Received 15 July 2021; revised 10 September 2021 and 28 September 2021; accepted 29 September 2021.
Date of publication 8 October 2021; date of current version 16 November 2021.

Digital Object Identifier 10.1109/JXCDC.2021.3118061

Combining Accuracy and Plasticity in Convolutional Neural Networks Based on Resistive Memory Arrays for Autonomous Learning

STEFANO BIANCHI[1] (Member, IEEE), IRENE MUÑOZ-MARTÍN[1] (Member, IEEE), ERIKA COVI[2], ALESSANDRO BRICALLI[3], GIUSEPPE PICCOLONI[3], AMIR REGEV[3], GABRIEL MOLAS[4] (Senior Member, IEEE), JEAN FRANÇOIS NODIN[4], FRANÇOIS ANDRIEU[4] (Senior Member, IEEE), and DANIELE IELMINI[1] (Fellow, IEEE)

[1]Dipartimento di Elettronica, Informazione e Bioingegneria and Italian Universities Nanoelectronics Team, Politecnico di Milano, 20133 Milan, Italy
[2]Namlab, 01187 Dresden, Germany
[3]Weebit Nano, Hod Hasharon 4527713, Israel
[4]Commissariat à l'Énergie Atomique et aux Énergies Alternatives (CEA), Laboratoire d'Électronique et de Technologie de l'Information (LETI), 38054 Grenoble, France

CORRESPONDING AUTHOR: D. IELMINI (daniele.ielmini@polimi.it)

This work was supported in part by the European Research Council (ERC) through the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement 648635 and in part by the Italian Minister for University and Research under Grant R164TYLBZP.

This article has supplementary downloadable material available at https://doi.org/10.1109/JXCDC.2021.3118061, provided by the authors.

ABSTRACT Nowadays, artificial neural networks (ANNs) can outperform the human brain's ability in specific tasks. However, ANNs cannot replicate the efficient and low-power learning, adaptation, and consolidation typical of biological organisms. Here, we present a hardware design based on arrays of SiO_x resistive switching random-access memory (RRAM), which allows combining the accuracy of convolutional neural networks with the flexibility of bio-inspired neuronal plasticity. In order to enable the combination of the stable and the plastic attributes of the network, we exploit the spike-frequency adaptation of the neurons relying on the multilevel programming of the RRAM devices. This procedure enhances the efficiency and accuracy of the network for MNIST, noisy MNIST (N-MNIST), Fashion-MNIST, and CIFAR-10 datasets, with inference accuracies of about 99%–89%, respectively. We also demonstrate that the hardware is capable of asynchronous self-adaptation of its operative frequency according to the fire rate of the spiking neuron, thus optimizing the whole behavior of the network. We finally show that the system enables fast and accurate filter retraining to overcome catastrophic forgetting, showing high efficiency in terms of operations per second and robustness against device non-idealities. This work paves the way for the theoretical modeling and hardware realization of resilient autonomous systems in dynamic environments.

INDEX TERMS Catastrophic forgetting, complementary learning systems, continual learning, convolutional neural networks (CNNs), resistive switching random-access memory (RRAM), spike-timing-dependent plasticity (STDP), supervised learning, unsupervised learning.

I. INTRODUCTION

IN nature, the organisms consolidate and elaborate the past information through neuropsychological processes that are in equilibrium with continual refinement and adjustment of the new incoming data. This ability of cohesion is the basis of the resilience that animals and humans show throughout their lives. In biological sciences, the theory of complementary learning systems explains the mutual contribution of hippocampus and neocortex to both strengthen the previous information and elaborate on the new incoming data for better precision and accuracy [1], [2]. In particular, the hippocampus accounts for the fast acquisition of the new experience, whereas the neocortex is specialized in giving robustness to the previous knowledge [3].

This mutual effort is what enables the efficient adaptation of biological organisms to evolving situations, an ability that goes under the name of ''resilience'' [4].

The artificial intelligence community has always tried to seek inspiration in such biological behavior for developing efficient computing machines. Artificial neural networks (ANNs) have recently demonstrated outstanding breakthroughs in tasks such as natural language processing [5] and object recognition, showing high accuracy in classification of large datasets [6]. This is mostly achieved by the supervised training technique based on backpropagation [7], where each weight of the neural network is progressively optimized in response to the presentation of labeled training information. However, since the early years of investigation on ''deep learning,'' supervised neural networks appeared too rigid and static with respect to the biological world since the high accuracy of recognition was related only to specific tasks previously prepared via time- and power-consuming training algorithms [8]. This lack of adaptation was summarized in the expression ''stability-plasticity dilemma,'' which highlights the difficulty of cohesion between consolidation and adaptation in ANNs, in contrast to humans and animals [9]. In particular, deep neural networks, such as feedforward multilayer perceptron (MLP) networks or convolutional neural networks (CNNs), suffer from the so-called ''catastrophic forgetting'' problem, which erases, during the training procedure of the new information, the previously learned data [10].

Several solutions have been proposed for overcoming this problem in ANNs, such as: 1) task-specific synaptic consolidation [11]; 2) replacement of the old redundant information, useless for achieving better accuracy, with new one [12]; and 3) allocation of additional neural resources [13]. However, all these techniques still appear too static toward the creation of real autonomous networks since they require too great complexity in order to be efficiently implemented in hardware [14].

The supervised approach to object and pattern classification is not the only one pursued in the field of artificial computing. More precisely, unsupervised bio-inspired Hebbian learning has recently known rapid progress in theoretical and applied research especially with spike-timing-dependent plasticity (STDP) [15]. The main advantage of Hebbian STDP is its intrinsic plasticity, at the expense of lower accuracy with respect to standard trained networks based on backpropagation [14]. Furthermore, optimal prediction and inference of continuously transforming environments seem to rely on a sort of short-term STDP, thus highlighting the timing-dependent plasticity as a good way to implement resilient neuromorphic systems [16]. STDP has been largely demonstrated in both CMOS technology [17] and memristive devices, such as with phase change memories (PCMs) [18], [19] and resistive switching random-access memory (RRAM) [20], [21]. In particular, the memristive devices used as synaptic elements are of utmost interest for the machine learning [22], [23], mainly for their compactness, stacking capability, and multilevel programming feature [24].

In the last few years, there have been some attempts to merge the supervised and unsupervised learning approaches in order to exploit the benefits of both stability and plasticity properties. These works offered an artificial approach to the biological complementary learning of hippocampus and neocortex [3]. In particular, it has been demonstrated that the merging of supervised and unsupervised learning blocks using a static CMOS logic to properly modify the convolutional results provides resilience to achieve plasticity inside a CNN [14], [22].

Here, we propose a SiO_x RRAM-based network capable of synthesizing, in hardware, a complementary learning system. The architecture relies on matrix-vector-multiplication (MVM) to accelerate the convolutional section and on bioinspired techniques in the unsupervised part, such as STDP and homeostatic spike-frequency adaptation (SFA). SFA is a biological neuronal mechanism [25], also studied for neuronal electrical models [26], which counteracts the intrinsic divergence of the Hebbian learning [27]. Differently, with respect to these previous works, we propose self-referential hardware capable of resilience without relying on external control signals. During inference, the bio-inspired section of the network clusters the non-trained information in original groups, for transfer learning. All the elements joining the same group share a set of common features. Such clustering activity provides the possibility of partial retraining of some filters to optimize the classification accuracy. We validate the system with MNIST, N-MNIST, Fashion-MNIST, and CIFAR-10 datasets via pure analytical simulations, Monte Carlo simulations, and dedicated hardware experiments with respect to the memristive section of the network. The hardware design mainly focuses on the synthesis of the filters, the convolution procedure, and the subsequent unsupervised classification via STDP. All the results are supported by analysis of circuital performances with experiments and simulations of the network efficiency in terms of ops/s, throughput, and latency. We finally demonstrate that the circuit plastically adapts its operative frequency for power saving and enables continual learning of up to 50% nontrained classes. This work allows us to overcome the catastrophic forgetting problem and proposes a hardware cohesion between the benefits of artificial and spiking neural networks, paving the way for the theory and hardware architectures of efficient autonomous neural networks. In the Supplementary Material of this work, a detailed explanation about the experimental setup is proposed along with further studies with respect to lifelong learning.

II. STRUCTURE OF THE NETWORK

The network relies on the use of multiple arrays of SiO_x RRAM devices, whose cross section is presented in Fig. 1(a). The RRAM devices are built on 130-nm technology CMOS wafers in the back end of line (BEOL) between the fourth and fifth metal layers. First, a TiN bottom electrode (BE) is created as an inert electrode. Afterward, SiO_x is deposited, followed by: 1) a Ti layer and 2) a TiN layer. The memory dots are, thus, obtained by etching. Then, a passivation layer

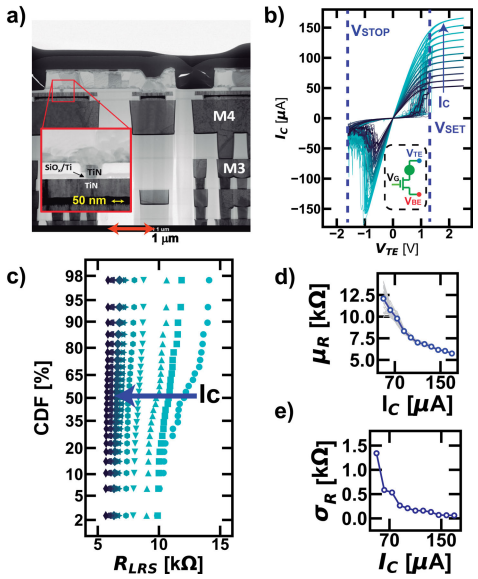

FIGURE 1. **(a) Cross section of the implemented RRAM devices at the microscope. (b) *I–V* characteristics at varying compliance current I_C. (c) Modulation of the LRS as a function of the compliance current during set operation. Note that (d) higher the average resistive state, (e) higher the standard deviation.**

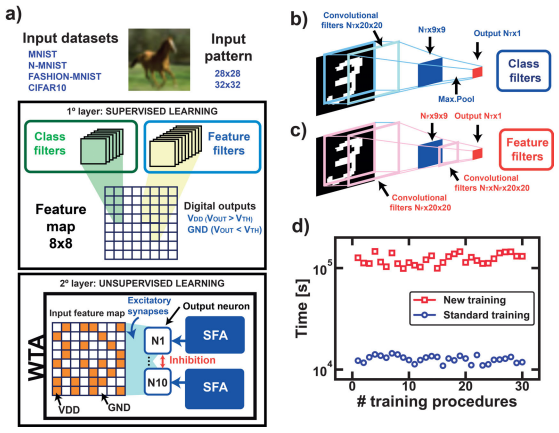

FIGURE 2. **(a) Schematic of the implemented network to get continual learning: the system is divided into two main blocks that correspond to the procedures of supervised learning and unsupervised learning, respectively. The supervised learning procedure uses convolutional filters to extract information from the input image. In particular, (b) class filters are trained to recognize only one specific class, while (c) feature filters contain general shapes that, for transfer learning, can be retrieved even in nontrained images. (d) Split and merge algorithm enables lifelong learning in neural networks but requires slower training procedures with respect to standard cases, hence more training cost. However, once new information appears at the input, the network is able to classify it without total retraining of the synaptic weights, thus avoiding the catastrophic forgetting typical of standard ANNs.**

is deposited. Finally, the top electrode (TE) contact is opened, and the fifth metal line is processed [28]. Fig. 1(b) shows the *I–V* characteristics of the devices under increasing conditions of compliance current I_C. The higher I_C, the wider the resistive window of the low resistive state (LRS) with respect to the high resistive state (HRS), as indicated in Fig. 1(c) and (d). Such window enlargement is proportional to a progressive reduction of the resistive level variability (e), thus leading to a better synaptic weight precision. All the measurement results present device-to-device variations since the neural network that we propose relies on several kilobits of RRAM elements.

A. CONVOLUTIONAL FILTERS

Fig. 2(a) shows the structure of the novel network from a high-level algorithmic point of view. The principal idea is to merge the benefits introduced in terms of accuracy by the CNN and the resilience typical of unsupervised learning in spiking neural networks. The first part of the network relies on two types of convolutional filters whose principal aim is to extract information from the input dataset: these two types of filters are named "class" and "feature" [14], [18]. The class filter is the result of a software training procedure whose aim is to give as convolutional output a digital "1" if and only if a specific class of the training dataset appears at the input; the convolutional training algorithm is implemented on a single layer convolutional network with max-pooling [see Fig. 2(b)]; and the only difference with respect to the usual training algorithm is that the training network is constrained to give "1" as output when the class corresponding to that neuron appears as training image at the input.

The feature filters, instead, are taken from a fully convolutional approach, where each filter directly maps a geometrical feature (a ray, a curve, an angle, and so on) [see Fig. 2(c)]. All the filters are first trained and then tested in order to group the best combination for correct inference. Such a training procedure is more expensive than the usual [see Fig. 2(d)] since it requires several parallel training processes and, thus, a worse overhead in terms of energy consumption. However, this split and merge algorithm gives the possibility of dividing the classical convolutional training in a set of simpler procedures specialized in bringing out a specific set of filters, which is positive in terms of energy consumption once retraining is taken into consideration, as explained in Section IV. During testing, in fact, the filters are convolved with the input in order to get a "found" or "not found" digital response. Such responses will then serve as input to the subsequent layers of the network in order to provide classification by exploiting unsupervised resilient STDP.

Note that the algorithm gives the possibility of overcoming the classical problems related to the static behavior of CNNs (catastrophic forgetting in presence of new information to learn) [29].

B. CONVOLUTION WITH RRAM ARRAYS

We used the SiO_x RRAM devices as synaptic elements for mimicking the MVM between the input and the filters. The MVM enhances the speed of the circuits with respect to standard CMOS procedures [24], [30]. Each filter is synthesized using binary RRAM devices, i.e., LRS or HRS

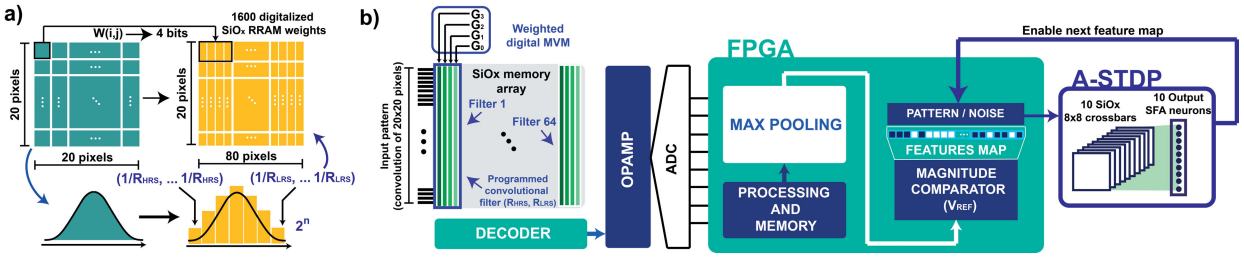

FIGURE 3. **(a) Digital implementation of the convolutional weights using 4 RRAM bits. (b) Digital MVM and consequent storage of the results into the FPGA by using, as a front end, a decoder, an operational amplifier, and an ADC. Once the feature maps are ready, they undergo STDP unsupervised learning for pattern classification. Note that it is possible to map negative values of the convolutional filters by means of RRAM-based 2's complement method.**

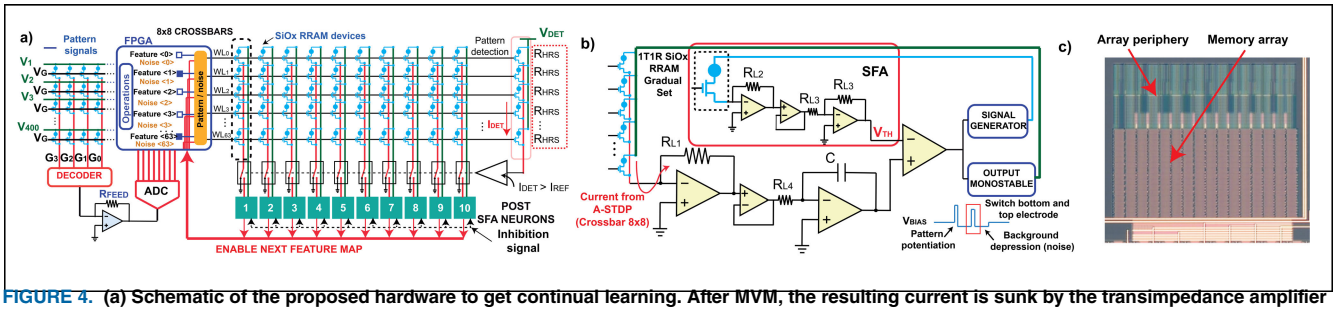

FIGURE 4. **(a) Schematic of the proposed hardware to get continual learning. After MVM, the resulting current is sunk by the transimpedance amplifier and then converted by the ADC. Once collected in the registers of the FPGA, these responses constitute an original set of responses, which is the new pattern presented to the STDP learning procedure. Note that the pattern of the STDP is triggered by the spike activity of the postneurons (POSTs). A further bitline is used to control the uncorrelated input signal that causes background depression [21]. (b) Structure of the CMOS neurons used for the STDP protocol: the current coming from the bitline is integrated. Once the neuron fires, a further RRAM device is gradually set in order to increase the threshold of the neuron, thus acting as a homeostatic control mechanism. (c) Top view of the memory array and of the peripheral circuitry. Note that peripheral electronics can be shared among different neurons in order to save area and power.**

[see Fig. 3(a)], to configure a digital word for every weight of the filter. Note that it is also possible to implement the 2's complement method to map positive and negative values of the synaptic weights. Every bitline is selected by a decoder, and the read current is sunk by a transimpedance amplifier [see Fig. 3(b)]. Then, the output voltage values are converted and processed by a field-programmable gate array (FPGA, in this case, the Xilinx Zync 7000 system on chip mounted in the Zedboard development kit), which processes the results of each bitline (from least to most significant) and executes a comparison with respect to a fixed threshold. In this way, for every input pattern, we obtain an original feature map, which should be ideally constant for those inputs belonging to the same class.

C. HOMEOSTATIC UNSUPERVISED LEARNING WITH RRAM ARRAYS

The feature map retrieved in the computation of the FPGA directly feeds the last layer of the network, which is designed to satisfy the STDP procedure with SFA, Fig. 4. Note that the RRAM devices are here used as both excitatory synapses [see Fig. 4(a)] and the internal state of each classification neuron [see Fig. 4(b)], a solution that could enable autonomous exploration of harsh environments by artificial agents [31], [32].

The excitatory synapses follow the digitalized STDP learning procedure: those synapses connecting an active position of the incoming pattern and the spiking output neuron are potentiated (set to LRS), while the other synapses (''background'' positions) are progressively depressed by using an uncorrelated signal, usually known as ''noise'' [21], [22], interspersed with the input feature map. Note that the internal state of each output neuron undergoes multilevel programming by proper modulation of the compliance current, I_C. The internal state of each neuron keeps track of the firing history of that neuron since the resistive value of the device is progressively increased every time that neuron fires [see Fig. 4(b)]. In fact, once the LRS gets progressively smaller in resistance, the threshold value of the neuron increases. In particular, this procedure: 1) counteracts the divergent growth of the synaptic weights under Hebbian learning by tuning the fire rate of the neuron, as it happens in biology [25] and 2) provides better specialization under winner-take-all (WTA) classification since every neuron is likely to fire to a specific pattern for which the STDP excitatory synapses have been specialized [31].

Note also that the peripheral electronics can be shared in order to reduce the area and power consumption [see Fig. 4(b)]. This advantage, considering also the in-memory computing approach, makes our solution feasible for large system synthesis, in contrast with pure bio-inspired CMOS-based approaches.

Considering the best set of filters after training, we achieve the final accuracies visible in Fig. 5. We also tested the inference ability of the network for two less-common datasets with respect to the usual MNIST and CIFAR-10: 1) the

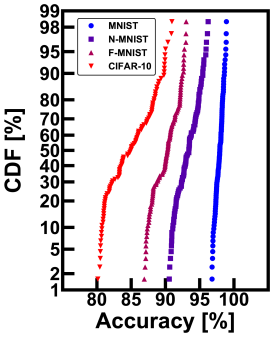

FIGURE 5. **Simulation of the full accuracies of the four datasets considering the non-idealities of the devices: a lower resistive window in the convolutional section, the LRS and HRS variabilities in the digitalized weights, and the switching errors during Hebbian learning.**

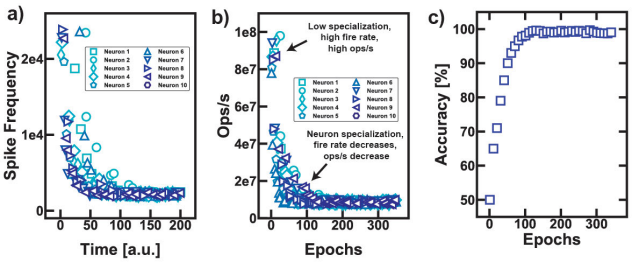

FIGURE 6. **(a) Decrease in the neuronal firing rate due to the spike frequency adaptation. (b) Calculated operations per second (ops/s) of the network during inference: note that the ops/s decreases at increasing homeostatic efficiency. (c) Increase in the classification accuracy during inference due to the spike frequency adaptation mechanism.**

N-MNIST, in which 1/9 of the pixels of each MNIST image is random and 2) the Fashion-MNIST, a collection of clothes images from Zalando [33].

In particular, the non-idealities of the devices (degradation of the resistive window in time, variabilities, not successful programming conditions, and so on) cause a significant drop in accuracy (from 1% of MNIST to 4.4% of CIFAR-10, on average), as shown by the distributions of Fig. 5.

III. HARDWARE EFFICIENCY

As shown in Fig. 4, the whole hardware setup works around the spike frequency block, which is the core element to provide efficient accuracy and plasticity for continual learning: whenever a neuron fires, the next feature map is ready to be presented to the unsupervised section; at the same time, the internal state of the firing neuron partially sets the corresponding control device, thus raising the threshold of that neuron and causing a progressive reduction of the firing rate. Thus, the number of operations per second (ops/s) is limited by the maximum speed of response of the unsupervised block and by the spike-frequency mechanism that progressively increases the unsupervised elaboration time of each feature map. The ops/s are calculated taking into consideration the synaptic-level MVM computation, and the clock cycles are needed to elaborate the information and the STDP learning procedure.

Fig. 6(a) shows the progressive frequency reduction during Hebbian learning of ten classes, highlighting the very first step of neuronal specialization (the firing rate decreases in time). Note that, to resemble the bio-inspired time for each epoch, the elaboration time should stay around 10 ms [21], which is too long to provide high throughput under inference. Reducing the epoch reference time, thus, is fundamental to achieve a substantial improvement of the network in terms of operations per second (ops/s). We have safely set the electronics in order to provide the fastest neuronal response of 40 μs. Note that this solution makes more area-efficient our design with respect to previous CMOS-based works [17] since respecting the biological timescale implies larger neuronal integration capacitances. Furthermore, the area consumption of memristors is limited to the selector, while CMOS synapses require a design with several transistors [17].

This means that the supervised block must perform the convolution of 64 filters of dimension 20×20, i.e., 81 shifts on the 28×28 input images, in less than 40 μs. Note that we perform, in parallel, the convolutional steps for each filter: we directly map the convolutional movement in hardware by programming for each filter its convoluted shape with respect to the fixed input. Considering also the consequent digitalization of each weight, we get a number of needed operations for every filter equal to $N_{filters} \times N_{bit} = 64 \times 4 = 256$.

Fig. 6(b) shows the estimated ops/s tendencies as a function of the epoch (i.e., feature map presentation at the input of the unsupervised block) also considering the further steps of digital conversion, pooling, and feature map definition. Note that the SFA block progressively reduces the fire rate toward a steady-state condition, causing a reduction of ops/s but an increased classification accuracy and neuronal specialization [see Fig. 6(c)].

Since the network is operated in a pipeline, we have a latency per input image in the order of two times the average response of the firing neurons, as indicated by the Monte Carlo simulations of Fig. 7(a). Such an approach enables the testing of the MNIST, Fashion-MNIST, and N-MNIST in less than 1 s, with an expected throughput of images in the order of 25 000 patterns/s. Of course, such result is strongly limited when an RGB dataset is used as a benchmark, as CIFAR-10, with a substantial increase in the latency and a consequent reduction of the throughput [see Fig. 7(b)]. The higher latency is due to both the RGB colors and the management of more data. Note that our architecture has a slightly lower efficiency in terms of throughput with respect to fully supervised approaches. However, this is balanced by the resilient properties of the network that we propose [30].

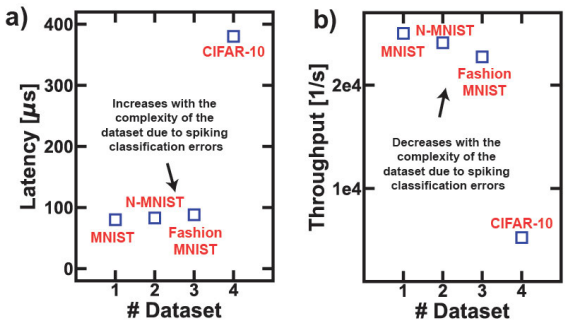

FIGURE 7. Monte Carlo simulations for (a) latency and (b) throughput of the inference machine as a function of the dataset under examination. Note that the CIFAR-10 requires a higher elaboration of data, hence increased latency and reduced throughput.

IV. DISCUSSION

A. CONTINUAL LEARNING

The plasticity introduced in the network by the STDP and the homeostatic control mechanism (via SFA) provides the resilience to accept new incoming feature maps for which the network has never been trained for (see Fig. 8). Here, the network has been previously trained with five classes (5–9), while the others are completely new during inference. However, these new classes provide original feature maps [see Fig. 8(a)] for transfer learning of the trained filters.

Generally, transfer learning refers to the use of previously acquired information in one specific domain to bring out a solution to a problem in a novel environment [34]. Standard approaches to transfer learning refer to a large domain of data that share invariant relational information for further classification capabilities [35], such as for zero- and one-shot learnings [36]. On the other hand, transfer learning is here used to enable a continual and resilient evolution of the network by exploiting the classification of new patterns during the unsupervised learning procedure.

For instance, all the images whose label is 4 (nontrained class) should excite a further output neuron to recognize the input patterns joining that new class. This procedure is experimentally demonstrated for the nontrained MNIST numbers from 0 to 4 in Fig. 8(b) and (c), where the synaptic time evolutions are shown. Fig. 8(d) also shows the scatter plot for a limited section of the firing responses. Note that, when the new patterns appear during inference, they are first misunderstood since the neurons have not been specialized yet and each neuronal threshold is randomly low. Once the fire rate reaches the steady state (lower firing rate since the neuronal threshold has increased), the errors diminish, and the accuracy improves, as already discussed in Fig. 6.

Since the new classes are classified by exploiting transfer learning, the feature maps may have slight variations from image to image. The activated pixels are sparse, and the corresponding average feature map is slightly blurred, as visible in the snapshots t_1, t_2, and t_3 of Fig. 9(a) for the CIFAR-10 dataset. This affects the final accuracy since, for instance, not every ''deer'' pattern hits the threshold of the neuron. Only limited variations ($<$15% or $<$20%) of the feature map density are tolerated by STDP, as indicated in Fig. 9(b). Such an effect gets more significant as the number of nontrained classes increases. This is mainly due to the decreasing efficiency of transfer learning and the increasing confusion between similar patterns during Hebbian learning [14]. Thus, we substantially have a constant decrease in the accuracy of nontrained classes and in the global accuracies, as the trained section gets progressively smaller. However, as indicated in Fig. 9(c), the accuracies of the nontrained classes were around 77% for MNIST and 65% for CIFAR-10 datasets, while the global accuracies range from 88% to 77%, respectively.

B. INCREMENTAL CONTINUAL LEARNING

It is possible to postprocess the filter responses after inference to get rid of those filters whose activity is not significant for the definition of the nontrained feature maps. For instance, if a ''feature filter'' individuates a particular shape in almost all the nontrained images, then the information given to the feature map is not useful. At the same time, a filter that sometimes gives a positive response for a pattern of a class and sometimes not for another pattern joining the same class is a source of confusion for the network (it causes misunderstanding for the unsupervised procedure and thus subsequent misclassification [see Fig. 9(b)]). During the continual learning of MNIST, for example, where the new classes are the numbers from 0 to 4, we have, on average, the responses of the convolution for the nontrained classes provided by the color map of Fig. 10(a). Note that some filters have an intermediate response (from 30% to 60%) to the filters, thus resulting as ''unstable points'' once the feature map of the corresponding new class arises. Those filters are harmful for the classification of the new classes since they create competition and misunderstanding among the nontrained patterns joining the same class.

Thus, as expressed in Fig. 10(b), we can get rid of some of those filters and proceed with the retraining in accordance with the clustered information during testing. Note that the synaptic retraining is focused on a limited section of the network since it regards fixed groups of synapses that are used as filters. Once, after a postprocessing study, a filter has to be eliminated and substituted, it is enough to erase that part of memory keeping the others untouched [see Fig.10(b)].

The retraining can be performed using the clustered information by the further classification neurons during the last inference activity. If, for instance, the 95% of responses of a neuron was due to the pattern joining the same new class, then those patterns are used to retrain the new filters, thus creating a set of responses that constitute the block of the continual learning activity. The global accuracies after retraining the filters are highlighted in Fig. 10(c), where a substantial improvement is highlighted. Note that the achieved accuracy is not the theoretically highest possible since the gradual

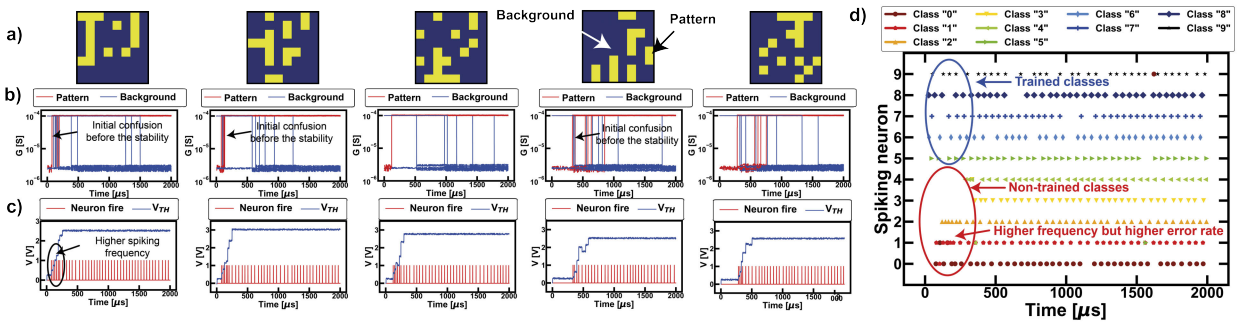

FIGURE 8. (a) Experimental examples of feature maps for five nontrained classes of MNIST with (b) synaptic evolution in time and (c) spike-frequency modulation effect. (d) Scatter plot of the fire activity of trained and nontrained neurons.

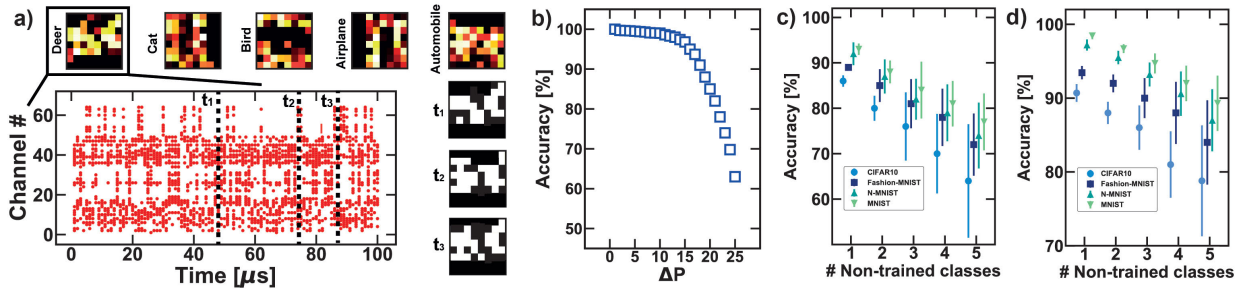

FIGURE 9. (a) Nontrained classes from the CIFAR-10 dataset. Note that the "deer" presents three substantial different patterns at t_1, t_2, and t_3. (b) This difference in terms of pattern density is the main source of misunderstanding during unsupervised STDP since it could lead to erroneous spiking activity. (c) Nontrained class accuracy as a function of the number of new classes. (d) Global accuracies as a function of the number of nontrained classes.

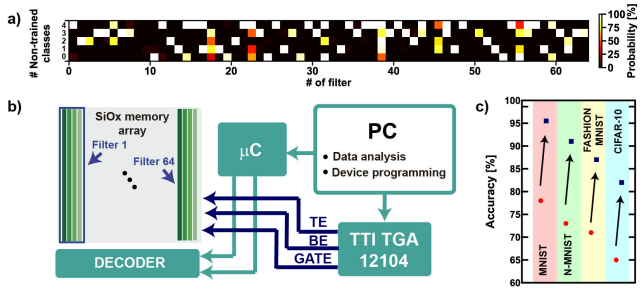

FIGURE 10. (a) Average probability of the responses for the nontrained classes as a function of the filters. Note that some filters (e.g., 23, 45, and 55) present a blurred response, which leads to useless information with respect to the classification of the new classes. Other filters, instead, have a too high probability of response (e.g., 38); hence, no classification improvement is obtained. (b) Such a limited number of filters can be properly reprogrammed to enhance the differentiation between the nontrained classes. (c) Increase in terms of overall accuracy after retraining.

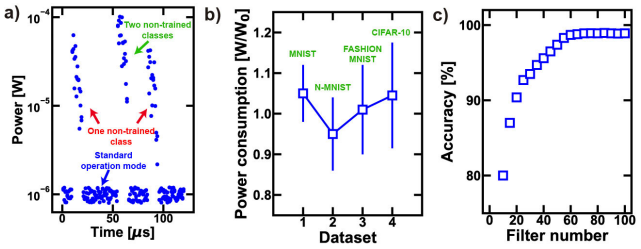

FIGURE 11. (a) Power consumption depends on the number of new classes to classify for continual learning. (b) Ratio of the power consumption of the conditions after and before the filter retraining. (c) Global accuracy as a function of the number of filters for the MNIST dataset.

retraining of the filters cannot be as efficient as the training of the entire dataset.

These circuital results suggest that the ability to forget and retrain is fundamental for the adaptation to changing environments, as already evidenced by other works under a pure theoretical framework [37]. However, note that a better forgetting property would be obtainable by using natural drifting devices as homeostatic elements, such as PCMs [38].

C. EFFICIENCY OF RETRAINING AND IMPROVEMENT OF THE ACCURACY

Every time a new set of patterns appears at the input, the further firing neurons require more energy to get specialization. However, the consumption levels always tend to come back to the steady-state condition [see Fig. 11(a)]. The procedure to achieve optimum continual learning in neural networks is straightforward since it requires erasing some filters and reprogram the corresponding devices [see Fig. 10(b)]. Thus, as evident in Fig. 11(b) by a comparison of the inference activity before and after the retraining, the average power consumption remains the same since the network has been only provided with different combinations of weights. Note also that the energetic efficiency is in tradeoff with respect to

the accuracy: the higher the number of filters (i.e., the energy and the area consumption), the higher the accuracy since more original feature maps per class are obtained. However, this trend saturates, as suggested by the simulations of Fig. 10(c) for the full testing of the MNIST dataset. The system gets marginal improvements after a certain number of filters (around 80), getting a final best ideal accuracy 1% lower with respect to the best achievable with standard neural networks [39]. This is the worst downside of the type of network that we propose, which can be directly referred to as the unsupervised classification accuracy.

D. IMPACT OF THE DEVICE NON-IDEALITIES

The failure of the programming of the devices strongly affects the behavior of the whole network, as we analyze in the following for the MNIST dataset. Fig. 12(a) shows the overall accuracy as a function of the compliance current I_C chosen to program the devices. Note that, as I_C lowers too much, the corresponding resistive window decreases, and the digitalized filter accuracy decreases accordingly. This is due to the fact the synaptic devices are binary weighted, and thus, greater variability in the resistive error causes a consequent worse inference.

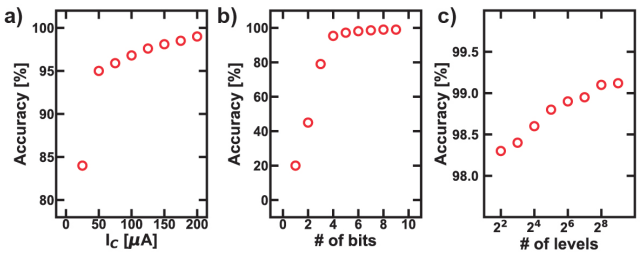

FIGURE 12. Accuracy for the MNIST dataset as a function of the compliance current used to program (a) convolutional weights, (b) number of bits used to map each weight, and (c) number of intermediate steps to implement the homeostatic increase in the neuronal threshold.

Increasing the number of bits used to digitalize the filters, we directly obtain a better accuracy of the network [see Fig. 12(b)]. On the other hand, the multilevel programming of the devices referred to the unsupervised section of the network does not require a precise resistance value, as typically observed in the neural computation [15], [21]. Note that, nevertheless, the device must always show a switching activity that can enable the homeostatic mechanism of spike frequency adaptation. However, such switching activity can be either ideally multilevel or binarized since the accuracy depends only slightly on the modulation of the internal thresholds [see Fig. 12(c)].

V. CONCLUSION

We presented a novel SiO_x RRAM-based network that combines the efficiency of supervised architectures and the adaptation of unsupervised learning to enable lifelong learning in artificial intelligent systems. The network provides high accuracies for full testing (almost 99% for MNIST and 89% for CIFAR10) and learning of up to 50% nontrained classes using spike-frequency modulation. We validated the results using ideal simulations, Monte Carlo analysis, and experimental measurements with SiO_x RRAM devices. We demonstrated optimized continual learning and optimum cohesion of stability and plasticity in the same ANN. The results highlight robustness against device non-idealities and suggest the possibility to create fully autonomous systems able to continually learn throughout their lives.

VI. ACKNOWLEDGMENT

Erika Covi was with the Dipartimento di Elettronica, Informazione e Bioingegneria and the Italian Universities Nanoelectronics Team, Politecnico di Milano, 20133 Milan, Italy. *(Stefano Bianchi and Irene Muñoz-Martín contributed equally to this work.)*

REFERENCES

[1] D. Kumaran, D. Hassabis, and J. L. McClelland, "What learning systems do intelligent agents need? Complementary learning systems theory updated," *Trends Cogn. Sci.*, vol. 20, no. 7, pp. 512–534, 2016, doi: 10.1016/j.tics.2016.05.004.

[2] J. L. McClelland, B. L. McNaughton, and R. C. O'Reilly, "Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory," *Psychol. Rev.*, vol. 102, no. 3, pp. 419–457, 1995, doi: 10.1037/0033-295X.102.3.419.

[3] R. C. O'Reilly, R. Bhattacharyya, M. D. Howard, and N. Ketz, "Complementary learning systems," *Cogn. Sci.*, vol. 38, no. 6, pp. 1229–1248, Dec. 2011, doi: 10.1111/j.1551-6709.2011.01214.x.

[4] C. Folke, S. R. Carpenter, B. Walker, M. Scheffer, T. Chapin, and J. Rockström, "Resilience thinking: Integrating resilience, adaptability and transformability," *Ecol. Soc.*, vol. 15, no. 4, p. 20, 2010, doi: 10.5751/es-03610-150420.

[5] J. Hirschberg and C. D. Manning, "Advances in natural language processing," *Science*, vol. 349, no. 6245, pp. 261–266, Jul. 2015, doi: 10.1126/science.aaa8685.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[7] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015, doi: 10.1038/nature14539.

[8] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Netw.*, vol. 113, pp. 54–71, May 2019, doi: 10.1016/j.neunet.2019.01.012.

[9] M. Mermillod, A. Bugaiska, and P. Bonin, "The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects," *Frontiers Psychol.*, vol. 4, p. 504, Aug. 2013, doi: 10.3389/fpsyg.2013.00504.

[10] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," *Psychol. Learn. Motivat.*, vol. 24, pp. 109–165, Dec. 1989, doi: 10.1016/S0079-7421(08)60536-8.

[11] K. James *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proc. Nat. Acad. Sci. USA*, vol. 114, no. 13, pp. 3521–3526, Mar. 2017, doi: 10.1073/pnas.1611835114.

[12] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "ICaRL: Incremental classifier and representation learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5533–5542, doi: 10.1109/CVPR.2017.587.

[13] A. A. Rusu *et al.*, "Progressive neural networks," May 2016, *arXiv:1606.04671*. [Online]. Available: http://arxiv.org/abs/1606.04671

[14] S. Bianchi, I. Muñoz-Martin, and D. Ielmini, "Bio-inspired techniques in a fully digital approach for lifelong learning," *Frontiers Neurosci.*, vol. 14, pp. 379–393, Apr. 2020, doi: 10.3389/fnins.2020.00379.

[15] G. Q. Bi and M. M. Poo, "Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type," *J. Neurosci.*, vol. 18, no. 24, pp. 10464–10472, 1998, doi: 10.1523/JNEUROSCI.18-24-10464.1998.

[16] T. Moraitis, A. Sebastian, and E. Eleftheriou, "Optimality of short-term synaptic plasticity in modelling certain dynamic environments," 2020, *arXiv:2009.06808*. [Online]. Available: https://arxiv.org/abs/2009.06808

[17] E. Chicca, F. Stefanini, C. Bartolozzi, and G. Indiveri, "Neuromorphic electronic circuits for building autonomous cognitive systems," *Proc. IEEE*, vol. 102, no. 9, pp. 1367–1388, Sep. 2014, doi: 10.1109/JPROC.2014.2313954.

[18] I. Muñoz-Martín, S. Bianchi, G. Pedretti, O. Melnic, S. Ambrogio, and D. Ielmini, "Unsupervised learning to overcome catastrophic forgetting in neural networks," *IEEE J. Explor. Solid-State Comput. Devices Circuits*, vol. 5, no. 1, pp. 58–66, Jun. 2019, doi: 10.1109/JXCDC.2019.2911135.

[19] S. Ambrogio *et al.*, "Unsupervised learning by spike timing dependent plasticity in phase change memory (PCM) synapses," *Frontiers Neurosci.*, vol. 10, p. 56, Mar. 2016, doi: 10.3389/fnins.2016.00056.

[20] M. Prezioso *et al.*, "Spike-timing-dependent plasticity learning of coincidence detection with passively integrated memristive circuits," *Nature Commun.*, vol. 9, no. 1, pp. 1–8, Dec. 2018, doi: 10.1038/s41467-018-07757-y.

[21] S. Bianchi *et al.*, "A compact model for stochastic spike-timing-dependent plasticity (STDP) based on resistive switching memory (RRAM) synapses," *IEEE Trans. Electron Devices*, vol. 67, no. 7, pp. 2800–2806, Jul. 2020, doi: 10.1109/TED.2020.2992386.

[22] I. Muñoz-Martín, S. Bianchi, S. Hashemkhani, G. Pedretti, and D. Ielmini, "Hardware implementation of PCM-based neurons with self-regulating threshold for homeostatic scaling in unsupervised learning," in *Proc. ISCAS*, Oct. 2020, pp. 1–5, doi: 10.1109/ISCAS45731.2020.9181033.

[23] S. B. Eryilmaz *et al.*, "Brain-like associative learning using a nanoscale non-volatile phase change synaptic device array," *Frontiers Neurosci.*, vol. 8, p. 205, Jul. 2014, doi: 10.3389/fnins.2014.00205.

[24] G. W. Burr *et al.*, "Neuromorphic computing using non-volatile memory," *Adv. Phys. X*, vol. 2, no. 1, pp. 89–124, Dec. 2016, doi: 10.1080/23746149.2016.1259585.

[25] G. G. Turrigiano, "The self-tuning neuron: Synaptic scaling of excitatory synapses," *Cell*, vol. 135, no. 3, pp. 422–435, Oct. 2008, doi: 10.1016/j.cell.2008.10.008.

[26] C. Bartolozzi and G. Indiveri, "Global scaling of synaptic efficacy: Homeostasis in silicon synapses," *Neurocomputing*, vol. 72, nos. 4–6, pp. 726–731, Jan. 2009, doi: 10.1016/j.neucom.2008.05.016.

[27] G. G. Turrigiano, "Homeostatic plasticity in neuronal networks: The more things change, the more they stay the same," *Trends Neurosci.*, vol. 5, pp. 221–227, May 1999, doi: 10.1016/s0166-2236(98)01341-1.

[28] W. Goes *et al.*, "A comprehensive oxide-based ReRAM TCAD model with experimental verification," in *Proc. IEEE Int. Memory Workshop (IMW)*, May 2021, pp. 1–4.

[29] S. Bianchi, I. Muñoz-Martin, G. Pedretti, O. Melnic, S. Ambrogio, and D. Ielmini, "Energy-efficient continual learning in hybrid supervised-unsupervised neural networks with PCM synapses," in *Proc. Symp. VLSI Technol.*, Jun. 2019, pp. T172–T173.

[30] S. Ambrogio *et al.*, "Equivalent-accuracy accelerated neural-network training using analogue memory," *Nature*, vol. 558, no. 7708, pp. 60–67, Jun. 2018, doi: 10.1038/s41586-018-0180-5.

[31] I. Muñoz-Martin *et al.*, "A SiO_x RRAM-based hardware with spike frequency adaptation for power-saving continual learning in convolutional neural networks," in *Proc. IEEE Symp. VLSI Technol.*, Jun. 2020, pp. 1–2, doi: 10.1109/VLSITechnology18217.2020.9265072.

[32] S. Bianchi, I. Muñoz-Martín, S. Hashemkhani, G. Pedretti, and D. Ielmini, "A bio-inspired recurrent neural network with self-adaptive neurons and PCM synapses for solving reinforcement learning tasks," in *Proc. ISCAS*, Oct. 2020, pp. 1–5.

[33] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," 2017, *arXiv:1708.07747*. [Online]. Available: http://arxiv.org/abs/1708.07747

[34] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[35] M. Palatucci, G. E. Hinton, D. Pomerleau, and T. M. Mitchell, "Zero-shot learning with semantic output codes," in *Proc. Int. Conf. Neural Inf. Process. Syst. (NIPS)*, vol. 22, Dec. 2009, pp. 1410–1418.

[36] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," in *Proc. 30th Adv. Neural Inf. Process. Syst. (NIPS)*, Jun. 2016, pp. 3630–3638.

[37] P. Panda, J. M. Allred, S. Ramanathan, and K. Roy, "ASP: Learning to forget with adaptive synaptic plasticity in spiking neural networks," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 8, no. 1, pp. 51–64, Mar. 2018.

[38] I. Muñoz-Martin, S. Bianchi, S. Hashemkhani, G. Pedretti, O. Melnic, and D. Ielmini, "A brain-inspired homeostatic neuron based on phase-change memories for efficient neuromorphic computing," *Frontiers Neurosci.*, vol. 15, p. 1054, Aug. 2021.

[39] L. Wan, M. Zeiler, S. Zhang, Y. L. Cun, and R. Fergus, "Regularization of neural networks using dropconnect," in *Proc. 30th Int. Conf. Mach. Learn. (ICML)*, vol. 28, May 2013, pp. 1058–1066, doi: 10.5555/3042817.3043055.