(Invited Paper)

Heterogeneous Manycore In-Memory Computing Architectures

Chukwufumnanya Ogbogu[†], Gaurav Narang[†], Biresh Kumar Joardar[*], Janardhan Rao Doppa[†], Partha Pratim Pande[†]

[†]Washington State University, Pullman, WA, USA, [*]University of Houston, TX, USA

[†]{c.ogbogu, gaurav.narang, jana.doppa, pande}@wsu.edu, [*]bjoardar@central.uh.edu

ABSTRACT

The growing use of deep learning has led to an increasing demand for hardware platforms that are computationally powerful, yet energy-efficient. In-memory computing (IMC) architectures using non-volatile memory, such as resistive random-access memory (ReRAM), present a promising alternative. In addition to ReRAM, there are a plethora of IMC devices. Each device offers different advantages and drawbacks in terms of power, latency, area, and non-idealities. However, IMCs lack general-purpose computing capability. For instance, ReRAM crossbars are not suited for high-throughput division, which is needed for implementing normalization layers. In this paper, we present architectures that combine both (IMC and general-purpose computing) in an optimized manner to derive the best out of both worlds. The heterogeneous architectures combine the high-throughput multiplications of IMCs with the general-purpose computing ability of floating-point devices (such as CPU, GPU, etc.) to implement both training and inferencing of various AI algorithms.

CCS CONCEPTS

Hardware → Emerging technologies; Hardware →Very large-scale integration design

KEYWORDS

DNN, Heterogeneous, in-memory computing, ReRAM, FeFET, SRAM.

ACM Reference format:

Chukwufumnanya Ogbogu, Gaurav Narang, Biresh Kumar Joardar, Janardhan Rao Doppa and Partha Pratim Pande. 2024. Heterogeneous Manycore In-Memory Computing Architectures: Invited Paper. In *Proceedings of the ACM/IEEE International Conference on Computer-Aided Design (ICCAD '24), October 27–31, 2024, Newark, New Jersey, USA. ACM, New York, NY, USA, 7 pages.*

1 Introduction

Deep Neural Networks (DNNs) are widely used to address complex challenges in a variety of application domains, including computer vision, natural language processing (NLP), and time-series sensor data analytics [1]. DNNs have hundreds of millions of trainable parameters, which need to be tuned using large and complex datasets. The high latency and energy cost of data-movement between the processing cores and memory units in traditional computing platforms based on the von-Neuman architecture (e.g., CPUs and GPUs) impose significant performance bottlenecks while executing DNN workloads, which is referred to as the "*memory wall"* challenge [2]. Consequently, there has been a growing demand for in-memory computing (IMC) platforms that seamlessly integrate both storage and computing, thereby enabling high-performance and energy-efficient acceleration of DNNs [3]. This is due to their ability to perform energy-efficient computation within the memory to eliminate unnecessary data movement, thus addressing the memory-wall challenge.

Recent work has studied the use of CMOS-based memory devices such as Static Random-Access-Memory (SRAM), and non-volatile memory (NVM) devices such as Resistive-Random-Access-Memory (ReRAM), Phase Change Memory (PCM), Ferroelectric-Field-Effect-Transistors (FeFET), and spintronic memory (MRAM) as suitable candidates for IMC-based platforms used for accelerating DNN workloads [2] [3] [4] [5] [6]. Architectures based on these IMC technologies offer significant speedup compared to traditional computing architectures. However, these devices have specific advantages and drawbacks in terms of power, area, latency, data retention, endurance, and other non-idealities, when used as the computing element in IMC-based architectures. For example, ReRAM devices have ~35× less area compared to SRAM cells. However, ReRAMs suffer from limited write endurance (10^6 -10^{12} programming cycles), whereas SRAMs have a high write endurance $>10^{17}$ cycles [7]. As a result, none of these IMC technologies are suited to handle the diversity in AI workloads by itself.

In addition, IMC-based architectures lack general purpose computing capability. For instance, IMC crossbar arrays can perform energy-efficient Matrix-Vector-Multiplication (MVM) operations very fast. However, they are not suited for high-throughput division, which is needed for implementing normalization layers or non-linear operations (such as SoftMax). On the other hand, general-purpose processors, such as CPU and GPU, can perform all kinds of mathematical tasks. However, CPUs/GPUs have significant area and power overhead, and are slower than IMC crossbar arrays for MVM operations, which form the primary backbone of most AI algorithms. Consequently, this necessitates the need for heterogeneous platforms that combine more than one type of IMC device and general-purpose processors on a single platform to achieve high-performance DNN acceleration, both for training and inferencing.

However, despite the benefits that heterogeneous computing platforms can potentially offer, integrating different types of memory and devices in a single platform presents unique challenges. Specifically, manufacturing technologies of IMC devices vary, and they are not always CMOS-compatible. As a result, fabricating a heterogeneous architecture that includes both IMC and CMOS ASICs in a single system leads to lower yield owing to the multiple design processes involved [9]. Recent work has proposed 2.5D and 3D heterogeneous integration technologies that enable the mapping of disparate technologies onto the same platform [9]. A suitable architecture should combine both IMC devices as well as general-purpose processors in an optimized manner to derive the best out of both worlds. However, existing implementations of 2.5D/3D heterogeneous architectures are not well optimized to ensure high-accuracy, energy-efficient and high-

ICCAD '24, October 27–31, 2024, New York, NY, USA

ACM ISBN 979-8-4007-1077-3/24/10.
https://doi.org/10.1145/3676536.3697138

TABLE I
COMPARISON OF VARIOUS IMC TECHNOLOGIES (†F is the minimum feature size [22])

Property	**SRAM** [3]	**DRAM** [23]	**MRAM** [27]	**PCM** [26]	**ReRAM** [7]	**FeFET** [25]
Multi-bit Cell	No	No	No	Yes	Yes	Yes
†Cell Area (F^2)	$150F^2$	$6F^2$	$32F^2$	$36F^2$	$4F^2$	$10F^2$ - $35F^2$
Write Energy (nJ)	~ 0.003	~ 0.05	~1	~6	3.9 - 5.3	0.01
Write Latency (ns)	~1	12 - 20	10 - 20	~150	51 - 54	<1
Write Endurance (cycles)	$>10^{17}$	$>10^{17}$	10^{12} - 10^{15}	10^{6} - 10^{9}	10^{10} - 10^{12}	10^{5} - 10^{8}
Leakage Energy	High	High	Low	Low	Low	Low

performance execution of DNN workloads [10]. This is because they do not consider how the characteristics of the IMC device could potentially impact the performance of DNN training and inference in their design optimization flow. For example, 3D architectures are known to give rise to thermal hotspots. This challenge is further exacerbated when general purpose processors such as GPUs are integrated in a 3D platform with IMC devices, which typically exhibit non-ideal behavior due to thermal noise [11]. As a result, this potentially degrades the inference accuracy of DNNs. Hence, to meet the high-accuracy and performance demands of DNNs, a hardware/software co-design approach needs to be explored to derive the benefits of heterogeneous IMC-based architectures.

In this paper, we first present the state-of-the-art IMC-based architectures and their different underlying technologies. Next, we discuss the challenges associated with DNN training and inference using existing IMC-based architectures, as well as the inherent reliability and non-ideal behavior of IMC devices. Finally, we present heterogeneous manycore IMC-based architectures as a solution to address these challenges.

2 IMC Architectures and Challenges

In this section, we first present some CMOS-based DNN hardware accelerators. Next, we discuss an overview of various IMC-based devices, their advantages, and limitations for accelerating DNN workloads. Finally, we present shortcomings of IMC-based AI accelerators to motivate the need for heterogeneous architectures.

2.1 Existing DNN Hardware Accelerators

DNN workloads primarily consist of highly parallelizable MVM operations, which can be accelerated via CPUs and GPUs. GPU-based processors with high-bandwidth memory remain the most widely used choice for DNN acceleration [12] [13]. However, the *memory-wall* challenge limits its applicability to large-scale DNN workloads [3]. Other CMOS-based AI accelerators based on FPGAs and ASICs such as Systolic Arrays have been proposed for the energy-efficient acceleration of DNN workloads [14] [15]. However, these methods are still hampered by performance bottlenecks, and significant energy consumption due to large amount of data movement to and from memory. Efficient Network-on-Chip (NoC) enabled manycore systems have been developed [11] [16]. These architectures aim to reduce the communication bottleneck by leveraging a multicast-enabled NoC to handle data movement during DNN training and inference. NoC enabled heterogeneous architectures with both CPU and GPUs have also been proposed [17]. 3D manycore architectures for AI workloads is another possibility [18]. Chiplet-based 2.5D architectures which are known for having lower fabrication costs compared to 3D counterparts have been proposed [19]. Despite the variety of CMOS-based architectures available for accelerating AI workloads, they are bottlenecked by the memory wall. Consequently, in-memory computing (IMC) paradigm has emerged as an excellent candidate for the energy-efficient acceleration of DNN workloads.

2.2 IMC Technologies

Several IMC technologies are used to develop high-performance AI accelerators. Table I compares the physical characteristics of the different IMC technologies used for accelerating DNN workloads.

SRAM-based IMC crossbar arrays have been proposed as suitable candidates for high-accuracy DNN training and inference [20] [21]. This is due to their low device variability, high write endurance ($>10^{17}$), low susceptibility to noise, and low write latency [3]. However, the SRAM crossbar suffers from high area overhead due to the 6T-cell structure with an area footprint of $150F^2$ [3] [22]. Moreover, SRAMs suffer from high leakage energy and have low-density storage (i.e., 1-bit per-cell). Hence, this makes IMC platforms built solely out of SRAM crossbars, unattractive for energy-efficient acceleration of DNN workloads on platforms with low form-factor.

Recent work has also leveraged **DRAM** technology for IMC-based architectures due to its small cell area [23]. However, DRAM suffers from high leakage power and refresh energy due to its volatile nature. Moreover, the 1T1C array structure of the DRAM cell lacks in-situ compute capability. Hence, they cannot enable parallel energy-efficient MVM operations without significant modifications (e.g. adding multiple row decoders) to its design [24]. Consequently, this has led researchers to explore non-volatile memory (NVM) devices as suitable candidates for DNN training and inference.

ReRAM-based IMC crossbars have been proposed for accelerating DNN workloads [4] [2]. The high-density storage enabled by multi-bit cell structure, small cell area and low leakage energy of the ReRAM device makes it suitable for energy-efficient execution of DNN workloads [7]. However, despite these advantages, ReRAM cells suffer from low write endurance, high write energy, and latency. Moreover, ReRAMs suffer from reliability issues such as faults and conductance drift due to temperature which can cause errors [18]. As a result, this limits the applicability of ReRAM-based architectures for reliable DNN training and inference.

FeFET-based devices have been explored as a possibility for IMC-based DNN accelerators [5] [6]. FeFET devices are particularly attractive due to their relatively low cell area compared to SRAMs, high read and write speeds, low write energy, and low leakage energy as shown in Table 1. Moreover, they exhibit relatively better temperature stability compared to ReRAM, making them less prone to errors [25]. However, they suffer from significantly lower write endurance as the ON/OFF ratio of the FeFET device deteriorates after repeated program/erase cycles [5]. This leads to read errors during DNN training and inference.

IMC devices made from Phase Change Materials (**PCMs**) have also been proposed as suitable candidates for DNN acceleration.

This is due to their high-density storage, high ON/OFF ratio, reasonably high endurance compared to FeFET devices and fast switching speeds. However, the high temperature sensitivity of PCMs is a major concern [26]. Moreover, PCMs require significantly high programming energy as shown in Table 1 [22]. This makes them unsuitable for certain scenarios such as for DNN training or fine-tuning which requires multiple programing cycles.

MRAM-based IMC architectures have been demonstrated as attractive candidates for energy-efficient acceleration of DNN workloads [27]. This is due to their high endurance (up to 10^{15} cycles) and low write latency (~10 ns). However, they also suffer from low ON/OFF ratio, and low-density storage (i.e., 1-bit per-cell). The low ON/OFF ratio of MRAMs causes soft errors in the form of erroneous read/write operations. This leads to a degradation in accuracy during DNN training and inference.

2.3 IMC Architectures for DNN Inferencing and Training

Recent developments in in-memory computing have led to hardware accelerators designed specifically for deep neural network (DNN) training and inference. These IMCs utilize a crossbar architecture to enable fast and energy-efficient matrix-vector multiplications (MVMs). ReRAM-based IMC architectures, where each tile contains specialized multiply-accumulate units (IMAs) that perform MVM operations have been proposed [4] [28]. Figure 1 illustrates an example of a ReRAM-based IMC architecture [4]. The tiles are interconnected through a NoC, optimizing data movement and facilitating high-bandwidth aggregation of intermediate results, while also employing pipelining to manage complex data dependencies during DNN training and inference [2].

In addition to ReRAM-based designs, SRAM-based IMCs have been explored, which utilize SRAM crossbar arrays for storage and computations [3]. Recent advancements also include MRAM-based accelerators offering floating point precision, multi-level FeFET devices for better performance in DNN inferencing and PCM-based accelerators designed for high-accuracy AI workloads [27] [5] [26]. These various technologies reflect the ongoing efforts to enhance performance of DNN accelerators through different material and device innovations. Heterogeneous IMC-based architectures aim to address some of these shortcomings and achieve improved performance and energy efficiency for DNN workloads.

2.4 Limitations/Challenges of IMC-based Architectures

Homogeneous IMC-based architectures have several challenges to overcome, as we present next.

Reliability challenges: IMC-based architectures are prone to functional errors arising from device and circuit non-idealities. Generally, the errors due to these non-idealities can be broadly categorized as; read errors and write errors. Read errors occur during the read mode of the IMC-based crossbar array. These errors are due to non-idealities such as IR-drop, conductance drift etc. Meanwhile, write errors commonly occur due to Stuck-at-faults (SA0 or SA1). Consequently, these non-idealities lead to erroneous MVM outputs, thus leading to DNN accuracy loss. Moreover, IMC devices suffer from conductance drift under the influence of thermal noise. These non-ideal effects must be accounted for in the design of a suitable IMC-based architectures.

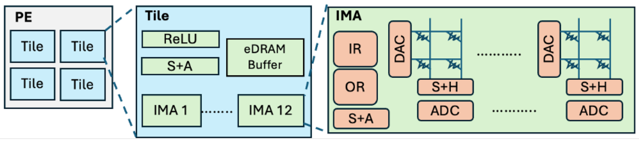


Figure 1. A typical hierarchical ReRAM-based IMC architecture, consisting of PEs, tiles, and in-situ multiply and accumulate units (IMAs).

Although IMC-based devices generally exhibit non-ideal behavior, it should be noted that each type of IMC device has its unique behavior under certain non-ideal conditions. For example, both ReRAM and FeFET devices suffer from conductance drift under high temperatures. However, ReRAM devices have an exponential dependence on temperature, while FeFET devices have a linear temperature dependence [25]. As a result, an increase in temperature would have a more severe impact on the predictive accuracy while executing DNN inference/training on ReRAM-based IMC architectures compared to the FeFET-based counterparts. It is crucial to consider the unique non-ideal properties of devices in the design optimization of a suitable heterogeneous architecture.

Circuit challenges: Next, analog IMC crossbars typically require high-resolution ADCs and DACs to interface with the digital domain. However, high-resolution ADCs (> 7-bits) consume significantly higher energy and area compared to the IMC crossbar itself [29]. As a result, ADC dominates the area and energy of IMC-based architectures. Moreover, IMC crossbar arrays suffer from IR-drop due to wire resistance and sneak-paths, which constitute the main sources of DNN accuracy degradation. As the crossbar array size increases, IR-drop becomes even more severe [30]. Moreover, larger crossbar array sizes are undesirable, as the resolution of the ADC increases with the crossbar size [29]. To tackle device and circuit non-idealities, MVM operations are computed at a much smaller granularity level than a full crossbar, referred to as Operation Units (OUs) [31]. For example, only nine wordlines (WLs) and eight bitlines (BLs) are activated concurrently within a 512×256 crossbar array to achieve a balance between achievable performance and reliability [32]. OU-based computation allows us to utilize lower-bit resolution ADC in each crossbar since fewer WLs are activated in each cycle. Nevertheless, for most state-of-the-art IMC-based crossbar arrays optimal selection of the crossbar size as well as OU size remain crucial in achieving a reasonable tradeoff between power, performance, area and DNN accuracy [31].

Training challenges: DNN training workloads pose additional requirements for designing crossbar-based systems, which includes a larger shared memory, and frequent crossbar writes. However, existing IMC-based DNN training accelerators (discussed in section 2.1 above) are not well optimized to accommodate these additional requirements. Unlike inference workloads, DNN training involves the back-propagation algorithm, which is sensitive to data precision and imposes additional design considerations. During training, these activations need to be stored temporarily for the computation of errors and gradients in the back-propagation phase, which is not needed for inference. Hence, IMC-based architectures aimed at DNN training have significantly higher storage requirements compared to their inference counterparts. To address this challenge, existing IMC-based architectures need to be equipped with high-bandwidth memory such as DRAM tiles for additional storage to enable large-scale DNN training. Moreover, the low precision of IMC devices poses a challenge during training, as gradients need to be computed in high-precision (32-bit floating point (FP32) precision). As a result, IMC-based architectures for training DNN workloads would also require high-precision processing elements (PEs) to support precision-critical portions of the workload such as back-propagation to ensure high accuracy.

Furthermore, during DNN training, the crossbar is written at the end of every step/batch corresponding to the weight update process. Large-scale DNN training with real-world datasets involves multiple epochs and training steps to achieve reasonable accuracy. This imposes a high write endurance requirement for devices used in IMC-based architectures designed for DNN training; several IMC technologies lack the required endurance to support DNN training (Table 1). AccuReD supports the execution of the back-propagation phase on general-purpose processors such as GPUs/CPUs [11]. Other work has proposed the use of SRAM-based crossbar arrays for the computation of gradients in the back-propagation phase [28].

Non-NVM operations: DNN workloads typically possess non-MVM layers such as Normalization, non-linear activation functions and pooling layers. These layers are often precision critical. The use of lower precision here would result in accuracy loss. However, these operations cannot be easily implemented on IMC-based crossbar arrays, as they are best suited for MVM operations only. As a result, existing approaches augment IMC-based architectures with on-chip CMOS-based logic and memory units to enable the implementation of non-MVM operations. However, this approach results in significant performance and energy overheads due to the high data movement between IMC and CMOS-based PEs during DNN training and inference. A suitable heterogeneous architecture must address this communication bottleneck.

From the above discussion, we see that despite the advantages, homogeneous IMC-based architectures may not be suited for accelerating diverse AI workloads. Hence, heterogeneous IMC-based architectures are viable alternatives for the end-to-end acceleration of DNN training and inference workloads.

3 Heterogeneous IMC Accelerators

In this section, we present heterogeneous IMC-based architecture for accelerating DNNs.

3.1 Heterogeneous Manycore IMC-based Architectures

To overcome the shortcomings of individual IMC technologies, prior work has proposed heterogeneous architectures that combine two or more IMC devices for accelerating DNN workloads. Various hybrid ReRAM/SRAM-based IMC-based architectures have been proposed to address the non-idealities of ReRAM devices and reduce the high area overhead of SRAM. Some of these methods involve encoding the MSBs using SRAMs, and ReRAMs for the LSBs of multi-bit weights, while maintaining high energy-efficiency [33]. Other methods involve the use of ReRAM and SRAM to perform the DNN forward- and back-propagation operations respectively, thereby mitigating the limited endurance challenge of ReRAM. In fact, a recent hybrid architecture incorporates SRAM macros to perform output compensation of the non-ideal output of ReRAM crossbars, thereby enabling robust DNN inference [8]. However, these methods do not consider the layer-wise characteristics of DNN workloads (e.g., number of neural layers, weights, activations, size of kernels etc.) while mapping neural layers to the heterogeneous IMC-based architectures.

In Figure 2, we show an example of how the layer-wise characteristics of the DenseNet40 workload determine the IMC device requirements during training. Here, we observe that initial layers (layers 0 - 14) in DenseNet40 process a higher number of activations than the latter layers, hence requiring more crossbars to store weights and activations. Therefore, these layers should be mapped to dense, low power PEs. These initial layers must also be placed closer to the heat sink to reduce thermal noise as these layers are more crucial for predictive accuracy. On the contrary, latter layers with comparatively fewer activations and smaller kernels, should be mapped on comparatively less dense PEs. These layer-wise characteristics must be considered while mapping DNN workloads to IMC platforms to ensure the high-performance and energy-efficient execution of DNN training and inference tasks.

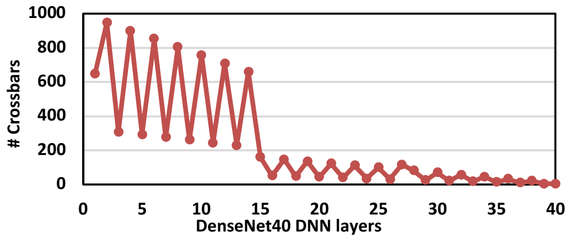


Figure 2: Layer-wise IMC crossbar array requirements for DenseNet40 trained with CIFAR-10 dataset.

HyDe is a recently proposed design and optimization methodology for mapping of deep neural network (DNN) layers to various PIM devices (SRAM, FeFET, PCM) in a hybrid platform, utilizing 2.5D chiplet-based heterogeneous integration [22]. HyDe maps each DNN layer to a suitable IMC device based on its characteristics. It leverages a scalarized single-objective optimization formulation and is aimed only at DNN inferencing. However, linear scalarization is known to perform poorly due to its inability to explore non-convex regions of the Pareto front. Other works such as HyperX have proposed a hybrid SRAM/ReRAM architecture, where the weights of some DNN layers remain static, and are mapped to ReRAMs, while the weights of other layers are mapped to SRAMs for fine-tuning [34]. Despite the advantages of heterogeneity, these existing solutions do not consider the challenges of integrating different IMC devices into a single platform. Hence, suitable heterogeneous IMC architectures for DNN training scenarios need to be explored.

3.2 3D Integration-enabled Heterogeneous Architectures

As mentioned above, IMC-based architectures are popular for accelerating both inference and training. However, DNN models typically consist of millions of parameters (weights and activations) which often cannot all be mapped onto a single planar tier consisting of IMC-based processing elements (PEs). Moreover, planar architectures provide limited design choices in terms of floor planning, i.e., how tiles are placed and the NoC is designed, which can lead to sub-optimal performance. Hence, 3D integration methods that stack planar tiers consisting of PEs connected to each other using through-silicon-via (TSV)- and monolithic 3D (M3D)-based vertical links have been proposed [18] [10]. Recent work such as AccuReD proposes a 3D heterogeneous architecture that enables high-performance execution of DNN workloads [11]. However, such heterogeneous architectures that leverage IMC devices mostly assume ideal behavior of the IMC device. Moreover, they only leverage one type of IMC device (ReRAM in this case). As a result, this leads to over-estimation of the DNN accuracy, and does not reap the benefits that other types of IMC devices can potentially offer.

3D Integration challenges for IMC architectures: Although 3D integration can enable high-performance DNN training as

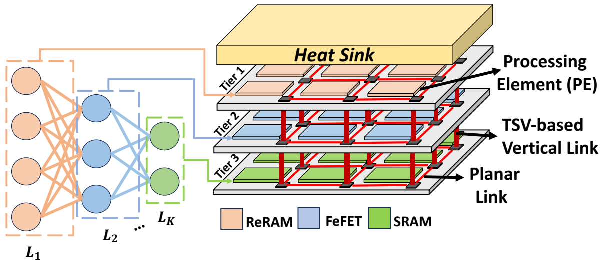


Figure 3: Illustration of layer-to-PE and PE-to-tier mapping of DNN workload with *K*-layers on to a 3D heterogeneous IMC-based architecture. Here, DNN layer L1 is mapped to ReRAM-based PEs and placed on Tier 1.

discussed earlier, stacking multiple high-power PEs vertically on top of one another increases temperature. Higher PE temperature introduces thermal noise in IMC devices, and ultimately affects the predictive accuracy [18]. Hence, we should avoid placing too many high-power consuming cores along one specific vertical column of the 3D architecture and away from the heat sink to reduce temperature hotspots. Further, different neural layers of a DNN workload possess unique characteristics such as kernel size, number of activations, precision, sensitivity to thermal noise etc., which all impact the predictive accuracy of the DNN. For example, the initial neural layers of a DNN workload are more crucial to the predictive accuracy, as they process more activations and thus, dissipate more power than the later layers [18]. Hence, the initial neural layers should not be mapped to PEs that are far away from the heat sink. Moreover, thermal noise has a varying impact on different IMC devices (as discussed in section 2.2). Thus, the accuracy-crucial layers should be mapped to IMC devices that are less susceptible to thermal noise. As an example, Figure 3 illustrates a three-tier 3D heterogeneous manycore architecture. Given the layer-wise characteristics of the DNN workload (as shown in Figure 2), and the physical properties of the IMC devices in the PEs, finding a suitable neural layer-to-PE and PE-to-tier mapping present a unique optimization problem. Hence, achieving an optimal mapping of the DNN layers to the IMC devices, as well as a suitable configuration of the IMC technology for each planar tier are necessary to achieve an acceptable balance between the predictive accuracy, area, latency, and power.

Design Optimization of 3D IMC architectures: To tackle the aforementioned challenges, the properties of the DNN neural layers, IMC device characteristics, as well as the PE to 3D planar tier mapping should be jointly considered to enable high-performance, energy-efficient, and reliable DNN training on heterogeneous IMC-based platforms. Consequently, this leads to a multi-objective optimization (MOO) problem of finding the suitable mapping of each neural layer to a PE with a suitable IMC device (i.e., either SRAM-/ReRAM-/FeFET-based PE), as well as its appropriate location in one of the planar tiers, that achieves the best latency (Lat), area (Ar), power (Pwr), and accuracy (Acc) trade-off. This MOO-formulation can be represented as:

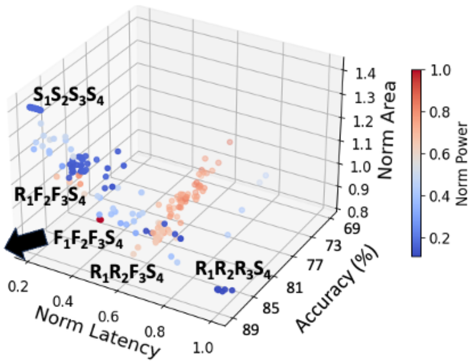
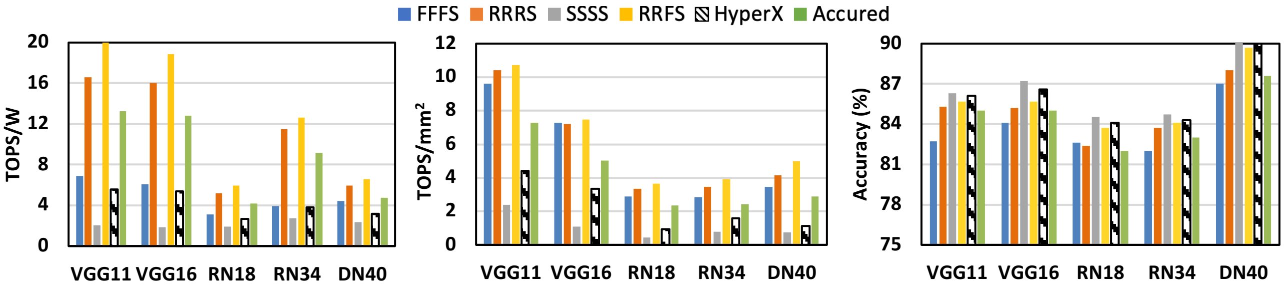


Figure 4: Layer-to-PE and PE-to-tier mapping trade-offs while running the DNN training task for ResNet34 model on CIFAR-10 dataset.

$$D^* = MOO(OBJ = Pwr, Ar, Lat, Acc) \quad (1)$$

where D^* is the set of Pareto optimal designs. The goal is to first find the Pareto optimal set $D^* \subseteq D$ using a MOO solver (e.g., AMOSA) [35]. The IMC devices largely vary in terms of design metrics such as area, power, latency, and temperature-dependent non-ideal effects etc. This variation provides the MOO solver with the scope of optimizing across multiple conflicting, yet crucial objectives namely: **latency**, **accuracy**, **area,** and **power**. We capture these objectives using three key performance evaluation metrics: energy-efficiency (TOPS/W), compute-efficiency (TOPS/mm^2), and DNN predictive accuracy. We select the best design d_{best} from the set of Pareto optimal designs that achieves the best performance in-terms of either TOPS/W or TOPS/mm^2, and negligible DNN accuracy loss (less than 1% accuracy loss compared to ideal accuracy).

Figure 4 shows a representative example of the Pareto front considering the four design objectives (latency, accuracy, area and power), for the ResNet34 DNN model trained with the CIFAR-10 dataset. In this illustration, the PE-to-tier mapping is represented by $\alpha = [t_1, t_2, \dots, t_z]$, where a planar tier t_z has PEs of one IMC device type – ReRAM (R), FeFET (F), SRAM (S). For example, the PE-to-tier mapping $[R_1, R_2, F_3, S_4]$ implies ReRAM-based PEs are mapped to the first two planar tiers (i.e., closer to the heat sink) Similarly, FeFET- and SRAM-based PEs are mapped to the 3rd tier and the 4th planar tier respectively. In this example, we have considered SRAM-, FeFET- and ReRAM-based IMC technologies,



(a) (b) (c)

Figure 5: Performance evaluation of the optimized 3D IMC architecture (RRFS) with state-of-the-art homogeneous and heterogeneous architectures in terms of (a) Energy-efficiency (TOPS/W), (b) Compute-efficiency (TOPS/mm^2), and (c) Accuracy of DNN workloads executed on CIFAR-10 dataset.

as examples to demonstrate the effectiveness of an optimized layer-to-PE and PE-to-tier mapping for a 3D heterogeneous IMC-based accelerator. It should be noted that other types of IMC devices such as PCMs and MRAMs can also be considered for heterogeneous systems. As shown in Figure 4, the Pareto optimal set of designs (D^*) are highlighted in black. Note that all the Pareto optimal configurations have the SRAM devices at the bottom tier, away from the heat sink. This SRAM tier is used for the gradient calculation during back-propagation phase [28]. Due to the necessity of the SRAM tier for the back-propagation, the homogeneous configurations where we have only one type of IMC device like FeFET or ReRAM, are: $[F_1, F_2, F_3, S_4]$ and $[R_1, R_2, R_3, S_4]$. Alternatively, the homogeneous configuration with only SRAM device is: $[S_1, S_2, S_3, S_4]$. These homogeneous architectures are referred to as FFFS, RRRS, SSSS respectively hereafter. The 3D heterogeneous IMC-based architecture $[R_1, R_2, F_3, S_4]$ (referred to as RRFS hereafter) exploits device heterogeneity with optimal layer-to-PE and PE-to-tier mapping and achieves the best trade-offs between power, latency, area and DNN accuracy compared to other Pareto optimal configurations.

In Figures 5(a) - 5(c), we evaluate the performance of the optimized 3D heterogeneous IMC-based architecture (RRFS) on five DNNs namely: VGG11, VGG16, ResNet18 (RN18), ResNet34 (RN34), and DenseNet40 (DN40) with the CIFAR-10 dataset in terms of TOPS/W, TOPS/mm^2 and DNN test accuracy. We compare the optimized RRFS heterogeneous architecture with three homogenous IMC-based architectures (FFFS, RRRS & SSSS) and state-of-the-art heterogeneous architectures (AccuReD and HyperX) [11] [34]. Here, AccuReD and HyperX architectures use $[R_1, R_2, GPU_3, GPU_4]$ and $[R_1, S_2, S_3, S_4]$ 3D tier configurations respectively. As shown in Figure 5(a) and 5(b), RRFS achieves up to 20 TOPS/W and 10.73 TOPS/mm^2 corresponding to 2.1× and 1.5× average improvement in terms of TOPS/W and TOPS/mm^2 respectively over the homogeneous architectures. Also, we demonstrate that the RRFS configuration achieves an average improvement of 3.1× (and 1.4×), and 2.7× (and 1.5×) over HyperX (and AccuReD) in terms of TOPS/mm^2 and TOPS/W respectively. As shown in Figure 5(c), the all-SRAM configuration (SSSS) achieves the highest accuracy due its high reliability, and less vulnerability to thermal issues in the 3D architecture. However, the homogeneous FeFET- and ReRAM-based counterparts (FFFS and RRRS) suffer up to 4% and 2.5% accuracy loss, due to the thermal noise in ReRAMs and FeFETs. Overall, the optimized RRFS configuration achieves less than 1% accuracy drop compared to the all-SRAM counterpart. Hence, the optimized 3D IMC architecture RRFS achieves the highest TOPS/W and TOPS/mm^2 with negligible loss in accuracy.

3.3 Heterogeneous Architectures for Transformers

Transformers have emerged as a focal point in the field of deep learning [36]. Transformers include various models that use multi-head attention layers e.g., large language models, vision transformers. The transformer architecture is composed of multiple sequential layers of encoder/decoder blocks, as shown in Figure 6. Each of these blocks consists of three major compute kernels: the multi-head attention (MHA), the feed-forward (FF) network, and the layer normalization block as shown in Figure 6.

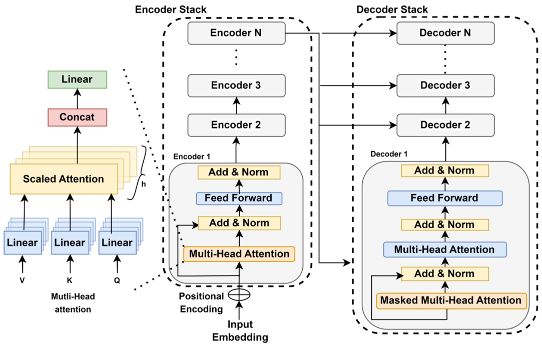


Figure 6: Transformer model architecture consisting of encoder and decoder stack of length *N*.

Designing hardware accelerators for transformer models is complex due to the variety of computing kernels used. Homogeneous IMC-based architectures struggle with performance and endurance issues, particularly with MHA kernels. The MHA kernels require dynamic operand multiplications, and thus necessitate a high frequency of write operations to IMC devices [37]. Since, IMC devices such as ReRAMs suffer from limited write endurance and high write latency, they are not suitable for MHA computation. In contrast, the FF network computation is independent of the input sequence length and can utilize IMC devices. To address these challenges, heterogeneous architectures have been proposed, such as Xformer, which uses SRAM to handle more frequently updated computation kernels [38]. CMOS-based solutions like TransPIM and HAIMA have also been proposed, that integrate DRAM with other memory types such as SRAM to improve performance [39] [40]. Recent developments also include 3D hybrid systems like H3D-Transformer consisting of FeFETs, SRAMs, and TPU cores [41]. Although, they demonstrate performance gains, they do not consider the challenges of thermal feasibility.

To address these challenges, we can extend the multi-objective formulation in (1) (discussed in Section 3.2) for transformers. The formulation must consider the requirements of the various transformer kernels and the properties of the IMC technologies to find a suitable heterogeneous IMC architecture, and this can be explored in future work.

4 Conclusion

In-memory computing (IMC) is an emerging paradigm that enables energy-efficient and high-performance acceleration of DNN workloads. However, standalone IMC technologies have their unique challenges, which limit their applicability to state-of-the-art artificial intelligence applications. To address this challenge, three-dimensional (3D) heterogeneous architectures facilitate the integration of more than one type of IMC device, as well as general-purpose processors into a single platform. However, when multiple IMC-based technologies are integrated to design a heterogeneous architecture, we need to address various challenges to establish suitable power, performance, area, and accuracy trade-offs. In this paper, we highlight the challenges of integrating multiple IMC devices in a 3D heterogeneous architecture. We also discuss the design space optimization using different IMC technologies and neural layer characteristics to achieve energy-efficient and high-performance acceleration of DNN workloads and Transformers.

ACKNOWLEDGMENTS

This work was supported, in part by the US National Science Foundation (NSF) under grants CNS-1955353 and CSR-2308530.

REFERENCES

[1] W. Liu et al. 2017. A survey of deep neural network architectures and their applications. *Neurocomputing,* vol. 234, 11-26.

[2] L. Song, X. Qian, L. Hai and Y. Chen. 2017. PipeLayer: A Pipelined ReRAM-Based Accelerator for Deep Learning. in *IEEE International Symposium on High-Performance Computer Architecture (HPCA)*.

[3] K. Roy, I. Chakraborty, M. Ali, A. Ankit and A. Agrawal. 2020. In-Memory Computing in Emerging Memory Technologies for Machine Learning: An Overview. in *IEEE Design Automation Conference (DAC)*.

[4] A. Shafiee et al. 2016. ISAAC: A Convolutional Neural Network Accelerator with In-Situ Analog Arithmetic in Crossbars Ali. in *International Symposium on Computer Architecture (ISCA)*.

[5] T. Soliman et al. 2023. First demonstration of in-memory computing crossbar using multi-level Cell FeFET. *Nature Communications,* vol. 14, no. 1, p. 6348.

[6] Y. Long et al. 2019. A Ferroelectric FET-Based Processing-in-Memory Architecture for DNN Acceleration. *IEEE Journal on Exploratory Solid-State Computational Devices and Circuits,* vol. 5.

[7] D. Niu et al. 2013. Design of cross-point metal-oxide ReRAM emphasizing reliability and cost. *In 2013 IEEE/ACM International Conference on Computer-Aided Design (ICCAD),* pp. 17-23.

[8] X. Peng et al. 2020. Benchmarking monolithic 3D integration for compute-in-memory accelerators: overcoming ADC bottlenecks and maintaining scalability to 7nm or beyond. in *IEDM*.

[9] A. Kaul et al. 2022. 3-d heterogeneous integration of rram-based compute-in-memory: Impact of integration parameters on inference accuracy. *IEEE Transactions on Electron Devices,* pp. 485-92.

[10] B. K. Joardar et al. 2020. AccuReD: High Accuracy Training of CNNs on ReRAM/GPU Heterogeneous 3D Architecture. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*.

[11] W. Xu et al. 2021. Parallelizing DNN training on GPUs: Challenges and opportunities. *In Companion Proceedings of the Web Conference* , pp. 174-178.

[12] C. Guo et al.. 2020. Balancing efficiency and flexibility for DNN acceleration via temporal GPU-systolic array integration. *In 2020 57th ACM/IEEE Design Automation Conference (DAC)*.

[13] R. Xu et al. 2023. A Survey of Design and Optimization for Systolic Array-based DNN Accelerators. *ACM Computing Surveys,* vol. 56, no. 1, pp. 1-37.

[14] E. Roorda et al. 2022. FPGA architecture exploration for DNN acceleration. *ACM Transactions on Reconfigurable Technology and Systems (TRETS),* vol. 15, no. 3, pp. 1-37.

[15] Z. Su et al. 2024. An Ultra-Low Cost and Multicast-Enabled Asynchronous NoC for Neuromorphic Edge Computing. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*.

[16] Y. Jiang et al. 2020. A unified architecture for accelerating distributed {DNN} training in heterogeneous {GPU/CPU} clusters. *In 14th USENIX Symposium on Operating Systems Design and Implementation (OSDI 20),* pp. 463-479.

[17] G. Narang, C. Ogbogu, J. Doppa and P. Pande. 2024. TEFLON: Thermally Efficient Dataflow-Aware 3D NoC for Accelerating CNN Inferencing on Manycore PIM Architectures. *ACM Transactions on Embedded Computing Systems*.

[18] H. Sharma, L. Pfromm, R. Topaloglu, J. Doppa, U. Ogras, A. Kalyanraman and P. Pande. 2023. Florets for Chiplets: Data Flow-aware High-Performance and Energy-efficient Network-on-Interposer for CNN Inference Tasks. *ACM Transactions on Embedded Computing Systems,* vol. 22(5s), pp. 1-21.

[19] C. Eckert et al. 2018. Neural Cache: Bit-Serial In-Cache Acceleration of Deep Neural Networks. in *45th Annual International Symposium on Computer Architecture (ISCA)*.

[20] S. Spetalnick and A. Raychowdhury. 2022. A Practical Design-Space Analysis of Compute-in-Memory With SRAM. *IEEE Transactions on Circuits and Systems I: Regular Papers,* vol. 69.

[21] A. Bhattacharjee, A. Moitra and P. Panda. 2023. HyDe: A Hybrid PCM/FeFET/SRAM Device-Search for Optimizing Area and Energy-Efficiencies in Analog IMC Platforms. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems,* vol. 13.

[22] S. Roy, M. Ali and A. Raghunathan. 2021. PIM-DRAM: Accelerating machine learning workloads using processing in commodity DRAM. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems,* vol. 11, no. 4, pp. 701-710.

[23] V. Seshadri et al. 2017. Ambit: In-memory accelerator for bulk bitwise operations using commodity DRAM technology *In Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture.*

[24] A. Keshavarzi, K. Ni, W. Van Den Hoek, S. Datta and A. Raychowdhury. 2020. FerroElectronics for Edge Intelligence. *IEEE Micro,* vol. 40.

[25] T. Kim et al. 2020. Evolution of phase-change memory for the storage-class memory and beyond. *IEEE Transactions on Electron Devices,* vol. 67, no. 4, pp. 1394-1406.

[26] A. Yusuf, T. Adegbija and D. Gajaria. 2024. Domain-Specific STT-MRAM-Based In-Memory Computing: A Survey. *IEEE Access,* vol. 12.

[27] X. Peng et al. 2020. DNN+NeuroSim V2.0: An end-to-end benchmarking framework for compute-in-memory accelerators for on-chip training. *arXiv:2003.06471*.

[28] C. Ogbogu, M. Soumen, B. Joardar, J. Doppa, D. Heo, K. Chakrabarty and P. Pande. 2023. Energy-Efficient ReRAM-Based ML Training via Mixed Pruning and Reconfigurable ADC. *IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED),* pp. 1-6.

[29] S. Lee, G. Jung, M. E. Fouda, J. Lee, A. Eltawil and F. Kurdahi. 2020. Learning to Predict IR Drop with Effective Training for ReRAM-based Neural Network Hardware. *In 2020 57th ACM/IEEE Design Automation Conference (DAC),* pp. 1-6.

[30] T.-H. Yang, H.-Y. Cheng, C.-L. Yang, I.-C. Tseng, H.-W. Hu, H.-S. Chang and H.-P. Li. 2019. Sparse reram engine: Joint exploration of activation and weight sparsity in compressed neural networks. *In Proceedings of the 46th International Symposium on Computer Architecture,* pp. 236-249.

[31] W. Chen et al. 2018. A 65nm 1Mb nonvolatile computing-in-memory ReRAM macro with sub-16ns multiply-and-accumulate for binary DNN AI edge processors. *In 2018 IEEE International Solid-State Circuits Conference-(ISSCC),* pp. 494-496.

[32] M. R. Haq Rashed, S. K. Jha and R. Ewetz. 2021. Hybrid Analog-Digital In-Memory Computing. in *IEEE/ACM International Conference On Computer Aided Design (ICCAD)*.

[33] G. Krishnan et al. 2022. Hybrid RRAM/SRAM in-Memory Computing for Robust DNN Acceleration. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* vol. 41.

[34] A. Kosta et al. 2022. HyperX: A Hybrid RRAM-SRAM partitioned system for error recovery in memristive Xbars. in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*.

[35] S. Bandyopadhyay, S. Saha, U. Maulik and K. Deb. 2008. A simulated annealing-based multiobjective optimization algorithm: AMOSA. *IEEE transactions on evolutionary computation,* vol. 12(3), pp. 269-283.

[36] T. Lin, Y. Wang, X. Liu and X. Qiu. 2021. A Survey of Transformers. in *ArXiv preprint ArXiv:2106.04554*.

[37] P. Dhingra et al. 2024. HeTraX: Energy Efficient 3D Heterogeneous Manycore Architecture for Transformer Acceleration. *arXiv preprint arXiv:2408.03397*.

[38] S. Sridharan, J. Stevens, K. Roy and A. Raghunathan. 2023. X-Former: In-Memory Acceleration of Transformers. *IEEE TVLSI*.

[39] M. Zhou, W. Xu, J. Kang and T. Rosing. 2022. TransPIM: A Memory-based Acceleration via Software-Hardware Co-Design for Transformer. in *HPCA*, Korea.

[40] Y. Ding et al. 2023. HAIMA: A Hybrid SRAM and DRAM Accelerator-in-Memory Architecture for Transformer. in *DAC*.

[41] Y. Luo and S. Yu. 2024. H3D-Transformer: A Heterogeneous 3D (H3D) Computing Platform for Transformer Model Acceleration on Edge Devices. in *ACM TOADES*.