S. L. Hill, I. Segev, and F. Schürmann, Reconstruction and simulation of neocortical microcircuitry, Cell **163**, 456 (2015).

[21] D. J. Amit and N. Brunel, Model of global spontaneous activity and local structured activity during delay periods in the cerebral cortex, Cereb. Cortex **7**, 237 (1997).

[22] C. Capone, M. di Volo, A. Romagnoni, M. Mattia, and A. Destexhe, State-dependent mean-field formalism to model different activity states in conductance-based networks of spiking neurons, Phys. Rev. E **100**, 062413 (2019).

[23] M. Carlu, O. Chehab, L. Dalla Porta, D. Depannemaecker, C. Héricé, M. Jedynak, E. Köksal Ersöz, P. Muratore, S. Souihel, C. Capone, Y.Zerlaut, A.Destexhe, and M.di Volo, A mean-field approach to the dynamics of networks of complex neurons, from nonlinear integrate-and-fire to Hodgkin–Huxley models, J. Neurophysiol. **123**, 1042 (2020).

[24] J. J. Hopfield, Neurons with graded response have collective computational properties like those of two-state neurons, Proc. Natl. Acad. Sci. USA **81**, 3088 (1984).

[25] P. S. Leon, S. A. Knock, M. M. Woodman, L. Domide, J. Mersmann, A. R. McIntosh, and V. Jirsa, The virtual brain: A simulator of primate brain network dynamics, Front. Neuroinform. **7**, 10 (2013).

[26] A. Renart, N. Brunel, and X.-J. Wang, in *Computational Neuroscience: A Comprehensive Approach*, edited by J. Feng (Chapman & Hall, Boca Raton, 2004), Chap. 15, pp. 431–490.

[27] S. di Santo, P. Villegas, R. Burioni, and M. A.Muñoz, Landau–ginzburg theory of cortex dynamics: Scale-free avalanches emerge at the edge of synchronization, Proc. Natl. Acad. Sci. USA **115**, E1356 (2018).

[28] H. R. Wilson and J. D. Cowan, Excitatory and inhibitory interactions in localized populations of model neurons, Biophys. J. **12**, 1 (1972).

[29] M. Tsodyks, K. Pawelzik, and H. Markram, Neural networks with dynamic synapses, Neural Comput. **10**, 821 (1998).

[30] R. Gütig, R. Aharonov, S. Rotter, and H. Sompolinsky, Learning input correlations through nonlinear temporally asymmetric Hebbian plasticity, J. Neurosci. **23**, 3697 (2003).

[31] G. Qiang Bi and M. Ming Poo, Synaptic modification by correlated activity: Hebb's postulate revisited, Annu. Rev. Neurosci. **24**, 139 (2001).

[32] C. Capone, C. Lupo, P. Muratore, and P. S. Paolucci, Burst-dependent plasticity and dendritic amplification support target-based learning and hierarchical imitation learning, in *Proceedings of the 39th International Conference on Machine Learning*, Proceedings of Machine Learning Research Vol. 162, edited by K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato (PMLR, Baltimore, Maryland, USA, 2022), pp. 2625–2637.

[33] B. Golosio, C. De Luca, C. Capone, E. Pastorelli, G. Stegel, G. Tiddia, G. De Bonis, and P. S. Paolucci, Thalamo-cortical spiking model of incremental learning combining perception, context and NREM-sleep, PLoS Comput. Biol. **17**, e1009045 (2021).

[34] G. Mongillo, O. Barak, and M. Tsodyks, Synaptic theory of working memory, Science **319**, 1543 (2008).

[35] S. Song, K. D. Miller, and L. F. Abbott, Competitive Hebbian learning through spike-timing-dependent synaptic plasticity, Nat. Neurosci. **3**, 919 (2000).

[36] G. Tiddia, B. Golosio, V. Fanti, and P. S. Paolucci, Simulations of working memory spiking networks driven by short-term plasticity, Front. Integr. Neurosci. **16**, 972055 (2022).

[37] R. Spiess, R. George, M. Cook, and P. U. Diehl, Structural plasticity denoises responses and improves learning speed, Front. Comput. Neurosci. **10**, 93 (2016).

[38] B. Pakkenberg, D. Pelvig, L. Marner, M. J. Bundgaard, H. J. G. Gundersen, J. R. Nyengaard, and L. Regeur, Aging and the human neocortex, Exp. Gerontol. **38**, 95 (2003).

[39] A. Roxin, N. Brunel, D. Hansel, G. Mongillo, and C. van Vreeswijk, On the distribution of firing rates in networks of cortical neurons, J. Neurosci. **31**, 16217 (2011).

[40] Y. Yang, X.-b. Wang, M. Frerking, and Q. Zhou, Spine expansion and stabilization associated with long-term potentiation, J. Neurosci. **28**, 5740 (2008).

[41] D. Muller, I. Nikonenko, P. Jourdain, and S. Alberi, LTP, memory and structural plasticity, Curr. Mol. Med. **2**, 605 (2002).

[42] https://github.com/gcc-mirror/gcc.

[43] https://www.gnu.org/software/gsl/.

[44] B. V. S. Vieth, JUSUF: Modular tier-2 supercomputing and cloud infrastructure at Jülich Supercomputing Centre, J. Large-Scale Res. Facilities **7**, A179 (2021).

[45] M. Rigotti, O. Barak, M. R. Warden, X.-J. Wang, N. D. Daw, E. K. Miller, and S. Fusi, The importance of mixed selectivity in complex cognitive tasks, Nature (London) **497**, 585 (2013).

[46] T. Pfeiffer, S. Poll, S. Bancelin, J. Angibaud, V. K. Inavalli, K. Keppler, M. Mittag, M. Fuhrmann, and U. V.Nägerl, Chronic 2P-STED imaging reveals high turnover of dendritic spines in the hippocampus in vivo, eLife **7**, e34700 (2018).

[47] M. Goldin, M. Segal, and E. Avignone, Functional plasticity triggers formation and pruning of dendritic spines in cultured hippocampal networks, J. Neurosci. **21**, 186 (2001).

[48] D. A. Butts, P. O. Kanold, and C. J. Shatz, A burst-based Hebbian learning rule at retinogeniculate synapses links retinal waves to activity-dependent refinement, PLoS Biol. **5**, e61 (2007).

[49] A. Payeur, J. Guerguiev, F. Zenke, B. A. Richards, and R. Naud, Burst-dependent synaptic plasticity can coordinate learning in hierarchical circuits, Nat. Neurosci. **24**, 1010 (2021).

[50] R. Coultrip, R. Granger, and G. Lynch, A cortical model of winner-take-all competition via lateral inhibition, Neural Netw. **5**, 47 (1992).

[51] L. Sergi, Teoria di campo medio dell'apprendimento nei sistemi neurali biologici attraverso la plasticità sinaptica strutturale, Master's thesis, Department of Physics, University of Cagliari, Italy (2023).

[52] M.-O. Gewaltig and M. Diesmann, Nest (Neural Simulation Tool), Scholarpedia **2**, 1430 (2007).

[53] G. Tiddia, B. Golosio, J. Albers, J. Senk, F. Simula, J. Pronold, V. Fanti, E. Pastorelli, P. S. Paolucci, and S. J.van Albada, Fast simulation of a multi-area spiking network model of macaque cortex on an MPI-GPU cluster, Front. Neuroinform. **16**, 883333 (2022).

[54] https://zenodo.org/doi/10.5281/zenodo.12067622.

[55] S. R. Schultz, Signal-to-noise ratio in neuroscience, Scholarpedia **2**, 2046 (2007).