Brain-inspired self-organization with cellular neuromorphic computing for multimodal unsupervised learning

Lyes Khacef, Laurent Rodriguez, Benoît Miramond

Université Côte d’Azur, CNRS, LEAT, France

firstname.lastname@univ-cotedazur.fr

Abstract—Cortical plasticity is one of the main features that enable our capability to learn and adapt in our environment. Indeed, the cerebral cortex has the ability to self-organize itself through two distinct forms of plasticity: the structural plasticity that creates (sprouting) or cuts (pruning) synaptic connections between neurons, and the synaptic plasticity that modifies the synaptic connections strength. These mechanisms are very likely at the basis of an extremely interesting characteristic of the human brain development: the multimodal association. The brain uses spatio-temporal correlations between several modalities to structure the data and create sense from observations. Thus, in spite of the diversity of the sensory modalities, like sight, sound and touch, the brain arrives at the same concepts. Moreover, biological observations show that one modality can activate the internal representation of another modality when both are correlated. To model such a behavior, Edelman and Damasio proposed respectively the Reentry and the Convergence Divergence Zone frameworks where bi-directional neural communications can lead to both multimodal fusion (convergence) and inter-modal activation (divergence). Nevertheless, these frameworks do not provide a computational model at the neuron level, and only few papers tackle this issue of bio-inspired multimodal association which is yet necessary for a complete representation of the environment. In this paper, we build a brain-inspired neural system based on the Reentry principles, using Self-Organizing Maps and Hebbian-like learning. We propose and compare different computational methods for unsupervised learning and inference, then quantify the gain of both convergence and divergence mechanisms in a multimodal classification task. The divergence mechanism is used to label one modality based on the other, while the convergence mechanism is used to improve the overall accuracy of the system. We perform our experiments on a constructed written/spoken digits database and a DVS/EMG hand gestures database. Finally, we implement our system on the Iterative Grid, a cellular neuromorphic architecture that enables distributed computing with local connectivity. We show the gain of the so-called hardware plasticity induced by our model, where the system’s topology is not fixed by the user but learned along the system’s experience through self-organization.

Index Terms—brain-inspired computing, convergence divergence zone, self-organizing maps, hebbian learning, multimodal classification, cellular neuromorphic architectures.

I. INTRODUCTION

Intelligence is often defined as the ability to adapt to the environment through learning. “A person possesses intelligence insofar as he has learned, or can learn, to adjust himself to his environment”, S. S. Colvin quoted in [1]. The same definition could be applied to machines and artificial systems in general. Hence, a stronger relationship with the environment is a key challenge for future Artificial Intelligence (AI) systems that interact in the real-world environment for diverse applications like object detection and recognition, tracking, navigation, etc. The system becomes an “agent” in which the so-called intelligence would emerge from the interaction it has with the environment, as defined in the embodiment hypothesis that is widely adopted in both developmental psychology [2] and developmental robotics [3]. In this work, we tackle the first of the six fundamental principles for the development of embodied intelligence as defined in [2]: the multimodality.

Indeed, biological systems perceive their environment through diverse sensory channels: vision, audition, touch, smell, proprioception, etc. The fundamental reason lies in the concept of degeneracy in neural structures [4], which is generally defined as the ability of biological elements that are structurally different to perform the same function or yield the same output [5]. In other words, it means that any single function can be carried out by more than one configuration of neural signals, so that the system still functions with the loss of one component. It also means that sensory systems can educate each other, without an external teacher [2]. The same principles can be applied for artificial systems, as information about the same phenomenon in the environment can be acquired from various types of sensors: cameras, microphones, accelerometers, etc. Each sensory-information can be considered as a modality. Due to the rich characteristics of natural phenomena, it is rare that a single modality provides a complete representation of the phenomenon of interest [6].

Multimodal data fusion is thus a direct consequence of the well-accepted paradigm that certain natural processes and phenomena are expressed under completely different physical guises [6]. Recent works show a growing interest toward multimodal association in several applicative areas such as developmental robotics [3], audio-visual signal processing [7], spacial perception [9], attention-driven selection [11] and tracking [12], memory encoding [13], emotion recognition [14], human-machine interaction [15], remote sensing and earth observation [16], medical diagnosis [17], understanding brain functionality [18], etc. Interestingly, the last mentioned application is our starting bloc: how does the brain handle...
multimodal learning in the natural environment? In fact, it is most likely the emergent result of one of the most impressive abilities of the embodied brain: the cortical plasticity which is enabled by self-organization.

In this work, we propose and explore several brain-inspired computational models of self-organization for multimodal unsupervised learning in neuromorphic systems. Section II describes the Reentry framework of Edelman [19] and the Convergence Divergence Zone framework of Damasio [20], two different theories in neuroscience for modeling multimodal association in the brain, and then review some of their recent computational models and applications. Section III details the proposed multimodal learning and inference algorithms, while section IV presents an extension of the Iterative Grid (IG) [21] which is applied to distribute the system’s computation in a cellular neuromorphic architecture for FPGA implementations. Then, section V presents the databases, experiments and results with the different case studies. Finally, section VI discusses the results and quantifies the gain of the so-called hardware plasticity through self-organization.

II. MULTIMODAL LEARNING: STATE OF ART

A. Brain-inspired approaches: Reentry and Convergence Divergence Zone (CDZ)

![Diagram of reentry and CDZ frameworks](image)

Fig. 1. Schematic representation of (a) reentry and (b) CDZ frameworks. The reentry paradigm states that unimodal neurons connect to each other through direct connections, while the CDZ paradigm implies hierarchical neurons that connect unimodal neurons.

Brain’s plasticity, also known as neuroplasticity, is the key to humans capability to learn and modify their behaviour. The plastic changes happen in neural pathways as a result of the multi-modal sensori-motor interaction in the environment [22]. But since most of the stimuli are processed by the brain in more than one sensory modality [23], how do the multimodal information converge in the brain? Indeed, we can recognize a dog by seeing its picture, hearing its bark or rubbing its fur. These features are different patterns of energy at our sensory organs (eyes, ears and skin) that are represented in specialized regions of the brain. However, we arrive at the same concept of the “dog” regardless of which sensory modality was used [24]. Furthermore, modalities can diverge and activate one another when they are correlated. Recent studies have demonstrated cross-modal activation amongst various sensory modalities, like reading words with auditory and olfactory meanings that evokes activity in auditory and olfactory cortices [25] [26], or trying to discriminate the orientation of a tactile grid pattern with eyes closed that induces activity in the visual cortex [27]. Both mechanisms rely on the cerebral cortex as a substrate. But even though recent works have tried to study the human brains ability to integrate inputs from multiple modalities [28] [29], it is not clear how the different cortical areas connect and communicate with each other.

To answer this question, Edelman proposed in 1982 the Reentry [19] [30]: the ongoing bidirectional exchange of signals linking two or more brain areas, one of the most important integrative mechanisms in vertebrate brains [19]. In a recent review [31], Edelman defines reentry as a process which involves a localized population of excitatory neurons that simultaneously stimulates and is stimulated by another population, as shown in Figure 1. It has been shown that reentrant neuronal circuits self-organize early during the embryonic development of vertebrate brains [32] [33], and can give rise to patterns of activity with Winner-Take-All properties [34] [35]. When combined with appropriate mechanisms for synaptic plasticity, the mutual exchange of signals amongst neural networks in distributed cortical areas results in the spatio-temporal integration of patterns of neural network activity. It allows the brain to categorize sensory inputs, remember and manipulate mental constructs, and generate motor commands [31]. Thus, reentry would be the key to multimodal integration in the brain.

Damasio proposed another answer in 1989 with the Convergence Divergence Zone (CDZ) [20] [36], another biologically plausible framework for multimodal association. In a nutshell, the CDZ theory states that particular cortical areas act as sets of pointers to other areas, with a hierarchical construction: the CDZ merges low level cortical areas with high level amodal constructs, which connects multiple cortical networks to each other and therefore solves the problem of multimodal integration. The CDZ convergence process integrates unimodal information into multimodal areas, while the CDZ divergence process propagates the multimodal information to the unimodal areas, as shown in Figure 1. For example, when someone talks to us in person, we simultaneously hear the speakers voice and see the speakers lips move. As the visual movement and the sound co-occur, the CDZ would associate (convergence) the respective neural representations of the two events in early visual and auditory cortices into a higher cortical map. Then, when we only watch a specific lip movement without any sound, the activity pattern induced in the early visual cortices would trigger the CDZ and the CDZ would retro-activate (divergence) in early auditory cortices the representation of the sound that usually accompanied the lip movement [23].
The bidirectionality of the connections is therefore a fundamental characteristic of both reentry and CDZ frameworks, that are likewise in many aspects. Indeed, we find computational models of both paradigms in the literature. We review the most significant ones to our work in Section II-B.

B. Models and applications

In this section, we review the recent works that explore brain-inspired multimodal learning for two main applications: sensori-motor mapping for embodied cognition and multisensory mapping for classification.

1) Sensori-motor mapping: In 2013, Lallee and Dominey [37] proposed the Multi-Modal Convergence Map (MMCM) that applies the Self-Organizing Map (SOM) [43] to model the CDZ framework. A hierarchy of SOMs is used to reduce the dimensionality of the input, using the coordinates of the most active unit of each unimodal map as input of the multimodal map. The MMCM was applied to encode the sensori-motor experience of a robot based on the language, vision and motor modalities. This “knowledge” was used in return to control the robot behaviour. The experiments were conducted in both a simulated and a real humanoid robot, the iCub [44]. In a nutshell, the MMCM provides an implemented framework in which multiple modalities are represented in distinct and converging maps. Activation in one modality can be used to generate a mental image in the other modalities. Lallee and Dominey demonstrated how this can be used to increase the performance of the iCub in the recognition of its hand in different postures.

In 2015, Droniou et al. [3] proposed an architecture based on Deep Neural Networks (DNNs), which is used by the iCub [44] to learn a task from multiple perceptual modalities: proprioception, vision and audition. The DNN is based on the auto-encoder paradigm for both reducing the dimensionality of data as in a standard auto-encoding approach and for clustering, adding a Softmax activation function [45] to make the compressed representation sparser and cluster the data. Globally, the system of Droniou et al. relates to the CDZ framework even if the actual purpose was not to provide a computational model for the theory. First, for a bi-modal task and given one modality alone, the network was able to infer a classification and a parametrization which can be used to reconstruct the missing modality. Second, the proposed network was able to exploit multimodal correlations to improve the representation of each modality alone.

In 2016, Escobar-Juarez et al. [22] proposed the Self-Organized Internal Models Architecture (SOIMA) that models the CDZ framework based on internal models, where sensory and motor information merge in a natural way and create a multi-modal representation [46]. The work focused on the pair formed by inverse-forward models. The inverse model is a controller that generates the motor command (Mt) needed to achieve a desired sensory state (St+1) given a current sensory state (St), while the forward model is a predictor that predicts the sensory state entailed (St+1) by some action of the agent (Mt) given a current sensory state (St). The necessary property of bidirectionality is once again pointed out by the authors. SOIMA relies on two main learning mechanisms: the first one consists in SOMs that create clusters of unimodal information coming from the environment. The second one codes the internal models by means of connections between the first maps using Hebbian learning that generates sensory-motor patterns. As in [37], a hierarchy of SOMs is used such that the inputs to the top multimodal map are the coordinates of the winning neurons in the unimodal maps. The SOIMA architecture was successfully experimented on a saccadic control and hand-eye coordination tasks.

In 2019, Zahra et al. proposed the Varying Density Self-Organizing Map (VDSOM) for characterizing sensorimotor relations in robotic systems with bidirectional connections. The proposed method relies on self-organizing properties through SOMs and associative properties through Oja’s learning [47] that enables it to autonomously obtain sensori-motor relations without any prior knowledge of either the motor (e.g. mechanical structure) or perceptual (e.g. sensor calibration) models. This solution relies on collecting data samples by motor babbling and is therefore suitable for various robotic manipulators without prior information about robot kinematics. Even though the paper [38] does not state so, the VDSOM is closer to the reentry paradigm, where direct bidirectional connections are learned amongst neurons.

2) Multi-sensory classification: In 2017, Parisi et al. [39] proposed a hierarchical architecture with Growing When Required (GWR) networks [48] for learning human actions from audiovisual inputs. The neural architecture consists of a self-organizing hierarchy with four layers of GWR for the unsupervised processing of visual action features. The fourth layer of the network implements a semi-supervised algorithm where actionword mappings are developed. This is done by binding co-occurring audiovisual inputs using bidirectional inter-layer connectivity, and thus learning multimodal representations of actions. The direct bidirectional connections follow the reentry paradigm.

In 2018, Jayaratne et al. [40] proposed a multisensory neural architecture that consists of multiple self-organizing neural layers of Growing SOMs (GSOM) [49] for modelling the respective cortical areas of each sensory modality, and inter-sensory associative connections representing the co-occurrence probabilities of the modalities. Here again, there is no hierarchy in the bidirectional connections, thus referring to the reentry paradigm. The system was implemented in Apache Spark [50] to distribute the GSOM computing with respect to data, i.e. distribute data across a cluster of computers to process them in parallel, and thus improving its scalability to big datasets. The system’s principle is to supplement the information on a single modality with the corresponding information on other modalities with the Tulips1 audio-visual dataset [51], exploiting the co-occurrence relationship across the modalities for a better classification accuracy.

In 2018, Rathi and Roy [41] proposed an STDP-based multimodal unsupervised learning for Spiking Neural Networks (SNNs). The goal of this work is to learn the cross-
modal connections between areas of single modality in Spiking Neural Networks (SNNs) to improve the recognition accuracy and make the system robust to noisy inputs. Each modality is represented by a specific SNN trained with its own data following the learning framework proposed in [52]. The SNN computation is distributed, but requires an all-to-all connectivity amongst neurons. This full connectivity goes against the scalability of the network as discussed in section IV. In addition, the cross-modal connections between the two SNNs are trained along with the unimodal connections. The cross-modal connections are sparsely connected following the reentry paradigm and initialized with random weights, and STDP is used to update these weights as both SNNs are presented with two inputs of the same class at same time. The correlation between neurons of different modalities is captured in the cross-modal connections, which assist the network in making the right decision by increasing the spikes for the correct class. The proposed method was experimented with a written/spoken digit classification task, and the collaborative learning results in an accuracy improvement of 2% compared to the best unimodal accuracy. Furthermore, the multimodal approach makes the network noise tolerant. The work of Rath and Roy [41] is the first to train SNNs with multimodal inputs, and is the closest to our work. Hence, a detailed comparison is presented in section V-D1.

In 2019, Cholet et al. [42] proposed a modular architecture for multimodal fusion using Bidirectional Associative Memories (BAMs), which were initially proposed by Kosko [53] as an adaptation of the Hopfield network [54] for hetero-association. The BAMs is composed of two fully and bidirectionally connected layers. The proposed architecture can be summarised in three stages: unimodal data are first processed by as many independent prototype-based Incremental Neural Networks (INNs) [55] as the number of modalities to be combined. The second stage consists of multiple BAMs that achieve the fusion of modalities by learning pairs of unimodal prototypes towards the integrative layer which builds an abstract representation. Finally, the third stage is a INN that performs a supervised classification. Even though we can see a form of hierarchy in the third stage with the INN that takes the BAMs as input for classification, the multimodal association itself is made with direct BAMs between unimodal representations, thus following the reentry paradigm.

3) Summary: Overall, the reentry and CDZ frameworks share two key aspects: the multimodal associative learning based on the temporal co-occurrence of the modalities, and the bidirectionality of the associative connections. We summarize the most relevant papers to our work in Table I, where we classify each paper with respect to the application, the brain-inspired paradigm, the learning type and the computing nature. We notice that sensori-mapping is based on unsupervised learning, which is natural as no label is necessary to map two modalities together. However, classification is based on either supervised or semi-supervised learning, as mapping multisensory modalities is not sufficient: we need to know the corresponding class to each activation pattern. We propose a labeling method in Section III-A2 based on a small labeled subset, so that we do not use any label in the learning process. The same approach is used in [41], but the authors rely on the complete labeled dataset, as further discussed in section V-D1. Finally, all previous works rely on the centralized Von Neumann computing paradigm, except [40] that attempts a partially distributed computing with respect to data, i.e. using the MapReduce computing paradigm to speed up computations. It is based on Apache Spark [56], mainly used for cloud computing. Also, STDP learning in [41] is distributed, but the inference for classification requires a central unit, as discussed in section V-D1. We propose a completely distributed computing on the edge with respect to the system, i.e. the neurons computing itself to improve the SOMs scalability for hardware implementation, as presented in Section IV.

Consequently, we chose to follow the reentry paradigm where multimodal processing is distributed in all cortical maps without dedicated associative maps for two reasons. First, from the brain-inspired computing perspective, more biological evidences tend to confirm the hypothesis of reentry as reviewed by [57], [58] and [59]. Indeed, biological observations high-

| Application | Work | Paradigm | Learning | Computing |
|-------------|------|----------|----------|-----------|
| Sensori-motor mapping | Lallee et al. [37] (2013) | CDZ | Unsupervised | Centralized |
| | Dromiou et al. [31] (2015) | CDZ | Unsupervised | Centralized |
| | Escobar-Juarez et al. [22] (2016) | CDZ | Unsupervised | Centralized |
| | Zahra et al. [38] (2019) | Reentry | Unsupervised | Centralized |
| Multi-sensory classification | Parisi et al. [39] (2017) | Reentry | Semi-supervised | Centralized |
| | Jayaratne et al. [40] (2018) | Reentry | Semi-supervised | Distributed (data level) |
| | Rath et al. [41] (2018) | Reentry | Unsupervised | Centralized * ** |
| | Cholet et al. [42] (2019) | Reentry * | Supervised | Centralized * |
| | Khacef et al. [this work] (2020) | Reentry | Unsupervised | Distributed (system level) |

* with an extra layer for classification. ** learning is distributed but inference for classification is centralized.
light a multimodal processing in the whole cortex including sensory areas [60] which contain multimodal neurons that are activated by multimodal stimuli [57] [61]. Moreover, it has been shown that there are direct connections between sensory cortices [62] [63], and neural activities in one sensory area may be influenced by stimuli from other modalities [58] [64]. Second, from a pragmatical and functional perspective, the reentry paradigm fits better to the cellular architecture detailed in Section IV and thus increases the scalability and fault tolerance thanks to the completely distributed processing [59]. Nevertheless, we keep the convergence and divergence terminology to distinguish between, respectively, the integration of two modalities and the activation of one modality based on the other.

III. PROPOSED COMPUTATIONAL MODEL

![Diagram of the proposed reentry model for multimodal association.](image)

The initial convergence zone model was proposed by Moll and Miikkulainen in 1997 [65], but it lacked the self-organizing and topographical property inherent to cortical maps [57]. Hence, we use SOMs and Hebbian-like learning in two times to perform multimodal learning: first, unimodal representations are obtained with SOMs and, second, multimodal representations develop through the association of unimodal maps via bidirectional synapses that can be seen as BAMs. The development of associations between co-occurring stimuli for multimodal binding has been strongly supported by neurophysiological evidence [66], and follow the reentry paradigm [51].

In this section, we summarise our previous work on SOM learning and labeling [67], then propose a new model for learning multimodal associations, labeling one modality based on the other and converge the two modalities with cooperation and competition for a better classification accuracy.

A. Unimodal learning: Self-Organizing Maps (SOMs)

With the increasing amount of unlabeled data gathered everyday through Internet of Things (IoT) devices and the difficult task of labeling each sample, DNNs are slowly reaching the limits of supervised learning [3] [68]. Hence, unsupervised learning is becoming one of the most important and challenging topics in Machine Learning (ML) and AI. The Self-Organizing Map (SOM) proposed by Kohonen [43] is one of the most popular Artificial Neural Networks (ANNs) in the unsupervised learning category [69], inspired from the cortical synaptic plasticity and used in a large range of applications [70] going from high-dimensional data analysis to more recent developments such as identification of social media trends [71], incremental change detection [72] and energy consumption minimization on sensor networks [73]. Since many variants of the SOM like the Dynamic SOM (DSOM) [74] and the Pruning Cellular SOM (PCSOM) [75] have been proposed, we name the original algorithm as the Kohonen SOM (KSOM).

1) SOM learning: The original KSOM algorithm introduced by Kohonen [43] is described in Algorithm 1. It is to note that \( t_f \) is the number of epochs, i.e. the number of times the whole training dataset is presented. The \( \alpha \) hyper-parameter value in Equation 1 is not important for the SOM training, since it does not change the neuron with the maximum activity. It can be set to 1 in Algorithm 1. All unimodal trainings were performed over 10 epochs with the same hyper-parameters as in our previous work [67]: \( \epsilon_i = 1.0, \epsilon_f = 0.01, \sigma_i = 5.0 \) and \( \sigma_f = 0.01 \).

2) SOM labeling: The labeling is the step between training and test where we assign each neuron the class it represents in the training dataset, a necessary step for any classification task based on unsupervised learning. We proposed in [67] a labeling algorithm based on few labeled samples. We randomly took a labeled subset of the training dataset, and we tried to minimize its size while keeping the best classification accuracy. Our study showed that we only need 1% of randomly taken labeled samples from the training dataset for the MNIST [76] classification.

The labeling algorithm can be summarized in four steps: first, we calculate the neurons activations to the input samples from the euclidean distance following Equation 1 where \( v \) is the input vector, \( w_n \) and \( a_n \) are respectively the weights vector and the activity of the neuron \( n \). The parameter \( \alpha \) is the width of the Gaussian kernel that becomes a hyper-parameter for the method, as further discussed in Section V. Second,
the Best Matching Unit (BMU), i.e. the neuron with the maximum activity is elected. Third, each neuron accumulates its normalized activation (simple division) with respect to the BMU activity in the corresponding class accumulator, and the three steps are repeated for every sample of the labeling subset. Fourth and finally, the label of each neuron is chosen according to the class accumulator that has the maximum activity.

B. Multimodal association: sprouting, Hebbian-like learning and pruning

Brain’s plasticity can be divided into two distinct forms of plasticity: the (1) structural plasticity that changes the neurons connectivity by sprouting (creating) or pruning (deleting) synaptic connections, and (2) the synaptic plasticity that modifies (increasing or decreasing) the existing synapses strength. We explore both mechanisms for multimodal association through Hebbian-like learning.

The original Hebbian learning principle proposed by Hebb in 1949 states that “when an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that As efficiency, as one of the cells firing B, is increased.” In other words, any two neurons that are repeatedly active at the same time will tend to become “associated” so that activity in one facilitates activity (1) in the other. The learning rule is expressed by Equation 7.

However, Hebb’s rule is limited in terms of stability for online learning, as synaptic weights tend to infinity with a positive learning rate. This could be resolved by normalizing each weight over the sum of all the corresponding neuron weights, which guarantees the sum of each neuron weights to be equal to 1. The effects of weights normalization are explained in [79]. However, this solution breaks up with the locality of the synaptic learning rule, and that is not biologically plausible. In 1982, Oja proposed a Hebbian-like rule that adds a “forgetting” parameter, and solves the stability problem with a form of local multiplicative normalization for the neurons weights, as expressed in Equation 8. In addition, Oja’s learning performs an on-line principal component analysis of the data in the neural network, which is a very interesting property in the context of unsupervised learning.

Nevertheless, Hebb’s and Oja’s rules were both used in recent works with good results, respectively in [22] and [38]. Hence, we applied and compared both rules. The proposed multimodal association model is detailed in Algorithm 2, where η is a learning rate that we fix to 1 in our experiments, and γ is deduced according to the number or the percentage of synapses to prune, as discussed in Section V. The neurons activities computing in the line 3 of Algorithm 2 are calculated in the same way as Equation 1.

C. Divergence for labeling

As explained in III-A2, neurons labeling is based on a labeled subset from the training database. We tried in [67] to minimize its size, and used the fewest labeled samples while keeping the best accuracy. We will see in Section V that depending on the database, we sometimes need a considerable number of labeled samples, up to 10% of the training set. In this work, we propose an original method based on the divergence mechanism of the multimodal association: for two modalities x and y, since we can activate one modality based on the other, we propose to label the SOMx neurons from the activity and the labels induced from the SOMx neurons, which are based on the labeling subset of modality x. Therefore, we only need one labeled subset of a single modality to label both modalities, taking profit of the bidirectional aspect of reentry. A good analogy to biological observations would be the retro-activation of the auditory cortical areas from the visual cortex, if we take the example of written/spoken digits presented in Section V. It is similar to how infants respond to sound symbolism by associating shapes with sounds [80].

Algorithm 1: Kohonen SOM algorithm

1: Initialize the network as a two-dimensional array of k neurons, where each neuron n with m inputs is defined by a two-dimensional position p_n and a randomly initialized m-dimensional weight vector w_n.
2: for t from 0 to T_f do
3: for every input vector v do
4: for every neuron n in the network do
5: Compute the afferent activity a_n:
6: end for
7: Compute the winner s such that:
8: end for
9: for every neuron n in the network do
10: Compute the neighborhood function h(σ(t, n, s)):
11: end for
12: Update the learning rate ϵ(t): ϵ(t) = ϵ_i × (e_f / ϵ_i)^(t/T_f)
13: Update the width of the neighboorhood σ(t):
14: end for
15: end for

Algorithm 2: Multimodal association

1: Initialize the SOM network as a two-dimensional array of k neurons, where each neuron n with m inputs is defined by a two-dimensional position p_n and a randomly initialized m-dimensional weight vector w_n.
2: for t from 0 to T_f do
3: for every input vector v do
4: for every neuron n in the network do
5: Compute the neighborhood function h(σ(t, n, s)):
6: end for
7: Compute the winner s such that:
8: end for
9: for every neuron n in the network do
10: Compute the neighborhood function h(σ(t, n, s)):
11: end for
12: Update the learning rate ϵ(t): ϵ(t) = ϵ_i × (e_f / ϵ_i)^(t/T_f)
13: Update the width of the neighboorhood σ(t):
14: end for
15: end for

The original Hebbian learning principle proposed by Hebb in 1949 states that “when an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that As efficiency, as one of the cells firing B, is increased.” In other words, any two neurons that are repeatedly active at the same time will tend to become “associated” so that activity in one facilitates activity (1) in the other. The learning rule is expressed by Equation 7.

However, Hebb’s rule is limited in terms of stability for online learning, as synaptic weights tend to infinity with a positive learning rate. This could be resolved by normalizing each weight over the sum of all the corresponding neuron weights, which guarantees the sum of each neuron weights to be equal to 1. The effects of weights normalization are explained in [79]. However, this solution breaks up with the locality of the synaptic learning rule, and that is not biologically plausible. In 1982, Oja proposed a Hebbian-like rule that adds a “forgetting” parameter, and solves the stability problem with a form of local multiplicative normalization for the neurons weights, as expressed in Equation 8. In addition, Oja’s learning performs an on-line principal component analysis of the data in the neural network, which is a very interesting property in the context of unsupervised learning.

Nevertheless, Hebb’s and Oja’s rules were both used in recent works with good results, respectively in [22] and [38]. Hence, we applied and compared both rules. The proposed multimodal association model is detailed in Algorithm 2, where η is a learning rate that we fix to 1 in our experiments, and γ is deduced according to the number or the percentage of synapses to prune, as discussed in Section V. The neurons activities computing in the line 3 of Algorithm 2 are calculated in the same way as Equation 1.

C. Divergence for labeling

As explained in III-A2, neurons labeling is based on a labeled subset from the training database. We tried in [67] to minimize its size, and used the fewest labeled samples while keeping the best accuracy. We will see in Section V that depending on the database, we sometimes need a considerable number of labeled samples, up to 10% of the training set. In this work, we propose an original method based on the divergence mechanism of the multimodal association: for two modalities x and y, since we can activate one modality based on the other, we propose to label the SOMx neurons from the activity and the labels induced from the SOMx neurons, which are based on the labeling subset of modality x. Therefore, we only need one labeled subset of a single modality to label both modalities, taking profit of the bidirectional aspect of reentry. A good analogy to biological observations would be the retro-activation of the auditory cortical areas from the visual cortex, if we take the example of written/spoken digits presented in Section V. It is similar to how infants respond to sound symbolism by associating shapes with sounds [80].

Algorithm 1: Kohonen SOM algorithm

1: Initialize the network as a two-dimensional array of k neurons, where each neuron n with m inputs is defined by a two-dimensional position p_n and a randomly initialized m-dimensional weight vector w_n.
2: for t from 0 to T_f do
3: for every input vector v do
4: for every neuron n in the network do
5: Compute the afferent activity a_n:
6: end for
7: Compute the winner s such that:
8: end for
9: for every neuron n in the network do
10: Compute the neighborhood function h(σ(t, n, s)):
11: end for
12: Update the learning rate ϵ(t): ϵ(t) = ϵ_i × (e_f / ϵ_i)^(t/T_f)
13: Update the width of the neighboorhood σ(t):
14: end for
15: end for

Algorithm 2: Multimodal association

1: Initialize the SOM network as a two-dimensional array of k neurons, where each neuron n with m inputs is defined by a two-dimensional position p_n and a randomly initialized m-dimensional weight vector w_n.
2: for t from 0 to T_f do
3: for every input vector v do
4: for every neuron n in the network do
5: Compute the neighborhood function h(σ(t, n, s)):
6: end for
7: Compute the winner s such that:
8: end for
9: for every neuron n in the network do
10: Compute the neighborhood function h(σ(t, n, s)):
11: end for
12: Update the learning rate ϵ(t): ϵ(t) = ϵ_i × (e_f / ϵ_i)^(t/T_f)
13: Update the width of the neighboorhood σ(t):
14: end for
15: end for
The proposed divergence method for labeling is detailed in Algorithm 3:

**Algorithm 3: Divergence algorithm for labeling**

1. **Initialize** \( \text{class}_{act} \) as a two-dimensional array of accumulators: the first dimension is the neurons and the second dimension is the classes.
2. **for** every input vector \( v_x \) of the \( x \)-modality labeling set with label \( l \) **do**
   3. **for** every neuron \( x \) in the \( SOM_x \) map **do**
   4. **Compute** the afferent activity \( a_x \):
      \[
      a_x = e^{-\frac{||v_x - w_{xy}||}{\sigma}}
      \]  
   5. **end for**
   6. **for** every neuron \( y \) in the \( SOM_y \) map **do**
   7. **Compute** the divergent activity \( a_y \) from the \( SOM_x \):
      \[
      a_y = \max_{x=0}^{n-1} (w_{xy} \times a_x)
      \]  
   8. **Add** the normalized activity with respect to the max activity to the corresponding accumulator:
      \[
      \text{class}_{act}[y][l] += a_y
      \]  
   9. **end for**
10. **end for**
11. **Normalize** the accumulators \( \text{class}_{act} \) with respect to the number of samples per class.
12. **for** every neuron \( y \) in the \( SOM_y \) map **do**
13. **Assign** the neuron label \( \text{neuron}_{lab} \):
    \[
    \text{neuron}_{lab} = \arg\max(\text{class}_{act}[y])
    \]  
14. **end for**

**D. Convergence for classification**

Once the multimodal learning is done and all neurons from both SOMs are labeled, we need to converge the information of the two modalities to achieve a better representation of the multi-sensory input. Since we use the reentry paradigm, there is no hierarchy in the processing, and the neurons computing is completely distributed based on the Iterative Grid detailed in Section IV. We propose an original cellular convergence method in three main steps:

- First, each neuron of the two SOMs computes its activity based on the afferent activity from the input.
- Second, each neuron updates its afferent activity via a multiplication with the lateral activity from the neurons of the other modality.
- Third and finally, all neurons compete to elect a winner, i.e., a global BMU with respect to the two SOMs.

Therefore, we have both a cooperation amongst neurons from different modalities and a global competition amongst all neurons. The proposed convergence method is detailed in Algorithm 4.

We explore different variants of the proposed convergence method regarding two aspects. First, both afferent and lateral activities can be taken as raw values or normalized values. We used min-max normalization that is therefore done with respect to the BMU and the Worst Matching Unit (WMU) activities. Second, the afferent activities update could be done for all neurons or only the two BMUs. In the second case, the global BMU cannot be another neuron but one of the two local BMUs, and if there is a normalization then it is only done for lateral activities (otherwise, the BMUs activities would be 1, and the lateral map activity would be the only relevant one). The results of our comparative study are presented and discussed in Section V.

**IV. CELLULAR NEUROMORPHIC ARCHITECTURE**

The centralized neural models that run on classical computers suffer from the Von-Neumann bottleneck due to the overload of communications between computing memory components, leading to an over-consumption of time and energy. One attempt to overcome this limitation is to distribute the computing amongst neurons as done in [52], but it implies an all-to-all connectivity to calculate the global information,
Cellular architectures for ANNs were common in early neuromorphic implementations and have recently seen a resurgence \[83\]. Such implementation is also referred as near-memory computing where one embeds dedicated coprocessors in close proximity to the memory unit, thus getting closer to the Parallel and Distributed Processing (PDP) paradigm formalized in the theory of ANNs. An FPGA distributed implementation model for SOMs was proposed in \[87\], where the local computation and the information exchange among neighboring neurons enable a global self-organization of the entire network.

Similarly, we proposed in \[21\] a cellular formulation of the related neural models which would be able to tackle the full connectivity limitation by iterating the propagation of the information in the network. This particular cellular implementation, named the Iterative Grid (IG), reaches the same behavior as the centralized models but drastically reduces their computing complexity when deployed on hardware. Indeed, we have shown in \[21\] that the time complexity of the IG is \(O(\sqrt{n})\) with respect to the number of neurons \(n\) in a squared map, while the time complexity of a centralized implementation is \(O(n)\). In addition, the connectivity complexity of the IG is \(O(n)\) with respect to the number of neurons \(n\), while the connectivity complexity of a distributed implementation with all-to-all connectivity \(O(n^2)\) is \(O(n^2)\). The principles of the IG are summarized in this section followed by a new SOM implementation over the IG substrata which takes into account the needs of the multimodal association learning and inference.

\[15\] A. The Iterative Grid (IG) substrata

Let’s consider a 2-dimensional grid shaped Network-on-Chip (NoC). This means that each node (neuron) of the network is physically connected (only) to its four closest neighbors. At each clock edge, each node reads the data provided by its neighbors and relays it to its own neighbors on the next one. The data is propagated (or broadcasted) in a certain amount of time to all the nodes. The maximum amount of time \(T_p\) which is needed to cover all the NoC (worst case reference) depends on its size: for a \(N \times M\) grid, \(T_p = N + M - 2\). After \(T_p\) clock edges, new data can be sent. A set of \(T_p\) iterations can be seen as a wave of propagation.

For the SOM afferent weights learning, the data to be propagated is the maximum activity for the BMU election, plus its distance with respect to every neuron in the map. The maximum activity is transmitted through the wave of propagation, and the distance to the BMU is computed in the same wave thanks to this finding: “When a data is iteratively propagated through a grid network, the propagation time is equivalent to the Manhattan distance between the source and each receiver.” \[21\]

The cellular propagation wave algorithm is detailed in Algorithm 5, where \(T_i\) is the iteration time that goes from 0 to \(T_p\). This \(T_i\) is to distinguish from \(t\) in Algorithm 1 which is relative to the training epoch. \(R\) the data stored in the node, \(D_j\) is the data given by the neighbor \(j\) with \(j \in [0;3]\) and the output buffers are memories used to keep the data consistency during the process. Each connection to neighbor

\begin{algorithm}
\begin{algorithmic}[1]
\State \textbf{for} every multimodal input vectors \(v_x\) and \(v_y\) \textbf{do}
\State \textbf{Do in parallel} every following step inter-changing modality \(x\) with modality \(y\) and vice-versa:
\State \textbf{Compute} the afferent activities \(a_x\) and \(a_y\):
\State \textbf{for} every neuron \(x\) in the SOM_x map \textbf{do}
\State \textbf{Compute} the afferent activity \(a_x\):
\State \hspace{1em} \(a_x = e^{-\frac{||w_{xy} - w_{xy}||}{\sigma}}\)
\State \textbf{end for}
\State \textbf{Normalize} (min-max) the afferent activities \(a_x\) and \(a_y\).
\State \textbf{Update} the afferent activities \(a_x\) and \(a_y\) with the lateral activities based on the associative synapses weights \(w_{xy}\):
\State \textbf{if} Update with max_update \textbf{then}
\State \textbf{for} every neuron \(x\) in the SOM_x map connected to \(n\) neurons from the the SOM_y map \textbf{do}
\State \hspace{1em} \(a_x = a_x \times \frac{\text{max}_{x=1}^{n}{(w_{xy} \times a_y)}}{n}\)
\State \textbf{end for}
\State \textbf{else if} Update with sum_update \textbf{then}
\State \textbf{for} every neuron \(x\) in the SOM_x map connected to \(n\) neurons from the the SOM_y map \textbf{do}
\State \hspace{1em} \(a_x = a_x \times \frac{\sum_{x=0}^{n}{(w_{xy} \times a_y)}}{n}\)
\State \textbf{end for}
\State \textbf{end if}
\State \textbf{Compute} the global BMU with the maximum activity between the SOM_x and the SOM_y.
\State \textbf{end for}
\end{algorithmic}
\end{algorithm}
nodes is provided with output double buffers, since we need to save the data of both current and previous clock edges.

Algorithm 5: Cellular propagation wave algorithm

1: $T_0$: Let $D_c$ the initial data of the cell.
2: compute $R \leftarrow g_1(D_c)$
3: Write $R$ on the output buffer.
4: for all $T_i$ do
5:   for all $D_j$ do
6:     compute $R_j \leftarrow f(D_j, T_i)$
7:   end for
8: compute $R \leftarrow g_4(R, R_0, R_1, R_2, R_3)$
9: Write $R$ on the output buffer.
10: Switch output buffers.
11: end for

A number of generic functions have been defined and explained in section [14-B]. In summary, the IG substrata allows to implement a cellular architecture able to distribute the centralized behavior of SOMs into each nodes of the NoC, transforming the connectivity complexity into a scalable time complexity in $O(\sqrt{n})$ with respect to the number of neurons $n$ regardless of the simulated SOM model [21].

B. Iterative Grid for Kohonen SOM model

The SOM implementation on the IG proposed in [21] has to be adapted to fit the needs of the multimodal association: (1) we add the Worst Matching Unit (WMU) activity needed for the activities min-max normalization in the convergence step, and (2) we use the Gaussian kernel in Equation 1 to transform the euclidean distances into activities. Therefore, the BMU is the neuron with the maximum activity and the WMU the neuron with the minimum one.

1) The BMU/WMU search wave: In order to compute the BMU and WMU search, we have to define $D_c$, $R$, $g_1$, $f$, and $g_4$. The figure 3 shows the BMU/WMU search IG implementation with a flowchart representation. Here, $D_c$ is the activity $A$ computed by the neuron before the wave, as defined in Equation 1. $R$ contains $A_{MIN}$ and $(A_{MAX}, T_M)$ with $A_{MIN}$ and $A_{MAX}$ the current known WMU and BMU activities respectively which are detected by the neuron at the iteration of propagation $T_M$. $g_1$ initializes $R$ with $A_{MIN} = A_{MAX} = D_c$ and $T_M = 0$. Because the propagation time $T$ is equivalent to the Manhattan distance, $T$ must be coherently coupled with $A_{MAX}$ for the learning equation computation. $i$ is the radius of the propagation which goes from 1 to half the perimeter of the grid. $f$ is very similar to $g_1$ and sets $R_j$ values to the respective value $[A_{MIN}, (A_{MAX}, T_i)]$. Finally, $g_4$ selects the minimum activity $A_{MIN}$ and the maximum one $A_{MAX}$ between $R, R_0, R_1, R_2, R_3$ and stores it as a result in the neuron’s output buffer. After this propagation wave, each neuron $n$ contains its own $[A_{MIN}, (A_{MAX}, T_M)]$ with $A_{MIN}$ and $A_{MAX}$ common for each of them and $T_M$ distinct values depending on their respective Manhattan distances to the BMU. Hence, Equation 2 is implemented without using a central controller or a full connectivity, but with a simple iterative cellular method based on a local connectivity.

2) The learning wave: From the winner propagation wave, every useful data is present in each neuron to compute the learning equation. No propagation wave is necessary at this step. We notice that the $A_{MIN}$ information is not necessary for the KSOM learning, but it is needed for the multimodal convergence step.

C. Hardware support for the iterative grid

The multi-FPGA implementation of the IG is a work in progress based on our previously implemented Neural Processing Unit (NPU) [89] [88]. As shown in Figure 4, the NPU is made of two main parts: the computation core and the communication engine. The computation core is a lightweight Harvard-like accumulator-based micro-processor where a central dual-port RAM memory stores the instructions and the data, both separately accessible from its two ports. A Finite
State Machine (FSM) controls the two independent ports of the memory and the Arithmetic and Logic Unit (ALU), which implements the needed operations to perform the equations presented in section IV-B. The aim of the communication engine is to bring the input stimuli vector and the neighbors activities to the computation core at each iteration. The values of the input vector flow across the NPUs through their $x_{in}$ and $x_{out}$ ports which are connected as a broadcast tree. The output activity ports of each NPU are connected to the four cardinal neighbors through a dedicated hard-wired channel.

Implemented on an Altera Stratix V GXEA7 FPGA, the resources (LUT, Registers, DSP and memory blocks) consumption is indeed scalable as it increases linearly in function of the size of the NPU network [89] [88]. We are currently working on configuring the new model in the NPU and implementing it on a more recent and adapted FPGA device, particularly for the communication part between multiple FPGA boards that will be based on [90].

In terms of scalable FPGA designs for neural networks, we find in the literature the work of Moore et al. [91] with the Bluehive project, a custom 64-FPGA machine made for large-scale real-time neural network simulation with a reconfigurable communication topology. Moore et al. showed that FPGAs are much better than current CPUs/GPUs for cellular architectures due to the low-latency and high-bandwidth communication needs. More recently, Wang et al. proposed an advanced multipurpose neuromorphic engine that breaks the Liebigs law, i.e. the problem that the performance of the system is limited by the component in shortest supply. The authors implemented an array of identical components, each of which can be configured as a leaky-integrate-and-fire (LIF) neuron, a learning-synapse (Spike timing dependent plasticity (STDP), or an axon with trainable delay (Spike Timing Dependent Delay Plasticity (STDDP)). Wang et al. also proposed an FPGA implementation of parallel and scalable neuromorphic cortex simulator [92], arranged in minicolumns and hypercolumns. Similarly to [88], the cortex simulator can be reconfigured for simulating different neural networks without any change in hardware structure by programming the memory. However, works in [91], [93] and [92] target SNNs with no on-chip learning for [91] and [92]. Our goal in terms of multi-FPGA communication is similar to [91], while neurons interconnections are different from [93] (central controller) and [92] (hierarchical communication).

Finally, the only cellular approach for implementing SOM model is proposed by Sousa et al. [87]. It is an FPGA implementation that shares the same approach as the IG with distributed cellular computing and local connectivity. However, the IG has two main advantages over the proposed cellular model in [87].

- Waves complexity: The smallest of 5 and neighborhood waves in [87] have been coupled into one wave called the winner wave, as the iterative grid is based on time to distance transformation to find the Manhattan distance between the BMU and each neuron. We have therefore a gain of about $2 \times$ in the time complexity of the SOM training.
- Sequential vs. combinatory architecture: The processes of calculating the neuron distances to the input vector, searching for the BMU and updating the weight vectors are performed in a single clock cycle. This assumption goes against the iterative computing paradigm in the SOM grid to propagate the neurons information. Hence, the hardware implementation in [87] is almost fully combinatory. It explains why the maximum operating frequency is low and decreases when increasing the number of neurons, thus being not scalable in terms of both hardware resources and latency.

D. Hardware support for multimodal association

For the multimodal association learning in Algorithm 2, the local BMU in each of the two SOMs needs both the activity and the position of the local BMU of the other SOM to perform the Hebbian-like learning in the corresponding lateral synapse. This communication problem has not been experimented in this work. However, this suppose a simple communication mechanism between the two maps that would implemented in two FPGAs where only the BMUs of each map send a message to each other in a bidirectional way. The message could go through the routers of the IG thanks to an XY-protocol to reach an inter-map communication port in order to avoid the multiplication of communication wires. It is to note that in this work, we follow the same approach as [22].
since we only reinforce the synaptic connections between the two unimodal BMUs for each sample. The other approach would be learning with all neurons, but it would create a bottleneck in the inter-map communication and thus drastically increase the learning time.

For the divergence and convergence methods in Algorithm 3 and Algorithm 4 respectively, the local BMU in each of the two SOMs needs the activity of all the connected neurons from the other SOM after pruning, i.e. around 20 connections per neuron. Because the number of remaining synapses is statistically bounded to 20%, the number of communication remains low in front of the number of neurons. Here again, we did not experiment on this communication mechanism but the same communication support could be used. Each BMU can send a request that contains a list of connected neurons. This request can be transmitted to the other map through the IG routers to an inter-map communication channel. Once on the other map, the message could be broadcasted to each neuron using again the routers of the IG. Only the requested neurons send back their activity coupled to their position in the BMU request. This simple mechanism suppose a low amount of communication thanks to the pruning that has been done previously. This inter-map communication can be possible if the IG routers support XY or equivalent routing techniques and broadcast in addition to the one of the propagation wave.

At this point of our work, each neuron is implemented in a NPU based on a previous implementation [88] with all the computing and memory resources, amongst which a list of the lateral synaptic weights with indexes to identify existing from pruned (or not sprouted) connections. Therefore, each lateral weight is saved by two neurons in the case of two SOMs.

V. EXPERIMENTS AND RESULTS

In this section, we present the databases and the results from our experiments with each modality alone, then with the multimodal association convergence and divergence, and we finally compare our model to three different approaches. All the results presented in this section have been averaged over a minimum of 10 runs, with shuffled datasets and randomly initialized neurons afferent weights.

A. DATABASES

The most important hypothesis that we want to confirm through this work is that the multimodal association of two modalities leads to a better accuracy than the best of the two. For this purpose, we worked on two databases that we present in this section.

1) Written/spoken digits database: The Mixed National Institute of Standards and Technology (MNIST) database [76] is a database of 70000 handwritten digits (60000 for training and 10000 for test) proposed in 1998. Even if the database is quite old, it is still commonly used as a reference for training, testing and comparing various ML systems for image classification. In [67], we applied Kohonen-based SOMs for MNIST classification with unsupervised learning, and achieved state-of-art performance with the same number of neurons (100) and only 1% of labeled samples for neurons labeling. However, the obtained accuracy of 87.36% is not comparable to supervised DNNs, and only two approaches have been used in the literature to bridge the gap: either use a huge number of neurons (6400 neurons in [52]) with exponential increase in size for linear increase in accuracy [41] which is not scalable for complex databases, or use unsupervised feature extraction followed by a supervised classifier (Support Vector Machine in [94]) which relies on the complete labeled dataset. We propose the multimodal association as a way to bridge the gap while keeping a small number of neurons and an unsupervised learning method from end to end. For this purpose, we use the classical MNIST as a visual modality that we associate to an auditory modality: Spoken-MNIST (S-MNIST).

We extracted S-MNIST from Google Speech Commands (GSC) [95], an audio dataset of spoken words that was proposed in 2018 to train and evaluate keyword spotting systems. It was therefore captured in real-world environments though phone or laptop microphones. The dataset consists of 105829 utterances of 35 words, amongst which 38908 utterances (34801 for training and 4107 for test) of the 10 digits from 0 to 9. We constructed S-MNIST associating written and spoken digits of the same class, respecting the initial partitioning in [76] and [95] for the training and test databases. Since we have less samples in S-MNIST than in MNIST, we duplicated some random spoken digits to match the number of written digits and have a multimodal-MNIST database of 70000 samples. The whole pre-processed dataset is available in [96].

2) DVS/EMG hand gestures database: To validate our results, we experimented our model on a second database that was originally recorded with multiple sensors: the DVS/EMG hand gestures database [97]. Indeed, the discrimination of human gestures using wearable solutions is extremely important as a supporting technique for assisted living, healthcare of the elderly and neuro-rehabilitation. For this purpose, we proposed in [98] a framework that allows the integration of multi-sensory data to perform sensor fusion based on supervised learning. The framework was applied for the hand gestures recognition task with five hand gestures: Pinky (P), Elle (E), Yo (Y), Index (I) and Thumb (T).

The dataset consists of 6750 samples (5400 for training and 1350 for test) of muscle activities via EletroMioGraphy (EMG) signals recorded by a Myo armband (Thalmic Labs Inc) from the forearm, and video recordings from a Dynamic Vision Sensor (DVS) using the computational resources of a mobile phone. The DVS is an event-based camera inspired by the mammalian retina [99], such that each pixel responds asynchronously to changes in brightness with the generation of events. Only the active pixels transfer information and the static background is directly removed on hardware at the frontend end. The asynchronous nature of the DVS makes the sensor low power, low latency and low-bandwidth, as the amount of data transmitted is very small. It is therefore a promising solution for mobile applications [100] as well as neuromorphic chips, where energy efficiency is one of the most important
Fig. 5. MNIST learning with KSOM: (a) neurons afferent weights; (b) neurons labels; (c) confusion matrix; we can visually assess the good labeling from (a) and (b), while (c) shows that some classes like 4 and 9 are easier to confuse than others, and that’s due to their proximity in the 784-dimensional space; (d) S-MNIST divergence confusion matrix; (e) DVS confusion matrix; (f) EMG divergence confusion matrix; the interesting characteristic is that the confusion between the same classes is not the same for the different modalities, and that’s why they can complement each other.

characteristics.

B. Unimodal classification

1) Written digits: MNIST classification with a KSOM was already performed in [67], achieving around 87% of classification accuracy using 1% of labeled images from the training dataset for the neurons labeling. The only difference is the computation of the $\alpha$ in Equation 1 for the labeling process. We proposed in [67] a centralized method for computing an approximated value of $\alpha$, but we consider it as a simple hyper-parameter for this work. We therefore calculate the best value off-line with a grid search since we do not want to include any centralized computation, and because we can find a closer value to the optimum, as summarized in Table II. The same procedure with the same hyper-parameters defined above is applied for each of the remaining unimodal classifications. Finally, we obtain 87.04% ± 0.64 of accuracy. Figure 5 shows the neurons weights that represent the learned digits prototypes with the corresponding labels, and the confusion matrix that highlights the most frequent misclassifications between the digits whose representations are close: 23.12% of the digits 4 are classified as 9 and 12.69% of the digits 9 are classified as 4. We find the same mistakes with a lower percentage between the digits 3, 5, and 8, because of their proximity in the 784-dimensional vector space. That’s what we aim to compensate when we add the auditory modality.

2) Spoken digits: Speech recognition is more and more present in human-computer interfaces like personal assistants (Google Assistant, Microsoft Cortana, Amazon Alexa, Apple Siri, etc.). The most commonly used acoustic feature in speech recognition is the Mel Frequency Cepstral Coefficients (MFCC) [101] [102] [103]. MFCC was first proposed in [104], which has since become the standard algorithm for representing speech features. It is a representation of the short-term power spectrum of a speech signal, based on a linear cosine transform of a log power spectrum on a nonlinear Mel scale of frequency. We first extracted the MFCC features from the S-MNIST data, using the hyper-parameters from [103]:
framing window size = 50ms and frame shift size = 25ms. Since the S-MNIST samples are approximately 1s long, we end up with 39 dimensions. However, it’s not clear how many coefficients one has to take. Thus, we compared three methods: [105] proposed to use 13 weighted MFCC coefficients, [106] proposed to use 40 log-mel filterbank features, and [103] proposed to use 12 MFCC coefficients with an additional energy coefficient, making it 13 coefficients in total. The classification accuracy is respectively 61.79% ± 1.19, 50.33% ± 0.59 and 75.14% ± 0.57. We therefore chose to work with a 39 x 13 dimensional features that are standardized (each feature is transformed by subtracting the mean value and dividing by the standard deviation of the training dataset, also called Z-score normalization) then min-max normalized (each feature is re-scaled to 0 – 1 based on the minimum and maximum values of the training dataset).

3) DVS hand gestures: In order to use the DVS events for gesture classification with conventional algorithms, we need to turn the stream of events into frames, as previously done in [98]. These frames are generated by accumulating the events occurring in a fixed time window of 200ms, so that they can be synchronized with the EMG signal. For each pixel, we count the number of events within the time windows regardless of their polarity, then we transform the event count frame into gray scale by min-max normalization. The event frames obtained from the DVS camera have a resolution of 128 x 128 pixels. Since the region with the hand gestures does not fill the full frame, we extract a 60 x 60 pixels patch that allows us to significantly decrease the amount of computation needed during learning and inference.

Even though unimodal classification accuracies are not the first goal, we need to reach a satisfactory performance. Since the dataset is small and the DVS frames are of high complexity with a lot of noise from the data acquisition, we either have to significantly increase the number of neurons for the SOM or to add an additional feature-extraction. We decided to use the second method with a Convolutional Neural Network (CNN)-based feature extraction to keep a reasonable number of neurons. One of our future improvements is to use unsupervised learning for feature extraction based on recent works in [107], [94] and [108], but it is out of the scope of this work. Hence, we use a supervised feature extraction based on the LeNet-5 topology [109] with one difference: the last convolution layer has only 12 filters instead of 120. In this way, we end up with extracted features of 972 dimensions, that we standardize and normalize before using as input for the SOM. We obtain an accuracy of 70.06% ± 1.15.

4) EMG hand gestures: For the EMG signal, we selected two time domain features that are commonly used in the literature [110]: the Mean Absolute Value (MAV) and the Root Mean Square (RMS) which are calculated over the same window of length 20ms, as detailed in [98]. With the same strategy as for DVS frames, we extract CNN-based features of 192 dimensions. The SOM reaches a classification accuracy of 66.89% ± 0.84.

C. Multimodal classification

After inter-SOM sprouting (Figure 6), training and pruning (Figure 7), we move to the inference for two different tasks: (1) labeling one SOM based on the activity of the other (divergence), and (2) classifying multimodal data with cooperation and competition between the two SOMs (convergence).

1) Divergence results: Table II shows unimodal classification accuracies using the divergence mechanism for labeling, with 75.9% ± 0.2 for S-MNIST classification and 65.56% ± 0.25 for EMG classification. As shown in Figure 7 we reach this performance using respectively 20% and 25% of the potential synapses for digits and hand gestures. Since the pruning is performed by the neurons of the source SOMs, i.e. the MNIST-SOM and DVS-SOM, pruning too much synapses causes some neurons of the S-MNIST-SOM and EMG-SOM to be completely disconnected from the source map, and therefore do not get any activity for the labeling process. Hence, the labeling is incorrect, with the disconnected neurons stuck with the default label 0. In comparison to the classical labeling process with 10% of labeled samples, we have a loss of only ~1.33% for EMG, and even a small gain of 0.76% for S-MNIST even though we only use 1% of labeled digits images. The choice of which modality to use to label the other is made according to two criteria: the source map must (1) achieve the best unimodal accuracy so that we maximize the separability of the transmitted activity to the other map, and it must (2) require the least number of labeled data for its own labeling so that we minimize the

| Database       | Digits | S-MNIST | DVS | EMG |
|----------------|--------|---------|-----|-----|
| **SOMs**       |        |         |     |     |
| Dimensions     | 784    | 507     | 972 | 192 |
| Neurons        | 100    | 256     | 256 | 256 |
| Labeled data (%) | 1      | 10      | 10  | 10  |
| Accuracy (%)   | 87.04  | 75.14   | ±0.1 | 70.06 ± 200 | 66.89 ± 1.0 |
| **Divergence** |        |         |     |     |
| Labeled data (%) | 1      | 10      | 10  | 0   |
| Gain (%)       | /      | +0.76   | /   | -1.33 |
| Accuracy (%)   | /      | 75.90   | /   | 65.56 |
| **Convergence**|        |         |     |     |
| Gain (%)       | +8.03  | +19.17  | +5.67 | +10.17 |
| Accuracy (%)   | 95.07  | 75.73   |     |     |

**TABLE II**

**CLASSIFICATION ACCURACIES AND CONVERGENCE/GAIN RATES.**
TABLE III
MULTIMODAL CLASSIFICATION ACCURACIES

| Learning | Update algorithm | Neurons activities | Digits | Hand gestures |
|----------|------------------|--------------------|--------|--------------|
|          |                  | All neurons | BMUs only | All neurons | BMUs only |
| Hebb     | Max              | 69.39 1 | 91.11 1 | 71.57 5 | 73.01 5 |
|          | Raw              | 66.15 1 | 91.76 10 | 75.20 4 | 75.69 4 |
|          | Norm             | 71.85 1 | 93.63 3 | 75.73 4 | 73.84 20 |
| Oja      | Max              | 94.79 4 | 91.17 1 | 71.35 3 | 73.96 10 |
|          | Raw              | 74.34 2 | 89.89 3 | 75.10 4 | 73.63 10 |
|          | Norm             | 91.59 15 | 89.32 30 | 73.75 4 | 74.22 30 |

Fig. 6. SOMs lateral sprouting in the multimodal association process: (a) Written/Spoken digits maps; (b) DVS/EMG hand gestures maps. We notice that less than half of the possible lateral connections are created at the end of the Hebbian-like learning, because only meaningful connections between correlated neurons are created. For (b), the even smaller number of connections is also related to the small size of the training dataset.

Fig. 7. Divergence and convergence classification accuracies VS. the remaining percentage of lateral synapses after pruning: (a) Written/Spoken digits maps; (b) DVS/EMG hand gestures maps. We see that we need more connections per neuron for the divergence process, because the pruning is done by the neurons of one of the two maps, and a small number of connections results in some disconnected neurons in the other map.

number of samples to label during data acquisition. Overall, the divergence mechanism for labeling leads to approximately the same accuracy than the classical labeling. Therefore, we perform the unimodal classification of S-MNIST and EMG with no labels from end to end.

2) Convergence results: We proposed eight variants of the convergence algorithm for each the two learning methods. For the discussion, we denote them as follow: Learning –
Fig. 8. Multimodal convergence classification: (a) Written/Spoken digits; (b) DVS/EMG hand gestures. The red and green lines are respectively the lowest and highest unimodal accuracies. Hence, there is an overall gain whenever the convergence accuracy is above the green line.

UpdateNeurons such that Learning can be Hebb or Oja, Update can be Max or Sum, Normalization can be Raw (the activities are taken as initially computed by the SOM) or Norm (all activities are normalized with a min-max normalization thanks to the WMU and BMU activities of each SOM), and finally Neurons can be BMU (only the two BMUs update each other and all other neurons activities are reset to zero) or All (all neurons update their activities and therefore the global BMU can be different from the two local BMUs). It is important to note that since we constructed the written/spoken digits dataset, we maximized the cases where the two local BMUs have different labels such as one of them is correct. This choice was made in order to better assess the accuracies of the methods based on BMUs update only, as both cases when the two BMUs are correct or incorrect at the same time lead to the same global results regardless of the update method. The convergence accuracies for each of the eight method applied on the two databases are summarized in Table III and Figure 8.

For the digits, we first notice that the Hebb’s learning with all neurons update leads to very poor performance, worse than the unimodal classification accuracies. To explain this behavior, we have to look at the neurons BMU counters during learning in Figure 9. We notice that some neurons, labeled as 1 in Figure 5, are winners much more often than other neurons. Hence, their respective lateral synapses weights increase disproportionately compared to other synapses, and lead those neurons to be winners most of the time after the update, as their activity is higher than other neurons very often during convergence. This behavior is due to two factors: first, the neurons that are active most of the time are those that are the fewest to represent a class. Indeed, we have less neurons labeled 1 compared to other classes, because the digit 1 have less sub-classes. In other words, the digit 1 has less variants and therefore can be represented by less prototype neurons. Consequently, those neurons are active more often because the number of samples for each class is approximately equal. Second, the Hebb’s learning is unbounded, leading the lateral synapses weights to increase indefinitely. Thus, this problem occurs less when we use Oja’s rule, as shown in Figure 8. We
notice that Oja’s learning leads to more homogenous results, and normalization often leads to a better accuracy. The best method using Hebb’s learning is \( Hebb - Max_{BMU}^{BMU} \) with 95.07\% ± 0.08, while the best method using Oja’s learning is \( Oja - Max_{All}^{Norm} \) with 94.79\% ± 0.11.

For the hand gestures, all convergence methods lead to a gain in accuracy even though the best gain is smaller than for digits, as summarized in Table II. It can be explained by the absence of neurons that would be BMUs much more often than other neurons, as shown in Figure 10. The best method using Hebb’s learning is \( Hebb - Sum_{All}^{BMU} \) with 75.73\% ± 0.91, while the best method using Oja’s learning is \( Oja - Sum_{Raw} \) with 75.10\%±0.9. In contrast with the digits database, here the most accurate methods are based on the Sum update. Thus, each neuron takes in account the activities of all the neurons that it is connected to. A plausible reason is the fact that the digits database was constructed whereas the hand gestures database was initially recorded with multi-modal sensors, which gives it a more natural correlation between the two modalities.

Overall, the best methods for both digits and hand gestures databases are based on Hebb’s learning, even though the difference with the best methods based on Oja’s learning is very small, and Oja’s rule has the interesting property of bounding the synaptic weights. For hardware implementation, the synaptic weights of the Hebb’s learning can be normalized after a certain threshold without affecting the model’s behavior, since the strongest synapse stays the same when we divide all the synapses by the same value. However, the problem is more complex in the context of on-line learning as discussed in Section VI. Quantitatively, we have a gain of +8.03\% and +5.67\% for the digits and the hand gestures databases respectively, compared to the best unimodal accuracies. The proposed convergence mechanism leads to the election of a global BMU between the two unimodal SOMs: it is one of the local BMUs for the \( Hebb - Max_{BMU}^{BMU} \) method used for digits, whereas it can be a completely different neuron for the \( Hebb - Sum_{Norm}^{All} \) used for hand gestures. In the first case,
Fig. 10. DVS/EMG hand gestures neurons BMU counters during multimodal learning and inference using Hebb−Sum method: (a) DVS SOM in learning; (b) EMG SOM neurons during learning; (c) DVS SOM neurons during inference; (d) EMG SOM neurons during inference.

since the convergence process can only elect one of the two local BMUs, we can compute the absolute accuracy in the cases where the two BMUs are different with one of them being correct. We find that the correct choice between the two local BMUs is made in about 87% of the cases. However, in both cases, the convergence leads to the election of a global BMU that is indeed spread in the two maps, as shown in Figures 9 and 10. Nevertheless, the neurons of the hand gestures SOMs are less active in the inference process, because we only have 1350 samples in the test database.

The best accuracy for both methods is reached using a subpart of the lateral synapses, as we prune a big percentage of the potential synapses as shown in Figure 7. We say potential synapses, because the pruning is performed with respect to a percentage (or number) of synapses for each neuron, and the neuron does not have the information of other neurons due to the cellular architecture. Thus, the percentage is calculated with respect to the maximum number of potential lateral synapses, that is equal to the number of neurons in the other SOM, and not the actual number of synapses. In fact, at the end of the Hebbian-like learning, each neuron is only connected to the neurons where there is at least one co-occurrence of BMUs, as shown in Figure 6. Especially for the hand gestures database, the sprouting leads to a small total number of lateral synapses even before pruning, because of the small number of samples in the training dataset. Finally, we need at most 10% of the total lateral synapses to achieve the best performance in convergence as shown in Figure 7. However, if we want to maintain the unimodal classification with the divergence method for labeling, than we have to keep 20% and 25% of the potential synapses for digits and hand gestures, respectively.

One interesting aspect of the multimodal fusion is the explainability of the better accuracy results. To do so, we plot the confusion matrices with the best convergence methods for the digits and hand gestures datasets in Figure 11. The gain matrices mean an improvement over the unimodal performance when they have positive values in the diagonal and negative values elsewhere. If we look at the gain matrix
Fig. 11. Written/Spoken digits confusion matrices using Hebb $- \text{Max}^{\text{BMU}_{\text{Norm}}}$ method: (a) convergence; (b) convergence gain with respect to MNIST; (c) convergence gain with respect to S-MNIST; DVS/EMG hand gestures confusion matrices using Hebb $- \text{Sum}^{\text{All}_{\text{Norm}}}$ method: (d) convergence; (e) convergence gain with respect to DVS; (f) convergence gain with respect to EMG.

Our results confirm that multimodal association is interesting because the strengths and weaknesses of each modality can be complementary. Indeed, Rathi and Roy [41] state that if the non-idealities in the unimodal datasets are independent, then the probability of misclassification is the product of the misclassification probability of each modality. Since the product of two probabilities is always lower than each probability, then each modality helps to overcome and compensate for the weaknesses of the other modality. Furthermore, multimodal association improves the robustness of the overall system to noise [41], and in the extreme case of losing one modality, the system could rely on the other one which links back to the concept of degeneracy in neural structures [4].

D. Comparative study

First, we compare our results with STDP approaches to assess the classification accuracy with a comparable number of neurons. Next, we confront our results with two different approaches: we try early data fusion using one SOM, then we use supervised perceptrons to learn the multimodal representations based on the two unimodal SOMs activities.

of the convergence method compared to the image modality, we notice two main characteristics: first, all the values in the diagonal are positive, meaning that there is a total accuracy improvement for all the classes. Second and more interestingly, the biggest absolute values outside the diagonal lie where there is the biggest confusion for the images, i.e. between the digits 4 and 9, and between the digits 3, 5 and 8, as previously pointed out in Section V-B1. It means that the auditory modality brings a complementary information that leads to a greater separability for the classes which have the most confusion in the visual modality. Indeed, the similarity between written 4 and 9 is compensated by the dissimilarity of spoken 4 and 9. The same phenomenon can be observed for the auditory modality, where there is an important gain for the digit 9 that is often misclassified as 1 or 5 in the speech SOM, due to the similarity of their sounds. Similar remarks are applicable for the hand gestures database with more confusion in some cases, which leads to a smaller gain.
TABLE IV
DIGITS CLASSIFICATION COMPARISON

| Learning | Model | # neurons | Labeled data (%) | Modality | Dataset | Accuracy (%) |
|----------|-------|-----------|------------------|----------|---------|--------------|
| STDP     | Diehl et al. [52] (2015) | 400 | 100 | Unimodal | MNIST | 88.74 |
|          | Hazan et al. [111] (2018) | 400 | 100 | Unimodal | MNIST | 92.56 |
|          | Rathi et al. [41] (2018) | 400 | 100 | Unimodal | MNIST | 86.00 |
|          | Rathi et al. [41] (2018) | 400 | 100 | Multimodal | MNIST + TI46 | 89.00 |
| Self-organization | Khacef et al. [this work] (2020) | 356 | 1 | Multimodal | MNIST + SMNIST | 95.07 |

* Labeled data are only used for the neurons labeling after unsupervised training.

1) SOMs vs. SNNs approaches for unsupervised learning: Table IV summarizes the digits classification accuracy achieved using brain-inspired unsupervised approaches, namely SOMs with self-organization (Hebb, Oja and Kohonen principles) and SNNs with STDP. We achieve the best accuracy with a gain of about 6% over Rathi and Roy [41], which is to the best of our knowledge the only work that explores brain-inspired multimodal learning for written/spoken digits classification. It is to note that we do not use the TI46 spoken digits database [112], but a subpart of Google Speech Google Speech Commands [95], as presented in section V-A1.

We notice that all other works use the complete training dataset to label the neurons, which is incoherent with the goal of not using labels, as explained in [67]. Moreover, the work of Rathi and Roy [41] differs from our work in the following points:

- The cross-modal connections are formed randomly and initialized with random weights. The multimodal STDP learning is therefore limited to connections that have been randomly decided, which induces an important variation in the network performance.
- The cross-modal connections are not bi-directional, thus breaking with the biological foundations of reentry and CDZ. Half the connections carry spikes from image to audio neurons and the other half carry spikes from audio to image neurons, otherwise making the system unstable.
- The accuracy goes down beyond 26% connections. When the number of random cross-modal connections is increased, the neurons that have learned different label gets connected. We do not observe such a behavior in our work, as shown in Figure 7.
- The decision of the multimodal network is computed by observing the spiking activity in both ensembles, thus requiring a central unit.

Nevertheless, the STDP-based multimodal learning is still a promising approach for the hardware efficiency of SNNs [113], and because of the alternative they offer for using even-based sensors with asynchronous computation.

2) SOM early data fusion: We find in the literature two main different strategies for multi-modal fusion [114] [42]: (1) score-level fusion where data modalities are learned by distinct models then their predictions are fused with another model that provides a final decision, and (2) data-level fusion where modalities are concatenated then learned by a unique model. Our approach can be classified as a classifier-level fusion which is closer to score-level fusion and usually produces better results than feature-level or data-level fusion for classification tasks [115] [116] [117]. However, it is worth trying to learn the concatenated modalities with one SOM having as much neurons as the two uni-modal SOMs, for a fair comparison. We use 361 and 529 neurons for digits and hand gestures respectively. We have few neurons more compared to the sum of the two uni-modal SOMs, as we want to keep the same square grid topology. We train the SOMs with the same hyper-parameters as for the uni-modal SOMs, and reach 90.68% ± 0.29 and 75.6% ± 0.32 accuracy for digits and hand gestures, respectively. We still have a gain compared to the uni-modal SOMs, but have an important loss of −4.39% for digits and a negligible loss of −0.13% for hand gestures compared to the proposed multimodal association. These results are coherent with the literature findings. Furthermore, the accuracy is not the only metric, as the memory footprint is an important factor to take in consideration when choosing a fusion strategy [118], especially for embedded systems. Indeed, since we target a hardware implementation on FPGA, the total number of afferent and lateral synaptic weights are parameters that require on-chip memory, which is very limited. With a simple calculation using the number of neurons and input dimensions, we find that we have a gain of 49.84% and 40.96% for digits and hand gestures respectively using the multimodal association compared to a data-level fusion strategy.

3) SOM coupled to supervised fusion: In order to have an approximation of the best accuracy that we could obtain with multimodal association, we used a number of perceptrons equal to the number of classes on top of the two uni-modal SOMs of the two databases, and performed supervised learning for the same number of epochs (10) using gradient descent (Adadelta algorithm). We obtain 91.29% ± 0.82 and 80.19% ± 0.63 of accuracy for the digits and hand gestures respectively. Surprisingly, we have a loss of −3.78% for the digits. However, we have a gain of 4.43% for the hand gestures. We argue that the hand gestures dataset is too small to construct robust multimodal representations through unsupervised learning, and that could explain the smaller overall gain compared to the digits dataset.
VI. DISCUSSION

A. A universal multimodal association model?

The development of associations between co-occurring stimuli for multimodal binding has been strongly supported by neurophysiological evidence \[66\] \[119\]. Similar to \[120\], \[121\] and \[39\] and based on our experimental results, we argue that the co-occurrence of sensory inputs is a sufficient source of information to create robust multimodal representations with the use of associative links between unimodal representations that can be incrementally learned in an unsupervised fashion.

In terms of learning, the best methods are based on Hebb's learning with a slightly better accuracy over Oja's learning, but the overall results are more homogeneous using Oja's learning that prevents the synaptic weights from growing indefinitely. The best results are obtained using $Hebb - Max_{Norm}^{BMU}$ with 95.07% ± 0.08 and $Hebb - Sum_{Norm}^{All}$ with 75.73% ± 0.91 for the digits and hand gestures databases, respectively. We notice that the $BMU$ method is coupled with the $Max$ update while the $All$ neurons method is coupled with the $Sum$ update, and the $Norm$ activities usually perform better than $Raw$ activities. However, we cannot have a final conclusion on the best method, especially since it depends on the nature of the dataset.

Moreover, the experimental results depend on the $\beta$ hyper-parameter, the Gaussian kernel width that has to be tuned for every database and every method. Thanks to the multiplicative update, the values of both SOMs are brought into the same scale which gives the possibility to elect the correct global $BMU$, and we get rid of a second hyper-parameter that would arise with a sum update method like in \[40\]. However, it is still time-taking in the exploration of the proposed methods for future works, even if it is a common limit when dealing with any ANN. Trying to find a more efficient approach for computing $\beta$ is part of our ongoing works.

Finally, multimodal association bridges the gap between unsupervised and supervised learning, as we obtain approximately the same results compared to MNIST using a supervised Multi-Layer Perceptron (MLP) with 95.73% \[113\] and S-MNIST using a supervised attention Recurrent Neural Network (RNN) with 94.5% \[122\] (even though this results was obtained on 20 commands). It can be an interesting approach to deeper explore, as we have in most cases the possibility to include multiple sensory modalities when dealing with the real-world environment.

B. Online multimodal learning

The multimodal association learning methods explored in this work are performed in two times: first, we train the SOMs for unimodal classifications, and second we create and reinforce bidirectional connections between the two maps based upon their activities on the same training dataset. In the context of on-line learning in a dynamic and changing environment, another approach would be to perform both Kohonen-like and Hebbian-like learning at the same time, continuously. For example, this approach is followed with STDP learning in \[41\].

For this purpose, the KSOM would be replaced by the DSOM. The reason is that the KSOM has a decaying learning rate and neighborhood width, so that the learning stabilizes after a certain number of iterations. As detailed in Algorithm[1] when $t = t_f$, the KSOM is almost unable to learn any change in the input stimuli, as $\epsilon_f << \epsilon_t$ and $\sigma_f << \sigma_t$. Therefore, the learning is stable but not dynamic. It can be considered as an off-line unsupervised learning algorithm. In contrast, the DSOM introduced by Rougier et al. \[74\] is a variation of the KSOM algorithm where the time dependency of the learning rate and neighborhood function has been replaced by the distance between the BMU and the input stimulus. Even if the DSOM is less accurate than the KSOM as we previously shown in \[67\], it is suitable for on-line learning.

Moreover, Oja’s learning would be the only alternative, as we need the forgetting parameter that bounds the synaptic weights and enables decaying, which is not available in Hebb’s learning. In addition, we would need a dynamic learning rate so that the multimodal association becomes stronger when the sample is well learned by the SOM, i.e. when the distance between the BMU and the sample is small. One way to do that is to add a Gaussian kernel to that distance, so that the multimodal binding becomes more relevant after the convergence of the SOMs, without any manual tuning on the hyper-parameters of the SOM. This is part of our ongoing works.

C. SOMA: Toward hardware plasticity

This work is part of the Self-Organizing Machine Architecture (SOMA) project \[123\], where the objective is to study neural-based self-organization in computing systems and to prove the feasibility of a self-organizing multi-FPGA hardware structure. We define computational models able to simultaneously self-organize at both computation and communication levels, and we want these models to be hardware-compliant, fault tolerant and scalable by means of cellular neuromorphic architectures.

In fact, the concept of the IG is supported in \[123\] as it states the following: “Changes initially are local: components only interact with their immediate neighbors. They are virtually independent of components farther away. But self-organization is often defined as global order emerging from local interactions”. Moreover, it states that “a self-organizing system not only regulates or adapts its behavior, it creates its own organization. In that respect it differs fundamentally from our present systems, which are created by their designer”. Indeed, the multimodal association through Hebbian-like learning is a self-organization that defines the inter-SOMs structure, where neurons are only connected to each other when there is a strong correlation between them. That’s a form of hardware plasticity. The hardware gain of the self-organization is therefore the gain in communication support, which is proportional to the percentage of remaining synapses for each neuron after learning and pruning.
VII. CONCLUSION AND FURTHER WORKS

We proposed in this work a brain-inspired computational model for multimodal unsupervised learning based on the reentry paradigm proposed by Edelman [19], with a generic model regardless of the number of maps and the number of neurons per map. Our system learns unimodal representations with Kohonen-based Self-Organizing Maps, then creates and reinforces the multimodal association through sprouting, Hebbian-like learning and pruning. It enables both structural and synaptic plasticities that are the core of neural self-organization. We exploited both convergence and divergence that are highlighted by Damasio [20], thanks to the bi-directional property of the multimodal representation in a classification task: the divergence mechanism is used to label one modality based on the other, and the convergence is used to introduce cooperation and competition between the modalities and reach a better accuracy than the best of the two unimodal accuracies. Indeed, our experiments show that the divergence labeling leads to approximately the same unimodal accuracy as when using labels, and we reach a gain in the multimodal accuracy of +8.03 for the written/spoken digits database and +5.67 for the DVS/EMG hand gestures database. Our model exploits the natural complementarity between different modalities like sight and sound as shown by the confusion matrices, so that they complete each other and improve the multimodal classes separability. Finally, our system is implemented on the Iterative Grid substrata to distribute the neural computing in a cellular neuromorphic architecture where each neuron is locally connected to its four neighbors. The system’s inter-map structure is therefore learned along the system’s experience through self-organization and not fixed by the user. It leads to a gain in the communication time that is proportional to the number of pruned lateral synapses for each neuron, that is about 80% of the possible connections. In addition to the convergence and divergence gains, the self-organization induces a form of hardware plasticity which has an impact on the hardware efficiency of the system, and that’s a first result that opens very interesting perspectives for future designs and implementations of self-organizing architectures inspired from the brain’s plasticity.

CONFLICT OF INTEREST STATEMENT

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.

AUTHOR CONTRIBUTIONS

LK, LR and BM conceived the idea. LK developed the code and performed the experiments, LR and BM supervised the work. LK, LR and BM wrote the paper.

FUNDING

This material is based upon work supported by the French Research Agency (ANR) and the Swiss National Science Foundation (SNSF) through SOMA project ANR-17-CE24-0036.

ACKNOWLEDGMENTS

The authors would like to acknowledge the 2019 Capoccia Neuromorphic Workshop and all its participants for the fruitful discussions.

DATA AVAILABILITY STATEMENT

The datasets for this study can be found in [96] and [97].

REFERENCES

[1] R. J. Sternberg, Handbook of Intelligence. Cambridge University Press, 2000.
[2] L. Smith and M. Gasser, “The development of embodied cognition: Six lessons from babies,” Artificial Life, vol. 11, no. 1-2, pp. 13–29, 2005. [Online]. Available: https://doi.org/10.1162/106454605778973
[3] A. Droniou, S. Ivaldi, and O. Sigué, “Deep unsupervised network for multimodal perception, representation and classification,” Robotics and Autonomous Systems, vol. 71, pp. 83 – 98, 2015, emerging Spatial Competences: From Machine Perception to Sensorimotor Intelligence. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889014002474
[4] G. M. Edelman, Neural Darwinism: The theory of neuronal group selection. New York, US: Basic Books, 1987.
[5] G. Edelman and J. Gally, “Edelman gm, gally ja. degeneracy and complexity in biological systems. proc natl acad sci usa 98: 13763-13768,” Proceedings of the National Academy of Sciences of the United States of America, vol. 98, pp. 13763–8, 12 2001.
[6] D. Lahat, T. Adali, and C. Jutten, “Multimodal data fusion: An overview of methods, challenges, and prospects,” Proceedings of the IEEE, vol. 103, no. 9, pp. 1449-1477, Sep. 2015.
[7] S. T. Shivappa, M. M. Trivedi, and B. D. Rao, “Audiovisual information fusion in humancomputer interfaces and intelligent environments: A survey,” Proceedings of the IEEE, vol. 98, no. 10, pp. 1692–1715, Oct 2010.
[8] B. Rivet, W. Wang, S. M. Naqvi, and J. A. Chambers, “Audiovisual speech source separation: An overview of key methodologies,” IEEE Signal Processing Magazine, vol. 31, no. 3, pp. 125–134, May 2014.
[9] A. Patti, A. Blanchard, M. Cardinaux, and P. Gaulier, “Gain-field modulation mechanism in multimodal networks for spatial perception,” in 2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012), Nov 2012, pp. 297–302.
[10] L. Fiack, N. Cuperlier, and B. Miramond, “Embedded and real-time architecture for bio-inspired vision-based robot navigation,” J. Real-Time Image Process., vol. 10, no. 4, pp. 609-722, Dec. 2015. [Online]. Available: https://doi.org/10.1007/s11554-014-0391-9
[11] S. Braun, D. Neil, J. Anumula, E. Ceolini, and S. Liu, “Attention-driven multi-sensor selection,” in 2019 International Joint Conference on Neural Networks (IJCNN), July 2019, pp. 1–8.
[12] D. Zhao and Y. Zeng, “Dynamic fusion of convolutional features based on spatial and temporal attention for visual tracking,” in 2019 International Joint Conference on Neural Networks (IJCNN), July 2019, pp. 1–8.
[13] A.-H. Tan, B. Subagdja, D. Wang, and L. Meng, “Self-organizing neural networks for universal learning and multimodal memory encoding,” Neural Networks, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0893608019302370
[14] Y. Zhang, Z. Wang, and J. Du, “Deep fusion: An attention guided factorized bilinear pooling for audio-video emotion recognition,” in 2019 International Joint Conference on Neural Networks (IJCNN), July 2019, pp. 1–8.
[15] M. Turk, “Multimodal interaction: A review,” Pattern Recognition Letters, vol. 36, pp. 189 – 195, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167865513002584
[16] C. Debes, A. Merentitis, R. Heremans, J. Hahn, N. Frangiadakis, T. van Kasteren, W. Liao, R. Bellens, A. Pizurica, S. Gauthama, W. Philips, S. Prasad, Q. Du, and F. Pacifici, “Hyperspectral and lidar data fusion: Outcome of the 2013 grss data fusion contest,” IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 7, 05 2014.
