COLLECTIVE CONTROL OF MODULAR SOFT ROBOTS VIA EMBODIED SPIKING NEURAL CELLULAR AUTOMATA

Giorgia Nadizar & Eric Medvet
University of Trieste, Trieste, Italy
giorgia.nadizar@phd.units.it,emedvet@units.it

Stefano Nichele & Sidney Pontes-Filho
Oslo Metropolitan University, Oslo, Norway
{stenic,sidneyp}@oslomet.no

ABSTRACT

Voxel-based Soft Robots (VSRs) are a form of modular soft robots, composed of several deformable cubes, i.e., voxels. Each VSR is thus an ensemble of simple agents, namely the voxels, which must cooperate to give rise to the overall VSR behavior. Within this paradigm, collective intelligence plays a key role in enabling the emerge of coordination, as each voxel is independently controlled, exploiting only the local sensory information together with some knowledge passed from its direct neighbors (distributed or collective control). In this work, we propose a novel form of collective control, influenced by Neural Cellular Automata (NCA) and based on the bio-inspired Spiking Neural Networks: the embodied Spiking NCA (SNCA). We experiment with different variants of SNCA, and find them to be competitive with the state-of-the-art distributed controllers for the task of locomotion. In addition, our findings show significant improvement with respect to the baseline in terms of adaptability to unforeseen environmental changes, which could be a determining factor for physical practicability of VSRs.

1 INTRODUCTION AND RELATED WORKS

Biological organisms are intrinsically modular at different scales (Lorenz et al., 2011). The collective self-organization at the cellular level results in the emergence of complex bodies and brains without any form of centralized control. Such modularity allows for mechanisms of local interaction which, in turn, result in collective learning and adaptation.

In the artificial domain, modular robotics (Alattas et al., 2019) provides a framework for the investigations of biologically-inspired principles of collective control through distributed coordination of the agents composing the robot (Cheney et al., 2014). In addition, modular robots allow for a high degree of reconfiguration and self-assembly (Pathak et al., 2019), as well as fault tolerance and modules reusability. However, in order to exploit such opportunities, there is the need for modular distributed controllers, possibly embedded in each module. Therefore, the overall behavior of the robot is the result of the collective interplay of distributed sensing, local communication, and actuation of interacting body modules. In addition, identical modules would facilitate the reusability of the parts and robustness in case of damage (Huang et al., 2020). In this work, we focus on a specific type of modular robots, namely Voxel-based Soft Robots (VSRs) (Hiller & Lipson, 2012). Since VSRs are robots made of interconnected soft blocks (voxels), each module may be considered an agent in a collective. As such, mechanisms of collective intelligence are desired. While such mechanisms of collective intelligence are rather popular in the context of swarm robotics (Hamann, 2018), e.g., via self-assembly of thousands of robots through local interactions (Rubenstein et al., 2014), they are less explored in the context of modular robotics.

One paradigm of distributed neural control through local interactions of identical cells is Neural Cellular Automata (NCA) (Li & Yeh, 2002; Nichele et al., 2017; Mordvintsev et al., 2020). In case of robots composed of identical modules, each NCA cell can be embodied in a robot module. Such approach would in theory facilitate a physical realization as no global wiring nor centralized control would be needed. NCA have been successfully used to grow and replicate CA shapes and structures with neuroevolution (Nichele et al., 2017) and with differentiable learning (Mordvintsev et al., 2020), to produce self-organising textures (Niklasson et al., 2021), to grow 3D artifacts (Sudhakaran et al., 2021), for regenerating soft robots (Horibe et al., 2021), and for controlling reinforcement learning agents (Variengien et al., 2021).

In this work, we introduce a novel embodied NCA model based on more biologically plausible Spiking Neural Network (SNN) controllers. We name it embodied Spiking Neural Cellular Automata (SNCA). SNNs incorporate neuronal and synaptic states in their neuron models, as well as the concept of time. SNCA open up several opportunities in the domain of modular robotics, such as mechanisms of homeostatic adaptation and local learning rules, e.g., spike-timing-dependent plasticity. In addition, nearby modules communicate natively through spikes which are not generated at every clock-cycle but only when the internal neural membrane potential reaches a specific threshold, which in turn changes the membrane potential of the post-synaptic neuron, either within the same robotic module or in a nearby module in case of modular robots. The advent of neuromorphic hardware, which natively supports SNNs execution and learning, may provide orders of magnitude improvement in energy consumption as compared to traditional neural networks (Blouw et al., 2019). Low energy consumption is considered to be an enabling factor for the physical realization of self-organizing VSRs.

This work is organized as follows: in Section 2 we introduce VSRs, their morphology, and the proposed NCA-based controllers. In Section 3 we describe SNNs and SNCA. In Section 4 we present the experimental results and discuss the insights they provide. Finally, in Section 5 we draw the conclusions.

2 Collective control of Voxel-Based Soft Robots

Voxel-Based Soft Robots (Hiller & Lipson, 2012) are a kind of modular soft robots, composed of several elastically deformable cubes (*voxels*). In this work, we experiment with a 2D version of VSRs, simulated in discrete time and continuous space (Medvet et al., 2020b). The way in which VSRs achieve movement is a direct consequence of the unique combination of softness and modularity. The global behavior derives from the collective and synergistic contraction and expansion of individual voxels, similarly to what happens in biological muscles. Because of modularity, a VSR can be considered as an ensemble of simple sub-agents, the voxels, which are physically joined to obtain a greater structure, and whose individual behaviors influence eachother and concur to the emergence of coordination. Therefore, to characterize a VSR we need to specify how the voxels are assembled and what are their properties (*morphology*), and how the voxels compute their control signals and communicate with one another (*controller*).

2.1 VSR morphology: assembling individual voxels

The morphology of a VSR specifies how individual voxels are assembled, their sensory equipment, and their physical properties. A VSR can be represented as a 2D grid of voxels, describing their spatial organization and assembly. Adjacent voxels in the grid are rigidly linked: not only does this allow to assemble a robot out of primitive modules, but it also forces mutual physical interactions resulting in an overall complex dynamics. In addition, each voxel can be equipped with sensors to enable proprioception and awareness of the surroundings. For each voxel, we use three types of sensors: (a) *area* sensors, perceiving the ratio between the current area of the voxel and its rest area, (b) *touch* sensors, sensing if the voxel is in contact with the ground or with another body, and (c) *velocity* sensors, which perceive the velocity of the center of mass of the voxel along the x- and y-axes. We normalize sensor readings to be defined in $[0, 1]^4$.

Concerning physical properties, we model voxels as compounds of spring-damper systems, masses, and distance constrains (Medvet et al., 2020c), whose parameters can be changed to alter features like elasticity or actuation power. Each voxel changes volume (actually area, in the 2D case) over time, due to passive interactions with other bodies and the active application of a control signal. The latter is computed at each simulation time step and is defined in $[-1, 1]$, where -1 corresponds to maximum requested expansion and 1 corresponds to maximum requested contraction. In the employed model (Medvet et al., 2020b), contraction and expansion correspond to linear variations of the rest-length of the spring-damper system, proportional to the control signal received.

2.2 VSR CONTROLLER: THE EMBODIED NEURAL CELLULAR AUTOMATA PARADIGM

A VSR controller derives from the ensemble of individual voxels controllers. However, achieving coordination while keeping the intelligent control of each voxel fully independent of the others is a difficult task. In fact, most studies involving VSRs either rely on independent, yet not intelligent, controllers based on trivial sinusoidal functions (Hiller & Lipson, 2012; Corucci et al., 2018; Kriegman et al., 2018), or sacrifice modularity and deploy a central neural controller that has access to all voxels (Talamini et al., 2019; Ferigo et al., 2021; Nadizar et al., 2021; 2022). To solve this issue, Medvet et al. (2020a) introduced the concept of distributed neural controllers, which exploit message passing between neighbors to allow the emerge of coordination thanks to collective intelligence. Here, we follow along the same direction, combining the key ideas of modularity and collective intelligence, with an approach based on Neural Cellular Automata (NCA) techniques (Li & Yeh, 2002; Nichele et al., 2017; Mordvintsev et al., 2020), in which the lookup table of each Cellular Automaton (CA) cell is replaced by an Artificial Neural Network (ANN). More in detail, we consider each voxel as a single *embodied* NCA cell (from now on, simply referred to as NCA), which has access to the local sensor readings and to some information coming from the neighbors, to compute the local actuation value and the messages directed towards adjacent voxels. Our approach, anyhow, has a substantial difference with the standard NCA architectures: namely, it is strongly bound to the VSR morphology employed, as we only instantiate NCA cells in correspondence to the voxels.

Formally, at every time step k, each NCA takes as input a vector $\boldsymbol{x}^{(k)} = \left[\boldsymbol{r}^{(k)}\ \boldsymbol{i}_{\uparrow}^{(k)}\ \boldsymbol{i}_{\leftarrow}^{(k)}\ \boldsymbol{i}_{\downarrow}^{(k)}\ \boldsymbol{i}_{\rightarrow}^{(k)}\right]$ and produces as output a vector $\boldsymbol{y}^{(k)} = \text{ANN}_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(k)}\right) = \left[a^{(k)}\ \boldsymbol{o}_{\uparrow}^{(k)}\ \boldsymbol{o}_{\leftarrow}^{(k)}\ \boldsymbol{o}_{\downarrow}^{(k)}\ \boldsymbol{o}_{\rightarrow}^{(k)}\right]$ where $\boldsymbol{r}^{(k)} \in \mathbb{R}^4$ is the local sensor reading, $\boldsymbol{i}_{\uparrow}^{(k)}, \boldsymbol{i}_{\leftarrow}^{(k)}, \boldsymbol{i}_{\downarrow}^{(k)}, \boldsymbol{i}_{\rightarrow}^{(k)}$, each defined in $\mathbb{R}^{n_c}$, are values coming from adjacent voxels (from above, left, below, right, respectively, and set to $\boldsymbol{0} \in \mathbb{R}^{n_c}$ if no voxel is present in the corresponding direction), $a^{(k)}$ is the actuation value, $\boldsymbol{o}_{\uparrow}^{(k)}, \boldsymbol{o}_{\leftarrow}^{(k)}, \boldsymbol{o}_{\downarrow}^{(k)}, \boldsymbol{o}_{\rightarrow}^{(k)}$, each defined in $\mathbb{R}^{n_c}$, are values directed to adjacent voxels (to above, left, below, right, respectively), and $\boldsymbol{\theta}$ are the parameters of the ANN constituting the NCA. Values output by a NCA at time k are used by an adjacent NCA at time $k+1$, e.g., given a NCA a that outputs $\boldsymbol{o}_{a,\rightarrow}^{(k)}$, the NCA b at its right will have $\boldsymbol{i}_{b,\leftarrow}^{(k+1)} = \boldsymbol{o}_{a,\rightarrow}^{(k)}$.

We experiment with three ways of instantiating the general scheme described above, *non-uniform directional* ($\cancel{\text{U}}$D), *uniform directional* (UD), and *uniform non-directional* (U$\cancel{\text{D}}$), which differ in the homogeneity of the individual cells (uniform vs. non-uniform) and in the information passed between voxels (directional vs. non-directional). The first two schemes are inspired by already existing forms of distributed controllers (Medvet et al., 2020a) and we consider them as baselines, whereas the U$\cancel{\text{D}}$-NCA is novel.

The most evident, yet conceptually simple, difference lies in the *uniformity*: in $\cancel{\text{U}}$-NCA, cells have a different ANN in each voxel (each with parameters $\boldsymbol{\theta}_i$), whereas in U-NCA all cells ANNs share the same parameters $\boldsymbol{\theta}$. Therefore, it follows that, for a given ANN architecture, the amount of parameters of $\cancel{\text{U}}$-NCA is n_{voxels} times the amount of parameters of U-NCA.

The second distinguishing element is *directionality*. In $\cancel{\text{D}}$-NCA, cells send the same output to all the adjacent cells, i.e., $\boldsymbol{o}_{\uparrow}^{(k)} = \boldsymbol{o}_{\leftarrow}^{(k)} = \boldsymbol{o}_{\downarrow}^{(k)} = \boldsymbol{o}_{\rightarrow}^{(k)} = \boldsymbol{o}^{(k)}$ and $\boldsymbol{y}^{(k)} = \left[a^{(k)}\ \boldsymbol{o}^{(k)}\right]$, whereas in D-NCA, cells send, in general, different outputs. The D-NCA hence corresponds to the one originally proposed by Medvet et al. (2020a). Contrarily, $\cancel{\text{D}}$-NCA are more adherent to the original concept of NCA as $\boldsymbol{o}^{(k)}$ can be interpreted as the current state of the cell.

The proposed types of NCA controllers can be employed with any type of ANN, either fully-connected feed-forward NNs, i.e., multi-layer perceptrons (MLP), or more biologically plausible NNs, such as the SNNs described in the following section.

3 Spiking Neural Networks as robotic controllers

Spiking Neural Networks (SNNs) are a type of ANNs in which biological resemblance plays a fundamental role (Gerstner & Kistler, 2002). Often referred to as the third generation of ANNs (Maass, 1997), SNNs are characterized by a more biologically and neuroscientifically faithful neural model than classical ANNs.

The key element of SNNs is the modeling of the evolution over time of the membrane potential of neurons. Modifications of such potential are caused by incoming neural stimuli, which can either be excitatory (increasing the potential) or inhibitory (decreasing it). Neural stimuli occur in the form of spikes over time, which can propagate along synapses in order to reach different neurons of the SNN, enabling information passing within the network. The generation of said spikes is called *firing*, and arises whenever the membrane potential of a neuron exceeds a given threshold. Despite the binary nature of spikes, the intensity of any stimulus received by a neuron is modulated by the strength of the synapse connecting the firing neuron (pre-synaptic neuron) and the neuron receiving the spike (post-synaptic neuron). Not unlike classical MLPs, synapses are modeled as weighted connections, where the weights play the main role in determining the behavior of the ANN, and can be subject to task-oriented optimization.

What is indeed essentially different between MLPs and SNNs, is the way information is encoded, which is a direct consequence of the peculiarities of the two models. In particular, in MLPs there is no notion of time, and information is encoded in the form of real values traveling along the synapses. Conversely, SNNs are bound to the concept of time to compute the evolution of the neural membranes and for the propagation of spikes in the network. Within this framework, information is embedded in the time distribution of spikes. Hence, additional tools are required to interpret spike trains as real values and vice versa.

Given their high biological resemblance, SNNs are extremely promising robotic controllers. In fact, faithfully mimicking the functioning of the nervous systems of living organisms could be an enabling factor for bringing the desirable traits of biological organisms to artificial agents, e.g., autonomy or adaptability. Moreover, the possibility of deploying SNNs on highly energy efficient neuromorphic hardware (Li et al., 2014) is an additional profitable feature, which could be of paramount importance with reference to energy constraints.

3.1 Discrete time Leaky Integrate and Fire model

Several spiking neuron models have been proposed (Izhikevich, 2004), which, despite differing in terms of biological plausibility and computational costs, all share the main concepts derived from neuroscience. Among them, we employ the computationally efficient Leaky Integrate and Fire (LIF) model, simulated in discrete time. The LIF model represents the neural membrane as a capacitor, whose potential can be increased or decreased by inputs (excitatory or inhibitory), and exponentially decays with time. At each neural simulation time step h, the membrane potential $v_j^{(h)}$ of a LIF neuron j is updated as:

$$v_j^{(h)} = v_j^{(h-1)} + \sum_{i=1}^{n} w_{i,j} s_i^{(h)} - \Delta t_h \lambda_v v^{(h-1)}, \tag{1}$$

with $w_{i,j} \in \mathbb{R}$ being the synaptic weight of the i-to-j synapse, n being the number of incoming synapses, $s_i^{(h)} \in \{0, 1\}$ carrying pre-synaptic neuron spike, and $\Delta t_h = 1/f_h$ being the neural simulation time resolution. After the update, and if the membrane potential $v_j^{(h)}$ exceeds a threshold $\vartheta_j^{(h)}$, the neuron j outputs a spike, i.e., $s_j^{(h)} = 1$, and the membrane potential is reset to its resting value v_{rest}, otherwise $s_j^{(h)} = 0$.

We enhance the LIF model introducing the biological concept of homeostatic plasticity. Homeostasis is a self regulatory mechanism present at various sites of living organisms, which aims at re-establishing equilibrium in contrast to strong stimuli that could unbalance a system (Turrigiano & Nelson, 2004). In our case, homeostasis operates as a firing rate regulator, acting on the threshold

$\vartheta_j^{(h)}$ of neurons, to prevent excessive or too scarce activity:

$$\vartheta_j^{(h)} = \min\left(\vartheta_j^{(h-1)}, \sum_{i=1}^{n} w_{i,j}\right) + \psi_j^{(h)}, \tag{2}$$

with $\psi_j^{(h)}$ being a parameter updated as:

$$\psi_j^{(h)} = \begin{cases} \psi_j^{(h-1)} + \psi_{\text{inc}} & \text{if } s_j^{(h-1)} = 1 \\ \psi_j^{(h-1)} - \psi_j^{(h-1)} \lambda_\psi \Delta t_h & \text{otherwise.} \end{cases} \tag{3}$$

3.2 The LIF model inside NCA

We employ the LIF model described above within the ANN for the NCA in our robots. We simulate both the robot mechanical models and the LIF SNNs in discrete time: however, we update the simulation of the LIF SNNs with a greater frequency. Namely, we update the mechanical model with a frequency $f_k = 60\,\text{Hz}$ (the default value of the 2D-VSR-Sim by Medvet et al. (2020b)) and the SNNs with a frequency $f_h = 16 f_k \approx 1\,\text{kHz}$ (as suggested by Izhikevich (2004)).

In practice, at each k, we build a spike train $\left(s^{(16k)}, \dots, s^{(16k+15)}\right) \in \{0,1\}^{16}$ to be fed to the SNNs from each element of the sensor reading $\boldsymbol{r}^{(k)}$ and we compute the actuation value $a^{(k)}$ considering the spikes emitted by the corresponding SNN output neuron up to $h = 16k$. Concerning the information traveling between pairs of SNNs, we simply copy the spike trains with a delay of 16 time steps in the SNN simulation, i.e., one time step in the robot simulation, consistently with the description given in Section 2.2. For performing the sensor reading and actuation value conversions, we take inspiration from rate coding (Bouvier et al., 2019), where real values are mapped to a frequencies, which are then used to generate spike trains (Wang et al., 2008).

For spike trains to be fed to input neurons corresponding to sensor readings, we set:

$$s^{(h)} = \begin{cases} 1 & \text{if } \exists n \in \mathbb{N} \text{ s.t. } h = h_{\text{last}} + n \left\lfloor \frac{f_h}{f^{(k)}} \right\rfloor \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

where h_{last} is the time step of the last spike to the neuron (initially set to 0) and $f^{(k)} = r^{(k)}(f_{\text{max}} - f_{\text{min}}) + f_{\text{min}}$, $r^{(k)} \in [0,1]$ being the element of the sensor reading. That is, we first linearly scale the scalar input to a frequency $f^{(k)} \in [f_{\text{min}}, f_{\text{max}}]$ and then we emit spikes at frequency $f^{(k)}$, i.e., one spike each $\left\lfloor \frac{f_h}{f^{(k)}} \right\rfloor$ time steps of the SNN simulation. We set $f_{\text{min}} = 5\,\text{Hz}$ and $f_{\text{max}} = 50\,\text{Hz}$ for biological plausibility: hence, with the maximum scalar input $r^{(k)} = 1$, we emit one spike each $\frac{1\,\text{kHz}}{50\,\text{Hz}} = 20$ time steps, whereas with the minimum input $r^{(k)} = 0$, we emit one spike each $\frac{1\,\text{kHz}}{5\,\text{Hz}} = 200$ time steps.

For the actuation value, we set:

$$a^{(k)} = 2\left(\frac{1}{f_{\text{max}}} \frac{f_h}{n_w} \sum_{k'=k-n_w+1}^{k} \sum_{h=16k'}^{16k'+15} s^{(h)}\right) - 1. \tag{5}$$

That is, we count the spikes in the last n_w robot simulation time steps, we linearly scale this value to $[0,1]$ considering the maximum possible frequency f_{max}, and then we linearly scale to $[-1,1]$. The reason why we consider n_w robot simulation time steps, instead of just the current one, is to have a better resolution of the actuation value and, hence, a smoother control. After preliminary experimentation, we set $n_w = 5$.

4 Experiments and results

We performed an extensive experimental campaign to investigate how coordination can emerge from different forms of collective control. We aimed at evaluating if we could improve the baselines of distributed control for VSRs (Medvet et al., 2020a) with our novel contribution, the U$\not{\text{D}}$-SNCA.

Therefore, we addressed the following research question: *"are U$\not{D}$-SNCA superior with reference to the baselines of distributed control?"*. To determine the effectiveness of a collective controller, we deployed it onto a VSR, and optimized its parameters using as quality measure the velocity achieved by the robot performing locomotion on a flat terrain. In addition, we also assessed the controllers adaptability, by measuring the VSR velocity, after the optimization, on a set of unseen terrains, i.e., terrains not used to optimize the controller parameters. With the extent of obtaining more general results, we experimented with three different morphologies.

4.1 U$\not{D}$-SNCA VS. BASELINE EMBODIED NCA

In order to provide an answer to the posed research question, we started by optimizing the parameters of three variants of NCA for each of the three considered morphologies, for a total of nine VSRs optimizations.

Concerning the NCA, we took into consideration the UD- and $\not{U}$D-NCA as baselines, and we compared them against the U$\not{D}$-SNCA. We used a MLP with $\tanh$ as activation function for both baseline NCA, while we equipped the SNCA with a fully-connected feed-forward SNN based on the LIF neural model augmented with homeostasis, with the following parameters: $v_{\text{rest}} = 0\,\text{mV}$, $\lambda_v = 0.01\,\text{s}^{-1}$, $\vartheta^{(0)} = 1\,\text{mV}$, $\psi^{(0)} = 0\,\text{mV}$, $\psi_{\text{inc}} = 0.2\,\text{mV}$, $\lambda_\psi = 0.01\,\text{s}^{-1}$. For both ANNs, we used 1 hidden layer, setting its size equal to the size of the input layer. We set $n_c = 1$ for both UD- and $\not{U}$D-NCA, and $n_c = 4$ for the U$\not{D}$-SNCA, in order to make the sizes of the ANNs output layers equal. Our choice of NCA hyper-parameters was driven by some exploratory experiments and by previous work involving SNNs (Pontes-Filho & Nichele, 2019) and NCA (Nichele et al., 2017; Mordvintsev et al., 2020).

Regarding the morphologies, we experimented with 3 VSRs, a biped ▦, a comb ▦, and a worm ▭. We chose these morphologies to test the NCA controllers versatility, because they resemble different forms of living organisms, which take advantage of their diverse body shapes to achieve diversified gaits. We relied on 2D-VSR-Sim (Medvet et al., 2020c) for the VSRs simulation, leaving all parameters to their default values. We made the code for the experiments publicly available at `https://github.com/giorgia-nadizar/VSRCollectiveControlViaSNCA`.

To optimize the NCA parameters, we resorted to evolutionary algorithms (EAs) as they can easily overcome the struggles posed by the non-differentiability in SNNs. In addition, EAs are well suited for ill-posed problems with many local optima, which makes them particularly appropriate for optimizing the parameters of robotic controllers. In this study, we used a simple form of evolutionary strategy (ES). At first, the population is initialized with n_{pop} individuals, i.e., numerical vectors $\boldsymbol{\theta}$, all generated by assigning to each element of the vector a randomly sampled value from a uniform distribution over the interval $[-1, 1]$. Subsequently, n_{gen} evolutionary iterations are performed, until reaching a total of n_{evals} fitness evaluations. On every iteration, the fittest quarter of the population is chosen to generate $n_{\text{pop}} - 1$ children, each obtained by adding values sampled from a normal distribution $N(0, \sigma)$ to each element of the element-wise mean $\boldsymbol{\mu}$ of all parents. The generated offspring, together with the fittest individual of the previous generation, end up forming the population of the next generation, which maintains the fixed size n_{pop}. We used the following ES parameters: $n_{\text{pop}} = 36$, $n_{\text{evals}} = 30\,000$, and $\sigma = 0.35$. We verified that evolution was in general capable of converging to a solution with the chosen values, despite the different sizes of the search spaces corresponding to each NCA configuration.

We optimized VSRs for the task of *locomotion* on a flat terrain, the goal being traveling as fast as possible along the positive x-axis. We assessed the performance of a VSR by measuring its average velocity v_x along the x-axis during a simulation of $30\,\text{s}$. We discarded the first $5\,\text{s}$ of each simulation to exclude the initial transitory from the velocity measurements. We used v_x as fitness measure for selecting the best individuals in the ES. For each of the 9 VSRs resulting from the combination of 3 NCA and 3 morphologies, we performed 10 independent evolutionary optimizations, i.e., with different random seeds, for a total of 90 runs.

Besides testing the VSR effectiveness upon parameters optimization, i.e., at the end of evolution, we also appraised their adaptability. We define a VSR controller as *adaptable*, if it is able to achieve good performance in locomotion in spite of environmental changes. To evaluate this in practice, we took each optimized VSR and re-assessed it on a set of unseen terrains, i.e., terrains which none of its ancestors ever experienced locomotion on. In particular, we experimented with the following

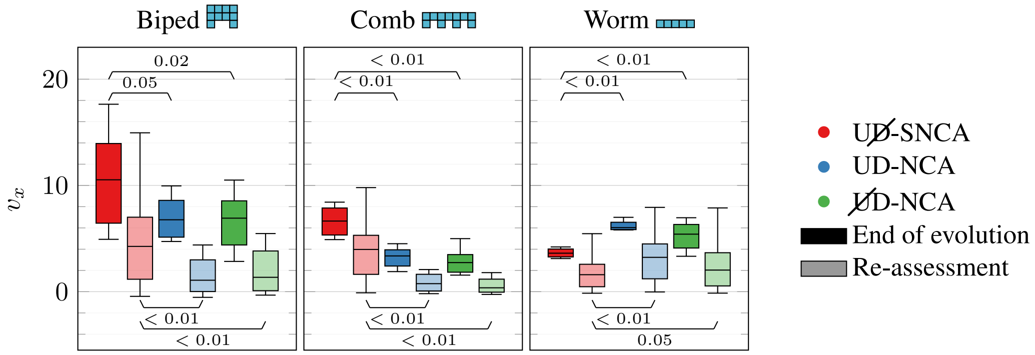

Figure 1: Box plots of the velocities v_x achieved by the best individuals at the end of evolution and upon re-assessment on unseen terrains for different VSR morphologies (plot columns) and embodied NCA controllers (color). Above pairs of boxes we report the p-values resulting from Mann-Whitney U tests with the null hypothesis of equality of the means.

terrains: (a) hilly with 6 combinations of heights and distances between hills, (b) steppy with 6 combinations of steps heights and widths, (c) downhill with 2 different inclinations, and (d) uphill with 2 different inclinations. As a result, we re-assessed each individual on a total of 16 different terrains; we define its adaptability as the average of the v_x on those terrains (each computed in a 30 s simulation, discarding the initial 5 s).

The results of our experimental evaluation are summarized in Figure 1. More in detail, for each of the considered VSR morphologies and NCA variants, we display the distribution of velocities achieved at the end of evolution by the best individuals, and their performance in terms of adaptability, i.e., the distribution of their average velocity on unseen terrains. In addition, above pairs of box plots, we report the p-values resulting from a two-sided Mann Whitney U statistical test; we consider, unless otherwise specified, $\alpha = 0.05$ as confidence level.

From Figure 1, we observe that for the biped and the comb morphologies, U$\cancel{D}$-SNCA are always significantly better than the baseline in terms of adaptability, and they are significantly better at the end of evolution in all but one case. However, the outcomes seem to be exactly opposite for the worm morphology, making it difficult to provide a general answer to the posed research question.

To further investigate on this apparent contradiction, we examined the behavior of a few evolved VSRs (videos are available at `https://giorgia-nadizar.github.io/VSRCollectiveControlViaSNCA/`) and found the reason behind the failure of the SNCA to be glaring for the worm morphology. In particular, we noticed that the NCA based on MLPs trigger a high frequency dynamics, resulting in a vibrating behavior, which for SNCA is prevented by homeostasis and $n_w = 5$ when converting spikes to actuation values. However, for the worm morphology, vibration appears to be the only effective gait, as these VSRs are not able to properly bend, having only one row of voxels at disposal. Conversely, the biped and comb morphologies have more complex structures, which allow the discovery of a wider range of efficacious gaits. In fact, when we inspected the behaviors of these two families of VSRs, we could notice a broader variety of gaits, with some tendencies to vibration among those controlled by MLP-based NCA. Avoiding vibrating behaviors, which have been shown to be a strong attractor in evolution (Medvet et al., 2021), is of paramount importance, as this type of movement severely hinders adaptability, and constitutes an insurmountable barrier for the physical practicability of VSRs, i.e., a form of reality gap (van Diggelen et al., 2021; Salvato et al., 2021). Even though it is possible to explicitly discourage vibrating behaviors, e.g., decreasing the actuation frequency, having a controller which avoids them by design is an undeniably significant accomplishment.

4.2 Strengths of the uniform non-directional SNCA

From the experimental outcomes, however, another question arises: *"what are the reasons behind the success of the U$\cancel{D}$-SNCA?"*. Namely, does the improvement lay in the non-directionality of the NCA or in the SNN employed?

To address the newly emerged points, we deepened our analysis with a supplementary experimental campaign, encompassing new combinations of NCA architectures, neural models, and morphologies, for a total of 12 additional VSRs to be optimized. Regarding the morphologies, we experimented with the biped and the comb, discarding the worm for the reasons highlighted in Section 4.1. For what concerns the controllers, we extended the previous experiments by evaluating all missing combinations of NCA architecture and neural models. Among the latters, we also included a SNN composed of LIF neurons for which we disabled homeostasis, keeping the value of the threshold fixed throughout the simulation to its initial value $\vartheta_i^{(h)} = \vartheta_i^{(0)} = 1\,\mathrm{mV}$. For each of the 12 new VSRs we repeated the experimental pipeline of Section 4.1.

We display the results, together with the outcomes of the previous experiments, in Table 1. Each cell of the table reports the median of the velocities achieved by VSRs at the end of evolution and upon re-assessment, grouped by morphology; each row corresponds to a NCA architecture, whereas we put neural models on the columns. We color cells proportionally to the median of velocities in order to better convey the information.

	End of evolution						Re-assessment					
	Biped			Comb			Biped			Comb		
	S-H	S	M	S-H	S	M	S-H	S	M	S-H	S	M
UD	6.5	5.5	8.0	3.8	5.9	3.8	3.7	2.3	2.2	2.6	3.8	1.4
~~U~~D	10.2	7.0	7.9	5.8	2.8	3.4	5.5	2.9	3.4	3.5	1.4	1.2
U~~D~~	13.1	13.0	9.3	7.5	5.9	8.5	5.5	5.5	2.2	5.1	3.9	4.7

Table 1: Medians of velocities v_x achieved by the best individuals at the end of evolution and upon re-assessment on unseen terrains for different morphologies. We put different NCA architectures on each row, and ANN models on the columns (M stands for MLP, S for SNN without homeostasis, S-H for SNN with homeostasis). Cells are colored proportionally to v_x.

From examining Table 1, we can investigate on the importance of the two aforementioned factors. First, to weigh the impact of the NCA architecture, we compare the medians of different rows for each column of the table. We observe that U~~D~~-NCA are not worse than both D-NCA in all but one case, and they always equal or outperform D-NCA if combined with SNNs. We speculate this descends from the fact that, especially in absence of agent specialization, i.e., in the case of U-NCA, it is easier for the prototype individual to learn to pass a single message to all its clone neighbors and correctly interpret the information received. In addition, we highlight that U~~D~~-NCA are less prone to triggering vibrating dynamics by design, and are thus more successful in combination with SNNs, which display and take advantage of the same trait.

Concerning the importance of the neural model, we note that SNNs, either with or without homeostasis, surpass MLPs in 10 out of 12 cases. To better appraise the influence of homeostasis in SNNs, we need to narrow our focus to the re-assessment results, where this neural model leads to neatly superior outcomes in all but one case, confirming its fundamental role in self-regulation and adaptation. Moreover, we can re-state that SNNs seem to be more naturally suited for being combined with U~~D~~-NCA, as both tend to move away from high frequency non-adaptable behaviors. Therefore, we can conclude that the superiority of our contribution lies in the successful combination of the novel U~~D~~-NCA architecture with SNNs with homeostasis.

5 Concluding remarks

We explored the paradigm of collective control of Voxel-based Soft Robots (VSRs), a form of simulated modular soft robots, appraising the emergence of coordination from the synergistic actuation of individual agents, i.e., the voxels constituting the VSR. Taking inspiration from NCA, a form of distributed neural control, and from state-of-the-art forms of embodied control of VSRs, we introduced the novel concept of embodied Spiking Neural Cellular Automata (SNCA), in which we used Spiking Neural Networks (SNNs) as elementary units. To evaluate the performance of the proposed SNCA as a robotic controller, we compared it against the state-of-the-art embodied controllers, optimizing the controller parameters of three different VSRs for the task of locomotion. Our experimental results show that the SNCA is not only competitive with the pre-existing controllers, but it also leads to significantly more adaptable agents, outperforming their rivals when faced with unforeseen circumstances. Moreover, we highlight a trend towards less reality-gap prone behaviors in VSRs controlled by SNCA, which paves the way for the physical practicability of such robots.

We believe our contribution can be considered as a starting point for several additional analyses, spanning across diverse research directions. Concerning SNNs, we plan to experiment with neuroplasticity in the form of unsupervised learning, to the extent of achieving greater generality and increased robustness of controllers (Qiu et al., 2020). In addition, we will address the problem of collective control with a cooperative coevolution strategy aimed at optimizing an ensemble of heterogeneous SNCA controllers (Potter & Jong, 2000).

REFERENCES

Reem J Alattas, Sarosh Patel, and Tarek M Sobh. Evolutionary modular robotics: Survey and analysis. *Journal of Intelligent & Robotic Systems*, 95(3):815–828, 2019.

Peter Blouw, Xuan Choo, Eric Hunsberger, and Chris Eliasmith. Benchmarking keyword spotting efficiency on neuromorphic hardware. In *Proceedings of the 7th Annual Neuro-inspired Computational Elements Workshop*, pp. 1–8, 2019.

Maxence Bouvier, Alexandre Valentian, Thomas Mesquida, Francois Rummens, Marina Reyboz, Elisa Vianello, and Edith Beigne. Spiking neural networks hardware implementations and challenges: A survey. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 15(2): 1–35, 2019.

Nicholas Cheney, Jeff Clune, and Hod Lipson. Evolved electrophysiological soft robots. In *ALIFE 14: The Fourteenth International Conference on the Synthesis and Simulation of Living Systems*, pp. 222–229. MIT Press, 2014.

Francesco Corucci, Nick Cheney, Francesco Giorgio-Serchi, Josh Bongard, and Cecilia Laschi. Evolving soft locomotion in aquatic and terrestrial environments: effects of material properties and environmental transitions. *Soft robotics*, 5(4):475–495, 2018.

Andrea Ferigo, Giovanni Iacca, Eric Medvet, and Federico Pigozzi. Evolving Hebbian Learning Rules in Voxel-based Soft Robots. 2021.

Wulfram Gerstner and Werner M Kistler. *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge university press, 2002.

Heiko Hamann. Swarm robotics: A formal approach. 2018.

Jonathan Hiller and Hod Lipson. Automatic design and manufacture of soft robots. *IEEE Transactions on Robotics*, 28(2):457–466, 2012.

Kazuya Horibe, Kathryn Walker, and Sebastian Risi. Regenerating Soft Robots Through Neural Cellular Automata. In *EuroGP*, pp. 36–50, 2021.

Wenlong Huang, Igor Mordatch, and Deepak Pathak. One policy to control them all: Shared modular policies for agent-agnostic control. In *International Conference on Machine Learning*, pp. 4455–4464. PMLR, 2020.

Eugene M Izhikevich. Which model to use for cortical spiking neurons? *IEEE transactions on neural networks*, 15(5):1063–1070, 2004.

Sam Kriegman, Nick Cheney, and Josh Bongard. How morphological development can guide evolution. *Scientific reports*, 8(1):1–10, 2018.

Xia Li and Anthony Gar-On Yeh. Neural-network-based cellular automata for simulating multiple land use changes using GIS. *International Journal of Geographical Information Science*, 16(4): 323–343, 2002.

Yi Li, Yingpeng Zhong, Jinjian Zhang, Lei Xu, Qing Wang, Huajun Sun, Hao Tong, Xiaoming Cheng, and Xiangshui Miao. Activity-dependent synaptic plasticity of a chalcogenide electronic synapse for neuromorphic systems. *Scientific reports*, 4(1):1–7, 2014.

Dirk M Lorenz, Alice Jeng, and Michael W Deem. The emergence of modularity in biological systems. *Physics of life reviews*, 8(2):129–160, 2011.

Wolfgang Maass. Networks of spiking neurons: the third generation of neural network models. *Neural networks*, 10(9):1659–1671, 1997.

Eric Medvet, Alberto Bartoli, Andrea De Lorenzo, and Giulio Fidel. Evolution of distributed neural controllers for voxel-based soft robots. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, pp. 112–120, 2020a.

Eric Medvet, Alberto Bartoli, Andrea De Lorenzo, and Stefano Seriani. 2D-VSR-Sim: A simulation tool for the optimization of 2-D voxel-based soft robots. *SoftwareX*, 12:100573, 2020b.

Eric Medvet, Alberto Bartoli, Andrea De Lorenzo, and Stefano Seriani. Design, validation, and case studies of 2D-VSR-Sim, an optimization-friendly simulator of 2-D Voxel-based Soft Robots. *arXiv preprint arXiv:2001.08617*, 2020c.

Eric Medvet, Alberto Bartoli, Federico Pigozzi, and Marco Rochelli. Biodiversity in evolved voxel-based soft robots. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 129–137, 2021.

Alexander Mordvintsev, Ettore Randazzo, Eyvind Niklasson, and Michael Levin. Growing neural cellular automata. *Distill*, 5(2):e23, 2020.

Giorgia Nadizar, Eric Medvet, Felice Andrea Pellegrino, Marco Zullich, and Stefano Nichele. On the effects of pruning on evolved neural controllers for soft robots. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pp. 1744–1752, 2021.

Giorgia Nadizar, Eric Medvet, Hola Huse Ramstad, Stefano Nichele, Felice Andrea Pellegrino, and Marco Zullich. Merging pruning and neuroevolution: towards robust and efficient controllers for modular soft robots. *The Knowledge Engineering Review*, 37, 2022.

Stefano Nichele, Mathias Berild Ose, Sebastian Risi, and Gunnar Tufte. CA-NEAT: evolved compositional pattern producing networks for cellular automata morphogenesis and replication. *IEEE Transactions on Cognitive and Developmental Systems*, 10(3):687–700, 2017.

Eyvind Niklasson, Alexander Mordvintsev, Ettore Randazzo, and Michael Levin. Self-Organising Textures. *Distill*, 6(2):e00027–003, 2021.

Deepak Pathak, Christopher Lu, Trevor Darrell, Phillip Isola, and Alexei A Efros. Learning to control self-assembling morphologies: a study of generalization via modularity. *Advances in Neural Information Processing Systems*, 32, 2019.

Sidney Pontes-Filho and Stefano Nichele. A Conceptual Bio-Inspired Framework for the Evolution of Artificial General Intelligence. *arXiv preprint arXiv:1903.10410*, 2019.

Mitchell A Potter and Kenneth A De Jong. Cooperative coevolution: An architecture for evolving coadapted subcomponents. *Evolutionary computation*, 8(1):1–29, 2000.

Huanneng Qiu, Matthew Garratt, David Howard, and Sreenatha Anavatti. Towards crossing the reality gap with evolved plastic neurocontrollers. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, pp. 130–138, 2020.

Michael Rubenstein, Alejandro Cornejo, and Radhika Nagpal. Programmable self-assembly in a thousand-robot swarm. *Science*, 345(6198):795–799, 2014.

Erica Salvato, Gianfranco Fenu, Eric Medvet, and Felice Andrea Pellegrino. Crossing the Reality Gap: a Survey on Sim-to-Real Transferability of Robot Controllers in Reinforcement Learning. *IEEE Access*, 2021.

Shyam Sudhakaran, Djordje Grbic, Siyan Li, Adam Katona, Elias Najarro, Claire Glanois, and Sebastian Risi. Growing 3D Artefacts and Functional Machines with Neural Cellular Automata. *arXiv preprint arXiv:2103.08737*, 2021.

Jacopo Talamini, Eric Medvet, Alberto Bartoli, and Andrea De Lorenzo. Evolutionary synthesis of sensing controllers for voxel-based soft robots. In *Artificial Life Conference Proceedings*, pp. 574–581. MIT Press, 2019.

Gina G Turrigiano and Sacha B Nelson. Homeostatic plasticity in the developing nervous system. *Nature reviews neuroscience*, 5(2):97–107, 2004.

Fuda van Diggelen, Eliseo Ferrante, Nihed Harrak, Jie Luo, Daan Zeeuwe, and AE Eiben. The Influence of Robot Traits and Evolutionary Dynamics on the Reality Gap. *IEEE Transactions on Cognitive and Developmental Systems*, 2021.

Alexandre Variengien, Sidney Pontes-Filho, Tom Glover, and Stefano Nichele. Towards self-organized control: Using neural cellular automata to robustly control a cart-pole agent. *Innovations in Machine Intelligence*, 1, 2021.

Xiuqing Wang, Zeng-Guang Hou, Anmin Zou, Min Tan, and Long Cheng. A behavior controller based on spiking neural networks for mobile robots. *Neurocomputing*, 71(4-6):655–666, 2008.