

Wireless On-Chip Communications for Scalable In-memory Hyperdimensional Computing

Robert Guirado^{†*}, Abbas Rahimi[†], Geethan Karunaratne[†], Eduard Alarcón^{*}, Abu Sebastian[†], and Sergi Abadal^{*}

[†]IBM Research – Zurich, Rüschlikon, Switzerland

^{*}Universitat Politècnica de Catalunya, Barcelona, Spain

Abstract—Hyperdimensional computing (HDC) is an emerging computing paradigm that represents, manipulates, and communicates data using very long random vectors (aka hypervectors). Among different hardware platforms capable of executing HDC algorithms, in-memory computing (IMC) systems have been recently proved to be one of the most energy-efficient options, due to hypervector manipulations in the memory itself that reduces data movement. Although implementations of HDC on single IMC cores have been made, their parallelization is still unresolved due to the communication challenges that these novel architectures impose and that traditional Networks-on-Chip and Networks-in-Package were not designed for. To cope with this difficulty, we propose the use of wireless on-chip communication technology in unique ways. We are particularly interested in physically distributing a large number of IMC cores performing similarity search across a chip, and maintaining the classification accuracy when each of which is queried with a slightly different version of a bundled hypervector. To achieve it, we introduce a novel over-the-air computing that consists of defining different binary decision regions in the receivers so as to compute the logical majority operation (i.e., bundling, or superposition) required in HDC. It introduces moderate overheads of a single antenna and receiver per IMC core. By doing so, we achieve a joint broadcast distribution and computation with a performance and efficiency unattainable with wired interconnects, which in turn enables massive parallelization of the architecture. It is demonstrated that the proposed approach allows to both bundle at least three hypervectors and scale similarity search to 64 IMC cores seamlessly, while incurring an average bit error ratio of 0.01 without any impact in the accuracy of a generic HDC-based classifier working with 512-bit vectors.

I. INTRODUCTION

Hyperdimensional computing (HDC) is an emerging computational framework and is based on the observation that key aspects of human memory, perception and cognition can be explained by the mathematical properties of hyperdimensional spaces comprising high-dimensional vectors known as hypervectors [1]. Hypervectors are defined as d -dimensional (where $d \geq 1,000$) (pseudo)random vectors with independent and identically distributed components. When the dimensionality is in the thousands, a large number of quasi-orthogonal hypervectors exist. This allows HDC to combine such hypervectors into new hypervectors using well-defined vector operations, such that the resulting hypervector is unique and with the same dimension. A number of powerful computational models are built on the rich algebra of hypervectors [2]–[5].

HDC has been employed in a range of applications such as cognitive computing [6]–[8], robotics [9], distributed com-

puting [10]–[12], communications [13]–[18], and in various aspects of machine learning. It has shown significant promise in machine learning applications that especially demand few-shot learning [19]–[23], in-sensor adaptive learning [24], [25], multimodal learning [26], [27], and always-on smart sensing [28]. By its very nature, HDC is extremely robust in the presence of failures, defects, variations, and noise, all of which are synonymous to ultra-low energy computation. It has been shown that HDC degrades very gracefully in the presence of various faults compared to baseline classifiers: HDC tolerates intermittent errors [29], permanent hard errors (in memory [30] and logic [31]), and spatio-temporal variations [32] in emerging technologies as well as noise and interference in the communication channels [15], [18]. These demonstrate robust operations of HDC under low signal-to-noise ratio and high variability conditions.

What these different HDC algorithms have in common is to operate on very large vectors, and therefore, are in need of architectures that handle such operations efficiently. For instance, HDC involves similarity searches across a set of stationary hypervectors in an associative memory, which are generally implemented in the form of dot-products. Due to this, in-memory computing (IMC) is a natural fit to HDC algorithms [32]. An IMC core departs from the von Neumann architectures which move data from a processing unit to a memory unit and vice versa by exploiting the possibility of performing operations (dot products, in our case) within the memory device itself [33]. This improves both the time complexity and the energy consumption of the architecture.

IMC systems have been proposed recently to execute HDC tasks using hypervectors as wide as 10,000-bit [32]. As further elaborated in Section II, IMC cores are capable of computing similarity searches through dot-products with unprecedented energy-efficiency, e.g., over $100\times$ energy saving compared to a digital accelerator [32]. However, the scaling of such architecture remains unclear due to the associated challenges. On the one hand, scaling up the architecture requires sharing a very large IMC core across many hypervectors—e.g., there will be a need to continually store and search over thousands hypervectors for representing novel classes in the incremental learning regime [19]—which poses a problem in terms of array impedances and programming complexity [34]. On the other hand, scaling out requires deploying multiple IMC cores to execute similarity searches in parallel. This implies

distribution and broadcasting hypervectors across a potentially large number of modules, which puts a large pressure on the system interconnect.

This paper focuses on the scaling out of IMC-based HDC systems and the interconnect challenge that comes with it. In highly parallel many-core systems, Networks-on-Chip (NoC) and Networks-in-Package (NiP) are typically used to interconnect the different processing elements and ensure a correct data orchestration. However, parallelizing several similarity searches for HDC is demanding, especially when it imposes all-to-one followed by one-to-all traffic patterns, a scenario for which conventional NoCs and NiPs suffer to provide a competitive performance. Hence, the interconnect becomes a bottleneck, severely limiting the scalability of the HDC architecture.

To address the scalability problem of IMC-based HDC architectures, in this paper we propose to use wireless communications technology. Wireless Network-on-Chip (WNoC) have shown promise in alleviating the bottlenecks that traditional NoC and NiP face, especially for collective traffic patterns and large-scale interconnection demands that are common in HDC [35]–[39]. To that end, WNoCs provide native broadcast capabilities. These properties are put in use for the proposed architecture, sketched in Fig. 1, with a novel approach that aims to answer the following question: *Given Q as a set of hypervectors that are superposed Over-The-Air (OTA), how could different physically distributed on-chip receivers reliably preform similarity search while each receiving a slightly different version of Q ?* To address it, we leverage the full electromagnetic knowledge of the chip package and engineer constellations to enable wireless OTA computations leading to a lightweight all-to-all concurrent communications at the chip scale. The resulting WNoC will be uniquely suited to the communication requirements of HDC operations while opportunistically bypassing the main limitations of wireless technology: the impact of relatively low aggregate bandwidth and high error rate are minimal thanks to the OTA approach and the inherent resilience of HDC algorithms to noise.

This paper makes the following three novel contributions. (i) For the first time, we use a wireless interconnect solution for HDC platform that allows scaling-out similarity search across multiple independent on-chip receiver modules. (ii) For the first time too, we enable more than one simultaneous transmitter to make use of OTA computation on a chip. (iii) We leverage a pre-characterization of the chip package to optimize OTA from multiple transmitters to multiple receivers. The proposed architecture is designed and evaluated at the electromagnetic level, demonstrating that it can support up to 64 receivers with 3 transmitters with an average bit error ratio (BER) of 0.01 and the maximum BER of 0.1, which do not have any impact in the accuracy of a generic HDC-based classifier operating with 512-bit hypervectors.

The rest of the paper is organized as follows. In Sec. II, we provide background on the topics of HDC, IMC, and wireless communications at the chip scale. In Sec. III, we motivate the problem by illustrating the scale-out of IMC-based

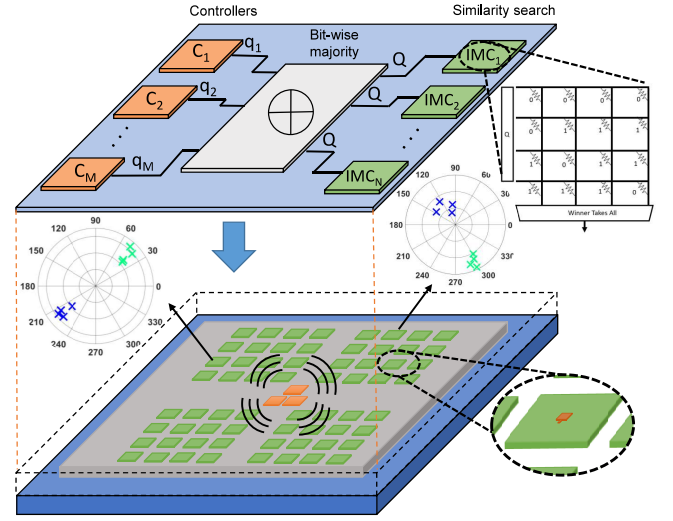


Fig. 1: Overview of the proposed many-core wireless-enabled IMC platform. Orange encoders map to our wireless TX, while green IMCs map to our wireless-augmented IMCs. Bit-wise majority operation maps to the wireless OTA computation.

HDC architectures and then propose the wireless solution. In Sec. IV, we depict the simulation methodology encompassing electromagnetic simulation, signal processing, and HDC-based learning. In Sec. V, we show the main results of the analysis. The paper is concluded in Sec. VI.

II. BACKGROUND

A. Hyperdimensional Computing

Here we focus on a variant of HDC models by making use of pseudo-random binary vectors of thousands of dimensions [1]. When using these binary hypervectors, it is easy to find nearly unlimited non-coincident quasi-orthogonal vectors with normalized Hamming distance close to 0.5. We call these random hypervectors atomic hypervectors. In classification tasks, one can further create an encoder to operate on these atomic hypervectors by binding, bundling (i.e., superposition), and permutation operations to obtain a composite hypervector describing an object or event of interest. The composite hypervectors, generated from various examples of the same class, can be further bundled together to create a single prototype hypervector representing a class. Particularly, the bundling operation for binary hypervectors is implemented as a logical bit-wise majority operation. The prototype hypervectors are stored in the associative memory.

In the inference stage, the query hypervectors of unknown objects/events are generated by following the same procedure as in the training stage. A query hypervector is later compared to the prototype hypervectors in the associative memory. Then, the chosen label is the one assigned to the prototype hypervector that has the highest similarity to the query vector. The robustness to failure is given by the spreading of information across thousands of dimensions. See [22] for more details.

B. In-memory Computing

IMC is a non von Neumann architecture that leverages the memory unit to perform in-place computational tasks, reducing the amount of data movement and therefore cutting down the latency and energy consumption associated with in-package communication [33]. That is, instead of fetching the data from the memory to the processing unit in order to carry out computations and store the results back to the memory, in IMC systems the operation is directly carried out in the computational memory, which requires less communication.

The latency produced by memory accesses is problematic in computing systems in general, but it can be more or less harmful depending on the particular application being executed, as it can limit the overall performance of the system. When this happens, and the memory accesses become the bottleneck, the term memory wall is commonly used, referring to the disparity between the processing speed and the ability of the memory to provide data to, or receive data from, the processing units. Several memory and architecture concepts have been designed and manufactured in the recent years to overcome these problems, such as high-bandwidth memory [40], 2.5D and 3D monolithic integration [41], interposers or hybrid memory cube [42]. However, from a complete architectural point of view, these are ad-hoc solutions that are not expected to solve the problem from the root, as the fundamental problem of moving large quantities of data from memory and back remains. Instead, the novel approach of IMC is being developed and appears as a promising candidate to overcome these challenges [33].

Resistance-based IMC cores, and more specifically those based on phase-change memory (PCM) devices, have recently shown promising results [43]. In a resistance-based IMC core, we can encode certain values as conductances of PCM devices placed in a mesh-like array. Then, by Ohm's law and Kirchhoff's law, a matrix-vector multiplication (MVM), essential to execute any machine learning algorithm, is as simple as tuning conductances to match the matrix values, inputting the vector as voltages from one side and finally reading the output currents from a perpendicular side.

Although IMC architectures are capable of executing various HDC operations [32], we are particularly interested in the similarity search in the associative memory. As shown in Fig. 2, since the prototype hypervectors P_i will be programmed in an IMC core, the similarity search through the dot product can be implemented as a MVM with the query hypervector Q as input vector. This allows performing a dot-product in $O(1)$ time complexity.

C. Wireless Network-on-Chip

NoCs are currently the *de facto* standard interconnects in modern multiprocessors due to their low latency and high throughput capabilities in systems with a few dozen processing cores. However, NoCs face significant challenges when scaling the architectures or when facing specific communication patterns such as broadcast or reductions. This has led to

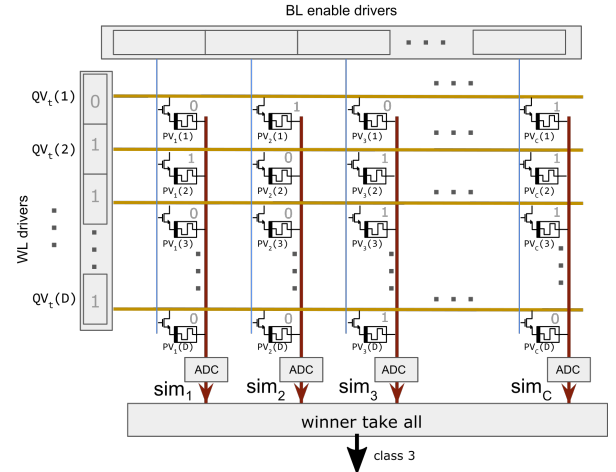


Fig. 2: Similarity search example in an IMC core. Since the prototype hypervector of the third column is the most similar one to the query vector Q , it will output more current than the others and its associated label will be chosen.

the point where systems are starting to be communication-bounded instead of computation-bounded. WNoCs have been introduced, among other alternatives, to overcome these issues. WNoCs are the result of augmenting cores or groups of cores with RF transceivers and antennas allowing them to communicate wirelessly through the chip package with all cores that are within range [44]–[46]. Even though this technology is still under development, proof-of-concept designs have been successfully implemented and tested [47].

Among the key advantages of WNoCs, one can find a natural support to broadcast communications, reduced latency, and an adaptive network topology [36], [39], [48], [49]. Hence, WNoCs can be especially advantageous if they are used to serve specific communication patterns that are very challenging to tackle using conventional NoCs [46]. This is of relevance in this work, as HDC algorithms being executed over IMC platforms make an intensive use of broadcast and reduction patterns, leading to important bottlenecks when scaled over traditional NoC/NiP platforms. In this case, the key strength of WNoCs lies on its use for broadcast communication, while it is in principle less suited to all-to-one reduction patterns. However, as we detail next, thanks to the proposed OTA computing solution, WNoCs become a perfect candidate to enabling the scalability of IMC-based HDC architectures.

III. TOWARDS WIRELESS-ENABLED SCALE-OUT HDC ARCHITECTURES

Although HDC has a great potential and IMC systems are used to execute it efficiently, the scaling of such systems, as essential as it is to satisfy the insatiable appetite of machine learning for computational resources, is still a pending matter. In architectural terms, IMC-based HDC systems can be scaled by either increasing the size of the IMC cores (scale-up) or by placing more cores in the system (scale-out).

On the one hand, scaling-up becomes complex as the required in-memory wire length blows up exponentially with

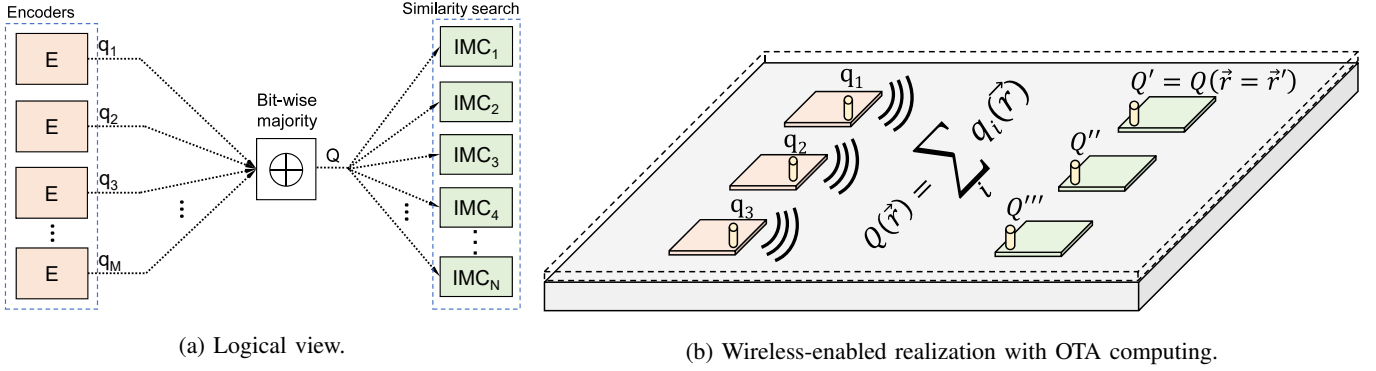


Fig. 3: Proposed scale-out approach of a HDC platform involving M encoders generating queries $q_1 \dots q_M$, the computation of a composite query Q via bit-wise majority, and N IMC cores performing similarity search over multiple copies of Q . In the wireless case, the IMC cores receive different versions of Q (Q' , Q'' , Q''') that are decoded minimizing the distance to Q .

the array size, leading to issues related to wire resistance and parasitic effects. Moreover, the complexity of weight programming also increases with the array size [34].

On the other hand, scaling-out is a technologically viable alternative. Fig. 3a shows a logical diagram of the desired scaled-out IMC architecture, capable of executing a HDC-based classifier. The M encoders at the left compute the different query hypervectors, which will be bundled later on through the majority operation. Each encoder can encode data from e.g., different sensory modalities [26], [27], or streaming channels [18]. This is highly desirable since by doing a bundling of M queries, we virtually increase the throughput by a factor of M . That is, we compress all the queries information in a single one instead of having M independent transmissions and redundant bundling at the similarity search cores. The N IMC cores, at the right of Fig. 3a, are in charge of comparing the composite query hypervector with all the prototype hypervectors they have stored, enabling the aforementioned scaling-out. By following this modular approach, a system as powerful as required by each application could be designed by varying M and N .

Challenges of wired scale-out. Notwithstanding, scaling out casts a significant pressure to the system interconnect. Firstly, the interconnection between the M encoders and a hypothetical circuit performing the bit-wise majority would result in heavy reduction M -to-1 traffic. Should the bundling operation be performed using a wired interconnect, we would have to add a centralized processing core with extra circuitry, which would not scale linearly with the number of encoders. Secondly, the interconnection between the bundling block and the N IMC cores follows a broadcast topology, which becomes slow and inefficient as N grows [36].

Even in the case of full co-integration of the encoders with specialized bundling circuitry and IMC cores, the system would need to provision a non-scalable amount resources. A lower cost modular alternative, proposed in other deep learning acceleration systems [50], is to build the architecture with specialized chiplets and to integrate them through an interposer. In this case, however, the interposer becomes a

bottleneck in terms of bandwidth and connectivity due to I/O pin limitations. This leads to multi-hop and serial-link schemes that add significant energy and latency per hop, i.e., ~ 1 pJ and ~ 20 ns [50], with hop counts typically scaling with \sqrt{N} for unicasts and with N for broadcasts [46].

In summary, wired scale-out of HDC platforms is challenging because: (i) the reduction (all-to-one) pattern generated by the bundling operation not only creates a communication bottleneck, but also acts as an implicit barrier; (ii) the broadcast (one-to-all) pattern of query distribution is inherently costly in chiplet-based systems; and (iii) both operations are sequential.

Proposed architecture. We tackle the three problems of wired scale-out at once by augmenting a many-core HDC platform with a WNoC. Fig. 3b shows the proposed WNoC implementation with M encoders augmented with wireless TXs and N IMC cores augmented with wireless RXs. The encoders broadcast, in a concurrent fashion and using a single channel, the different queries to be bundled. As a result of the wave propagation, each receiver will obtain a slightly different version of the superposition of all transmitted signals, which will be decoded using the channel state information, which is quasi-static and known a priori. Hence, the final majority result is known in the RXs per each TX bit combination. That is, we can pre-assign different decision regions that map the received superposed symbols to their logical majority per each RX, as illustrated in Fig. 4. See Sec. IV for more details.

In summary, the proposed architecture is built upon three key observations:

- *Given the controlled package scenario, OTA computing can be leveraged.* In particular, the majority operations required by the bundling of hypervectors can be performed over-the-air (OTA) with low error thanks to a pre-characterization of the channel.
- *The inherent broadcast nature of wireless communication allows to implement single-hop in-package transfers.* This, together with the OTA bundling, allows for a seamless parallelization of the similarity search over multiple associative memories at the chip scale while completely eliminating the communication bottleneck.

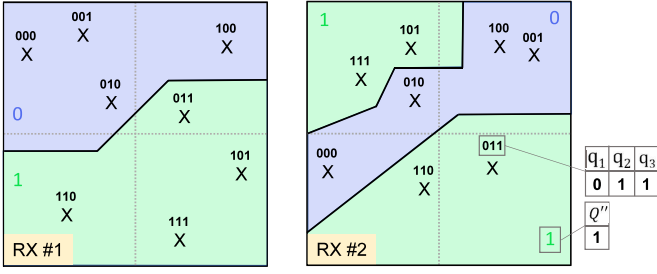


Fig. 4: Example of decision regions of over-the-air (OTA) majority computation for three transmitters $\{q_1, q_2, q_3\}$ at two distinct receivers. Blue/green regions map to 0/1.

- *The resilience of the HDC paradigm to errors makes it highly tolerant to poor BER conditions.* Indeed, a drawback of wireless technology in general and OTA computing in particular is that it can suffer from relatively high error rates, leading to inefficient designs. However, as we show later in the paper, HDC is inherently resistant to such conditions and allows to scale the proposed approach to tens of IMC cores.

IV. METHODOLOGY

The main contribution of this work is the validation of the OTA on-chip computing concept and scalability assuming a realistic chip package. Fig. 5 summarizes the procedures followed to evaluate the proposed approach. First, a package has been modelled in CST Studio [51] together with its corresponding chiplets, as also shown in Fig. 5. The operating frequency is 60 GHz, compatible with the on-chip environment [45]. Symbols are transmitted with an amplitude of 0 dBm per antenna [47], and the phase is discretized in 45 degree steps. Both time-domain and frequency-domain simulations for a simultaneous excitation of all TXs have been performed. The results have been post-processed to extract delay spread, path-loss data and phase data. Next, this has been used in MATLAB to perform a constellation search. That is, among all the different possible symbol phases and for all TX bit-combinations, the ones reporting the best BERs have been chosen. Finally, the error rate figures have been used in an HDC framework in order to characterize the impact of the wireless channel in the overall architecture in terms of classification accuracy.

Source coding. The way the TX encode the bits of their queries is by varying their phases. That is, all TX symbols will have same amplitude but different phases. We sweep a discrete set of 8 phases in the TXs in order to characterize the electromagnetic behaviour in each case and to find the best separable phase combinations. That is, we consider as RX constellation the aggregation of all the possible TX combinations. When choosing the optimal TX phases (two per sender, each one assigned to the binary 1 or 0), however, we have two points to consider: first, we have to meet the independent phase requirement. That is, we have to make sure that each TX only uses two phases and that the phase of each

TX is independent of each other; secondly, the TX phases affect all RXs, meaning that, when we fix the symbol phases we fix the received constellation for all receivers. This implies that a joint optimization considering all RXs is needed.

As an instance of the proposed approach and for illustration purposes, let us consider three TXs. In that case, we have a constellation with $2^3 = 8$ symbols for each RX. In order to map the eight symbols to their binary majority result, four corresponding to $\text{maj}(\cdot) = 1$ and four corresponding to $\text{maj}(\cdot) = 0$, decision regions are computed using the K -means clustering algorithm with $K = 2$. We make sure that each cluster contains four symbols and that the combination of TX phases allows the mapping to the majority result. Fig. 6 shows an example of this method in three distinct RXs: on top, we show the received signals considering all possible bit combinations in the TXs and for all the swept phases, whereas, on bottom, we see the chosen constellations. Further, Fig. 7 shows the chosen transmitted phases for the case under study and how they are mapped in a particular receiver.

Error rate assessment. Once the candidate clusters are obtained, we compute the BER of each constellation in each RX, for all the different possible symbol phases, and choose the cluster that leads to the lowest average BER across RXs. In all cases, the BER has been evaluated considering the centroids of each binary cluster as ideal received symbols, and using the analytical expression of error rate of BPSK,

$$\text{BER}^{\text{BPSK}} = 0.5 \cdot \text{erfc}\left(\frac{0.5 \cdot d_c}{\sqrt{N_0}}\right), \quad (1)$$

where $\text{erfc}(\cdot)$ is the complementary error function, d_c is the distance among centroids and N_0 is the noise spectral density.

Bundling and accuracy evaluation. Once the final TX phases have been chosen considering the best average BER, an in-house Python HDC is used to evaluate its impact on the accuracy. Every associative memory connected to an RX stores 100 different prototype hypervectors, i.e., 100 different classes, each with 512-bit that suffices for the scenario considered in this paper. Errors coming from the OTA computations are modeled as uncorrelated bit flips over the query hypervectors.

While the baseline bundling consists on simply computing the bit-wise logical majority result across the different TX bits, we also consider a permuted bundling. This bundling consists on permuting the queries in the TXs prior to applying the majority operation to them. By permuting the hypervectors we obtain two benefits. First, this allows the identification of the transmitter of the detected class from the composite query. If we make each transmitter to apply a 1-bit cyclic permutation to its query before sending it to the wireless channel, the detected bundled hypervectors will contain the information of such permuted versions. Then, each receiver can expand its prototype hypervector set with their permuted versions, each corresponding to a different transmitter signature. The second direct benefit of permuting the hypervectors is that it helps increasing the quasi-orthogonality between them, which has a direct impact in accuracy, since the TXs share a common codebook of hypervectors.

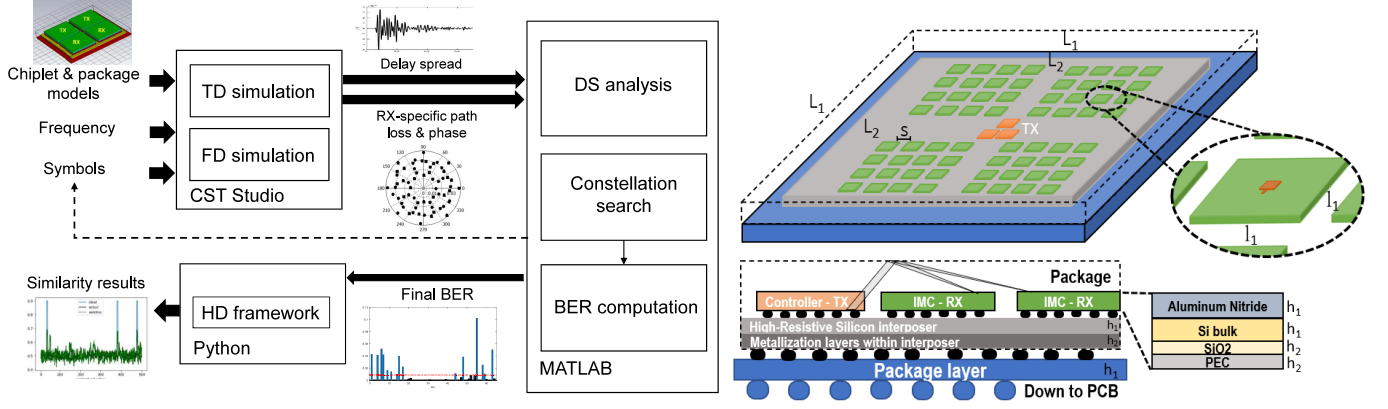


Fig. 5: Overview of the evaluation methodology and layout of a sample architecture with 3 TXs and 64 RXs. The package is enclosed in a metallic lid and empty spaces are filled with vacuum. $h_1 = 0.1$ mm; $h_2 = 0.01$ mm; $l_1 = 7.5$ mm; $s = 3.75$ mm; $L_1 = 33$ mm; $L_2 = 30$ mm.

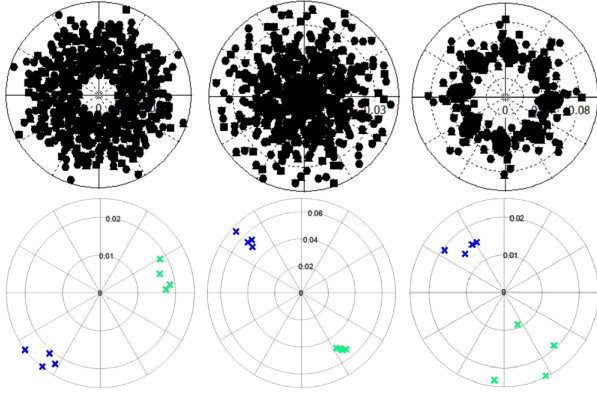


Fig. 6: Sweep of all possible phase combinations (top) and chosen to minimize the error rate of the majority computation (bottom). Blue/green symbols map to logical 0/1.

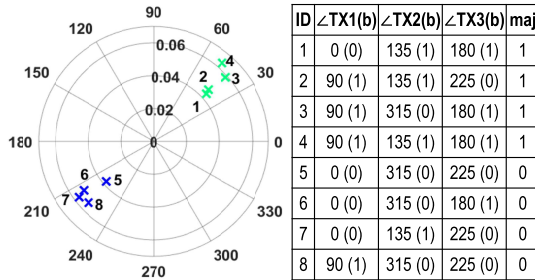


Fig. 7: Constellation and truth table with transmitted phases/bits for a specific RX. Blue/green symbols map to logical 0/1.

V. RESULTS AND DISCUSSION

After applying the proposed methodology and the careful optimization of the TX symbols as illustrated in Fig. 6, we obtained the TX phases shown in Fig. 7 for our 3-TX system. The assessment of the error rate considering the chosen TX phases is summarized in Fig. 8, which plots the BER of each particular receiver in the 64-RX system under study. As it can be seen, the BER values are very much dependent on the particular receiver, with values lower than 10^{-5} in a significant amount of cases, but also with a worst-case BER of ~ 0.1 . In

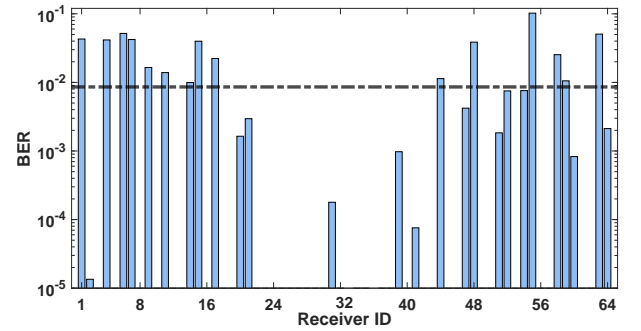


Fig. 8: Resulting BER values per each individual RX in the architecture. The dashed line indicates the average value.

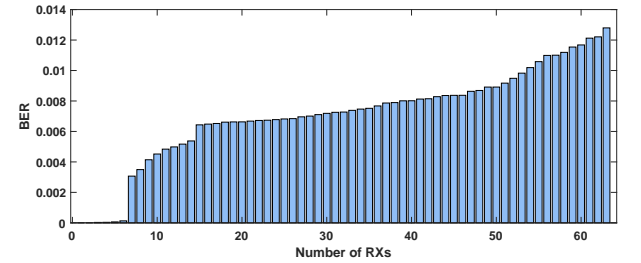


Fig. 9: Architecture scalability in a 3 TXs scenario.

average, the error rate is below 0.01. Time-domain simulations, not shown for the sake of brevity, further confirm that the OTA computation can be done at multi-Gb/s rates.

To understand how the error rate could scale with the number of receivers, we re-simulate the entire architecture with a varying number of RX cores and computing the average BER obtained in each case. As shown in Fig. 9, the average BER generally increases with the number of receivers for which we are optimizing the architecture. This is expected since, when accommodating more constellations in our optimal TX phases search, we are imposing more conditions and hindering the joint optimization across all receivers.

Next, to evaluate the performance of the proposed architecture, we execute a typical HDC-based classification task by introducing the wireless error figures in the HDC chain.

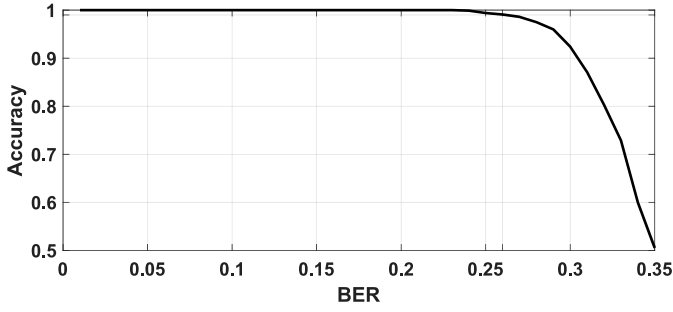


Fig. 10: Impact on the accuracy of a classification task when increasing the error rate of the encoder-to-search interconnect.

TABLE I: Accuracy results in an IMC for the analyzed bundling techniques, a variable number of TXs, both for an ideal channel (no errors) and for a channel with BER equivalent, in average, to that obtained with 64 RXs.

Baseline Bundling	Channel	Number of bundled hypervectors					
		1	3	5	7	9	11
	Ideal	1	0.966	0.902	0.803	0.704	0.543
Permuted Bundling	Channel	Number of bundled hypervectors					
		1	3	5	7	9	11
	Ideal	1	1	1	1	0.995	0.978
Baseline Bundling	Channel	Number of bundled hypervectors					
		1	3	5	7	9	11
	Wireless	1	0.966	0.9	0.801	0.699	0.537
Permuted Bundling	Channel	Number of bundled hypervectors					
		1	3	5	7	9	11
	Wireless	1	1	1	1	0.994	0.963

First, we illustrate the impact of errors on the classification by performing a generic classification task test over 100 prototype hypervectors of 512 bits, with increasing error rates. As Fig. 10 depicts, the class accuracy remains above 99% even when we apply bit flips equivalent to a BER of 0.26. This means that the noise robustness provided by the HDC properties relaxes the error link conditions, ensuring a correct behaviour under the worst-case wireless scenarios, as we show next.

Fig. 11a and Fig. 11b show the similarity search result for the baseline bundling and permuted bundling cases, respectively, after comparing the composite query hypervector against a set of 100 prototype hypervectors. The figures show how a single query has capacity enough to successfully accommodate several queries via bundling (blue line), and that the error introduced by the wireless OTA computation reduces the similarity but does not introduce any classification errors (green line). Table I shows the numerical results of the final class accuracy for the executed task, comparing an ideal channel without errors with our wireless channel with a sizable BER. The effect of the wireless channel is practically irrelevant in terms of accuracy, as predicted by Fig. 10. Moreover, the permuted bundling significantly improves the baseline bundling, confirming that the proposed approach supports the aggregation of a dozen hypervectors over the air and the parallelization of similarity search over tens of IMCs.

VI. CONCLUSION

In this work, we introduced an OTA on-chip computing concept capable of overcoming the scalability bottleneck present in wired NoC architectures when scaling out IMC-based HDC systems. By using a WNoC communication layer, a number of encoders is able to concurrently broadcast HDC queries

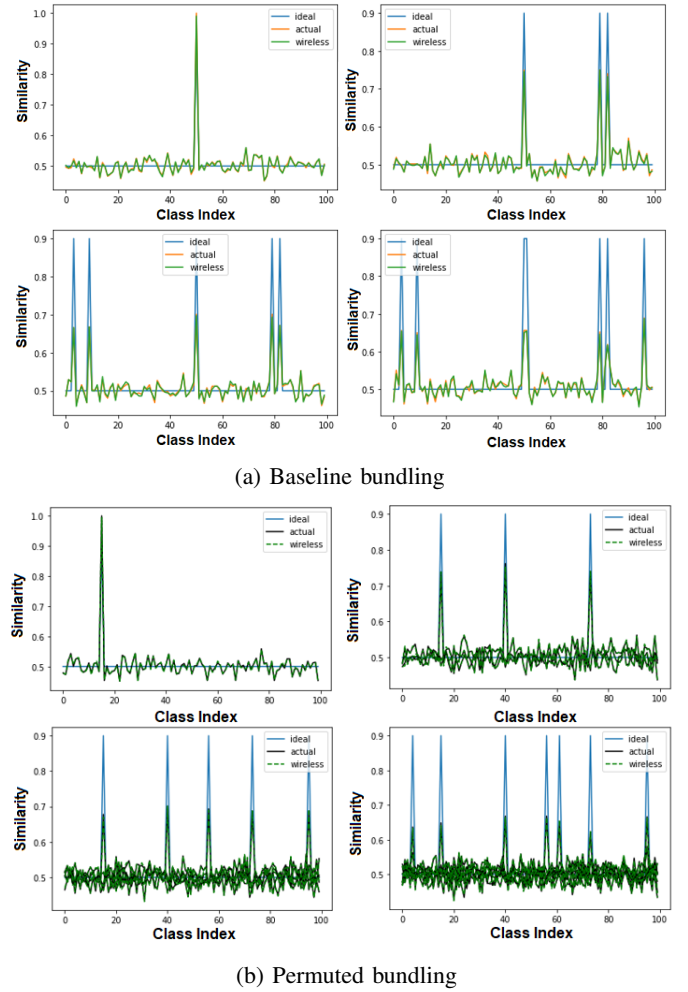


Fig. 11: Similarity results comparison for different forms of bundling and number of bundled hypervectors. We show bundling of one, three, five and seven hypervectors.

towards all the IMC cores within the architecture. Then, a pre-characterization of the propagation environment allows to map the received constellations to the computed composite query, in each core, based on a decision region strategy. Through a proper correspondence between the TX phases, the received constellation and the decision region, we have shown that the opportunistic calculation of the bit-wise majority of the transmitted HDC queries is possible with low error. We demonstrated the concept and shown its scalability up to 11 TXs and 64 RXs, obtaining the BER of the OTA approach and later employing it to evaluate the impact of the WNoC errors in a HDC classification task. Overall, we conclude that the quality of the WNoC links are solid enough to have a negligible impact on the application accuracy, mostly thanks to the great error robustness of HDC.

ACKNOWLEDGMENT

Authors gratefully acknowledge funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 863337 (WiPLASH).

REFERENCES

- [1] P. Kanerva, "Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors," *Cognitive Computation*, vol. 1, no. 6, pp. 623–641, 1995.
- [2] T. A. Plate, "Holographic reduced representations," *IEEE Transactions on Neural Networks*, vol. 6, no. 3, pp. 623–641, 1995.
- [3] T. A. Plate, *Holographic Reduced Representations: Distributed Representation for Cognitive Structures*. Center for the Study of Language and Information, Stanford, 2003.
- [4] D. A. Rachkovskij, "Representation and processing of structures with binary sparse distributed codes," *IEEE Transactions on Knowledge and Data Engineering*, vol. 3, no. 2, pp. 261–276, 2001.
- [5] R. W. Gayler, "Vector symbolic architectures answer Jackendoff's challenges for cognitive neuroscience," in *Proceedings of the IACS '03*, 2003.
- [6] T. A. Plate, "Analogy Retrieval and Processing with Distributed Vector Representations," *Expert Systems*, vol. 17, no. 1, pp. 29–40, 2000.
- [7] S. V. Slipchenko *et al.*, "Analogical Mapping using Similarity of Binary Distributed Representations," *Information Theories and Applications*, vol. 16, no. 3, pp. 269–290, 2009.
- [8] P. Kanerva, "What We Mean When We Say 'What's the Dollar of Mexico?': Prototypes and Mapping in Concept Space," in *Proceedings of the AAAI Fall Symposium '10*, 2010, pp. 2–6.
- [9] P. Neubert *et al.*, "An Introduction to Hyperdimensional Computing for Robotics," *KI - Künstliche Intelligenz*, vol. 33, no. 4, pp. 319–330, 2019.
- [10] D. Verma *et al.*, "Towards A Distributed Federated Brain Architecture using Cognitive IoT Devices," *Proceedings of COGNITIVE*, 2017.
- [11] C. Simpkin *et al.*, "A Scalable Vector Symbolic Architecture Approach for Decentralized Workflows," in *Proceedings of COLLA*, 2018.
- [12] R. Tomsett *et al.*, "Demonstration of Dynamic Distributed Orchestration of Node-RED IoT Workflows Using a Vector Symbolic Architecture," in *Proceedings of IEEE SMARTCOMP '19*, 2019, pp. 464–467.
- [13] P. Jakimovski *et al.*, "Collective communication for dense sensing environments," in *Proceedings of the IE'11*, 2011, pp. 157–164.
- [14] D. Kleyko *et al.*, "Dependable mac layer architecture based on holographic data representation using hyper-dimensional binary spatter codes," in *Multiple Access Communications*, 2012.
- [15] H.-S. Kim, "HDM: Hyper-Dimensional Modulation for Robust Low-Power Communications," in *IEEE International Conference on Communications*, 2018.
- [16] C. W. Hsu *et al.*, "Collision-tolerant narrowband communication using non-orthogonal modulation and multiple access," in *Proceedings of the GLOBECOM*, 2019.
- [17] C.-W. Hsu *et al.*, "Non-orthogonal modulation for short packets in massive machine type communications," in *Proceedings of the GLOBECOM '20*, 2020.
- [18] M. Hersche *et al.*, "Near-channel classifier: symbiotic communication and classification in high-dimensional space," *Brain Informatics*, vol. 8, no. 1, p. 16, Aug 2021.
- [19] M. Hersche *et al.*, "Constrained Few-shot Class-incremental Learning," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 1–19.
- [20] G. Karunaratne *et al.*, "Robust high-dimensional memory-augmented neural networks," *Nature Communications*, vol. 12, 2021.
- [21] A. Burrello *et al.*, "One-shot learning for iecg seizure detection using end-to-end binary operations: Local binary patterns with hyperdimensional computing," in *Proceedings of the IEEE BioCAS*, 2018, pp. 1–4.
- [22] A. Rahimi *et al.*, "Efficient Biosignal Processing Using Hyperdimensional Computing: Network Templates for Combined Learning and Classification of ExG Signals," *Proceedings of the IEEE*, vol. 107, no. 1, pp. 123–143, 2019.
- [23] A. Rahimi *et al.*, "Hyperdimensional Computing for Blind and One-Shot Classification of EEG Error-Related Potentials," *Mobile Networks and Applications*, 2017.
- [24] A. Moin *et al.*, "A Wearable Biosensing System with In-sensor Adaptive Machine Learning for Hand Gesture Recognition," *Nature Electronics*, vol. 4, no. 1, pp. 54–63, 2021.
- [25] S. Benatti *et al.*, "Online Learning and Classification of EMG-Based Gestures on a Parallel Ultra-Low Power Platform Using Hyperdimensional Computing," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 13, no. 3, pp. 516–528, 2019.
- [26] E. Chang *et al.*, "Hyperdimensional Computing-based Multimodality Emotion Recognition with Physiological Signals," in *Proceedings of the IEEE AICAS*, 2019.
- [27] A. Mitrokhin *et al.*, "Symbolic Representation and Learning with Hyperdimensional Computing," *Frontiers in Robotics and AI*, pp. 1–11, 2020.
- [28] M. Eggimann *et al.*, "A 5 μ W Standard Cell Memory-based Configurable Hyperdimensional Computing Accelerator for Always-on Smart Sensing," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 10, pp. 4116–4128, 2021.
- [29] A. Rahimi *et al.*, "A Robust and Energy Efficient Classifier Using Brain-Inspired Hyperdimensional Computing," in *Proceedings of the IEEE/ACM ISLPED*, 2016.
- [30] H. Li *et al.*, "Hyperdimensional Computing with 3D VRRAM In-Memory Kernels: Device-Architecture Co-Design for Energy-Efficient, Error-Resilient Language Recognition," in *Proceedings of the IEEE IEDM*, 2016.
- [31] T. Wu *et al.*, "Brain-Inspired Computing Exploiting Carbon Nanotube FETs and Resistive RAM: Hyperdimensional Computing Case Study," in *Proceedings of the IEEE ISSCC*, 2018.
- [32] G. Karunaratne *et al.*, "In-memory hyperdimensional computing," *Nature Electronics*, vol. 3, pp. 327–337, 2020.
- [33] A. Sebastian *et al.*, "Memory devices and applications for in-memory computing," *Nature Nanotechnology*, vol. 15, 03 2020.
- [34] S. Yu *et al.*, "Scaling-up resistive synaptic arrays for neuro-inspired architecture: Challenges and prospect," in *Proceedings of the IEEE IEDM*, 2015.
- [35] S. Laha *et al.*, "A New Frontier in Ultralow Power Wireless Links: Network-on-Chip and Chip-to-Chip Interconnects," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 2, 2015.
- [36] M. Ahmed *et al.*, "An asymmetric, one-to-many traffic-aware mm-wave wireless interconnection architecture for multichip systems," *IEEE T. on Emerging Topics in Computing*, 2020.
- [37] S. Jog *et al.*, "One protocol to rule them all: Wireless network-on-chip using deep reinforcement learning," in *Proceedings of the NSDI '21*, 2021, pp. 973–989.
- [38] A. Ganguly *et al.*, "Interconnects for DNA, Quantum, In-Memory and Optical Computing: Insights from a Panel Discussion," *IEEE Micro*, 2022.
- [39] S. Abadal *et al.*, "Graphene-based wireless agile interconnects for massive heterogeneous multi-chip processors," *arXiv preprint arXiv:2011.04107*, 2020.
- [40] J. Kim *et al.*, "HBM: Memory solution for bandwidth-hungry processors," in *2014 IEEE Hot Chips 26 Symposium (HCS)*, 2014.
- [41] M. Shulaker *et al.*, "Three-dimensional integration of nanotechnologies for computing and data storage on a single chip," *Nature*, vol. 547, pp. 74–78, 2017.
- [42] J. T. Pawlowski, "Hybrid memory cube (HMC)," in *2011 IEEE Hot Chips 23 Symposium (HCS)*, 2011.
- [43] R. Khaddam-Aljameh *et al.*, "HERMES-core—a 1.59-TOPS/mm² PCM on 14-nm CMOS in-memory compute core using 300-ps/LSB linearized CCO-based ADCs," *IEEE Journal of Solid-State Circuits*, 2022.
- [44] H. M. Cheema *et al.*, "The last barrier: on-chip antennas," *IEEE Microwave Magazine*, vol. 14, no. 1, pp. 79–91, 2013.
- [45] X. Timoneda *et al.*, "Engineer the channel and adapt to it: Enabling wireless intra-chip communication," *IEEE Transactions on Communications*, vol. 68, no. 5, pp. 3247–3258, 2020.
- [46] R. Guirado *et al.*, "Dataflow-Architecture Co-Design for 2.5D DNN Accelerators using Wireless Network-on-Package," in *Proceedings of the ASP-DAC '21*, 2021, pp. 806–812.
- [47] X. Yu *et al.*, "Architecture and design of multichannel millimeter-wave wireless noc," *IEEE Design Test*, vol. 31, no. 6, pp. 19–28, 2014.
- [48] S. Abadal *et al.*, "Opportunistic beamforming in wireless network-on-chip," in *Proceedings of the IEEE ISCAS*, 2019.
- [49] M. F. Imani *et al.*, "Metasurface-programmable wireless network-on-chip," *Advanced Science*, 2022.
- [50] Y. S. Shao *et al.*, "Simba: Scaling deep-learning inference with multi-chip-module-based architecture," in *Proceedings of the MICRO-52. ACM*, 2019, p. 14–27.
- [51] "CST Microwave Studio," [Online]. Available: <https://www.cst.com>. Accessed 28-September-2021.