Unsupervised Cognition

Alfredo Ibias*, Hector Antona, Guillem Ramirez-Miranda and Enric Guinovart

Avatar Cognition

Barcelona, Spain

Email: {alfredo*, hector, guillem, enric}@avatarcognition.com

Eduard Alarcon

Universitat Politècnica de Catalunya - BarcelonaTech

Barcelona, Spain

Email: eduard.alarcon@upc.edu

* Corresponding author

Abstract—Unsupervised learning methods have a soft inspiration in cognition models. To this day, the most successful unsupervised learning methods revolve around clustering samples in a mathematical space. In this paper we propose a state-of-the-art, primitive-based, unsupervised learning approach for decision-making inspired by a novel cognition framework. This representation-centric approach models the input space constructively as a distributed hierarchical structure in an input-agnostic way. We compared our approach with both current state-of-the-art unsupervised learning classification, and with current state-of-the-art cancer type classification. We show how our proposal outperforms previous state-of-the-art. We also evaluate some cognition-like properties of our proposal where it not only outperforms the compared algorithms (even supervised learning ones), but it also shows a different, more cognition-like, behaviour.

Index Terms—Unsupervised learning, Incremental learning, Cognitive systems, Explainable AI, Computational intelligence.

I. INTRODUCTION

Unsupervised learning is a huge field focused on extracting patterns from data without knowing the actual classes present in it. Due to this particularity, the field is full of methods that cluster data based on its mathematical representation. This hampers their applicability to data whose mathematical relationships do not directly correlate with its cognitive relationships, relationships that cognitive agents (like humans) find between the data. For example, the MNIST dataset has a clear cognitive relationship between its different samples: the number they represent. However, when transformed into numerical values for clustering, their relationships fade out in favour of relationships between their encodings, that do not necessarily correspond with the cognitive ones.

In this field, there are multiple algorithms that focus on the unsupervised classification problem. However, due to their soft inspiration in cognition models, most of them address the problem from an optimisation perspective. This approach requires building a mapping between any input and a valid output (ideally, the best output), and thus it is dividing the input space into subspaces. In that regard, the representations

become spatial, in the sense that the classes are represented by subspaces of an infinite space, independently of the similarity between the inputs that fall in that subspace.

In contrast, novel theories about how the brain works propose that the brain models the world in a constructive way, that is, it generates constructive representations of the world [1]–[3]. A constructive representation would be an abstraction or archetype of a class, in the sense that, it would be a representation to which any (or at least some) elements of the class are similar to. This implies that, to assign a class to an input, it has to be similar enough to one of the already learned representations and, if it is not similar enough to any of them, it can not be classified. Mathematically speaking, the difference between both approaches is that the first, traditional one focuses on splitting a representation space, and the second, novel one focuses on building a set of representations.

To empirically evaluate this new approach, we need to develop a new unsupervised learning algorithm. To develop such algorithm, we decided to follow a novel cognition framework [4] that is based in the previously mentioned theories of the brain. This framework presents the *Self-Projecting Persistence Principle* (SPPP), that defines how latent information is present in reality, how it persist in time, and how it projects itself to the world through manifestations. As such, this framework presents the basic cognition task as building abstractions based on manifestations captured from latent information, with the goal that such abstractions approximate the original latent information. To fulfil such task, the framework proposes to process manifestations into constructive representations through a primitive-based processing. Finally, this primitive-based processing is scaled to build a whole Cognitive Architecture.

Our aim in this paper is not to develop the full Cognitive Architecture, but just the Perception [5] part. This Perception should recognise inputs without needing a label, and hence it is equivalent to an unsupervised learning algorithm. The previously mentioned cognition framework [4] defines some requirements for developing such an algorithm: it has to be input-agnostic, primitive-based, scalable, and representation-centric. With these requirements, in this paper we propose a

novel unsupervised learning algorithm for decision-making we call *Cluster*. We expect this algorithm to be a building block towards a full cognition algorithm based on the previously mentioned cognition framework.

In this paper we present the fundamentals of the Cluster as well as one of its modulators: the *Spatial Attention* modulator. This modulator will auto-regulate the spatial discriminability of the algorithm. Additionally, we developed our proposal to be transparent and explainable, as it is desirable that any solution can describe its representations and explain its decisions. Finally, a perk of focusing on generating constructive representations is that our algorithm is able to state if a new input does not correspond to any previously seen pattern, that is, it can say "I do not know".

We compared our proposal to the main unsupervised classification algorithms: K-Means for tabular data and Invariant Information Clustering (IIC) [6] for image data. We compared it with different configurations of K-Means and IIC and for four different static datasets (two tabular and two image datasets), for an unsupervised classification task. The results show a clear advantage of our proposal, being able to deal with both tabular and image data with a decent performance. We also performed a sate-of-the-art comparison with a medical dataset, in which we beat the current state-of-the-art for classifying cancer types. Finally, we performed some experiments to evaluate cognition-like properties. In this case we compared our proposal to not only K-Means and IIC, but also the K-NN clustering supervised method. The comparison consisted on recognising MNIST digits even when removing random pixels. In that experiment there is a clear advantage of our proposal over the compared algorithms that shows how building constructive representations produces a different behaviour, and thus has the potential to have cognition-like properties. Given these results we conclude that our proposal is a state-of-the-art disruptive unsupervised learning algorithm for decision-making, with different, more promising properties than traditional algorithms.

The rest of the paper is organised in a related work resume at Section II, a proposal description in Section III, an empirical evaluation at Section IV, a discussion in Section V, a threats to validity analysis at Section VI, and a resume of the conclusions and future work at Section VII.

II. RELATED WORK

There are multiple algorithms for unsupervised learning developed along the years, from generic clustering algorithms like K-Means [7], to more specific, usually Artificial Neural Network-based, algorithms that deal with only one task. In this second category we can find algorithms that deal with topics as unrelated as representation learning [8], [9], video segmentation [10], speech recognition [11] or community detection [12]. However, none of them try to build constructive representations, but instead they divide a mathematical representation of the input space into clusters that represent the different classes present in the input space.

Among these clustering algorithms, there are few that stand out, specially for the task of unsupervised classification.

One of them is K-Means due to its performance clustering tabular data. This algorithm clusters the samples based on their closeness in the mathematical space of their encodings. Another one is Invariant Information Clustering (IIC) [6] due to its performance clustering images. This algorithm takes an image, transforms it with a given transform, and then run both of them over two Artificial Neural Networks with the goal of learning what is common between them. To that effect, it aims to maximise the mutual information between encoded variables, what makes representations of paired samples the same, but not through minimising representation distance like it is done in K-Means. In any case, both algorithms stand out due to their performance in their respective domains, but none of them is able to obtain good accuracy across domains. Thus, we will use them as baseline for comparison purposes, even though they cannot be applied in all cases.

Finally, regarding brain-inspired methods that try to model the input space, the only research we are aware of is the Hierarchical Temporal Memory [13] (HTM) and SyncMap [14], although they are algorithms suited for learning sequences instead of static data, and HTM is not unsupervised. Thus, as far as we are aware, ours is the first proposal of a brain-inspired, primitive-based, unsupervised learning algorithm for modelling static data.

III. THE PROPOSAL

Our proposal, based on the novel cognition framework presented at [4], is composed by: an Embodiment, a Cluster, and a Spatial Attention modulator. The goal of the Embodiment is to transform the input space into Sparse Distributed Representations (SDRs), the goal of the Cluster is to process those SDRs and model the input space generating constructive representations, and the goal of the Spatial Attention modulator is to auto-regulate the Cluster.

A. The Embodiment

The goal of our unsupervised learning algorithm is to model the input space generating constructive representations. To do so, it requires a representation-oriented universal data structure. Recent research has shown that such data structure is Sparse Distributed Representations [13], [15] (SDRs), which allows for input-agnostic representations of inputs independently of their data type. This has been proven to be the actual way in which the brain processes its inputs [13], [16]. Thus, our algorithm will work only with SDRs.

To translate inputs to SDRs we need an encoder architecture. To interpret the SDRs the Cluster generates we need a decoder architecture too. Both architectures conform the *Embodiment* of the Cluster, as displayed in Figure 1. In our case, as in our experiments we only explore tabular and image datasets, we only present four kinds of encoder decoder pairs: one for float point numbers, one for categorical data, one for grey images, and one for colour images. All these encoders are lossless and thus allow us to recover the encoded data.

The grey images translation to SDR is straightforward: a grey image's SDR is a flattened version of the image (in which each pixel is a dimension) with the values normalised to be
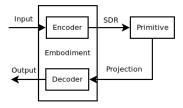
Fig. 1. Global Schema

between 0 and 1. In the case of colour images, we perform the same transformation to each one of the RGB channels and we concatenate their SDRs to form the image SDR.

For floating point numbers their translation to SDR is a bit more nuanced. We take the input space of the number (that is, the possible values it can take) and divide it into bins. Those bins will be the dimensions of the SDR. Then, each number will fall into one of those bins. However, in order to allow some overlap between numbers (what is fundamental for finding similarities using the Cluster), we also activate as many bins around the number bin as another parameter we call *bin overlap*. With this, the SDR is a long list of zeros and some ones around the bin where the number falls. By default, we use an overlap of 10% of the number of bins, that by default is set to 100 bins. In the case of categorical data we create one bin per category and set the overlap to 0%, following a one-hot encoding.

Having these representations, we define the SDR representation of a tabular input as a concatenation of the SDR representations of each entry, adjusting the indices to put one entry representation after another. Using this same methodology, we can compose multiple input types into one SDR. Although this is not the goal of this paper, these compositions could potentially help our algorithm to deal with datasets in which the input has many different types (for example, keyboard keys and video, like the recently released MineRL dataset [17]), although proving that would be matter of future work.

Here it is important to remark that, due to the transformation of any input and any output into SDRs, the algorithm always deals with the same data structures, and thus it is optimised to learn them independently of what they represent. Moreover, this makes our algorithm input-agnostic, as any kind of data is potentially transformable into an SDR.

B. The Cluster

Once we have an SDR representation of the inputs (and a way to recover the values from the SDR representation), we need to process them. To that end, and following the requirements outlined at Section I, we need a primitive-based processing. We defined a basic primitive we called *Footprint*, and we defined its scalability grouping multiple Footprints into a set we called *Cell*, and grouping multiple Cells into a hierarchy we called *Cluster*, following the terminology present at [4].

1) The Footprint: Our most basic processing unit, that is, our primitive, is the Footprint. A Footprint is an internal representation with two basic functions. A Footprint contains a data SDR, and can contain (for evaluation purposes only) a

---

Algorithm 1 Footprint Update (all ops are element-wise)

Require: FP: a Footprint, In: an input
1: $fp \leftarrow FP.SDR \quad inp \leftarrow In.SDR$
2: $n \leftarrow FP.N$ {Recall #inputs mixed into the Footprint}
3: $tmp \leftarrow fp * n$
4: $tmp \leftarrow tmp + inp$
5: $FP.SDR \leftarrow tmp/(n+1)$
6: $FP.N \leftarrow n+1$

Algorithm 2 Footprint Activation (all ops are element-wise)

Require: FP: a Footprint, In: an input
1: $fp \leftarrow FP.SDR \quad inp \leftarrow In.SDR$
2: $FP.ARCHETYPE \leftarrow (inp + fp)/2$
3: $FP.PROJECTION \leftarrow (inp * 2) - fp$
4: **return** $FP.ARCHETYPE, \ FP.PROJECTION$

Fig. 2. (left) An example of Footprint: the combination of the 1's of the MNIST dataset.
(right) An example of Cell: the Footprints of the 60,000 samples of the MNIST dataset.

metadata SDR (i.e. an SDR representing a label). A Footprint also has an *updating function* and an *activation function*. The updating function modifies the data and metadata SDRs when necessary mixing the Footprint SDRs with an external SDR (Algorithm 1), while the activation function computes the *Archetype* and *Projection* of such Footprint (Algorithm 2). An example of a Footprint is displayed at Figure 2 left.

The Archetype and the Projection are the two outputs of a Footprint after processing an input, and they are derived from the SPPP [4]. The Archetype aims to be an abstracted version of the input, while the Projection aims to be a concreted version of an Archetype. Both are the manifestations (and thus self-projections) of the Footprint at different perspectives, and they allow for the Footprint to be connected to other Footprints.

Finally, as it is clear from Algorithm 1, our approach to build constructive representations consists on merging similar inputs to build the final representation. This is an aggregative approach to abstractions that is not very common on the field, but whose benefits will be clear in the following sections.

2) The Cell: The main limitation of Footprints is that they can only store one representation. Thus, to be able to have multiple representations, we needed to create multiple Footprints. To organise them, we grouped them inside a structure called Cell, that coordinates the Footprints to avoid redundancies. Whenever a Cell receives an input, it has to decide which Footprint will process it to produce the outputs, and it does so trough similarity: the most similar Footprint is the one that is activated, and thus the one that will process the input, and whose Archetype and Projection will be the Cell's Archetype and Projection. Thus, a Cell contains a set
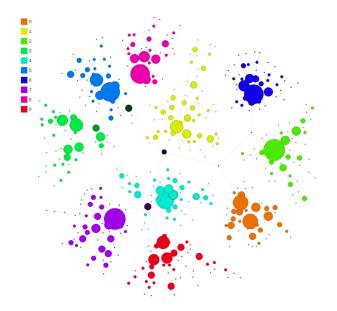
Fig. 3. An example of Cluster: the Cells of the hierarchy present in the 60,000 samples of the MNIST dataset. The Seed Cell is the central black node.

of Footprints and a threshold common to all its Footprints. This threshold is used to decide whether an input is similar to an existing Footprint or not, and will be deeply defined in the next section. An example of a Cell is displayed at Figure 2 right.

To decide if a new input is considered similar to an existing Footprint, we use a Similarity Function to get an score of the similarity of both SDRs, and if that score is over the Cell's threshold, then both the input and the Footprint are considered similar to each other. When there are more than one Footprint with a similarity over the Cell's threshold, we consider similar only the one with higher similarity. This similar Footprint is then the *active* Footprint.

To compute the previously mentioned similarity score between SDRs we need a *Similarity Function* that compares SDRs. This Similarity Function should take two SDRs and return a similarity value stating how similar we consider them to be. The specific Similarity Function we developed for this paper is a variation of the euclidean distance, but we also tested using the euclidean distance between vectors and the differences in results are minimal. An important remark here is that the Similarity Function should use only the data part of both SDRs to compute the similarity in order for our method to be fully unsupervised, leaving the metadata part outside of any decision making.

3) The Cluster: The core of our algorithm revolves around properly building and organising Footprints. To organise them, they are grouped inside Cells, but a Cell only allows us to have representations at the same level of abstraction. To have multiple levels of abstractions, we need to have multiple Cells organised in a tree-like hierarchical structure, and that is what we call Cluster. An example of Cluster is displayed at Figure 3.

At the beginning of training there is only one Cell in the Cluster (which we call Seed Cell), that starts with no Footprints. When new inputs are processed during training,

new Footprints are generated and the Seed Cell grows. And when an input is considered similar to an existing Footprint, then the need for a hierarchy arises. In such a case, a new Cell is created as a child of the Seed Cell, and it is associated with that Footprint. Thus, a Cell can have as many children as Footprints, and each child Cell has an associated parent Footprint.

With this organisation, a Cluster's goal is to organise Cells in a subset hierarchy, in which the Seed Cell contains the Footprints representing more inputs, and the leaf Cells contain Footprints representing only one input. The idea is that any child Cell will subdivide the subset of inputs represented by its parent Footprint. To that end, is fundamental the fact that each Cell has its own similarity threshold, what allows for a better discrimination policy as we will see in following sections.

4) Processing an Input: Now let us show how a new input is processed by the Cluster. To follow this description, a general schema of this algorithm is displayed at Figure 4. As we can see in the schema, our input processing method has three phases. The first phase is a filtering one, in which we look for the Footprint most similar to the input. The second phase is an abstracting one, in which we go up the Cluster generating an abstraction of the input using the Cell's Archetype. In this phase is where the Footprint update also happens. Finally, the third phase is a concreting one, in which we take the generated abstraction from the previous phase and filter it down the Cluster to find the Footprint most similar to the abstraction of the input using the Cell's Projection.

We start the filtering phase (left side of Figure 4) with an SDR that is a new input (this SDR contains both a data part and a metadata part). This input is then compared to all the Footprints present in the Cluster, and we select, from the Footprints that surpass their Cell's similarity threshold, the Footprint that gets the highest similarity. If a Footprint is selected, then we check if it has a child Cell, and if it has it, we create a new Footprint copy of the input in such Cell. If the Footprint does not have a child Cell, then one is created that contains two new Footprints: one is a copy of the input, the other is a copy of the parent Footprint. If no Footprint is selected, then a new Footprint copy of the input will be created in the Seed Cell and will be selected. If we are in evaluation mode, no Cell or Footprint are created in this phase.

In the abstracting phase (centre side of Figure 4), the selected Footprint is activated and executes its updating and activation functions in that order. The Archetype produced by the activation function is the output of the Cell in this phase. This output is then passed up as input to its parent Cell, where the parent Footprint is activated and executes its updating and activation functions with its child Cell's Archetype. This way a Footprint update is performed with an aggregation of the input and the active child Footprints. This process is repeated until it has been done in the Seed Cell, and the Archetype generated by the Seed Cell is considered the Archetype of the Cluster. If we are in evaluation mode, no Footprint is updated in this phase.

Finally, in the concreting phase (right side of Figure 4), the Cluster's Archetype is used to perform the inverse of the abstracting phase, but this time without learning. That is,
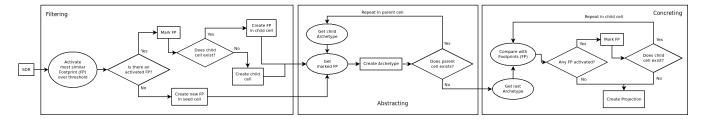
Fig. 4. The Cluster's Training Schema

the Archetype is provided to the Seed Cell as input to be compared against the existing Footprints, and if any Footprint is activated, that is, its similarity with the input is over the Cell's threshold, its activation function is executed. The Projection produced by the activation function is the output of the Cell in this phase. This output is then passed down as input to the corresponding child Cell to repeat the process. When there is no child Cell, or no Footprint is activated, the SDRs of the last activated Footprint are considered the Projection of the Cluster. If there is no Footprint activated in the Seed Cell then there is no Projection and an "I do not know" answer is provided.

The final Projection is supposed to be a very concrete version of the Cluster's Archetype, which is an abstraction of the original input. Then, the output of the whole method is this Projection, that includes a data SDR with a representation of the input and a metadata SDR with the additional information about the data SDR provided (like a label). This Projection is later processed by the decoder of the Embodiment to retrieve the final label of the input. A resume of the global algorithm is displayed at Figure 1. It is important to note that, if we are in evaluation mode, then the update function of the Footprints is not executed to not modify the internal representations, and no new Footprints neither Cells are created. Thus, if there is no Footprint activated in the Seed Cell in the third phase then there is no Projection and our method answers an "I do not know".

Here it is important to remark the relevance of the second and third phases: they allow generalisation. Without them, our algorithm would be a set of copies of the inputs, deciding the class by the most similar one. That is, it would be a convoluted implementation of the K-Nearest Neighbours algorithm with $K = 1$. Adding these phases allows us to build constructive representations, that in turn may be more similar to potential inputs than the already processed ones. Doing the updates in an intelligent way, we have more potential for generalisation. This also allows us to avoid overfitting in subsequent epochs, because the aggregated representation will never be equivalent to an individual input.

To end with the Cluster, it is important to note that, during training, the label is provided in the metadata SDR to be included in the Footprints, but it is not used for comparing Footprints, and thus it is avoided for building and organising the internal representations. Later, when in evaluation mode, the label of each Footprint would be a mixture of the labels of the inputs that updated such Footprint. When classifying, the returned label will be the strongest label of the Projection.

Algorithm 3 Spatial Attention

Require: FPS: a Cell's Footprints
Require: SA: a Cell's Spatial Attention
1: $SA \leftarrow 0.0$
2: $l \leftarrow length(FPS)$
3: $n \leftarrow 0$
4: $maxSim(fp) = max([similarity(fp, ot) \ for \ ot \ in \ FPS])$
5: $minSim(fp) = min([similarity(fp, ot) \ for \ ot \ in \ FPS])$
6: **for** i in $1 : l$ **do**
7: $\quad fp \leftarrow FPS[i].SDR$
8: $\quad SA \leftarrow SA + ((maxSim(fp) + minSim(fp))/2)$
9: $\quad n \leftarrow n + 1$
10: **end for**
11: $SA \leftarrow SA/n$
12: **return** SA

Finally, it is important to note how the Cluster is representation-centric, as the whole algorithm focuses on generating the right internal representations of the inputs, without explicit guide by the classification goal. We aim that these representations would produce the right Projections for classification, but we do not base our updating in how well they are classifying. Instead, we focus our learning on building representations that make sense and can be considered proper abstractions of the individual inputs that made them. This is useful to detect patterns, as different instances of the same pattern will eventually collide into one representation, building an abstraction of such pattern.

C. The Spatial Attention Modulator

In the previous Section, a threshold was used to decide if two SDRs were considered similar or not. This threshold can be set arbitrarily, but that would hamper the performance of the Cluster and would generate multiple extra parameters of the model. Thus, a way to automatise the threshold selection was needed, and that is the role of the Spatial Attention Modulator. The whole role of this modulator is to measure the variability of the input space of the Cell and dynamically set a similarity threshold. The algorithm we developed to set such threshold is the average of the mean similarities between the Footprints of the Cell, and its pseudo-code is displayed at Algorithm 3.

The rationale behind using this approach is that any input space has a certain variability, and the right threshold will be that one that sits in the middle of such variability. Such variability is unknown, but the Footprints have captured part of it in the form of aggregations. Thus, each aggregation (that is, each Footprint) represents a class and the average of the similarities between them is the "captured" variability.

TABLE I
CHARACTERISTICS OF THE EXPERIMENTAL SUBJECTS

Name	Type	# Features	# Samples
Wisconsin Breast Cancer	Tabular (Numerical)	30	569
Pima Indians Diabetes	Tabular (Numerical)	8	768
MNIST	Image (B&W)	28×28	$60,000 + 10,000$
CIFAR10	Image (RGB)	32×32	$50,000 + 10,000$
Cancer Type	Tabular (Numerical)	1500	398

This actually allows for child Cells to have higher thresholds than their parent Cells, as their input space are limited to those inputs that are similar to their associated parent Footprint. This generates an increase in discrimination power the further down the Cluster a Footprint is. In turn, this develops a distributed hierarchy, in which each Cell processes a different subdomain of the input domain.

IV. EMPIRICAL EVALUATION

To evaluate our proposal, we performed three different experiments: a comparison in classification task versus other unsupervised learning algorithms, a state-of-the-art comparison over a medical dataset, and a comparison in cognition-like capabilities versus other clustering algorithms. All the experiments were run in an Ubuntu laptop with an Intel Core i9-13900HX at 2.60GHz with 32 cores, 32Gb of memory, and a NVIDIA GeForce RTX 4060 with 8Gb of VRAM.

A. Experimental Subjects

Our experimental subjects for these experiments were five datasets: two tabular datasets full of numerical values, two image datasets, and a medical data dataset. Those datasets are the widely known Wisconsin Breast Cancer dataset [18], [19], Pima Indians Diabetes dataset [20], MNIST dataset [21], CIFAR10 dataset [22], and Cancer Type dataset (extracted from The Cancer Genome Atlas (TCGA) database [23]). The different properties of these datasets are presented in Table I.

We divided these datasets into a training set and a test set. For Wisconsin Breast Cancer and Pima Indians Diabetes we split the samples into 70% for the training set and 30% for the test set. In the case of the MNIST and CIFAR10 datasets, they come with 10,000 samples for test. Thus, we took as training set all the samples from the training dataset and the test set are those 10,000 test samples. The used Embodiments are the ones described in Section III-A, with an overlap of 10% for Wisconsin Breast Cancer and of 20% for Pima Indians Diabetes due to their respective characteristics.

B. Experiments

1) Learning Curves Experiment: The first experiment we performed aimed to test how well our proposal deals with a classification task compared to other unsupervised learning algorithms. For tabular data we compared to K-Means with as many centroids as labels, and with the number of centroids that the elbow method [24], [25] proposes. To evaluate its classification power, each cluster was assigned the label that was most repeated between the training elements of that cluster. In the case of our proposal, the label selected is the one

associated to the Projection. When comparing over the image datasets (MNIST and CIFAR10), the Invariant Information Clustering (IIC) algorithm was computed. In this case, the IIC algorithm was setup with the recommended parameters set by the authors for each dataset, and we compared different number of epochs (1, 10, 100) because we could not try the author recommended number of epochs ($3,200$ for MNIST and $2,000$ for CIFAR10) or any higher number of epochs due to resource constraints.

To compare these algorithms, we executed them over the experimental subjects computing the learning curve. That means, we trained the algorithms with the first 150 samples of the training set, we evaluated them with the samples used to train, and then we tested them over the whole test set. Then we trained them with the first 151 samples of the training set, evaluated them with the samples used to train, and tested them over the whole test set again, and so on and so forth. We repeated this process, adding 1 training sample each time, until the whole training set was used for training. We display the resulting learning curves in Figure 5. Due to the size of the datasets, when computing the image datasets results we executed the experiments for the first 200 samples, from then on each 100 samples until the $2,000$ sample, from then on each $1,000$ samples until the $10,000$ sample, and from then on we computed the result with the whole dataset.

The results of this experiment clearly show that our proposal is a better option for unsupervised classification. As we can observe, for tabular data our alternative is on par with K-Means for the Pima Indian Diabetes dataset (loosing by a 1.3%) and for the Wisconsin Breast Cancer dataset (winning by a 1.17%). When we move to image data, we can observe how our proposal is on par with IIC (loosing by 1.75% for MNIST with 100 epochs, but winning by 6.05% for CIFAR10 and by 7.07% for MNIST with 10 epochs).

We want to explicitly remark the fact that our proposal is able to obtain very good accuracies with fewer samples. For contrast, IIC needs around $1,700$ training samples to obtain an stable accuracy over 70% in test MNIST, while our proposal needs less than 200 samples. Moreover, our proposal does not need multiple epochs to obtain such results: it only goes through the training samples once, although more epochs also improve results (as shown by the "10 epochs" lines). Finally, if we compare with IIC with only 1 training epoch, then IIC is not able to overcome our proposal in any scenario, what shows the performance improvement and data efficiency of our approach.

2) State-Of-The-Art Experiment: Now, for our state-of-the-art experiment, we compared our algorithm over the Cancer Type dataset with the state-of-the-art methods evaluated at [26]. We performed exactly the same experiment: cancer type classification with five-fold cross-validation. Here being unsupervised was not a requisite, but even with our unsupervised approach we managed to get the results displayed at Table II, where it is clear how our proposal outperforms the other algorithms (except for the F1 macro measure).

This experiment is crucial to show how our proposal can produce state-of-the-art results in certain scenarios, even when competing against Artificial Neural Network-based models. It
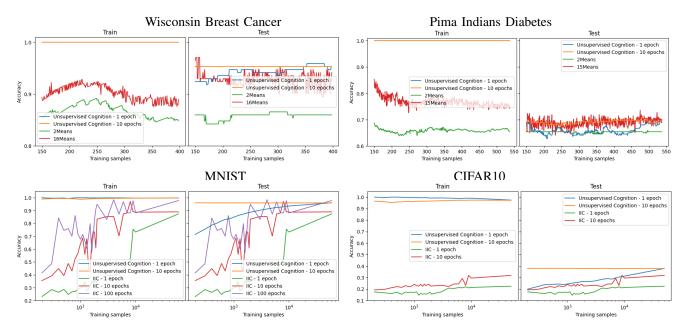
Fig. 5. Learning curves comparison for the different datasets

TABLE II
RESULTS OF MULTI-CLASS CLASSIFICATION BETWEEN CANCER
SUB-TYPES. OTHER METHODS RESULTS ARE FROM [26]

Model	Accuracy	F1 weighted	F1 macro
GCN [27]	0.73	0.721	0.525
GAT [28]	0.733	0.725	0.552
MOGONET [29]	0.712	0.717	0.614
MVGNN [26]	0.735	0.725	**0.636**
Unsupervised Cognition	**0.746**	**0.737**	0.513

Fig. 6. Distortion curves comparison for the different clustering algorithms

also shows how it can be used in real-world scenarios and not only in toy examples like the ones used for the learning curves experiment.

3) Cognition-like Capabilities Experiment: Finally, in our last experiment we wanted to explore the cognition-like capabilities of our proposal, compared to other clustering algorithms, using noise distortion [30]. To that end, inspired by [31], we devised an experiment using the MNIST dataset that consists on training the algorithms with the first $10,000$ samples of the training set, and then take the $10,000$ samples from the test set and start taking out pixels. That is, for different percentages (from 0% to 100% with a step of 2%), we remove that percentage of pixels (that is, we set them to black) from all the samples of the test set. Then, we evaluate all the algorithms over that test set and compute both the accuracy curve and the area under such curve. We did the same experiment also using the $10,000$ train samples, in order to also evaluate such distortion curve over the already experienced samples. The selected clustering algorithms are: our proposal, our proposal capped to have only 1 Cell, K-Means with 10 centroids, K-Means with 105 centroids, IIC with 1 epoch, IIC with 100 epochs, K-NN with 11 neighbours and K-NN with 1 neighbour. We display the resulting distortion curves at Figure 6.

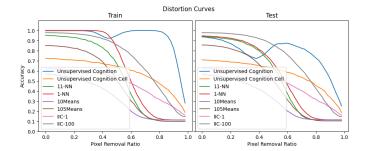The idea behind this experiment is that, even after removing

some pixels from an image, humans are able to recognise numbers. Moreover, if given some specific pixels, and after being told that such pixels represent a number, humans are able to fill in the number. Thus, we understand that recognising and/or reconstructing numbers is a capability of cognitive systems, one that we would desire in any Artificial Intelligence algorithm. Therefore, the goal here is to analyse how well each algorithm is able to recognise and reconstruct numbers from a set of pixels. As the pixels are removed at random, it is expected that after some removal percentage not even humans are able to recognise them, but the more pixels are removed, the better the concept of a number is understood if the number is correctly recognised. Thus, this experiment is expected to set a difference between those algorithms that have an optimisation approach and those that have a modelling one.

As we can observe, our proposal has better distortion curves than any other alternative. In numbers, our proposal obtains an Area Under the Curve (AUC) of 92.60 over train and of 76.23 over test. This AUC is way ahead of the next best one, obtained by the IIC with 100 epochs, that obtains an AUC for both train

and test of 72.28. Moreover, as we can observe in Figure 6, the other tested methods have a steep descend when there is more than $50 - 60\%$ of pixels removed, while our proposal keeps getting high accuracy until the very end, when it is almost impossible to recognise the numbers even for a human (around an 80% of pixels removed). This contrast allows us to conclude that the behaviour presented by our proposal is fundamentally different than the behaviour of the alternatives.

We understand that this is the effect of developing the algorithm around the idea of modelling the input space generating constructive representations, as this is the fundamental difference between the algorithms. This effect also implies that our algorithm is finding a different kind of relationships between the samples than the pure numerical or pattern based ones, showing cognition-like properties. Analysing it more in deep, we think that one of the keys of this behaviour is the discriminative hierarchy of patterns. This hierarchy comes from the subdivision of the input space into subspaces through the automatic replication of the Cells, and it allows for a more robust representation of concepts, and thus a better adaptation to noise. Having different levels of representations allows for a better matching between noisy samples and the internal representations, as they can be more similar to some of the intermediate representations than to the lower, literal ones.

C. Ablation Studies

In this Section we analyse the effects of the different parameters of our proposal. Let us start by stating that the main "parameter" of our proposal is the Embodiment. In this paper we have presented very basic Embodiments, and our algorithm works decently well with them. However, a fine tuned embodiment can cause huge increases in performance. For example, during our experiments with the Pima Indian Diabetes dataset we discovered that our initial embodiment (with a 10% overlap) did not obtain the best results. Thus, after testing multiple overlaps we settled in the 20% overlap. In general, for numerical data, the overlap between numbers is a fundamental parameter, because usually with a 0% overlap the performance is low, then it quickly raises with a small overlap and eventually falls down when the overlap is too big.

Other "parameters" of our proposal are the similarity and Spatial Attention functions. We presented the ones that we have discovered that produce the best results, after trying a lot of alternatives like the similarity between the input and the aggregation of the Footprints of the Cell for the Spatial Attention function or the Jaccard distance for the similarity function. Furthermore, we understand that alternative functions can be developed with the potential to improve furthermore the results, but looking for those improved functions is matter of future work.

Finally, we would like to reflect on the idea that our proposal does not have real parameters, at least not in the sense of the ones one can find in other Artificial Intelligence methods. There is no "magic" number that needs to be fine tuned, but instead all parameters of this kind correspond to the Embodiment and the input preprocessing. We consider this to be a huge advantage of our proposal, as this allows it to be used with much less required knowledge and expertise.

V. DISCUSSION AND LIMITATIONS

In this Section we would like to discuss the transparency and explainability of our algorithm, its capability of saying "I do not know", and its limitations.

Regarding transparency and explainability, it is fundamental to note that, as our algorithm has an internal hierarchical organisation of Sparse Distributed Representations (SDRs), it is possible to recall how our algorithm decided which label corresponds to the input. To that effect, we need the decoder from the Embodiment to transform the internal SDRs into understandable outputs. Thus, we can interpret any decision as a filtering from the Seed Cell, based on its Footprints, and down the hierarchy until the last Footprint that was activated. Then, its representation is the Projection, and the strongest label of that Projection is the selected label.

Regarding the capability of our algorithm to say "I do not know", it is easily derived from our threshold setup. If a new input does not surpass the threshold for any Footprint, that is, its similarity with each one of the Footprints in the Cluster is lower than their associated thresholds, then our algorithm returns a value stating it cannot associate that input to any knowledge it has learned. This is in fact used during training to generate new Footprints. Moreover, that answer is not only an "I do not know the label", but it actually means that it does not have a model for such input, so it can not return any Projection of it neither. This is an important and novel feature in an unsupervised learning algorithm. Its importance lies in the fact that saying "I do not know" ensures the user understands that the algorithm was not trained to recognise the pattern that was given, instead of falsely providing an answer and hallucinating [32], [33].

Finally, regarding the limitations of our proposal, its main one is the high memory costs involved compared to other alternatives due to the storage of a huge number of SDRs. We are aware that this limitation can hamper its scalability and applicability over very huge datasets and we are working in ways to diminish it, from developing growth inhibition and death mechanisms for the Footprints and Cells, to improving our Embodiments to generate smaller SDRs.

A secondary but also important limitation is the fact that our proposal is not an optimisation method. This implies that its focus is not to generate the best answer, or to cluster in the best way possible, like other algorithms. Instead, it is focused on building meaningful representations, that are useful to represent the input space, and we expect that this focus will produce, incidentally, a good classifier. This in turn hampers our classification capabilities, and thus our results, but anyway we managed to obtain the good results presented in this paper.

VI. THREATS TO VALIDITY

In this section we discuss the possible threats to the validity of our results. The first kind are the threats to internal validity, that can explain our results due to uncontrolled factors. The main threat in this category is the possibility of having a faulty code. To reduce this threat we have carefully tested each piece of code used in our experiments and developed unit tests for them, and we have relied on widely tested libraries like

scikit for the K-Means and K-NN algorithms, and the authors implementation for IIC. Another threat in this category is the impact of randomisation in the comparison results. As our proposal is fully deterministic, randomisation does not affect it, and any subsequent run arises the exact same values. Thus, for the compared methods we ran them multiple times and provided the results from the best execution. Finally, the last threat in this category is the use of unsupervised learning algorithms for a classification task, task usually associated to supervised learning ones. We are aware that the K-Means algorithms was not developed for classification tasks, but at the same time we needed a way to compare the results of our algorithm with theirs, having into account that characterising the "clusters" that ours generates is not viable. Thus, we decided to use classification tasks, as they allowed to compare how well the algorithms detected the underlying relationship between the inputs, even if none of them uses the label for training purposes.

The second kind of threats are the ones to external validity, that hamper the generality of our results to other scenarios. In our case the only threat in this category is the small scale experimental setup, having compared against two methods over four datasets. However, we have performed small comparisons with other methods and datasets too, and obtained similar results. Moreover, we have included also a state-of-the-art comparison for a medical dataset to show the capability of our proposal of obtaining state-of-the-art results beating multiple other algorithms, include Artificial Neural Network-based ones. Nonetheless, the comparison of our proposal to less well known algorithms is matter of future work.

Finally, the last kind of threats are the construction validity ones, hampering the extrapolation of our results to real-world scenarios. In this case, the range of possible scenarios is potentially infinite, and this threat cannot be fully addressed, but as explained before, the exploration of how our proposal behaves in other scenarios is matter of future work.

VII. Conclusions

Current well known unsupervised learning methods have a dim capability of extracting cognition-like relationships due to their optimisation oriented setup. The biggest exponent of this field is K-Means, that clusters samples based only on the mathematical distance between them. In this paper we have proposed an alternative, input-agnostic, representation-centric, unsupervised learning algorithm for decision-making that extracts cognition-like relationships between samples through constructive representations.

Our proposal transforms the inputs into SDRs, and then generates an internal representation of those SDRs in order to later recall that representation when asked about the class of an specific input. We tested our proposal against K-Means and IIC for unsupervised classification tasks in four datasets, and show that our proposal is equivalent to them, even although it only process each sample once. Moreover, we have compared it with the state-of-the-art for identifying cancer types, and overcome them. Finally, we have evaluated how well it can discover cognition-like relationships compared to other clustering

algorithms, and we have found that it is better than the three main clustering algorithms: K-Means, IIC and K-NN. This is important because it means that our proposal does not only have a different, better behaviour than unsupervised learning algorithms, but also than supervised learning clustering ones.

As future work, we would like to explore how our proposal performs in other datasets and against other unsupervised learning algorithms, and perform an in deep analysis of the relevance of each "parameter" of our model. We would also like to develop new Embodiments for different input types, like sound. We would like to explore the extension of our algorithm with other modulators too, like a conditioning modulator that allows us to have a reinforcement learning-like algorithm, or a temporal modulator that allows us to process sequences. We would like to explore different algorithms to compute the similarity function or the spatial attention function too. Finally, we would like to extend our proposal with growth inhibition and death mechanisms for the Footprints and Cells, in order to reduce its memory costs.

References

[1] J. Hawkins and S. Blakeslee, *On Intelligence*. USA: Times Books, 2004.

[2] J. Z. Leibo, J. Cornebise, S. Gómez, and D. Hassabis, "Approximate hubel-wiesel modules and the data structures of neural computation," *CoRR*, vol. abs/1512.08457, 2015. [Online]. Available: http://arxiv.org/abs/1512.08457

[3] D. Yon, C. Heyes, and C. Press, "Beliefs and desires in the predictive brain," *Nature Communications*, vol. 11, no. 1, p. 4404, Sep 2020.

[4] A. Ibias, G. Ramirez-Miranda, E. Guinovart, and E. Alarcón, "From manifestations to cognitive architectures: A scalable framework," in *Artificial General Intelligence - 17th International Conference, AGI 2024, Seattle, WA, USA, August 13-16, 2024, Proceedings*, ser. Lecture Notes in Computer Science, vol. 14951. Springer, 2024, pp. 89–98.

[5] J. E. Laird, C. Lebiere, and P. S. Rosenbloom, "A standard model of the mind: Toward a common computational framework across artificial intelligence, cognitive science, neuroscience, and robotics," *AI Mag.*, vol. 38, no. 4, pp. 13–26, 2017.

[6] X. Ji, A. Vedaldi, and J. F. Henriques, "Invariant information clustering for unsupervised image classification and segmentation," in *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, 2019, pp. 9864–9873.

[7] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. 28, no. 2, pp. 129–136, 1982.

[8] F. Wang, H. Liu, D. Guo, and F. Sun, "Unsupervised representation learning by invariance propagation," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[9] Y. Shin, C. Tran, W. Shin, and X. Cao, "Edgeless-gnn: Unsupervised representation learning for edgeless nodes," *IEEE Trans. Emerg. Top. Comput.*, vol. 12, no. 1, pp. 150–162, 2024.

[10] N. Araslanov, S. Schaub-Meyer, and S. Roth, "Dense unsupervised learning for video segmentation," in *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, 2021, pp. 25 308–25 319.

[11] A. Baevski, W. Hsu, A. Conneau, and M. Auli, "Unsupervised speech recognition," in *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, 2021, pp. 27 826–27 839.

[12] J. Gao, J. Chen, B. M. Oloulade, R. Al-Sabri, T. Lyu, J. Zhang, and Z. Li, "Commgnas: Unsupervised graph neural architecture search for community detection," *IEEE Trans. Emerg. Top. Comput.*, vol. 12, no. 2, pp. 444–454, 2024.

[13] Y. Cui, S. Ahmad, and J. Hawkins, "The HTM spatial pooler - A neocortical algorithm for online sparse distributed coding," *Frontiers Comput. Neurosci.*, vol. 11, p. 111, 2017.

[14] D. V. Vargas and T. Asabuki, "Continual general chunking problem and syncmap," in *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*. AAAI Press, 2021, pp. 10 006–10 014.

[15] S. Ahmad and J. Hawkins, "How do neurons operate on sparse distributed representations? A mathematical theory of sparsity, neurons and active dendrites," *CoRR*, vol. abs/1601.00720, 2016. [Online]. Available: http://arxiv.org/abs/1601.00720

[16] P. Foldiak, "Sparse coding in the primate cortex," *The handbook of brain theory and neural networks*, 2003.

[17] W. H. Guss, B. Houghton, N. Topin, P. Wang, C. Codel, M. Veloso, and R. Salakhutdinov, "Minerl: A large-scale dataset of minecraft demonstrations," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*. ijcai.org, 2019, pp. 2442–2448.

[18] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[19] W. Wolberg, W. Street, and O. Mangasarian, "Breast Cancer Wisconsin (Diagnostic)," UCI Machine Learning Repository, 1995.

[20] J. W. Smith, J. E. Everhart, W. Dickson, W. C. Knowler, and R. S. Johannes, "Using the adap learning algorithm to forecast the onset of diabetes mellitus," in *Proceedings of the annual symposium on computer application in medical care*. American Medical Informatics Association, 1988, p. 261.

[21] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[22] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[23] J. N. Weinstein, E. A. Collisson, G. B. Mills, K. R. Shaw, B. A. Ozenberger, K. Ellrott, I. Shmulevich, C. Sander, and J. M. Stuart, "The cancer genome atlas pan-cancer analysis project," *Nature genetics*, vol. 45, no. 10, pp. 1113–1120, 2013.

[24] R. L. Thorndike, "Who belongs in the family?" *Psychometrika*, vol. 18, no. 4, pp. 267–276, Dec 1953.

[25] F. Liu and Y. Deng, "Determine the number of unknown targets in open world based on elbow method," *IEEE Trans. Fuzzy Syst.*, vol. 29, no. 5, pp. 986–995, 2021.

[26] Y. Ren, Y. Gao, W. Du, W. Qiao, W. Li, Q. Yang, Y. Liang, and G. Li, "Classifying breast cancer using multi-view graph neural network based on multi-omics data," *Frontiers in Genetics*, vol. 15, p. 1363896, 2024.

[27] X. Li, J. Ma, L. Leng, M. Han, M. Li, F. He, and Y. Zhu, "Mogcn: a multi-omics integration method based on graph convolutional network for cancer subtype analysis," *Frontiers in Genetics*, vol. 13, p. 806842, 2022.

[28] X. Xing, F. Yang, H. Li, J. Zhang, Y. Zhao, M. Gao, J. Huang, and J. Yao, "An interpretable multi-level enhanced graph attention network for disease diagnosis with gene expression data," in *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2021, pp. 556–561.

[29] T. Wang, W. Shao, Z. Huang, H. Tang, J. Zhang, Z. Ding, and K. Huang, "Mogonet integrates multi-omics data using graph convolutional networks allowing patient classification and biomarker identification," *Nature communications*, vol. 12, no. 1, p. 3445, 2021.

[30] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning (still) requires rethinking generalization," *Commun. ACM*, vol. 64, no. 3, pp. 107–115, 2021.

[31] M. Assran, Q. Duval, I. Misra, P. Bojanowski, P. Vincent, M. G. Rabbat, Y. LeCun, and N. Ballas, "Self-supervised learning from images with a joint-embedding predictive architecture," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*. IEEE, 2023, pp. 15 619–15 629.

[32] P. A. Ortega, M. Kunesch, G. Delétang, T. Genewein, J. Grau-Moya, J. Veness, J. Buchli, J. Degrave, B. Piot, J. Pérolat, T. Everitt, C. Tallec, E. Parisotto, T. Erez, Y. Chen, S. E. Reed, M. Hutter, N. de Freitas, and S. Legg, "Shaking the foundations: delusions in sequence models for interaction and control," *CoRR*, vol. abs/2110.10819, 2021. [Online]. Available: https://arxiv.org/abs/2110.10819

[33] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and P. Fung, "Survey of hallucination in natural language generation," *ACM Comput. Surv.*, vol. 55, no. 12, 2023.

Alfredo Ibias received B.Sc. degrees in Computer Science and in Mathematics from Complutense University of Madrid, Spain, and an M.Sc. degree in Formal Methods in Computer Science and a Ph.D. degree in Computer Science from the same university. For 4 years he did research around the development of AI methods for uncommon scenarios. Currently he is working as an AI researcher at Avatar Cognition. His main research area is the development of a general AI based on novel theories of the brain.

Hector Antona received B.Sc. degrees in Computer Science and in Telecommunications Engineering from Universitat Politècnica de Catalunya (UPC), Spain, and an M.Sc. degree in Advanced Telecomunications Technologies from the same university. Currently he is working as an AI researcher at Avatar Cognition. His main research area is the applicability of novel AI methods.

Guillem Ramirez-Mirandaz received B.Sc. degree in Computer Science from Universitat Politécnica de Barcelona, Spain. For 3 years he did research on performance analysis and optimisation of high-performance computing applications. Currently, he is working as a developer and researcher at Avatar Cognition and pursuing a B.A. degree in Philosophy at Universidad Nacional de Estudios a Distancia (UNED), Spain. His main research area is the development of a general AI based on novel theories of the brain.

Enric Guinovart received B.Sc. degree in Computer Science from Universitat Politècnica de Catalunya (UPC), Spain. He has been working in the industry for 20 years as AI consultant (among other roles). In 2018 he funded Avatar Cognition, where he currently works as co-CEO, CTO and CRO. His main research area is the development of a general AI based on novel theories of the brain.

Eduard Alarcon received the M.Sc. (National award) and Ph.D. degrees (honors) in Electrical Engineering from the Technical University of Catalunya (UPC BarcelonaTech), Spain, in 1995 and 2000, respectively. Since 1995 he has been with the Department of Electronics Engineering at the School of Telecommunications at UPC, where he became Associate Professor in 2000 and is currently full professor. Visiting professor at CU Boulder and KTH. He has co-authored more than 450 scientific publications, 6 books, 8 book chapters and 12 patents. He has been involved in different National, European (H2020 FET-Open, Flag-ERA, ESA) and US (DARPA, NSF, NASA) R&D projects within his research interests including the areas of on-chip energy management and RF circuits, energy harvesting and wireless energy transfer, nanosatellites and satellite architectures for Earth Observation, nanotechnology-enabled wireless communications, Quantum computing architectures and Artificial Intelligence chip architectures. He has received the GOOGLE Faculty Research Award (2013), SAMSUNG Advanced Institute of Technology Global Research Program gift (2012), and INTEL Doctoral Student Honor Programme Fellowship (2014). Professional officer responsibilities include elected member of the IEEE CAS Board of Governors (2010-2013) and Vice President for Technical Activities of IEEE CAS (2016-2017, and 2017-2018). Editorial duties include Senior founding Editorial Board of the IEEE Journal on Emerging topics in Circuits and Systems, of which he was Editor-in-Chief (2018-2019).