



## OPEN ACCESS

## EDITED BY

Matteo Cirillo,  
University of Rome Tor Vergata, Italy

## REVIEWED BY

Paolo Moretti,  
University of Erlangen Nuremberg, Germany  
Thomas Cusick,  
University at Buffalo, United States

## \*CORRESPONDENCE

P. Milani,  
✉ paolo.milani@unimi.it  
D. E. Galli,  
✉ davide.galli@unimi.it

<sup>†</sup>These authors have contributed equally to this work

RECEIVED 20 March 2024

ACCEPTED 25 April 2024

PUBLISHED 30 May 2024

## CITATION

Martini G, Tentori E, Mirigliano M, Galli DE, Milani P and Mambretti F (2024), Efficiency and controllability of stochastic boolean function generation by a random network of non-linear nanoparticle junctions.  
*Front. Phys.* 12:1400919.  
doi: 10.3389/fphy.2024.1400919

## COPYRIGHT

© 2024 Martini, Tentori, Mirigliano, Galli, Milani and Mambretti. This is an open-access article distributed under the terms of the [Creative Commons Attribution License \(CC BY\)](#). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

# Efficiency and controllability of stochastic boolean function generation by a random network of non-linear nanoparticle junctions

G. Martini<sup>1†</sup>, E. Tentori<sup>1,2†</sup>, M. Mirigliano <sup>1</sup>, D. E. Galli <sup>1\*</sup>, P. Milani <sup>1\*</sup> and F. Mambretti<sup>3,4</sup>

<sup>1</sup>CIMAINA and Department of Physics, University of Milan, Milan, Italy, <sup>2</sup>Padova Neuroscience Center, Università degli Studi di Padova, Padova, Italy, <sup>3</sup>Dipartimento di Fisica e Astronomia, Università degli Studi di Padova, Padova, Italy, <sup>4</sup>Atomistic Simulations group, Italian Institute of Technology, Genova, Italy

Amid efforts to address energy consumption in modern computing systems, one promising approach takes advantage of random networks of non-linear nanoscale junctions formed by nanoparticles as substrates for neuromorphic computing. These networks exhibit emergent complexity and collective behaviors akin to biological neural networks, characterized by self-organization, redundancy, and non-linearity. Based on this foundation, a generalization of  $n$ -inputs devices has been proposed, where the associated weights depend on all the input values. This model, called *receptor*, has demonstrated its capability to generate Boolean functions as output, representing a significant breakthrough in unconventional computing methods. In this work, we characterize and present two actual implementations of this paradigm. One approach leverages the nanoscale properties of cluster-assembled Au films, while the other utilizes the recently introduced Stochastic Resistor Network (SRN) model. We first provide a concise overview of the electrical properties of these systems, emphasizing the insights gained from the SRN regarding the physical processes within real nanostructured gold films at a coarse-grained scale. Furthermore, we present evidence indicating the minimum complexity level required by the SRN model to achieve a stochastic dynamics adequate to effectively model a novel component for logic systems. To support our argument that these systems are preferable to conventional random search algorithms, we discuss quantitative criteria based on Information-theoretic tools. This suggests a practical means to steer the stochastic dynamics of the system in a controlled way, thus focusing its random exploration where it is most useful.

## KEYWORDS

receptor, stochastic resistor network, unconventional computing, boolean functions, nanostructured films

# 1 Introduction

The urgent need of a substantial improvement in environmental footprint, associated with the capability of performing complex tasks with ever-increasing amounts of data, has raised the interest on Unconventional Computing Systems (UCS) as a viable alternative to data processing based on CMOS technology and von Neumann architecture [1, 2]. One of the common traits of UCS is the exploitation of the complexity emerging from generic underlying physical substrates for computation. Among several approaches, we just recall molecular, optical, chemical and *in materia* computing [3–12]. To this end, a new paradigm recently emerged in this field is the receptron (reservoir perceptron), a generalization of perceptron [13, 14]. While perceptron's weights associated with each input are independent and must be individually adjusted to give the desired output [15], in the receptron model a network of highly interconnected nonlinear objects adjusts its conduction pathway topology depending on the input stimuli, such that the weighting process is not just responding separately to each stimulus but is sensitive to their co-location. Network's weights are, then, functions of a convolution of the spatial location of the input topology. Thus, a receptron can be used as a binary classification tool to map the inputs from different electrodes in two possible sets labeled by 0 and 1. If the inputs are also binary, the device can be employed as a Boolean function generator [16]. The potential of such a device resides in its capability to generate a complete set of Boolean functions (including those non-linearly separable) of  $n$  variables for classification tasks [17], without having been explicitly programmed for that. Practical realizations of the receptron include devices made of a nanostructured metallic film interconnecting a generic pattern of electrical contacts. In particular, a promising hardware implementation of the receptron is represented by cluster-assembled nanostructured Au films, which have recently been shown to have a complex resistive switching activity together with potentiation behavior [17–21]. Such devices, which have been extensively characterized experimentally, are here discussed - together with an abstract model of receptron - as a relevant example of this new computing paradigm. The internal nanostructure of these gold films is repeatedly altered by the application of input stimuli, and we anticipate that this property candidates such objects as suitable devices for computation.

The Boolean function generation is designed as a dynamic process, thanks to the possibility of a plastic rearrangement of the nanostructured conductive medium: the nanojunctions' network that constitute the film can be reconfigured via the administration of stimuli of amplitude larger than a threshold voltage  $V_{th}$ , during the device *reprogramming* [20, 22, 23]. The specific modification of the conductive paths will then depend on the particular electrode configuration [17]. Subsequently, in the *compute* step, the sample resistance is probed via the application of sub-threshold voltage input signals, which do not alter the structure and topology of the conduction paths. Thanks to the extremely large number of conductive states available to the system, the device can be effectively used as a binary classifier: it behaves as a nonvolatile, reconfigurable function generator without any previous training [16]. The stochasticity of the output calls for the identification of a

few hyperparameters of the experimental setup that can be tuned to improve the predictability of the Boolean outputs.

An abstract model known as Stochastic Resistor Network (SRN) enables a thorough examination of the electrical properties of this nanostructured system, via a large three-dimensional (3D) regular resistor network [18]. The conductance of a small portion of the film is represented as an edge in an abstract graph conceptualization. Each edge weight can evolve to a new discretized conduction value, according to stochastic local physical rules, shaping the collective dynamics of the system. Probabilistic updates are deduced from empirical effects, including local thermal dissipation and nonlinear conduction mechanisms. Going beyond one-to-one mapping of all the conductive junctions between gold nanoparticles, the SRN offers a coarse-grained representation of the intricate dynamics within the film's internal structure. Integrating the network with very high conductive edges that act as electrodes readily provides an implementation of the receptron model. The system inherently exhibits diverse dynamic behaviors tailored to different magnitudes of external electrical stimulation.

Here we formally define the problem of characterizing and governing the receptron stochasticity, striving to gain improved performances, together with a detailed description of its implementation. With the insights provided by the SRN model, we assess the susceptibility to the parameters that regulate the main dynamical features of the receptron reprogramming dynamics. The comparison between the simulations and the experimental system suggests the constructive characteristics that are needed for a receptron to sample the Boolean function space with increased efficiency. Using tools derived from Information theory, we prove that, to some extent, the effectiveness of the receptron in generating Boolean functions can be boosted via specific reprogramming protocols.

## 2 Materials and methods

### 2.1 The receptron

The digital receptron [13] is a reprogrammable, nonlinear threshold logic gate. Digital inputs  $\{x_i\}$  undergo weighting before being thresholded to obtain a binary output, akin to a conventional perceptron. However, unlike a simple linear combination through diagonal weights  $\{w_i\}$ , here the weighting involves also cross terms ( $w_{ij}, w_{ijk}, \dots$ ) of a sparse weight tensor with rank equal to the number of inputs (refer to [13] for a formal definition of the receptron model). In the case of three inputs, its functioning is described by the following equation:

$$Out = \vartheta \left( \sum_{i=1}^3 w_i x_i + \sum_{i>j=1}^3 w_{ij} x_i x_j + w_{123} x_1 x_2 x_3 - T \right) \quad (1)$$

where  $\vartheta(x)$  is the Heaviside step function,  $T$  the threshold. Thanks to the mixing components, its output is not restricted to linearly separable functions, in contrast to the perceptron. A receptron functioning is programmed by the set of internal weights: computation is the result of the convolution between the memorized weights configuration and the probing stimulus (Figure 1A). A reprogramming procedure guides the system to

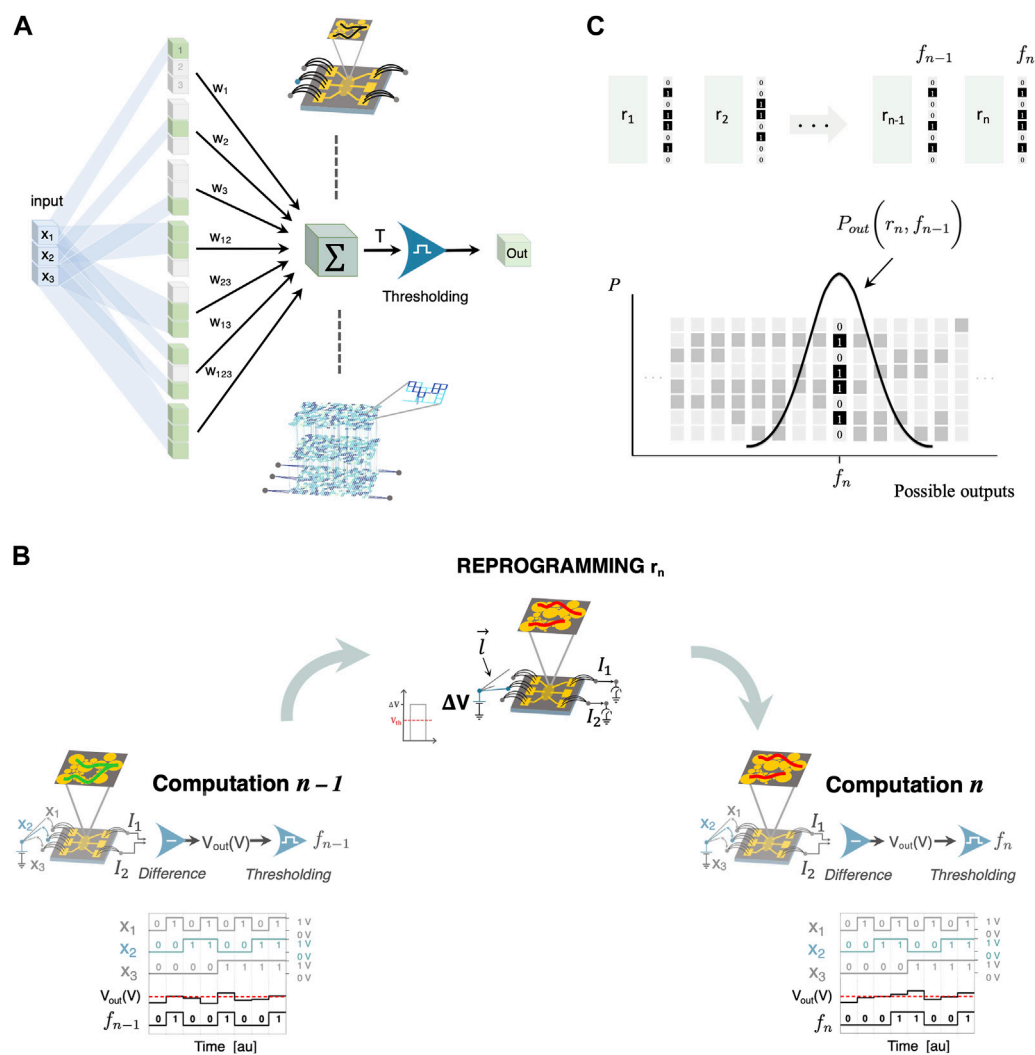


FIGURE 1

(A) Schematic representation of computation taking place in the receptor with three Boolean inputs ( $x_1, x_2, x_3$ ) and illustrations of the receptor implementations, experimental (top) and simulated network (bottom). Receptor operation is governed by a set of internal weights, which are associated both to the inputs and their spatial arrangement. Weights can be conceptualized as elements of a sparse weight tensor (non-zero weights are represented here as the arrows column) with a rank equivalent to the number of inputs. Here,  $\Sigma$  depicts the summation of the weighted inputs. The outcome of the convolution is thresholded to obtain a Boolean output. Illustrations of the experimental and the simulated receptor are shown above and below  $\Sigma$ ; the devices are equipped with three inputs on the left, and with two output channels (right), whose currents flowing through are subtracted to provide the digital output. (B) Schematic representation of Computing and Reprogramming phases of the experimental receptor. During Computation  $n-1$ , the nanostructure's internal configuration (highlighted in green) leads to the generation of a Boolean function. Subsequently, the reprogramming of the device - induced by high voltage stimuli - alters the internal structure, leading to transitions between internal conductive configurations (highlighted in red). Reprogramming functions as a re-weighting process: during Computation  $n$ , the system is now capable of generating a new Boolean function. (C) Representation of a reprogramming sequence. Receptor weights can be altered due to a reprogramming, depending on the internal inputs, resistive state of the system and the features of the reconfiguration (summarized by  $r$ ). After the  $n$ -th reprogramming ( $r_n$ ), the output function  $f_n$  is drawn from a probability distribution  $P_{out}$  depending on the reprogramming features and the previous system state (see Eq. 3).

update the weight tensor, enabling it to perform a different computation.

While the additional possibilities offered by the nonlinearity have already been discussed elsewhere [13], little attention has been given to the reprogramming procedure itself, especially in relation to common experimental limits, like intrinsic long-term dynamics and sample-to-sample variability. The two experimental receptor implementations proposed so far, optical [13] and electrical [17], both rely on substrates for the weighting of the inputs, that cannot be reconfigured in a deterministic manner: such complex weighting media evolve according to a stochastic

dynamics that cannot be completely captured by a set of differential equations. Here we focus on the electrical implementation [16] of a 3-bit (input) receptor to show how an intrinsically random reprogramming of a complex system can still be controlled and characterized to improve the performance of the computing device. Even if complete randomness may only be an extreme case scenario, the results presented here can be used to deal with all systems whose complicated modeling prevents a controlled and precise tailoring of the internal state or where sample to sample variability exceeds the tolerance of such a mathematical description.

## 2.2 Experimental and simulated receptrons

As anticipated, in this work we make use of an experimental and an *in silico* implementation of an electrical receptron (see Eq. 1), each featuring a three-channel input. Both realizations exploit a central non-ohmic conducting medium for the weighting process: two electrical currents  $I_1$  and  $I_2$  flowing out such a complex network are subtracted to obtain an analog value which is then thresholded (performed at the software level). The internal set of weights is encoded as the resistive state of the experimental/simulated nanostructured network: these can be reprogrammed by triggering a *resistive switching* phenomenon, i.e., a plastic rearrangement of the resistive network through the delivery of high voltage pulses [17].

### 2.2.1 Experimental receptron implementation

In the experimental implementation of the receptron [17], a three-input configuration exploits the electrical behavior of a multi-electrode nanostructured Au film. The assembly of nanoscopic (mean size around 6 nm) clusters of gold atoms on a flat silicon-oxide substrate results in a defect-rich (mainly grain boundaries) film whose intricate conductive paths result from the interaction of several branched aggregates, which grow as thickness is increased until the formation of spanning aggregates when the percolation threshold is exceeded [20, 21, 24]. The significant decrease in electrical conductivity resulting from such a high number of defects is accompanied by a multitude of resistance states that can be reached by inducing rearrangements of the nanostructure. As local temperature increases due to joule heating, cluster aggregates can experience significant morphological changes leading to macroscopically different current pathways [22, 25].

The electrical setup (Figure 1B) consists of three relays on the left, connected to respective electrodes, enabling the switching between voltage supply and open circuit. Two relays, connected to the extremal output electrodes on the right, allow the switching between ground terminals, via a digital multimeter for current measurement, and open circuit. As for many memristive systems [27], the reprogramming and computation are both the result of electrical stimuli, which only differ in magnitude. When computing, a low voltage ( $\Delta V = 1$  V) drives an electrical current through the substrate, indirectly probing its resistive state (Figure 1B - bottom panels, where the input signal is determined by  $x_1, x_2, x_3$ , which can take the 0/1 values); when reprogramming, higher voltages ( $\Delta V > 5$  V for the millimeter-size devices used here) are used to trigger the network reconfiguration. To probe the function implemented, the  $2^3 = 8$  possible input combinations are subsequently tested, each one leading to as many outputs (digitized via thresholding), i.e., to a newly generated Boolean function (see Figure 1B - Computation  $n - 1$ , bottom panel). A successive reprogramming alters the nanostructure of the gold film, possibly changing input-output current preferred pathways (see Figure 1B - Reprogramming, red curves), with the formation/disruption of grain boundaries and other defects. At time  $n$ , we can measure the new “computational state” again with a serial measurement of the  $2^3 = 8$  possible input combinations, resulting in a new Boolean function (see Figure 1B - Computation  $n$ , bottom panel).

### 2.2.2 In-silico simulation of the receptron

The simulated network uses the SRN model [18], which involves a stochastic evolution of resistors to represent the distribution of conductive regions within a cluster-assembled gold film [17, 27]. Each link represents a coarse-grained portion of the percolating network which can assume one of four possible conductance levels: an insulating one or three distinct conductive ones. At each simulation step, every link can stochastically switch to another level using Monte Carlo (MC) moves (see [Supplementary Material](#) and Ref. [18]): these mimic local thermal dissipation near crystalline orientation mismatches and nonlinear phenomena across the band gap described for the experimental system [17, 20, 21, 28]. Briefly, each link conductance  $\sigma_{ij}$  can be probabilistically downgraded based on the local power dissipation ( $W_{ij}^{(d)} = \frac{\Delta V_{ij}^2}{R_{ij}}$ ), and upgraded by absorbing power from its neighbors ( $W_{ij}^{(a)} \sim \frac{\sum_{N_{neigh}(kl)} \Delta V_{kl}^2}{R_{kl}}$ ). Then, the absolute value of the voltage at each link is compared with a threshold value, which determines whether a downgrade/upgrade of  $\sigma_{ij}$  has to be attempted. Since such moves only depend on the square or the absolute value of the voltage across the link, both sets of update rules exhibit symmetry with respect to local potential polarity (always taken as positive in our simulations).

After each MC move, Kirchhoff's equations are used to solve the network [29, 30]. Note that in our SRN complex physical phenomena arise from probabilistic update rules, which draw inspiration from microscopic-scale physics. This key aspect distinguishes our system's evolution from other models that directly replicate potentiation mechanisms (see, e.g., [31]). The SRN shares some common features (the regular grid, the way the network is solved, the local Joule effect) with other models like the simpler Random Fuse Model (RFM) [32], which was nonetheless conceived to study only the breaking process of materials traversed by current. Conversely, the stochastic evolution of the SRN, the ability to reproduce more physical effects than just the breaking and reforming of connections [33] and the easy adaptability to mimic a real experimental setup constitute major differences with respect to the old RFM.

Here, the simulated network exploits three electrode-nodes (ENs) to establish connections between the source and the network through as many groups of permanent-conductive links (ELs). These ELs are strategically positioned to mimic three input electrodes and can be connected to the source node through three permanent switch-links (SLs). SLs serve as switches between ENs and the source, allowing different configurations. The network's multi-channel setup, together with its 3D structure, is represented in [Supplementary Figure S1](#), which also features two output electrodes for connection to the ground. At each simulation step, it is consistently feasible to measure the resistance of every network component and the current passing through each edge, particularly the current exiting the network during computation (as illustrated in Figure 3A). This setup enables the simulation of computation and reprogramming phases, both of which adhere to the same logic as the experimental implementation of the receptron (see [Supplementary Material](#) for the definition of the threshold voltage  $V_{th}$  in the simulated case).

Note that our analyses benefit from the regularity of the network, which has a simpler geometry if compared with

recently introduced models for nanowire networks [34]. Such models feature a much more complex organization of the nodes and links, but we do believe that this choice would not modify our findings since those are not related in any way to the peculiar adjacency matrix of the simulated network.

## 2.3 Receptor reprogramming

The experimental and simulated receptor reprogramming is the result of current flow during the application of high voltage stimuli. Here we identify a series of control parameters that, by shaping the distribution of electrical currents, are instrumental in influencing the evolution of the physical system and, consequently, the dynamics of the weights. Overall, the effect of reprogramming is determined by the complex interplay of the voltage stimulus magnitude, polarity, and localization (that is, the boundary conditions that constrain the current flow). We will refer to these parameters with  $r$ :

$$r = (\Delta V, \pm, \vec{l}) \quad (2)$$

Being  $\Delta V \in [\Delta V_{min}, \Delta V_{max}]$ ,  $\pm \in \{+, -\}$  and  $\vec{l} = \{0, 1\} \times \{0, 1\} \times \{0, 1\} = \{0, 1\}^3$ . Here,  $\Delta V_{min/max}$  stands for the maximum voltage applied to the system when reprogramming,  $\pm$  is referred to the polarity of the latter, while  $\vec{l}$  indicates the state of each switchable connection, that is which electrodes are involved in the reprogramming of the physical substrate (See Figure 1B).

As a first order approximation, we assume that the state transition process is Markovian [35]: this ansatz is validated by the decay of the autocorrelation function for the output (see Supplementary Material) which has a characteristic low correlation step, between 1 and 4. Given this assumption, the output function  $f_n$  after the  $n$ th reconfiguration phase is drawn from a probability distribution  $P_{out}$  which depends on the previous state (i.e., the previous function  $f_{n-1}$ ) and how the reconfiguration has been performed, summarized by  $r_n$ :

$$f_n \in P_{out}(r_n, f_{n-1}) \quad (3)$$

as schematically represented in Figure 1C.

We will determine which parameters contained in  $R$  have a greater influence on  $P_{out}$  proving that a careful control over the latter gives a probabilistic control over the output function distribution, acting like a sort of control knob which may be optimized to enhance the functional generation efficiency of the device.

## 2.4 Calculating mutual information between reprogramming parameters and receptor outputs

The goal of identifying the most effective reprogramming hyperparameters can be achieved by a specific reprogramming protocol, wherein alternating cycles of reprogramming and output computation are conducted.  $R = \{r_1, r_2, \dots, r_n\}$  collects all the characteristics of the sequence of reprogrammings performed on the physical substrate and  $F = \{f_1, f_2, \dots, f_n\}$  containing the function implemented following the respective reprogramming.

We quantify the receptor susceptibility to diverse reprogramming protocols by computing the Mutual Information between the reprogramming sequence  $R$  and output functions sequence  $F$ ,  $MI(R, F)$ , and the entropy of the output,  $H(F)$ . The Mutual Information

$$MI(R, F) = H(F) - H(F|R)$$

quantifies the amount of reduction in uncertainty about the output given knowledge of the input [36], and a significantly non-vanishing ratio of  $MI(R, F)/H(F)$  would imply that it is possible to extract relevant information about the output based on the chosen reprogramming scheme. This approach allows for efficient characterization of the device's functioning, as it avoids the need for extensive grid-search experiments that could be difficult to perform on a statistically significant batch of devices, due to inter-device variability.

In our study, we want to verify that the following is significantly true:

$$H(F) \leq H(F|R) \leftrightarrow MI(R, F) > 0$$

To compute the entropies, we have used the natural logarithm and applied the correction term proposed by [37]. This correction term is given by:

$$H(\theta) = -\sum_i p_i \ln(p_i) + \frac{B^* - 1}{2N}$$

where “ $\cdot$ ” is a generic observable,  $B^*$  is the number of states with  $P_i \neq 0$  and  $N$  is the total number of observations.

In experiments and analogously in simulations we have estimated  $p_i$  using a frequentist approach.  $F$  is defined to be implemented output function represented as a decimal number, and it depends on the specific thresholding that is implemented, while  $H(R)$  is clearly independent from this post-processing. In particular, we take  $F$  belonging to the set of the 256 possible Boolean functions of 3 variables, and the simultaneous occurrence of  $R$  and  $F$  is thus, simply, the event of reading  $F$  as output after having written with the  $R$  scheme. We point out that the normalization used to compute the probability clearly depends on the single experiment/simulation performed. I.e., if in a simulation of the SRN model we record  $B$  distinguished Boolean functions, among the 256 possible ones, the occurrence of each of them will be normalized by  $B$  (and not by 256).

### 2.4.1 Significant Mutual Information test

Statistically dependent input and output sets are expected to have higher Mutual Information than would occur by chance. To determine the significance of the Mutual Information measured, we used a standard  $p_{value}$  significance test. This involves comparing the measured MI to a null hypothesis, which represents the Mutual Information that would be expected if  $R$  and  $F$  were independent. If the measured MI is significantly different from the null hypothesis, it can be concluded that the dependence between  $R$  and  $F$  is statistically relevant. As a null model we permuted the elements  $f \in F$  keeping the  $r \in R$  fixed. This approach preserves  $H(R, F)$  and  $H(F)$ , changing only the value of joint entropy between inputs and outputs  $H(R, F)$ . We permuted  $R$  and calculated  $MI(R, F) n_{perm} = 5000$  times for each



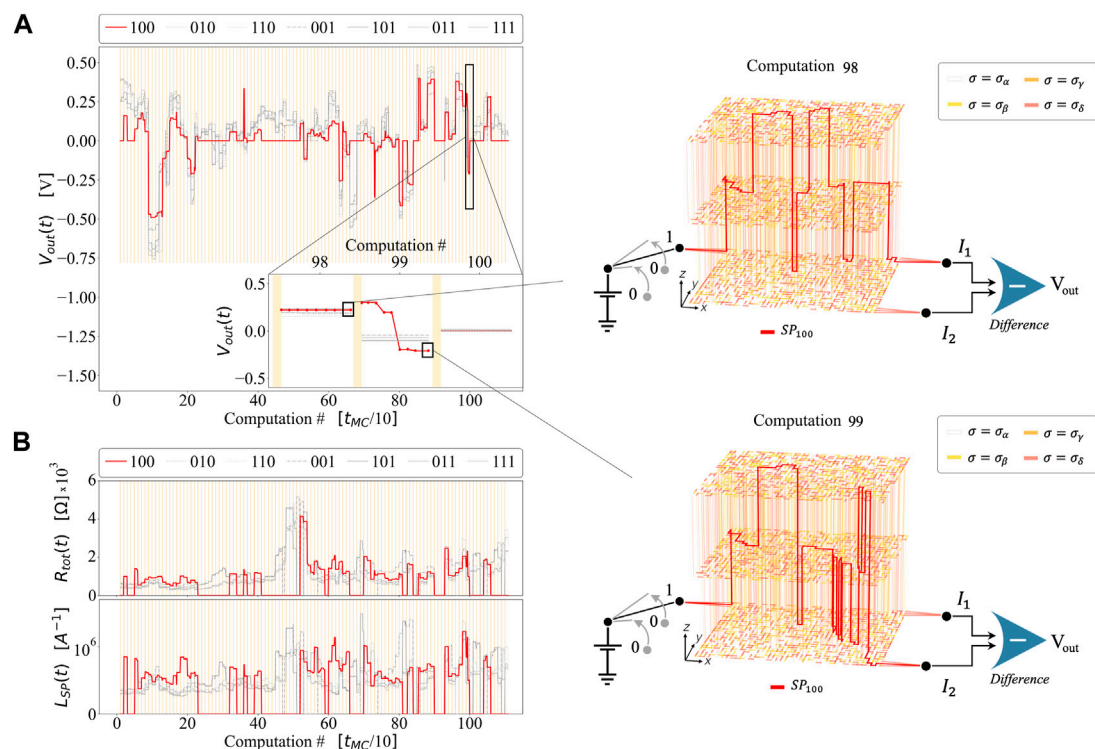


FIGURE 2

Simulation of the SRN model, in which each reprogramming is followed by a computation, repeating the process in series 110 times. During the simulation, the polarity and localization of the input channels in the reprogramming remain constant. Each computation lasts 10 MC steps. (A) Left panel: Evolution of  $V_{out}(t_{MC})$  during the whole procedure of 110 reprogrammings, each followed by a computation. The red curve is the one corresponding to the (100) computations, vertical orange lines indicate that reprogramming occurs between two computations. Right panels: snapshots of the network, corresponding to two computations (100) interspersed by reprogramming phases. The input-output shortest path is highlighted in red. (B) Evolution of  $R_{tot}(t_{MC})$  and the SP length  $L(t_{MC})$  for computations. Red curves correspond to outputs associated to the input (100). Note that for representational purposes only we represent the infinities of  $R_{tot}(t_{MC})$  and  $L(t_{MC})$  curves as zeros.

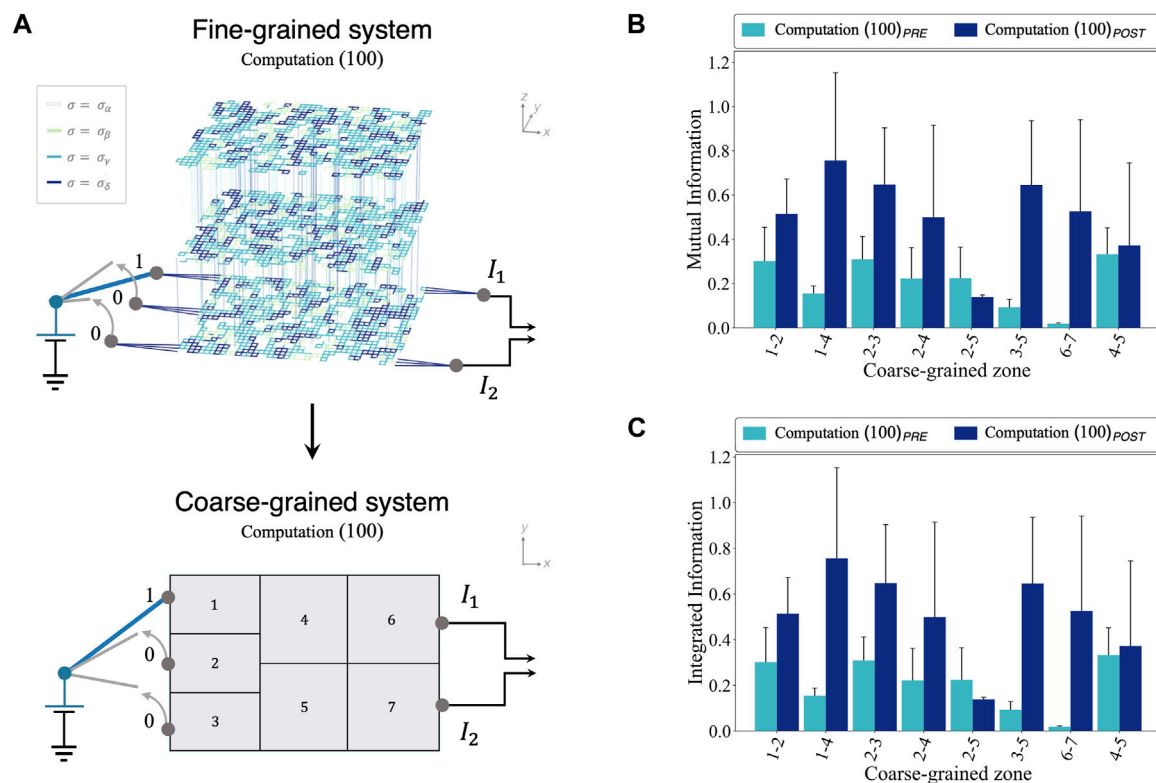
threshold. Mutual Information deemed significant if MI calculated for the null model were higher than actual values  $n_{viol}$  up to 5 times ( $\alpha = 0.001$ ). The  $p$ -value is then obtained as  $p_{value} = \frac{n_{viol}}{n_{perm}}$ . All MI estimates whose  $p$ -value was higher than  $\alpha$  have been rejected (MI = 0).

## 3 Results

### 3.1 Internal dynamics of the Stochastic Resistor Network

SRN simulations offer an important perspective on what is happening to the complex network during reprogramming. In the forthcoming example, we explore the effect on the analog weighted output,  $V_{out}$ , of a selective modification of a single reprogramming parameter, *ceteris paribus*. In practice, we have performed several reprogrammings with a reprogramming localization fixed to (1,0,0), a positive polarity and a random voltage magnitude: after each such reprogramming we sampled the implemented Boolean function by scanning over all digital inputs. In the language of Eq. 2, each set of reprogramming parameters is defined by  $r = (\Delta V, +, (1, 0, 0))$  with  $\Delta V$  uniformly selected in [15 V, 35 V].

Figure 2 illustrates the relationship between the outputs and the network restructuring. Figure 2A presents an example (focusing on 110 MC steps) of  $V_{out}$  curves, for each output reading during computation stimulus; these exhibit significant variations due to the alternation of reprogrammings, depicted by orange vertical lines. The common trend among the curves suggests a certain level of mutual correlation. Notably, these features closely resemble experimental results [16, 18]. The small peculiarities of each of the outputs can be attributed to both the specific network configuration and the precise positions of the channels through which the incoming current flows. Specifically, the repeated reprogramming with the same localization (1,0,0) has greater effects on the outputs associated with this input (red vs. gray curves). The distribution of the analog output value associated to (1,0,0) reaches lower  $V_{out}$  values compared to the others. This indicates that repeated reprogramming with fixed localization leads to a resistance increase near the current pathways explored during the corresponding computation. This, in turn, has a lesser impact on other pathways. To prove this, we obviate the need for an intricate analysis of the overall resistive connection distribution; instead, we employ the calculation of current's shortest paths (SPs) within the network as a suitable proxy. The length of the SP has in fact a consistent relationship with macroscopic resistance patterns, making them good microscopic indicators of bulk current flow in



**FIGURE 3** Left: network coarse-graining procedure (A). The coarse-grained system is obtained by dividing the network into seven parallelepipeds, each of which is mapped onto a two-dimensional sub-region. In both the schematic representations, the computation 100 is depicted. Right: The effect of a (1,0,0) reprogramming on Mutual Information MI (B) and Integrated Information II (C) between two or more coarse-grained sub-regions.

the system. Using the metrics introduced in [18], the SRN model allows for the measurement of the lengths of all the possible current paths between the input and output at each time step, calculated as  $L = \sum_{i,j} \frac{1}{I_{ij}}$ , with  $(i, j)$  identifying the pairs of nodes. Among these paths, an optimization algorithm [38, 39] allows to calculate the SPs followed by the current between the network's input and output. As an example, Figure 2B depicts the lengths of the shortest conductive paths within the network, depending on the reading localization. To enhance clarity, we set  $L$  values that would otherwise result in infinite length to zero. Notably,  $L$  exhibits a distinct trend, especially during computations with input (100).

Shortest Paths, however, are not captured by their mere length: a visual inspection of their evolution is in fact quite instructive on the effects of specific reprogrammings and the topology of the induced network. An example is provided in the right panels of Figure 2A. Here we show how the reprogramming leaves the first section intact while inducing a notable change in the second part of the SPs for the (100) input combination configuration (red curve). It is precisely this latter variation that justifies the small difference between the outputs that decorate the overall trend. Even more interestingly, we observe a significant variability of SPs as a result of rearrangements. This indicates that extensive modifications of the inner structure are repeatedly occurring, in striking contrast to what is reported for a nanowire network (see, for example, [40]), where a single path was progressively strengthened. The repeated variation of outputs after reprogramming phases subtends a continuous change of the current

SPs: such dynamics guarantees the reconfigurability of the device *ad infinitum* and preserves the variability of information processing resulting from such a ceaselessly mutating substrate.

Far from complete randomness, however, such an evolution strongly depends on how the reprogramming is performed, thus on  $r$ . To prove this, we have analyzed, thanks to the Information Theory framework presented in [41, 42], the way in which reprogramming affects the information processing. The system is divided into 7 coarse-grained sub-regions (as shown in Figure 3A) to analyze the time-evolution of the electrical properties of each zone. Mutual Information (MI) associated with the electric current is used to describe the interactions among complementary areas of the system, while Integrated Information (II) is exploited to evaluate the reciprocal integration among the building blocks of a sub-region [41, 42]. These measures are built on top of the entropy  $H$ , which is calculated from the discrete probability distribution of the average conductance in each sub-region (with 10 distinct states available to each coarse-grained zone; see Methods for details).

As shown in Figures 3A–C reprogramming with  $r = \{15V, +, 100\}$  leads to many changes in the information flows between zones when reading the output associated to the (100) combination in input. The pair one to four exhibits a major increase in II, beyond statistical compatibility; this was somehow predictable, since it is the region lying closer to the open input channel. The growth of II (3–5) is instead interpretable as an augmented segregation of those regions (where the effect of the reprogramming is smaller) with respect to

the rest of the system. Conversely, we do expect regions 6 and 7 to be always highly integrated after a reprogramming, regardless of its details, as it happens in this case. For all the other pairs considered, the II value before and after the reprogramming is statistically compatible, so it means that (with the limited available data) the effect of the reprogramming is modest.

On the other hand, since Mutual Information expresses the association between a part of the system and all the rest, we expect it to display a generalized growth after the reprogramming. This happens in most cases; the increase in 1–4 is marked, signaling an activation of such zones. Also 2–3, 2–5 and 3–5 exhibit a significant modification, probably because they are in closer contact with the 1–4 region where most of the variations in the electrical conduction happen. Supported by empirical evidence from this analysis, we can however state that different reprogramming input patterns result in partially altered levels of reciprocal correlations among sub-regions in the network, but more evidently in higher integration in the sub-regions closer to the open input channels. The spatial analysis of II and MI in the SRN can provide hints for the manipulation of hyperparameters of the reprogramming process, such as the amplitude, polarity, and localization of stimuli, in a way which is not applicable to the experimental electrical repton.

## 3.2 Characterization of the repton sampling efficiency

The indications provided by information theory can now be applied to the practical computation that we expect from repton, finally analyzing its performances as a functional generator. To obtain a digitized output from an analog one, a thresholding procedure is exploited, where among a set of trial threshold values is retained the one which provides the greatest number of different generated functions [17]. The efficiency of the generation process will of course depend on the specific hardware implementation and on how the (stochastic) reprogramming is performed. We are first interested in determining which constructive characteristics have a higher impact on efficiency, while the effect of a specific reprogramming will be the object of the following section. A random exploration, where  $r$  is sampled from a uniform probability distribution stretching over the whole boundaries of the parameter space, allowed us to average out the contribution from the characteristics of the reprogramming.

The fact that a single repton can generate any function [13], does not *per se* guarantee that it will do it quickly enough: such unconstrained opportunities would be useless if the device could not reach the desired target in a reasonable number of reprogrammings. In fact, the prohibitive scaling of the number of  $N$ -input Boolean functions ( $2^{2^N}$ ) would really make it impossible for any functional generator to investigate every alternative in a satisfactory amount of time: for instance, a generator operating 24 h a day at GHz frequencies would take centuries just to exhibit all 6-input Boolean functions. As a matter of principle, the capabilities of these devices could be significantly boosted not just by increasing the rate of generation of new functions, but also by making reprogramming more specific, that is limiting the random search to a functional neighborhood of the specific target.

We start by investigating which factors allow the maximization of the number of functions explored, while the next section will deal with targeting a specific area of the output functions space. The Boolean-function generation efficiency  $\varepsilon$  is a quantity already established for the experimental device [17], which for the case of 3 bits reduces to:

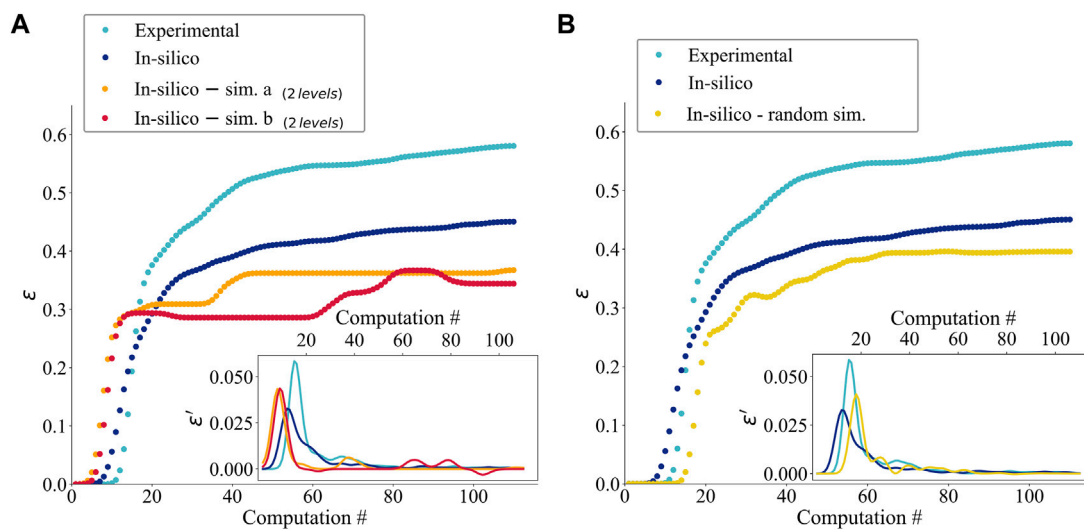
$$\varepsilon = \frac{256}{\sum_{i=0}^{255} C(i)} \quad (4)$$

$C(i)$  being the number of reptions that, combined via OR or XOR, can implement the  $i$ th boolean function. We will briefly recall here the significance of such a definition. Given the infeasibility for a single generator to cover the whole (Boolean) functional space in a computationally profitable time, the idea is then to see whether a minimal set of pre-defined simple combinations (OR and XOR; see below for details) of the functions generated allows to also retrieve the missing ones. When this is done, we not only have an idea of how quickly the network reaches different possibilities but, most importantly, we also keep into account the difference between the functions generated: if the exploration is restricted to a very narrow region of functional space, we may miss entire classes of outputs which are impossible to generate even after the aforementioned combinations. The concept of a complete set [17], which formalizes the intuitive idea of a basic set of functions which is needed to implement any other via linear combinations is thus naturally introduced. The generation of an ever-increasing number of functions will then ease obtaining the missing ones: the efficiency  $\varepsilon$  is thus a measure of the average number of reptions needed to implement a boolean function, simply counting the fraction of Boolean functions already generated.

We have identified key mechanisms that influence this variability, acting on the evolution laws that are governing the dynamics of the computer-simulated repton while using the physical one as a reference. Figure 4A compares distinct, i.e., regulated by a different dynamics, SRN-based implementations of a repton, to be contrasted with the cyan experimental curve: the first type of network is composed of links that can explore 4 levels of conductance (blue curve), while in the second case only 2 states are possible (orange and red curves). In the latter cases, network's links can access either a conductive level (respectively  $0.01 \Omega^{-1}$  and  $0.04 \Omega^{-1}$ ) or a nonconductive one. The inset shows the smoothed derivatives of the efficiency curves. As a result of reprogrammings, the  $\varepsilon$  curves jump to a non-zero value after a few steps due to the exploration of a complete set of functions. As shown, when only two levels of conductance are allowed, the efficiency of the network initially increases rapidly, but then levels off and reaches a plateau. Despite the network being reconfigured, the system here is unable to generate new Boolean functions. The limited number of conductance levels available to each of the (nonetheless highly abundant) degrees of freedom prevents the system from achieving higher levels of efficiency.

The derivatives of the compared curves (inset) highlight that there is also a discrepancy between the saturation points and the different trends as a function of the computing step. While the qualitative behavior is shown to be insensitive to the details of the statistical evolution, some smaller quantitative differences arise in the shorter term for the derivative and in the longer term for the plateau value. These were used to settle the number of  $\sigma$  levels in simulations to 4 but also provides key insights into the functioning of such a system: its sampling





**FIGURE 4**  
**(A)** Boolean function generation efficiencies (see Eq. 4) obtained for a an experimental receptron and different computer-simulated receptron, where we used a 4-levels SRN model (blue curve) and two 2-levels SRN model, each resulting by a random search process of a succession of 110 computations and reprogrammings. Inset: Derivatives of efficiency curves. **(B)** Efficiency and Derivatives computed for a physical receptron (turquoise curve), a computer-simulated receptron using an SRN model (blue curve), and a simulated receptron with randomly evolving topology (yellow curve) [29]. All curves have been smoothed using a 1D gaussian kernel.

efficiency strongly depends on the number of states accessible to each of the large number of conductances constituting the network.

Figure 4B compares the efficiency limiting values for both real and virtual receptrons: a physical receptron (cyan curve), a receptron simulated with the SRN model (blue curve), and a simulated receptron with randomly evolving topology - i.e., each link can be updated to one of the 4 conductance levels with uniform probability - (yellow curve). All the devices have similar efficiency trends, with an initial efficient exploration of the Boolean function space and a subsequent decrease of the variability and then of the curve slope. Even the simulated receptron with randomly evolving topology reaches a qualitatively comparable level of variability as the other two. Intuitively, the most significant differences between the various devices emerge after a few reprogramming steps: while it is relatively easy to generate new functions when only a tiny fraction of all the possibilities have already been visited, it becomes increasingly harder as reprogramming events start to accumulate. Thus, the key for a good receptron does not just lie in the number of degrees of freedom (which is enormous also in the case of the 2-level system) but rather in the availability of a wide set of local energy minima resulting in different outputs, in stark contrast with systems characterized by a *Winner Takes it All* dynamics [40], as we noted previously.

As discussed previously, however, the rate of generation of new functions is not a sufficient quantifier for the performance of this device: its ability to adapt to different reprogramming stimuli is critical if we are to limit its exploration to a much narrower target region of functional space. Being a global average,  $\epsilon$  does not consider the second aspect, which is precisely where the physical behavior plays a crucial role, ensuring in principle the possibility to modify the statistical properties of the outputs: this is why we need to introduce controllability. We will now turn our attention to the quantification of this adaptability of the device, providing examples of parameters which determine different functional exploration.

### 3.3 Evaluation of the receptron controllability

Here we investigate the link between a specific reprogramming protocol and the resulting distribution of the outputs: as we anticipated,  $r$  was sampled from a uniform probability distribution covering the whole possibilities in the set specified by the selected  $r$ , thus its effect is averaged out. The results presented this far already evidence that such a choice does not prevent the system from exploring a wide variety of configurations, therefore efficiently generating a large set of Boolean functions. What follows will prove that even coarse control on the characteristics of the reprogramming gives in turn some influence on the statistics exhibited by the system, something we will refer to as *controllability*. Fine-tune control on the probability of each of the possible output functions would instead require a tremendous number of parameters. Such a line of reasoning prompts us with a generalized concept of controllability as a trade-off between the number of parameters and the time required to obtain a given output. In fact, the stochastic nature of an ideal receptron guarantees that, in principle, the desired function will be at some point implemented: a scarce control over the reprogramming will enlarge the functional space subject to exploration and slow down convergence to the result but will not prevent its realization. In mathematical terms, the output function is a random variable that depends on the way reprogrammings are performed, as mentioned in Eq. 2. We will now quantify the relation between different possible  $r$  sets and the full set of generated functions in  $n$  reprogramming steps,

$$F = \{f_1, f_2, \dots, f_n\}$$

A quantitative way to assess the receptron susceptibility to diverse reprogramming protocols consists in measuring the reprogramming/computing Mutual Information MI ( $R, F$ ) [41,

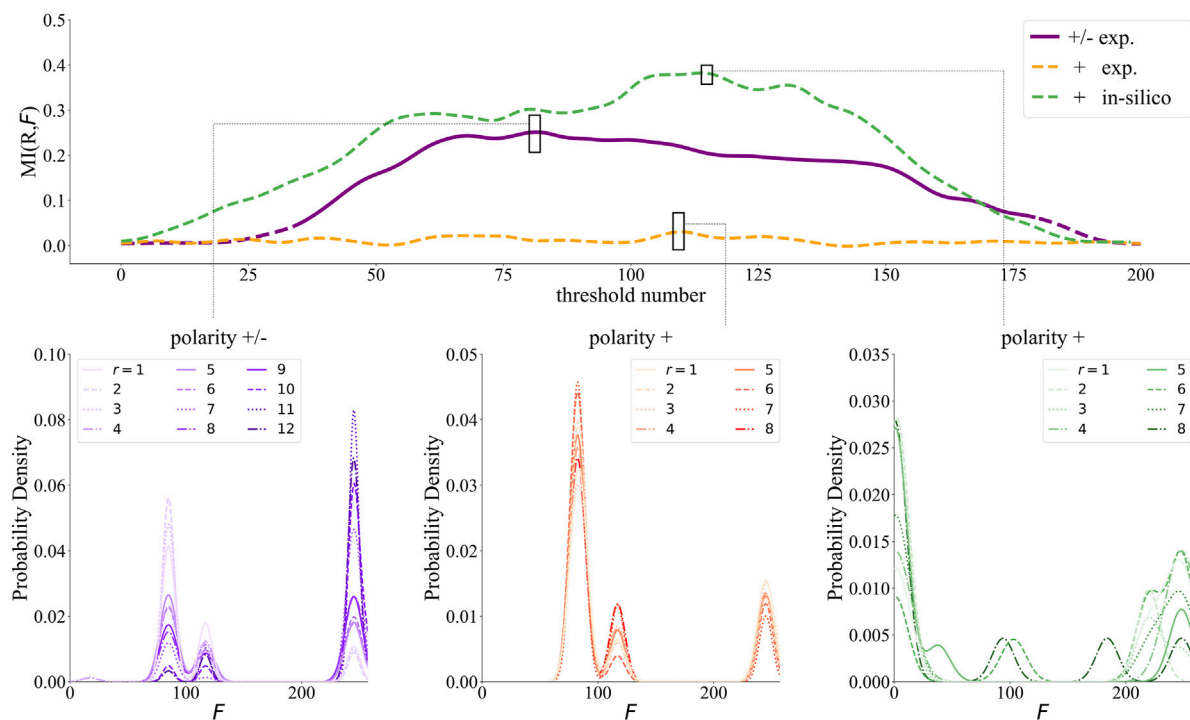


FIGURE 5

**(Upper panel)** Computation/reprogramming Mutual Information,  $MI(R, F)$  for different choices of  $r$ , indexed in an arbitrary way, on the same sample as a function of the chosen threshold (x-axis). The solid line indicates when  $MI$  is significant, while the dashed line indicates nonsignificance. For the experimental receptor the purple curve is obtained using positive and negative voltages during reprogramming steps, while the orange one (as the green one for the SRN model) had the polarity fixed to just positive values.  $MI(R, F)$  curves have been smoothed with a Gaussian Kernel with  $\sigma = 2$  threshold numbers. **(Bottom left panel)** Distributions of output functions per each reprogramming features set  $r$ , obtained after reprogramming steps with alternate polarity for the real device. Here the protocol involves 500 alternating reprogramming and computation phases, with 12 different possible  $r$ , wherein polarity can **(Bottom central panel), (Bottom right panel)** Distributions of output functions per each reprogramming features set  $r$ , obtained after a reprogramming protocol with only positive applied voltages for experimental and simulated receptors. The distributions have been smoothed using Kernel Density Estimation (KDE) with the same bin width.

42]: this will give indications on the effect of different stimuli sequences on the output function distributions. The Mutual Information between  $r$  and  $F$ ,  $MI(R, F) = H(F) - H(R)$ , allowed us to determine under which conditions the output entropy,  $H(F)$ , is significantly larger than the conditional output entropy,  $H(R) = H(F, R) - H(R)$  (See Section 2.4 for further details). When this occurs, the knowledge (i.e., control) over  $R$ , i.e., the reprogramming steps applied to the system, gives information about  $F$ : we can practically influence the result and increase the likelihood of a certain output function. Two different reprogramming protocols were employed: one where negative and positive polarity of the voltage is alternated, and another one where only positive voltage pulses are administered to the system. The entropy and Mutual Information curves were constructed using 200 evenly spaced thresholds between maximum and minimum  $V_{out}$  values. The statistical significance of the obtained curves is computed with a standard-value significance test, to assess the likelihood that the obtained  $MI(R, F)$  values are higher than those obtained for a random process.

We recall here that, while Eq. 2 well describes the possible ways to perform reprogramming in the experimental system, for the SRN model in its actual flavor the polarity is not a real degree of freedom: network configuration's updates just depend on  $|\Delta V|$ . This has not been a limiting factor, so far: the electrical

characterization of the SRN and properties like efficiency are not affected by this ingredient. For the  $MI(R, F)$  analysis, however, the computational receptor implementation can only be compared with the experimental one at fixed polarity.

The results of this analysis are shown in Figure 5: the value of  $MI(R, F)$  as a function of the threshold, for different sets of reprogramming protocols, is presented in the upper panel. A solid line indicates a statistically significant  $MI$  curve, while a dashed one marks data points that failed the significance test. We emphasize that non-significant  $MI$  values are to be considered null and are plotted in their original values for display purposes only. First, it is evident that in the experimental case with only positive polarity reprogramming steps (orange curve), every threshold choice for  $V_{out}$  yields a non-significant  $MI$  (orange curve), indicating that the parameters varied between different reprogramming steps were not sufficient to induce a statistically significant dependence of the outputs on  $R$ . This is confirmed by the SRN simulations carried out in analogous conditions, which are not vanishing but also not significant in the whole range (green curve). Conversely, when the reprogramming procedure features both positive and negative electrical potentials (purple curve), there is a wide set of threshold choices for which  $MI(R, F)$  is statistically significant (solid curve). This configuration demonstrates how the applied voltage sign contributes to the

establishment of a statistically significant relationship between the reprogramming and the resulting output signal.

For a deeper understanding of what it does mean to have a non-significant  $MI(R, F)$  (i.e., vanishing), it is rather instructive to consider what happens for the extremal thresholds. Whenever the threshold lies very far from the analog weighted outputs' mean, all thresholded outputs will be either 0 or 1, thus the output function will be forced to  $f = 0 \vee f = 255 \forall t$  (a Boolean function consisting of all zeros or ones, respectively): evidently, the specific series of reprogramming procedures,  $r$ , has no impact on the generated functions. Thus, even if the output function will be easily predicted, we have no *control* over the specific output function distribution: this first marks the distinction between our capability to predict  $o$  (*predictability*) and our influence (through reprogramming procedures) on its distribution (*controllability*).

Intermediate thresholds allow the maximization of the MI curves. Notably, our experimental results reveal two different scenarios: the receptron consistently explores the same region within the space of Boolean functions for both fixed and variable polarity reprogramming protocols, resulting in non-significant and significant MI respectively. This is visually represented in Figure 5 - Bottom left and central panels, where we observe multimodal distributions with peaks precisely centered on the same output functions. We emphasize that the whole series of experimental measurements were conducted on the same device, with the primary difference being the random chronological order of the reprogrammings  $R$  in each protocol. When we examine reprogramming protocols with a fixed positive polarity parameter (Figure 5 - Bottom central panel), our analysis shows that the output multi-mode distributions follows a consistent pattern, with the primary peaks reaching the same heights for each reprogramming mode. In this case, the output functions are not easily predicted (being the result of the system's randomic dynamics), but still we cannot influence the outputs statistical likelihood by choosing specific reprogramming steps, which is rephrased in terms of a complete *lack of controllability*. More interestingly, when polarity truly makes a difference (Figure 5 - Bottom left panel), a different phenomenon emerges. In this case,  $r$  modes significantly influence the output distributions, resulting in varying peak heights. This indicates that the outcome of each reprogramming mode favors certain outputs over others, a fact supported by our significance analysis of MI. This evidence strongly implies that adopting reprogramming protocols with a targeted order of polarity may provide greater control over the resulting output, thereby challenging the Markovianity hypothesis. Significant nonzero  $MI(R, F)$  values are found whenever different reprogramming steps are specific, being capable of restricting the generation of Boolean functions to a subset of all the possibilities. In the context of simulations, it is noteworthy that each reprogramming mode induces Boolean output distributions that significantly differ in the location of the peaks, as depicted in Figure 5 - right panel. This finding in contrast with the experimental observations can be attributed to a system size effect: each input channel localization appears to cause changes that are too extensive and dramatic compared to the physical system. While this leads to a broader exploration of the output space in general, the outcomes of such reprogramming steps are not controllable, as suggested by the significance test.

These examples highlight the difference between predictability and controllability: even if the system pertains to a degree of

randomness (meaning it is not fully predictable), we can influence the statistics of the expected outcome via the specific features of reprogramming steps. The two quantities, however, evolve independently: as we have shown, we can have negligible controllability both in the case of high (for a trivial threshold setup) and low predictability (for positive-only polarity). Changing the polarity appears to be decisive for a nonzero controllability to emerge: this could be ascribed to a series of complex memory effects inside the cluster-assembled film, ranging from capacitive charge accumulation near the electrodes to bistable junction-switching, which are still under investigation. Conversely, we observe that the consequences of changing the reprogramming localization and the exact tension value have a smaller impact on the final outcome for different reasons. On the one hand, the numerical value of  $\Delta V$ , beyond a given threshold, does not matter much (see also [18]); on the other hand, one expects a change in the reprogramming localization to be relevant but only limited to a sub-region of the system, as evidenced for the SRN by Figure 2. The results presented so far only allow us to limit the vast range of possible reprogramming protocols, but still, it is not feasible to quantitatively plan the reprogramming protocol to reach a specific target. To this extent, possible future improvements of the SRN, including an increased system responsivity after a polarity change, might help in further focusing on particular reprogramming protocols. All the analyses described so far leverage the Markovian hypothesis that a given reprogramming influences only the following one. However, we would expect higher levels of Mutual Information by considering as  $r$  the quantitative description of the reprogramming characteristics for  $n > 1$  such steps: an analysis of this kind, however, requires further statistics, both in terms of the number of different reprogramming protocols and, fixing  $r$ , the number of reprogramming steps.

A future step will consist in the identification of a reasonable trade-off between increasing the accuracy of our predictions and the growing cost of such an approach. The analysis of temporal autocorrelation in our datasets suggests that the consequences of a reprogramming might, in principle, affect up to a few of the following readouts. The outputs' trajectory in functional space is in fact the result (integral) of the contributions from several reprogramming steps: each one may be responsible for significant variations, but of course the starting point is itself dependent to some extent on the past history. In any case, the results presented so far, under assumption of Markovianity, already contribute to set some boundaries: we have underlined that only some hyperparameters are decisive to condition the system's subsequent output. We argue the effect of such hyperparameters to have a larger relevance than the number of  $R$  sets explored.

## 4 Conclusion

We have successfully modeled the complex behavior of a physical receptron undergoing reprogramming and computing processes during the stochastic generation of Boolean functions. In particular, we have extended the characterization of the Boolean function generation process of a receptron to include not just efficiency measures but also a crucial controllability analysis. Aiming to quantitatively assess the effects of reprogramming on the network regions, we have employed Information

Theory tools, highlighting the variable interacting strength with which different regions of the film react to external perturbations. We have discussed the requirements in terms of system size, thus number of junctions, for a sufficient level of complexity in the network: a large number of interconnected building blocks is needed together with the possibility for each of them to explore at least a few states (the discrete conductance levels). This complexity is reflected in the efficiency parameter, which captures the receptor intrinsic variability in generating Boolean output functions, together with the possibility to control its output, albeit in a statistical way.

Assessing the degree of controllability for a receptor is a key step in view of using it as a paradigm for innovative approaches to computing. Its inherent stochasticity, which lies at the heart of its flexibility and effectiveness, can in fact be limited by acting on the reprogramming process. We have identified reprogramming voltage polarity as a key parameter to steer the output functions distribution in the experimental receptor. The information obtained with this method provides the user with all the fundamental tools needed to calibrate the balance between predictability and controllability of the output values, with crucial influence on the device effectiveness as a Boolean function generator. Leveraging entropy-based measurements, we have developed a quantitative and robust criterion to determine which parameters allow the Mutual Information between the reprogramming protocol and the output to be significantly non-vanishing.

Even if qualitative, the agreement between the SRN model and the physical substrate is remarkable, the difference arising from the deep diversity of the two platforms, especially concerning fine details. Despite its considerable computational weight, the SRN is a coarse-grained physics-inspired abstract model, quite limited in size, much simpler than its experimental counterpart. Our simulations suggest that it is also more fragile upon repeated reprogramming steps: in the future, a systematic exploration of the grid of the SRN hyperparameters can undoubtedly help to increase the durability of the network.

Moreover, we plan to explore different strategies to implement a realistic dependence on the polarity of the applied tension in upcoming SRN developments, to better capture the experimentally assessed sensitivity to such hyperparameter. On the one hand, current SRN predictions concerning controllability provide generic indications about the relationship between a given reprogramming protocol and the resulting output vectors. On the other hand, while such results have a statistical robustness not comparable to the one typical of the experimental device, the simulation has a higher degree of reproducibility and often a lower cost than performing extensive experiments. Therefore, computational investigation can restrict the region of the phase space to be explored, allowing for more specific experimental campaigns. In addition, the simulations of the resistor network are useful to get insights about the experimental device functioning at a mesoscopic level: for instance, the visual inspection of current pathways, or the analysis of the reciprocal correlation among the different sub-regions. This kind of information is not retrievable from the mere analysis of the experimental receptor.

## Data availability statement

The raw data supporting the conclusion of this article will be made available by the authors, without undue reservation.

## Author contributions

ET: Conceptualization, Data curation, Formal Analysis, Funding acquisition, Investigation, Methodology, Software, Visualization, Writing–original draft. GM: Conceptualization, Data curation, Formal Analysis, Investigation, Methodology, Visualization, Writing–original draft. MM: Conceptualization, Methodology, Writing–review and editing. DG: Conceptualization, Funding acquisition, Methodology, Supervision, Writing–review and editing. PM: Conceptualization, Funding acquisition, Methodology, Software, Supervision, Writing–review and editing. FM: Conceptualization, Investigation, Methodology, Software, Supervision, Writing–original draft.

## Funding

The author(s) declare that financial support was received for the research, authorship, and/or publication of this article. The study was also supported by the CINECA agreement with Università degli Studi di Milano 2019–2020, CINECA grants *IscraC RENNA* 2019 and *IscraC iRENNA* 2021 and INDACO-UNITECH 2020–2022. These organizations provided high-performance computing resources and technical support that were crucial for the successful completion of the research.

## Acknowledgments

The authors thank Michele Allegra for his contributions during the development of this work, providing valuable insights and suggestions that helped shape the final product.

## Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

## Supplementary material

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/fphy.2024.1400919/full#supplementary-material>



## References

- Teuscher Christof. Unconventional computing catechism. *Front Robotics AI* (2014) 1:10. doi:10.3389/frobt.2014.00010
- Dale M, Miller JF, Stepney S. *Advances in unconventional computing: volume 1: theory*. Berlin, Heidelberg: Springer (2017).
- Schrauwen B, Verstraeten D, Van Campenhout J. An overview of reservoir computing: theory, applications and implementations. In: ESANN 2007 Proceedings - 15th European Symposium on Artificial Neural Networks; April 25-27, 2007; Bruges, Belgium (2007).
- Verstraeten D, Schrauwen B, D'Haene M, Stroobandt D. An experimental unification of reservoir computing methods. *Neural Networks* (2007) 20(3):391–403. doi:10.1016/j.neunet.2007.04.003
- Tanaka G, Yamane T, Héroux JB, Nakane R, Kanazawa N, Takeda S, et al. Recent advances in physical reservoir computing: a review. *Neural Networks* (2019) 115: 100–23. doi:10.1016/j.neunet.2019.03.005
- Dale M, Miller JF, Stepney S, Trefzer MA. A substrate-independent framework to characterize reservoir computers. *Proc R Soc A: Math Phys Eng Sci* (2019) 475(2226): 20180723. doi:10.1098/rspa.2018.0723
- Banda P, Teuscher C, Lakin MR. Online learning in a chemical perceptron. *Artif Life* (2013) 19(2):195–219. doi:10.1162/ARTL\_a\_00105
- Miller JF, Harding SL, Tufte G. Evolution-in-materio: evolving computation in materials. *Evol Intelligence* (2014) 7(1):49–67. doi:10.1007/s12065-014-0106-6
- Larger L, Baylón-Fuentes A, Martinenghi R, Udaltsov VS, Chembo YK, Jacquot M. High-speed photonic reservoir computing using a time-delay-based architecture: million words per second classification. *Phys Rev X* (2017) 7(1):011015. doi:10.1103/PhysRevX.7.011015
- Miller JF, Downing K. Evolution in materio: looking beyond the silicon box. In: Proceedings - NASA/DoD Conference on Evolvable Hardware, EH (2002).
- Indiveri G. Introducing 'neuromorphic computing and engineering. *Neuromorphic Comput Eng* (2021) 1(1):010401. doi:10.1088/2634-4386/ac0a5b
- Li Q, Diaz-Alvarez A, Iguchi R, Hochstetter J, Loeffler A, Zhu R, et al. Dynamic electrical pathway tuning in neuromorphic nanowire networks. *Adv Funct Mater* (2020) 30(43). doi:10.1002/adfm.202003679
- Paroli B, Martini G, Potenza MAC, Siano M, Mirigliano M, Milani P. Solving classification tasks by a perceptron based on nonlinear optical speckle fields. *Neural Networks* (2023) 166:634–44. doi:10.1016/j.neunet.2023.08.001
- Minsky M, Papert S. *Perceptron: an introduction to computational geometry*. 19. Cambridge: The MIT Press (1969). expanded edition.
- Lecun Y, Bengio Y, Hinton G. Deep learning. *Nature* (2015) 521(7553):436–44. doi:10.1038/nature14539
- Martini G, Mirigliano M, Paroli B, Milani P. The Receptron: a device for the implementation of information processing systems based on complex nanostructured systems. *Jpn J Appl Phys* (2022) 61:SM0801. doi:10.35848/1347-4065/ac665c
- Mirigliano M, Paroli B, Martini G, Fedrizzi M, Falqui A, Casu A, et al. A binary classifier based on a reconfigurable dense network of metallic nanojunctions. *Neuromorphic Comput Eng* (2021) 1(2):024007. doi:10.1088/2634-4386/ac29c9
- Mambretti F, Mirigliano M, Tentori E, Pedrani N, Martini G, Milani P, et al. Dynamical stochastic simulation of complex electrical behavior in neuromorphic networks of metallic nanojunctions. *Sci Rep* (2022) 12(1):12234. doi:10.1038/s41598-022-15996-9
- Minnai C, Bellacicca A, Brown SA, Milani P. Facile fabrication of complex networks of memristive devices. *Sci Rep* (2017) 7(1):7955. doi:10.1038/s41598-017-08244-y
- Mirigliano M, Borghi F, Podestà A, Antidormi A, Colombo L, Milani P. Non-ohmic behavior and resistive switching of Au cluster-assembled films beyond the percolation threshold. *Nanoscale Adv* (2019) 1(8):3119–30. doi:10.1039/c9na00256a
- Mirigliano M, Decastri D, Pullia A, Dellasega D, Casu A, Falqui A, et al. Complex electrical spiking activity in resistive switching nanostructured Au two-terminal devices. *Nanotechnology* (2020) 31(23):234001. doi:10.1088/1361-6528/ab76ec
- Tarantino W, Colombo L. Modeling resistive switching in nanogranular metal films. *Phys Rev Res* (2020) 2(4):043389. doi:10.1103/PhysRevResearch.2.043389
- López-Suárez M, Melis C, Colombo L, Tarantino W. Modeling charge transport in gold nanogranular films. *Phys Rev Mater* (2021) 5(12):126001. doi:10.1103/PhysRevMaterials.5.126001
- Nadalini G, Borghi F, Košutová T, Falqui A, Ludwig N, Milani P. Engineering the structural and electrical interplay of nanostructured Au resistive switching networks by controlling the forming process. *Sci Rep* (2023) 13:19713. doi:10.1038/s41598-023-46990-4
- Casu A, Chiodoni A, Ivanov YP, Divitini G, Milani P, Falqui A. *In situ* TEM investigation of thermally induced modifications of cluster-assembled gold films undergoing resistive switching: implications for nanostructured neuromorphic devices. *ACS Appl Nano Mater* (2024) 7(7):7203–12. doi:10.1021/acsanm.3c06261
- Strukov DB, Stanley Williams R. Exponential ionic drift: fast switching and low volatility of thin-film memristors. *Appl Phys A* (2009) 94(3):515–9. doi:10.1007/s00339-008-4975-3
- Durkan C, Welland ME, Welland ME. Analysis of failure mechanisms in electrically stressed Au nanowires. *J Appl Phys* (1999) 86(3):1280–6. doi:10.1063/1.370882
- Mirigliano M, Milani P. Electrical conduction in nanogranular cluster-assembled metallic films. *Adv Phys* (2021) 6. doi:10.1080/23746149.2021.1908847
- Kagan M. On equivalent resistance of electrical circuits. *Am J Phys* (2015) 83(1): 53–63. doi:10.1119/1.4900918
- Rubido N, Grebogi C, Baptista MS. General analytical solutions for DC/AC circuit-network analysis. *Eur Phys J ST, Spec Top* (2017) 9. doi:10.1140/epjst/e2017-70074-2
- Montano M, Milano G, Ricciardi C. Grid-graph modeling of emergent neuromorphic dynamics and heterosynaptic plasticity in memristive nanonetworks. *Neurom. Comput. Eng* (2023) 2:014007. doi:10.1088/2634-4386/ac4d86
- de Arcangelis L, Redner S, Herrmann HJ. A random fuse model for breaking processes. *J de Physique Lettres* (1985) 46(13):585–90. doi:10.1051/jphyslet:019850046013058500
- Costagliola G, Boisa F, Pugno NM. Random fuse model in the presence of self-healing. *New J Phys* (2020) 22:033005. doi:10.1088/1367-2630/ab713f
- Zhu R, Hochstetter J, Loeffler A, Diaz-Alvarez A, Nakayama T, Lizier JT, et al. Information dynamics in neuromorphic nanowire networks. *Sci Rep* (2021) 11:13047. doi:10.1038/s41598-021-92170-7
- Ching W-K, Ng MK. Markov chains. In: *Models, algorithms and applications* (2006).
- Cover TM, Thomas JA. Entropy, relative entropy, and mutual information. In: *Elements of information theory* (2005). doi:10.1002/047174882x.ch2
- Roulston MS. Estimating the errors on measured entropy and Mutual Information. *Physica D* (1999) 125(3–4):285–94. doi:10.1016/S0167-2789(98)00269-3
- Dijkstra EW. A note on two problems in connexion with graphs. *Numer Math (Heidelberg)* (1959) 1(1):269–71. doi:10.1007/BF01386390
- NetworkX. NetworkX. Available from: <https://github.com/networkx/networkx> (Accessed October 26, 2023).
- Manning HG, Niosi F, da Rocha CG, Bellew AT, O'Callaghan C, Biswas S, et al. Emergence of winner-takes-all connectivity paths in random nanowire networks. *Nat Commun* (2018) 9(1):3219. doi:10.1038/s41467-018-05517-6
- Sporns O, Tononi G, Kötter R. The human connectome: a structural description of the human brain. *PLoS Comput Biol* (2005) 1(4):e42. doi:10.1371/journal.pcbi.0010042
- Tononi G, Sporns O, Edelman GM. A measure for brain complexity: relating functional segregation and integration in the nervous system. *Proc Natl Acad Sci U S A* (1994) 91(11):5033–7. doi:10.1073/pnas.91.11.5033