Hyperdimensional Computing for Chip Testing – Towards Learning Fast from Little Data

Milestone Report – GS-IMTR

Paul R. Genssler

October 12, 2022

GS-IMTR
Graduate School
Intelligent Methods for Test and Reliability

Advisor / main examiner: Jun.-Prof. Hussam Amrouch

Co-examiner: Prof. Jens Anders

Mentor: Steffen Templin

1 Introduction

This report gives a comprehensive summary of the state of GS-IMTR project 10 (P10) "Hyperdimensional Computing for Chip Testing – Towards Learning Fast from Little Data" as part of the Milestone Presentation. P10 focuses on investigating the emerging brain-inspired machine learning method hyperdimensional computing (HDC), with focus on (but not limited to) its applicability to the area of semiconductor test and reliability. Given the unique challenges this area provides, P10 aims at developing novel approaches to overcome the limitations of existing machine-learning algorithms and tackle existing and upcoming challenges for semiconductor test and reliability.

1.1 Unique Challenges for Test and Reliability

Technology scaling is reaching atomic limits where displacing few charge carries within a transistor may disturb its entire functionality and behavior. As a matter of fact, quantum effects become dominant at extreme feature sizes (i.e., 5 nm and 3 nm). This, in turn, makes device measurements inherently noisy and the behavior of transistor itself becomes stochastic. Such large inherent uncertainties, in fact, impose serious challenges for deep learning when it comes to chip testing and defect classification.

The number of prototype chips for any new technologies is typically limited at the beginning, restricting the number of available samples that can be acquired for training. If only a limited number of samples can be acquired, their benefit for the training has to be as high as possible to maximize the training efficiency. Furthermore, to determine which samples improve the classification accuracy the most, active learning-based approaches can be employed. However, to fully utilize the potential of active leaning in semiconductor testing, training must be very fast for quick feedback in a short feedback loop.

Furthermore, sub-7 nm transistors enable the creation of overwhelmingly complex system on chip (SoC) in which virtually unlimited corner cases may exist. The measurements and tests of such cases are additionally influenced by external factors like temperature and voltage fluctuations but also seemingly randomly, e.g., by the precise timing of asynchronous events in the system. This, in turn, results in seldom problems in which reproducing them during integrated circuit (IC) testing (i.e., repeating and creating the same conditions that have induced them) is very challenging if not impossible. Because those problems are seldom, only few samples (i.e., little data) will be available. Hence, model training might not be feasible with typical machine learning (ML) approaches.

Beside seldom problems where available data is limited, there are other kinds of problems in IC testing where data is also very limited not due to the difficulty in producing such data but due to the prohibitively high associated cost. During wafer testing, undetected wear-out signs in probing cards (i.e., probe tip degradation) can lead to shortcuts during wafer testing. Hence, catastrophic breakdowns happen due to burning effects because a large current suddenly flows through the probe tip. While, in theory, producing a large dataset for learning is possible through enforcing failures (i.e., accelerating wear-out and burning effects), the huge cost associated with such experiments prevents that, each probe card costs tens to hundreds of thousands of Dollars.

1.2 Hyperdimensional Computing

The idea of hyperdimensional (HD) computing was first introduced by Kanerva [20]. Later, concrete implementations were proposed, like holographic reduced representation [21], binary spares distributed code [22], or multiply-add-permute [23]. This section will introduce HD computing based on the multiply-add-permute approach and summarizes [24] and [25]. For more mathematical background, the work by Kanerva [24] is recommended.

All values in HD computing are represented by vectors of dimension d. The elements of the vector could be of any type, but real numbers or binary values are the most common ones. In this project, we will focus primarily on binary vectors as they allow for more efficient implementations. However, the concepts outlined in this section also apply for other types of vectors, e.g., with real numbers. In HDC, the space is not only two or three dimensional, but has many thousand dimensions, hence the term hyperdimensional. Using such a large space has various advantages, like robustness to noise, either of the computational device or the data [26].

The d-dimensional vectors are generated randomly. So binary vectors are equal to a random string of

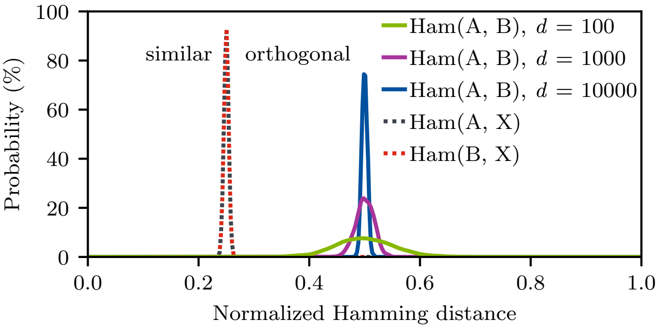

Figure 1: Normalized Hamming distance of vectors in HD space depending on dimension d and for $X = A + B$. While A and B are orthogonal to each other, X is similar to both.

zeros and ones of length d. To determine how close two vectors are, the normalized Hamming distance[1] is calculated. Two vectors are *orthogonal*, if the normalized Hamming distance is approximately 0.5. The probability that two randomly generated vectors are orthogonal is depicted in Fig. 1 and increases with a higher d. Two vectors are called *similar*, if the normalized Hamming distance is approximately 0.25 or smaller.

Item Memory The initial mapping of values and properties like Dollar and currency into HD space is a one-time process. A random bit vector of dimension d is generated and assigned to each value or property. The label-vector mapping is saved in the item memory (IM) and reused if the same value has to be mapped into HD space again. If the label for a vector is needed, the vector is used as a query in the IM. The query-vector is compared to all stored vectors and the closest match taken. The large Hamming distance of the vectors to each other makes this type of storage very robust to noise. Even if almost half of the bits of the query vector are wrong, the closest match is still the original vector. All other vectors have a larger distance. Therefore, the IM also sometimes also referred to as a clean-up memory. A query with a noisy vector results in the original "clean" vector.

Associate Memory The associate memory (AM) is structurally the same as the IM. In contrast to the IM, the AM does not store randomly generated vectors, but the trained vectors. During training, a vector is computed from the training data, labeled and stored in the AM. This is repeated for all classes/labels, each being represented by a vector. Later during inference, the computation is repeated (with the new input data) and a vector close to the trained vector used as the query. Thanks to the noise resistance, even noisy queries yield the correct vector-label pair.

Addition The pointwise addition, also called bundling operation, combines two or more vectors into a new vector X that is maximal similar to the inputs. In other words, the Hamming distance between X and any inputs is at a minimum as illustrated in Fig. 1.

For binary vectors, a majority function is commonly used. An example is given in Equation (1) for three input vectors. For each bit, the number of ones and zeros is compared and the majority decides the bit in the resulting vector. If the number of input vectors is even and the number of ones and zeros is equal, the output bit is assigned randomly. Alternatively, two of the inputs are XOR-ed and the result used as a tie breaker. The addition is a lossy operation and cannot be inverted.

$$\begin{array}{r} A = 0\,1\,0\,0\;\;0\,0\,1\,1 \\ +\,B = 0\,1\,1\,0\;\;1\,1\,0\,1 \\ +\,C = 1\,0\,1\,0\;\;0\,0\,1\,1 \\ \hline X = 0\,1\,1\,0\;\;0\,0\,1\,1 \end{array} \tag{1}$$

[1] The number of bits that is different between the two operands. It is normalized to the length of the vector, i.e., the dimension d.

Multiplication The pointwise multiplication, takes two vectors and produces a new vector that is orthogonal to both inputs. It is commonly implemented as a bitwise XOR as shown in Equation (2).

$$\begin{array}{r} A = 0\,1\,0\,0\;\;0\,0\,1\,1 \\ \oplus B = 0\,1\,1\,0\;\;1\,1\,0\,1 \\ \hline X = 0\,0\,1\,0\;\;1\,1\,1\,0 \end{array} \tag{2}$$

In contrast to the addition, the XOR operation, i.e., the multiplication, is invertible. Note that $Z \oplus Z = 0$, in other words, a vector is its own multiplicative inverse. Hence, it is possible to reconstruct the first input vector A from the second input B and the output vector X (and vise versa). The steps are outlined in Equation (3). This enables the semantic of binding two vectors together and be represented by the resulting vector.

$$\begin{aligned} X &= A \oplus B \qquad\quad | \oplus B \\ X \oplus B &= A \oplus B \oplus B \\ X \oplus B &= A \oplus 0 = A \end{aligned} \tag{3}$$

Permutation The permutation operation ρ alters a vector Y, e.g., by a linear shift. Thus, the permuted vector $\rho(Y)$ is orthogonal to Y. In the example given in Equation (4), the normalized Hamming distance of $\rho(Y)$ and Y is 0.75. With a higher d, it is expected to be closer to 0.5. The permutation operation, which is easy to invert, comes in handy to construct more complex data types.

$$\begin{aligned} Y &= 1\,1\,0\,1\;\;0\,1\,1\,0 \\ \rho(Y) &= 0\,1\,1\,0\;\;1\,0\,1\,1 \end{aligned} \tag{4}$$

Data Structures and Examples The described operations enable typical data structures like mappings or sequences. In the example *Dollar of Mexico*, a mapping is used, which is commonly referred to as a record. A record R binds a property P to a value V and bundles multiple of these pairs:

$$R = P_0 \oplus V_0 + P_1 \oplus V_1 + P_2 \oplus V_2 + \dots$$

Each country is encoded as a record and thus as a single vector. The encoding of properties P like population and currency are stored in the IM. Possible values V like $350M$, $120M$ (population size), Dollar, and Mexican Peso (currency) are stored in the IM as well. Using those values and properties, the two countries can be encoded in a record R. Such a single encoding step is already the training and hence referred to as *one-shot learning.* The trained country vectors are stored in the AM. For iterative learning with many samples representing the same class, their encoded vectors are average into a single class vector. During inference, the same encoding process is repeated and the vector used as a query in the AM. Once the closest match is found, its label is the class for the query.

2 Published Research, 15 Works in Total (5 Journals, 10 Conferences)

With the support of the Graduate School Intelligent Methods for Test and Reliability (GS-IMTR) [7] at the University of Stuttgart, funded by Advantest, we published research at a variety of workshops, conferences, and journals. In publications in which Paul R. Genssler is the first author, research is usually heavily linked to the topic of P10. However, we also had the opportunity to contribute to publications of collaborations, which are usually more loosely related to the topic of P10. Authorship of Paul R. Genssler in cited publications is also highlighted in the references section at the end of this report, including 5 publications in journals and 10 publications at conferences. Since the start of P10, Paul R. Genssler has contributed to 10 publications (3 journals and 7 conferences), as depicted in Fig. 2.

2.1 First-Author Publications

Typical neural network-based ML algorithms require large datasets and many iterations to achieve their high accuracy. However, in such amounts of data are not always available, e.g., due to the high cost of

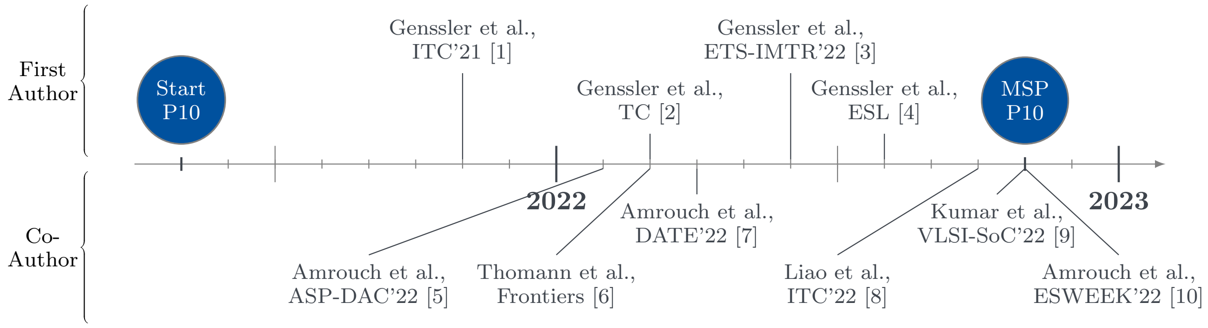
First Author
Co-Author
Start P10
MSP P10
2022
2023
Genssler et al., ITC'21 [1]
Genssler et al., TC [2]
Genssler et al., ETS-IMTR'22 [3]
Genssler et al., ESL [4]
Amrouch et al., ASP-DAC'22 [5]
Thomann et al., Frontiers [6]
Amrouch et al., DATE'22 [7]
Liao et al., ITC'22 [8]
Kumar et al., VLSI-SoC'22 [9]
Amrouch et al., ESWEEK'22 [10]

Figure 2: Timeline of GS-IMTR project 10, showing all accepted first author publications (top) and contributions to other works (bottom) from the start of the project up to the milestone presentation date.

labeling it or simply creating the samples. HDC addresses this challenges, which we have demonstrated in our first work.

Our work *"Brain-Inspired Computing for Wafer Map Defect Pattern Classification"* [1], presented at the *IEEE International Test Conference (ITC'21)*, is the first to propose HDC in the area of semiconductor test. Wafer testing is an important step to discard defective chips early saving costs. Although every defect reduces the yield, they are expected and tolerated. However, a cluster or pattern of defects in the wafer map hints at a systematic problem in the production process. The wafer maps provide visual details critical to recognizing the errors. Test engineers have to check every wafer map for such patterns to aid in the root cause analysis. Such a manual step is costly, prone to error, and slow. Our HDC-based to reduce labeling effort and learn from little data is shown in Fig. 3.

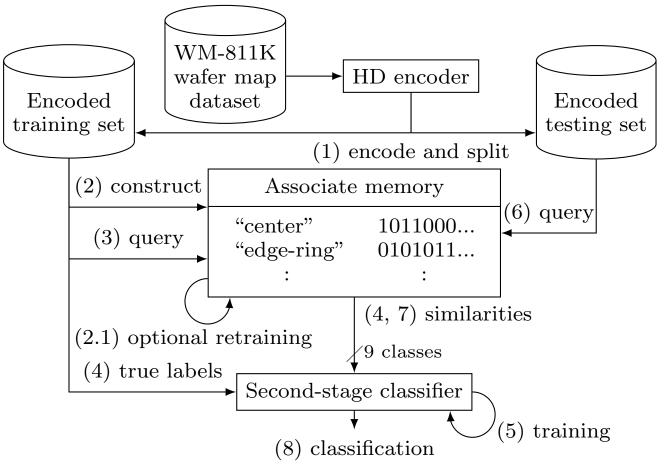

Figure 3: HDC flow for wafer map defect pattern classification. Figure from [1].

The typical encoding-training-testing flow is enhanced with an additional training step. After the AM is constructed from the training dataset (2), it is used again as a query for the AM (3). The resulting similarity metrics for all classes (4), combined with the true labels (4), are used to train the second-stage classifier (5). During inference, the unknown wafer map is also provided to the AM (6) and the similarities are forwarded to the second-stage classifier (7, 8). With our work, we show that learning from little data is possible with HDC. In fact, we demonstrate that a single expert-provided image is sufficient for binary classification with 94 % accuracy for a subset of the dataset. For the full dataset, we achieve 95 % average accuracy, which is comparable to convolutional neural networks (CNNs) yet our approach is 46X faster. This highlights not only the potential of HDC as a classifier for active learning but even more so for the whole area of semiconductor testing.

Our work *"Brain-Inspired Computing for Circuit Reliability Characterization"* [2], published in *IEEE Transactions on Computers*, addresses early and rapid characterization of degradation effects impacting the circuits' transistors. In this work, we are the first to employ HDC in the area of semiconductor reliability. With current sub-10 nm technologies, the reliability of these chips is a major challenge. The processes in the foundry have to be as precise as possible to manufacture transistors where to smallest parts are only a few atoms in size. Even small deviations can produce less reliable devices, reducing yield, a critical metric for the foundry and their clients. Characterizing such small deviations in the manufactured circuits precisely is necessary to improve their processes, thus the reliability, and ultimately increase performance. Determining the amount of degradation, the drift away from the desired properties, allows for a prediction of the remaining lifetime or on-the-fly performance tuning. In this work, we demonstrate how brain-inspired HDC can be employed to address these challenges. In fact, our approach can be applied to different applications within the field of reliability characterization. Depending on the specific application, different advantages of HDC are exploited.

To train the HDC model, data can be generated through (a) simulations of a multi-transistor circuit. The physics-based single transistor model is typically provided by the foundry. In the simulation, the degradation can be set freely and this parameter value together with the resulting voltage transfer characteristics (VTC) is the training data for one class. Alternatively, (b) prototype devices can be measured. Like any other semiconductor, such devices are affected by process variation. Hence, multiple samples per device have to be taken to overcome such effects. To generate distinct training data per class, a device is subjected to aging stress causing premature degradation. Through physics simulation, the amount of induced degradation is known and used as a label together with the measured VTC as training data for one class. Once the model is trained with a range of classes, the VTCs of a circuit can be measured and its degradation inferred.

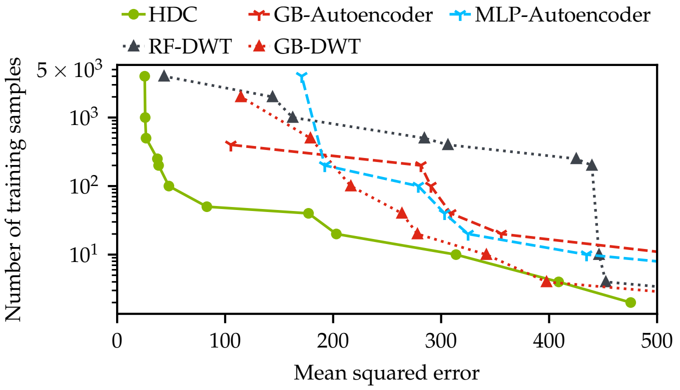

Figure 4: Pareto-optimal solutions for a number training samples compared with inference accuracy expressed mean squared error. Only the best performing traditional machine learning methods are shown. Figure from [2].

A key promise of HDC is its ability to learn from little data. In Fig. 4, the Pareto-optimal solutions for the best-performing traditional methods are compared with HDC. HDC outperforms the traditional ML methods in terms of error for a given budget of training samples. In particular, in the range from ten to 1000 total training samples, HDC has a significantly lower mean squared error, more than 4x at 100 training samples. We have also tested the approach with other circuits that SRAM cells. The results demonstrate that larger circuits tend to yield lower errors, because the more transistors they have, the lower the impact of process variation becomes and in turn the impact of degradation increases.

Our work *"Brain-Inspired Hyperdimensional Computing: How Thermal-Friendly for Edge Computing?"* [4], published in *IEEE Embedded Systems Letters*, explores the promise of HDC is a highly efficient implementation for embedded systems like wearables. While fast implementations have been presented, other constraints have not been considered for edge computing. In this work, we aim at answering how thermal-friendly HDC for edge computing is. Devices like smartwatches, smart glasses, or even mobile systems have a restrictive cooling budget due to their limited volume. Although HDC operations are simple, the vectors are large, resulting in a high number of CPU operations and thus a heavy load on the entire system potentially causing temperature violations. In this work, the impact of HDC on the chip's temperature is investigated for the first time. We measure the temperature and power consumption of a commercial embedded system and compare HDC with conventional CNN as shown in Fig. 5.

To improve the inference accuracy, the concept of retraining has been proposed [27]. After an initial training cycle and class hypervector generation, the training data itself is used for inference. If a sample is

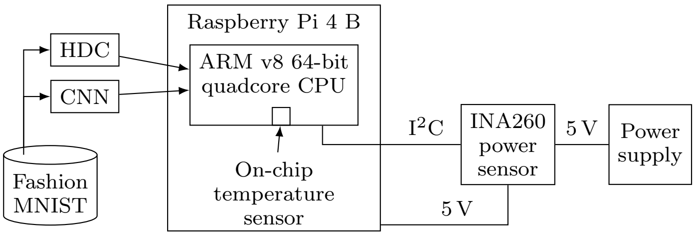

Figure 5: Experimental setup to measure temperature and power of an embedded system. In this case a Raspberry Pi 4. Fashion MNIST serves as a realistic workload to test HDC and a CNN. Figure from [4].

incorrectly classified, then the class hypervector is adjusted to be more similar to the sample. The process is repeated. Each iteration is referred to as an epoch.

OnlineHD [28] is a state-of-the-art HDC implementation using floating point hypervectors. Version 0.1.2 is built on top of Python's PyTorch framework and provides efficient implementations for the underlying mathematical operations during encoding. The similarity computation is written in C++. As a comparison, a CNN is implemented also with the PyTorch framework. Both methods achieve about the same inference accuracy. GPU acceleration is not available. All tasks, such as training and inference, are continuously repeated for four hours. This allows the system to accumulate heat and to minimize environmental influences.

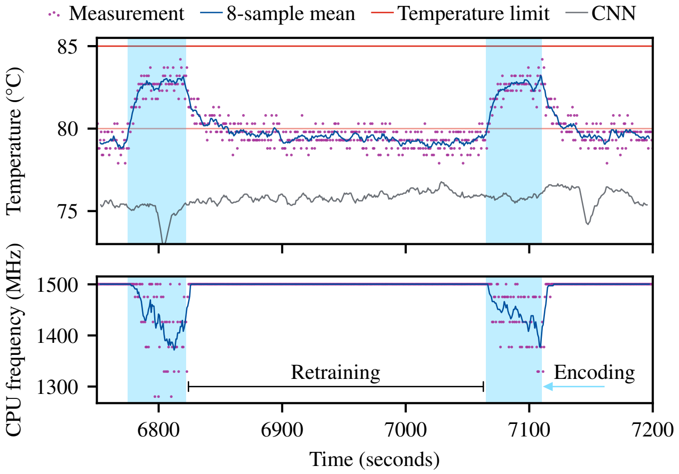

Figure 6: Encoding during repeated training in HDC causes CPU throttling. Despite that, HDC is still faster in training the model. Figure from [4].

During (re)training, the CNN does not exceed 79 °C with 75.6 °C on average as shown in Fig. 6. In contrast, HDC reaches temperatures of up to 84.2 °C, 6.7 °C more than CNN on average. HDC is frequently close to the maximum temperature of 85 °C during the encoding of the data set. Hence, dynamic voltage and frequency scaling (DVFS) is employed by the SoC, reducing the clock frequency up to 47 % (800 MHz). During retraining, the temperature drops to about 79 °C. No thermal buffers are created for the next encoding cycle. The power consumption is similar for both models with 5.2 W on average. Therefore, the operations used by HDC cause more heat compared to the CNN. Yet, despite the CPU throttling, HDC completes a training cycle 14.9 % faster. However, inference accuracy on the test set is 4 % lower with HDC. Our work unveils that HDC faces a temperature and power challenge in low-power systems such as wearables and hence it is less thermal-friendly in edge computing. These results motivated further research in accelerating HDC, especially in the area of in-memory computing to address the overhead of data transfers.

Besides the aforementioned conferences and journals, our work has been presented as part of invited talks at the 27th Asia and South Pacific Design Automation Conference (ASP-DAC'22) [5], at the 2022 International Conference on Hardware/Software Codesign and System Synthesis (CODES) at ESWEEK, and at the IFIP/IEEE 30th International Conference on Very Large Scale Integration (VLSI-SoC'22)

[9]. Furthermore, the work was presented at the Design, Automation & Test in Europe Conference & Exhibition, DATE 2022 [7] and at the Workshop on Intelligent Methods for Test and Reliability (IMTR'22) at the IEEE European Test Symposium (ETS'22) [3].

2.2 Co-Authorship in Collaborations

The work *"Design Close to the Edge in Advanced Technology using Machine Learning and Brain-Inspired Algorithms"* [5] is a collaboration between P9 and P10 within the GS-IMTR and has been published as an invited special session paper at ASP-DAC'22. In advanced technology nodes, transistor performance is increasingly impacted by different types of design-time and run-time degradation. First, variation is inherent to the manufacturing process and is constant over the lifetime. Second, aging effects degrade the transistor over its whole life and can cause failures later on. Both effects impact the underlying electrical properties of which the threshold voltage is the most important. To estimate the degradation-induced changes in the transistor performance for a whole circuit, extensive SPICE simulations have to be performed. However, for large circuits, the computational effort of such simulations can become infeasible very quickly. Furthermore, the SPICE simulations cannot be delegated to circuit designers, since the required underlying transistor models cannot be shared due to their high confidentiality for the foundry. In [5], we tackle these challenges at multiple levels, ranging from transistor to memory to circuit level. We employ machine learning and brain-inspired algorithms to overcome computational infeasibility and confidentiality problems, paving the way towards design close to the edge. Our reliability estimation results, as one part of the work, are shown in Fig. 7.

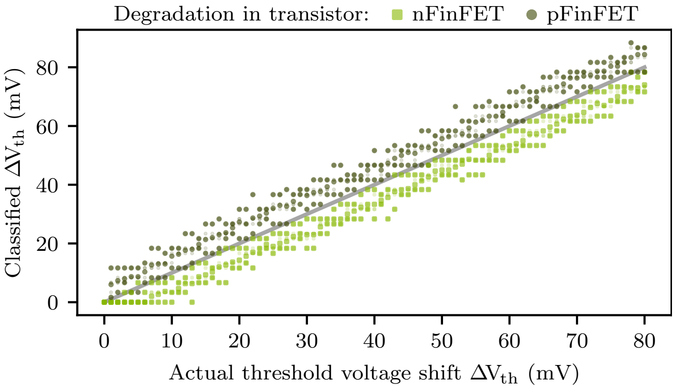

Figure 7: Classification results for the ΔV_{th} based on the measured static noise margin (SNM). On average, the inferred value deviates by 5.8 mV. Figure from [5].

In [4], we have shown that the amount of data transfers challenges embedded systems. In our work *"All-in-Memory Brain-Inspired Computing using FeFET Synapses"* [6], published in Frontiers in Electronics, we propose a novel in-memory design to reduce data transfers and avoid the von Neumann bottleneck. On the hardware level, analog Processing-in-Memory (PiM) schemes are used to build platforms that eliminate the compute-memory gap to overcome the von Neumann bottleneck. PiM can be efficiently implemented with ferroelectric transistors (FeFET), an emerging non-volatile memory technology. However, PiM and FeFET are heavily impacted by process variation, especially in sub 14 nm technology nodes, reducing the reliability and thus inducing errors. HDC is robust against such errors.

Nevertheless, the analog nature of PiM schemes necessitates the conversion of results to digital, which is often not considered. Yet, the conversion introduces large overheads and diminishes the PiM efficiency. In that work, we propose an all-in-memory scheme performing computation *and* conversion at once, utilizing programmable FeFET synapses to build the comparator used for the conversion. An overview of our proposed FeFET-based AM is shown in Fig. 8. The AM is subdivided in D/N N-bit blocks, each storing N bits of a class hypervector in N TCAM cells. Each block consists of two parts: the TCAM cells to store the bits and to perform the XOR operation with the query bits, and the FeFET-based synaptic comparator to convert the voltage from the match line (ML) into the digital domain for further accumulation. Due to process variation, this ML voltage as well as the synapses can produce incorrect partial Hamming distances. In our evaluation, we show that this error rate can be as high as 80 %. Yet, the inference accuracy of HDC is reduced by 0.5 % to 1.0 % for large dimensions.

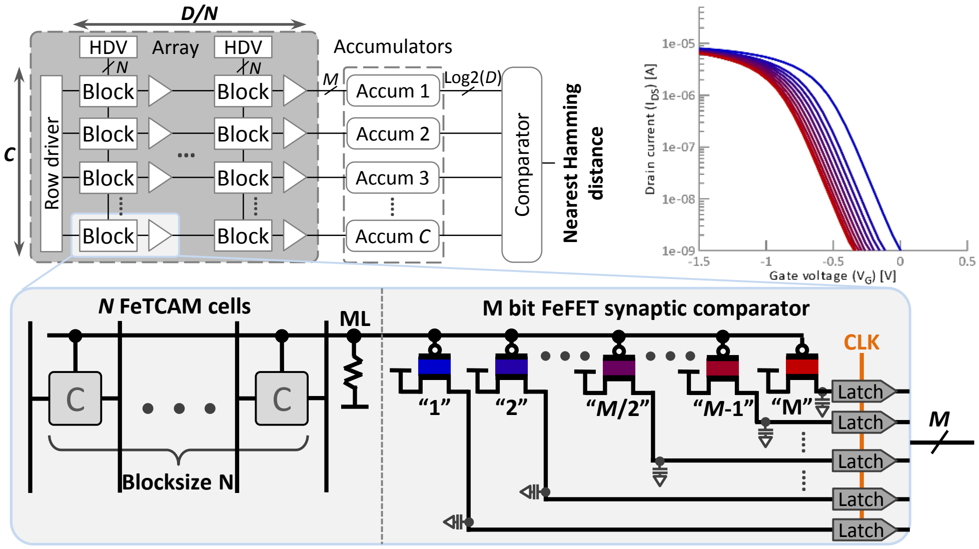

Figure 8: Sketch of possible layout with C rows for the class hypervector each of dimension D cut into blocks of size N. The $I_D - V_G$ plot on the top right with the intermediate states of the p-type FeFET synapse shows their individual states. In the bottom, a single Fe-TCAM block with N cells on the left and our proposed synaptic comparator decoding the analog discharge rate of a N cell TCAM block into digital on the right. Figure from [6].

Finally, the work *"Wafer Map Defect Identification Based on the Fusion of Pattern and Pixel Information"* [8] is a collaboration between P6 and P10 within the GS-IMTR and has been presented at IEEE International Test Conference (ITC'22). The experience gained in P10 from the work in the previous year [1] could be applied to this work. Our work proposes a multi-task learning framework based on neural networks that fuses the information of the entire wafer map as well as the state of each individual die to enhance the defect pattern classification capability. More precisely, in addition to a common learning objective of predicting the defect pattern for the entire wafer map, a novel second learning task is proposed that takes account the state information of dies on wafer maps (i.e., pass or fail of each die). As a result, the proposed model is trained towards simultaneously predicting defect patterns as well as predicting the state of dies. In other words, it is trained to reconstruct the wafer map in a pixel level. The multi-task learning process is expected to regularize the main learning procedure (i.e., prediction of defect patterns) and yield more reliable identification performance.

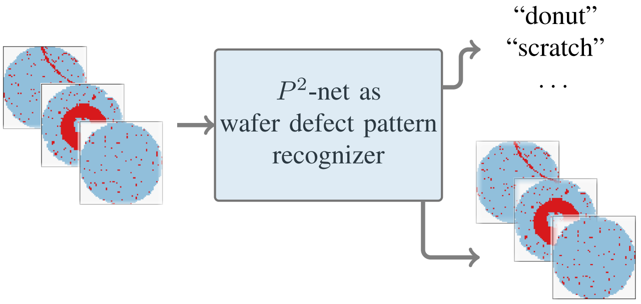

Figure 9: An overview of the proposed generic framework for classifying defect patterns in wafer maps. Our proposed P2-Net is jointly trained to predict pattern-level labels (e.g. donut and scratch) and reconstruct the original wafer map based on a pixel-wise prediction. Figure from [8].

In general, our method had comparable or better classification accuracy than the other recent state-of-the-art approaches, clearly showing the effectiveness of the proposed novel architecture. Although DMC [29] achieved slightly better performance (about 0.7 % only) than our method, DMC required heavy data augmentation, while our method was directly trained on the original dataset so that notable training time can be saved. One interesting observation is that the random forest as a conventional machine learning algorithm achieved similar accuracy with other DL-based methods. The main reason is the imbalanced dataset as mentioned before; i.e., the samples affiliated with the "none" pattern were dominant and the metric accuracy was thus not representative enough.

3 Conclusion and Future Work

Within the first 18 months of P10, we explored many important directions in the context of HDC. Most importantly, we demonstrated in multiple publications [1–3] that HDC can be applied in the area of semiconductor test and reliability. With our latest contributions to address graph-based problems with HDC [11], we will soon be able to also investigate circuit netlists and propose novel approaches in the field of security. Further, reliability research of transistor aging effects will be accelerator through the ML-based tools developed in this project [12]. Our research will keep addressing the unique challenges in the area of semiconductor test and reliability, such as little data, fast learning, and robustness against noise and errors.

Publications Accepted and Under Review

[1] **P. R. Genssler** and H. Amrouch, "Brain-inspired computing for wafer map defect pattern classification," in *IEEE International Test Conference (ITC'21)*, 2021.

[2] **P. R. Genssler** and H. Amrouch, "Brain-inspired computing for circuit reliability characterization," IEEE Transactions on Computers, 2022.

[3] **P. R. Genssler** and H. Amrouch, "Brain-inspired hyperdimensional computing for semiconductor test and reliability," in *Workshop on Intelligent Methods for Test and Reliability (IMTR'22)*, 2022.

[4] **P. R. Genssler**, A. Vas, and H. Amrouch, "Brain-inspired hyperdimensional computing: How thermal-friendly for edge computing?" IEEE Embedded Systems Letters, 2022.

[5] H. Amrouch, F. Klemme, and **P. R. Genssler**, "Design close to the edge in advanced technology using machine learning and brain-inspired algorithms," in *27th Asia and South Pacific Design Automation Conference (ASP-DAC'22)*, 2022.

[6] S. Thomann, H. L. G. Nguyen, **P. R. Genssler**, and H. Amrouch, "All-in-memory brain-inspired computing using fefet synapses," Frontiers in Electronics, vol. 3, 2022.

[7] H. Amrouch, J. Anders, S. Becker, M. Betka, G. Bleher, P. Domanski, N. Elhamawy, T. Ertl, A. Gatzastras, **P. R. Genssler**, S. Hasler, M. Heinrich, A. v. Hoorn, H. Jafarzadeh, I. Kallfass, F. Klemme, S. Koch, R. Küsters, A. Lalama, R. Latty, Y. Liao, N. Lylina, Z. P. Najafi-Haghi, D. Pflüger, I. Polian, J. Rivoir, M. Sauer, D. Schwachhofer, S. Templin, C. Volmer, S. Wagner, D. Weiskopf, H.-J. Wunderlich, B. Yang, and M. Zimmermann, "Intelligent methods for test and reliability," in *Design, Automation & Test in Europe Conference & Exhibition, DATE 2022*, 2022.

[8] Y. Liao, R. Latty, **P. R. Genssler**, H. Amrouch, and B. Yang, "Wafer map defect identification based on the fusion of pattern and pixel information," in *IEEE International Test Conference (ITC'22)*, 2022.

[9] S. Kumar, S. Chatterjee, S. Thomann, **P. R. Genssler**, Y. S. Chauhan, and H. Amrouch, "Cross-layer fefet reliability modeling towards robust hyperdimensional computing," in *IFIP/IEEE 30th International Conference on Very Large Scale Integration (VLSI-SoC'22)*, 2022.

[10] H. Amrouch, M. Imani, X. Jiao, Y. Aloimonos, C. Fermuller, D. Yuan, D. Ma, H. Errahmouni, **P. R. Genssler**, and P. Sutor, "Brain-inspired hyperdimensional computing for ultra-efficient edge ai," in *Proceedings of the 2022 International Conference on Hardware/Software Codesign and System Synthesis*, 2022.

[11] R. Wang, **P. R. Genssler**, H. Amrouch, and X. Jiao, "Hyperdimensional computing-based graph classification," 2022, under review.

[12] **P. R. Genssler**, H. E. Barkam, K. Pandaram, M. Imani, and H. Amrouch, "Modeling and predicting transistor aging under workload dependency using machine learning," IEEE Transactions on Circuits and Systems I: Regular Papers (TCAS-I), 2022, under review.

[13] **P. R. Genssler**, V. Van Santen, J. Henkel, and H. Amrouch, "On the reliability of fefet on-chip memory," IEEE Transactions on Computers, vol. 71, no. 4, pp. 947–958, 2022.

[14] C. Hakert, K.-H. Chen, H. Schirmeier, L. Bauer, **P. R. Genssler**, G. von der Brüggen, H. Amrouch, J. Henkel, and J.-J. Chen, "Software Based Read and Write Wear-Leveling for Non-Volatile Main Memory," ACM Transactions on Embedded Computing Systems (TECS), 2021.

[15] V. M. van Santen, S. Thomann, C. Pasupuleti, **P. R. Genssler**, N. Gangwar, U. Sharma, J. Henkel, S. Mahapatra, and H. Amrouch, "Bti and hcd degradation in a complete 32× 64 bit sram array– including sense amplifiers and write drivers–under processor activity," in *2020 IEEE International Reliability Physics Symposium (IRPS)*, 2020.

[16] V. M. van Santen, **P. R. Genssler**, O. Prakash, S. Thomann, J. Henkel, and H. Amrouch, "Impact of self-heating on performance, power and reliability in finfet technology," in *2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2020.

[17] C. Hakert, M. Yayla, K.-H. Chen, G. von der Brüggen, J.-J. Chen, S. Buschjäger, K. Morik, **P. R. Genssler**, L. Bauer, H. Amrouch, et al., "Stack usage analysis for efficient wear leveling in non-volatile main memory systems," in *2019 ACM/IEEE 1st Workshop on Machine Learning for CAD (MLCAD)*, 2019.

[18] H. E. Barkam, S. Yun, **P. R. Gensler**, Z. Zou, C.-K. Liu, H. Amrouch, and M. Imani, "Hdgim: Hyperdimensional genome sequence matching on unreliable highly-scaled fefet," in *Design, Automation & Test in Europe Conference & Exhibition, DATE 2023*, 2023, under review.

[19] S. Thomann, **P. R. Genssler**, and H. Amrouch, "Hw/sw co-design for reliable in-memory brain-inspired hyperdimensional computing," IEEE Transactions on Computers, 2022, under review.

Other References

[20] P. Kanerva, Sparse distributed memory. MIT press, 1988.

[21] P. Smolensky, "Tensor product variable binding and the representation of symbolic structures in connectionist systems," Artificial Intelligence, vol. 46, no. 1, pp. 159–216, 1990.

[22] D. A. Rachkovskij and E. M. Kussul, "Binding and normalization of binary sparse distributed representations by context-dependent thinning," Neural Comput., vol. 13, no. 2, pp. 411–452, 2001.

[23] R. W. Gayler, "Multiplicative binding, representation operators & analogy (workshop poster)," 1998.

[24] P. Kanerva, "Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors," Cognitive Computation, vol. 1, pp. 139–159, 2009.

[25] L. Ge and K. K. Parhi, "Classification using hyperdimensional computing: A review," IEEE Circuits and Systems Magazine, vol. 20, no. 2, pp. 30–47, 2020.

[26] A. X. Manabat, C. R. Marcelo, A. L. Quinquito, and A. Alvarez, "Performance analysis of hyperdimensional computing for character recognition," in *2019 International Symposium on Multimedia and Communication Technology (ISMAC)*, 2019, pp. 1–5.

[27] M. Imani, D. Kong, A. Rahimi, and T. Rosing, "VoiceHD: Hyperdimensional computing for efficient speech recognition," in *2017 IEEE Int. Conf. on Rebooting Computing (ICRC)*, 2017, pp. 1–8.

[28] A. Hernandez-Cane, N. Matsumoto, E. Ping, and M. Imani, "OnlineHD: Robust, efficient, and single-pass online learning using hyperdimensional system," in *2021 Design, Automation Test in Europe Conference Exhibition (DATE)*, 2021, pp. 56–61.

[29] T.-H. Tsai and Y.-C. Lee, "A light-weight neural network for wafer map classification based on data augmentation," IEEE Transactions on Semiconductor Manufacturing, vol. 33, no. 4, pp. 663–672, 2020.