DeepNVM++: Cross-Layer Modeling and Optimization Framework of Nonvolatile Memories for Deep Learning

Ahmet Inci, Mehmet Meric Isgenc, and Diana Marculescu, *Fellow, IEEE*

***Abstract*—Nonvolatile memory (NVM) technologies, such as spin-transfer torque magnetic random access memory (STT-MRAM) and spin-orbit torque magnetic random access memory (SOT-MRAM), have significant advantages compared to conventional SRAM due to their nonvolatility, higher cell density, and scalability features. While previous work has investigated several architectural implications of NVM for generic applications, in this work, we present *DeepNVM++*, a *framework* to characterize, model, and analyze NVM-based caches in GPU architectures for deep learning (DL) applications by combining technology-specific circuit-level models and the actual memory behavior of various DL workloads. We present both *iso-capacity* and *iso-area* performance and energy analysis for systems whose last-level caches rely on conventional SRAM and emerging STT-MRAM and SOT-MRAM technologies. In the iso-capacity case, STT-MRAM and SOT-MRAM provide up to 3.8× and 4.7× energy-delay product (EDP) reduction and 2.4× and 2.8× area reduction compared to conventional SRAM, respectively. Under iso-area assumptions, STT-MRAM and SOT-MRAM provide up to 2× and 2.3× EDP reduction and accommodate 2.3× and 3.3× cache capacity when compared to SRAM, respectively. We also perform a scalability analysis and show that STT-MRAM and SOT-MRAM achieve orders of magnitude EDP reduction when compared to SRAM for large cache capacities. Our comprehensive cross-layer framework is demonstrated on STT-/SOT-MRAM technologies and can be used for the characterization, modeling, and analysis of *any* NVM technology for last-level caches in GPUs for DL applications.**

***Index Terms*—Convolutional neural networks (CNNs), deep learning (DL), deep neural networks (DNNs), GPU architectures, magnetic random access memory (MRAM), nonvolatile memory (NVM), spin-orbit-torque MRAM (SOT-MRAM), spin-transfer-torque MRAM (STT-MRAM), SRAM.**

I. Introduction

Over the last decade, the performance boost achieved through CMOS scaling has plateaued, necessitating sophisticated computer architecture solutions to gain higher performance in computing systems while maintaining a feasible power density. These objectives, however, are concurrently challenged by the limitations of the performance of memory resources [1]. In contrast to the initial insight of Dennard on power density [2], deep CMOS scaling has exacerbated static power consumption, causing the heat density of ICs to reach catastrophic levels unless properly addressed [3]–[5].

As computers suffer from memory and power-related limitations, the demand for data-intensive applications has been on the rise. With the increasing data deluge and recent improvements in GPU architectures, deep neural networks (DNNs) have achieved remarkable success in various tasks, such as image recognition [6], [7], object detection [8], and chip placement [9] by utilizing inherent massive parallelism of GPU platforms. However, DNN workloads continue to have large memory footprints and significant computational requirements to achieve higher accuracy. Thus, DNN workloads exacerbate the memory bottleneck which degrades the overall performance of the system. To this end, while deep learning (DL) practitioners focus on model compression techniques [10]–[12], system architects investigate hardware architectures to overcome the memory bottleneck problem and improve the overall system performance [13]–[20]. We note the current trend of GPU architectures is toward increasing last-level cache capacity as shown in Fig. 1. Our analysis shows that the conventional SRAM technology incurs scalability problems as far as power, performance, and area (PPA) is concerned [19], [21]–[23]. Nonvolatile memory (NVM) technology is one of the most promising solutions to tackle the memory bottleneck problem for data-intensive applications [24].

However, because much of emerging NVM technology is not available for commercial use, there is an obvious need for a framework to perform design space exploration for these emerging NVM technologies for DL workloads.

In this work, we present *DeepNVM++*, an extended and improved framework [17] to characterize, model, and optimize NVM-based caches in GPU architectures for DL workloads. Without loss of generality, we demonstrate our framework for spin-transfer torque magnetic random access memory

Manuscript received 8 November 2020; revised 22 March 2021; accepted 16 October 2021. Date of publication 11 November 2021; date of current version 20 September 2022. This work was supported in part by NSF CCF under Grant 1815899, and in part by NSF CSR under Grant 1815780. This article was recommended by Associate Editor F. Liu. *(Corresponding author: Ahmet Inci.)*

Ahmet Inci is with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: ainci@andrew.cmu.edu).

Mehmet Meric Isgenc was with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213 USA. He is now with Apple Inc., Cupertino, CA, USA (e-mail: misgenc@alumni.cmu.edu).

Diana Marculescu is with the Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, TX 78712 USA, and also with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: dianam@utexas.edu).

Digital Object Identifier 10.1109/TCAD.2021.3127148

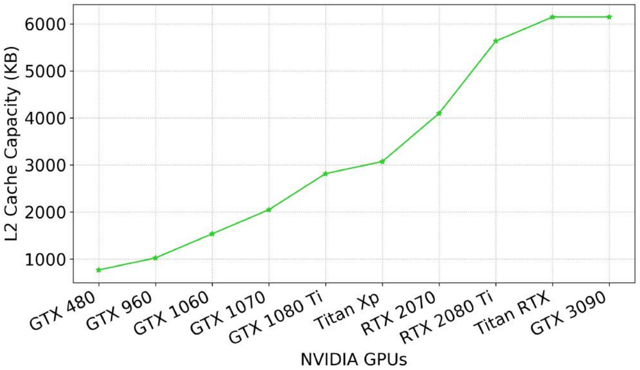


Fig. 1. L2 cache capacity in recent NVIDIA GPUs [25].

(STT-MRAM) and spin-orbit torque magnetic random access memory (SOT-MRAM), keeping in mind that it can be used for any NVM technology, GPU platform, or DL workload. Our cross-layer analysis framework incorporates both circuit-level characterization aspects and the memory behavior of various DL workloads running on an actual GPU platform. *DeepNVM++* enables the evaluation of *power, performance, and area* of NVMs when used for last-level (L2) caches in GPUs and seeks to exploit the benefits of this emerging technology to improve the performance of DL applications.

To perform *iso-capacity* analysis, we carry out extensive memory profiling of various DL workloads for both training and inference on existing GPU platforms. For the *iso-area* analysis, existing platforms cannot be used for varying cache sizes, so we rely on architecture-level simulation of GPUs to quantify and better understand last-level cache capacity and off-chip memory accesses. In both cases, our framework automatically combines resulting memory statistics with circuit and microarchitecture-level characterization and analysis of emerging NVM technologies to gauge their impact on DL workloads running on future GPU-based platforms.

II. Related Work and Article Contributions

Although 16 nm has become a commonplace technology for high-end customers of foundries, an intriguing inflection point awaits the electronics community as we approach the end of the traditional density, power, and performance benefits of CMOS scaling. To move beyond the computing limitations imposed by staggering CMOS scaling trends, magnetic random access memory (MRAM) has emerged as a promising candidate.

The enabling technology of MRAM consists of magnetic tunnel junction (MTJ) pillars that can store data as a resistive state. An MTJ pillar consists of a thin oxide film sandwiched by two ferromagnetic layers. One of these ferromagnetic layers has a fixed magnetization which serves as a reference layer. The magnetization of the other layer can be altered by changing the direction of the current that flows through the pillar. If the magnetization of the free layer and the reference layer is in parallel, the device is in the low resistance state. If the magnetization of layers is in opposite directions, the device is in the high resistance state [26].

STT bitcells [27] use an MTJ pillar as their core storage element and an additional access transistor to enable read and write operations. Although STT bitcells offer nonvolatility, low read latency, and high endurance [28], the write current is also high [29]–[31], which increases power consumption. To this end, SOT bitcells have been proposed to overcome the write current challenges by isolating the read and write paths [32]. Because the read disturbance errors are much less likely in SOT bitcells, both read and write access devices can be tuned in accordance with the lower current requirements [33], [34]. The read and write current requirements of STT and SOT bitcells can have a crucial impact on the eventual MRAM characteristics because they affect the CMOS access transistors, bitcell area, and peripheral logic. Thus, a comparison of these bitcells and the traditional SRAM merits a meticulous analysis that takes these factors into account.

Prior work has proposed effective approaches to overcome the shortcomings of emerging NVM technologies such as using hybrid SRAM and NVM-based caches that utilize the complementary features of different memory technologies [35]–[38], relaxing nonvolatility properties to reduce the high write latency and energy [39]–[42], and implementing cache replacement policies [43]–[45] for higher level caches, such as L1 caches and register files. However, NVM technology appears to be a better choice for lower level caches such as L2 or L3 caches due to its long write latency and high cell density. Higher level L1 caches are latency-sensitive and optimized for performance, whereas last-level caches are capacity-sensitive and optimized for a high hit rate to reduce off-chip memory accesses. Therefore, NVM-based caches provide a better use case for replacing SRAM in last-level caches due to their high cell density when compared to SRAM-based caches. To this end, we evaluate the PPA of NVM technology when used for last-level caches in GPU platforms.

While prior work has shown the potential of NVM technologies for generic applications to some extent, there is a need for a cross-layer analysis framework to explore the potential of NVM technologies in GPU platforms, particularly for DL workloads. The most commonly used modeling tool for emerging NVM technologies is *NVSim* [46], a circuit-level model for performance, energy, and area estimation. However, *NVSim* is not sufficient to perform a detailed cross-layer analysis for NVM technologies for DL workloads since it does not take architecture-level analysis and application-specific memory behavior into account. To this end, prior work has proposed cross-layer evaluation frameworks for nontraditional architectures such as processing-in-memory-based analog and digital architectures [47]–[49]. However, there is still a need for a cross-layer analysis framework to perform design space exploration of NVM technologies for GPU architectures for DL workloads. In this article, we incorporate *NVSim* with our cross-layer modeling and optimization flow including novel architecture-level iso-capacity and iso-area analysis flow to perform design space exploration for conventional SRAM and emerging NVM caches for DL workloads running on GPU architectures. We make the following contributions.

1) *Circuit-Level Bitcell Characterization:* We perform detailed circuit-level characterization combining a commercial 16-nm CMOS technology and prominent STT [50] and SOT [51] models from the literature to

iterate through our framework in an end-to-end manner to demonstrate the flexibility of *DeepNVM++* for future studies.

2) *Microarchitecture-Level Cache Design Exploration:* We use *NVSim* [46] to perform a fair comparison between SRAM, STT-MRAM, and SOT-MRAM by incorporating the circuit-level models developed in 1) using 16-nm technology and choosing the best cache configuration for each of them.
3) *Iso-Capacity Analysis:* To compare the efficacy of MRAM caches to conventional SRAM caches, we perform our novel iso-capacity analysis based on *actual platform profiling* results for the memory behavior of various DNNs by using the *Caffe* framework [52] on a high-end NVIDIA 1080 Ti GPU (implemented in 16-nm technology) for the ImageNet dataset [53].
4) *Iso-Area Analysis:* Because of their different densities, we compare SRAM and NVM caches in an iso-area analysis to quantify the benefits of higher density of NVM technologies on DL workloads running on GPU platforms. Since existing platforms do not support resulting iso-area cache sizes, we extend the GPGPU-Sim [54] simulator to run DL workloads and support larger cache capacities for STT-MRAM and SOT-MRAM.
5) *Scalability Analysis:* Finally, we perform a thorough scalability analysis and compare SRAM, STT-MRAM, and SOT-MRAM in terms of PPA to project and gauge the efficacy of NVM and SRAM-based caches for DL workloads as cache capacity increases.

To the best of our knowledge, putting everything together, *DeepNVM++* is the *first comprehensive framework* for cross-layer characterization, modeling, and analysis of emerging NVM technologies for DL workloads running on GPU platforms. Our results show that in the iso-capacity case, STT-MRAM and SOT-MRAM achieve up to 3.8× and 4.7× *energy-delay product (EDP) reduction* and 2.4× and 2.8× *area reduction* compared to SRAM baseline, respectively. In the iso-area case, STT-MRAM and SOT-MRAM achieve up to 2× and 2.3× *EDP reduction* and accommodate 2.3× and 3.3× *cache capacity* compared to SRAM, respectively.

The remainder of this article is organized as follows. In Section III, we describe the details of our methodology from circuit to microarchitecture-level characterization, modeling, and analysis to obtain SRAM, STT-MRAM, and SOT-MRAM cache parameters. We also detail our iso-capacity and iso-area analysis methodology. In Section IV, we show experimental results demonstrating the efficiency of STT-MRAM and SOT-MRAM over the conventional SRAM for iso-capacity and iso-area cases. Furthermore, we perform a scalability analysis and show the PPA of SRAM, STT-MRAM, and SOT-MRAM. Next, we discuss the implications of the results shown in this article in Section V. Finally, Section VI concludes this article by summarizing the results.

III. Methodology

In this section, we present our cross-layer analysis framework, as shown in Fig. 2. First, we show our detailed circuit-level characterization analysis using CMOS, STT, and SOT device models (Section III-A). After developing bitcell models, we present our microarchitecture-level cache design methodology to obtain cache area, latency, and energy results (Section III-B). Next, we describe our iso-capacity analysis flow in which we gather actual memory statistics through GPU profiling (Section III-C). Finally, we detail our iso-area analysis in which we extend GPGPU-Sim to run DL workloads and support larger cache capacities for STT-MRAM and SOT-MRAM (Section III-D).

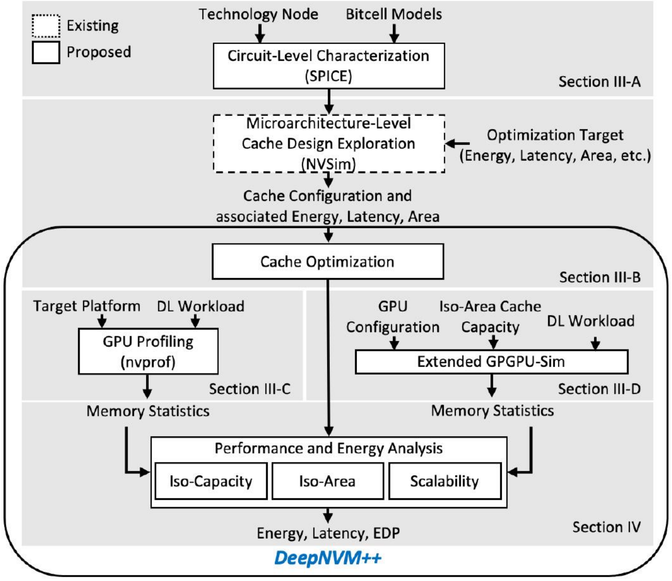


Fig. 2. Overview of the cross-layer analysis flow.

A. Circuit-Level NVM Characterization

A vast majority of work in the literature uses simple bitcell models [33] to assess the PPA of corresponding cache designs. Because bitcells are the core components of the memory, the methodology to calculate the bitcell latency, energy, and area is crucial for accurate comparisons. To this end, we use a commercial 16-nm bitcell design as a baseline as we model the STT and SOT bitcells. This technology node also matches the fabrication technology of the GPU platform that we use to gather actual memory statistics in Section III-C.

The key bitcell parameters needed for cache modeling are read and write currents and latency values for high-to-low and low-to-high resistive transitions. These parameters can be optimized by tuning the size of the access transistors. While larger access transistors enable faster reads and writes, they increase the energy consumption and the bitcell layout size. The optimal sizing of the access transistor and the array architecture varies based on the bitcell type. The access transistor sizing optimization is crucial since it impacts the eventual PPA characteristics of the bitcell and the cache. To address the array architecture differences between STT and SOT MRAM for a fair comparison, we performed transient simulations.

For our simulations, we used perpendicular to the plane STT [50] and SOT [51] models and a commercial 16-nm FinFET model that takes post-layout effects into account. To

TABLE I
STT-MRAM AND SOT-MRAM BITCELL PARAMETERS AFTER DEVICE-LEVEL CHARACTERIZATION

	STT-MRAM	SOT-MRAM
Sense Latency (ps)	650	650
Sense Energy (pJ)	0.076	0.020
Write Latency (ps)	8400 (set) / 7780 (reset)	313 (set) / 243 (reset)
Write Energy (pJ)	1.1 (set) / 2.2 (reset)	0.08 (set) / 0.08 (reset)
Fin Counts	4 (read/write)	3 (write) + 1 (read)
Area (normalized)	0.34*	0.29*

*: Area is normalized with respect to the foundry SRAM bitcell

find the latency and energy parameters, we used parametrized SPICE netlists wherein the read/write pulse widths were modulated to the point of failure. Furthermore, we swept a range of fin counts for the access devices to find the optimal balance between the latency, energy, and area. For the transient SPICE simulations, we picked the FinFET models corresponding to the worst delay and power scenarios. To calculate the bitcell area for the 16-nm layout design rules, we used the bitcell area formulations provided in prior work [55].

We summarize the obtained bitcell parameters in Table I. The sensing delay is measured from wordline activation to the point where the bitline voltage difference reaches 25 mV. The sense energy is the integration of the power consumed over the sensing time window. For both magnetic flavors, the sense delay is similar; however, SOT-MRAM is more energy efficient in terms of read operation owing to the separation of the read/write terminals. The write latency in this context refers to the time between the arrival of the write enable signal to the access transistor and a complete magnetization change for the MTJ. The write latencies for STT and SOT bitcells are significantly different, as expected. This difference can be seen in the energy values as well. The access device is more than double the width of the technology minimum device in order to enable a larger current flow to the STT bitcell, causing the 1T1R STT bitcell to occupy a larger area than the 2T1R SOT bitcell. The isolation of the read and the write terminals in the SOT bitcell allows for a smaller write access device. The area values are normalized by the foundry bitcell area. We highlight the significant area difference and demonstrate its impact on the cache characteristics in Section III-B. We use these bitcell parameters for energy-delay-area product (EDAP) optimized cache design exploration as discussed in the next section.

B. Microarchitecture-Level Cache Design Exploration

In order to demonstrate the impact of using STT and SOT bitcells in L2 caches, we use *NVSim* [46], a circuit-level analysis framework that delivers energy, latency, and area results. After developing *NVSim*-compatible bitcell models as described in Section III-A, we analyzed a range of cache capacities (1–32 MB) for all possible configurations and cache access types to demonstrate the potential of STT-MRAM and SOT-MRAM as the cache capacity tends to grow. Such a scalability study will help in determining the benefits of switching from conventional SRAM to NVM-based caches in future GPU platforms as depicted by the trend in Fig. 1.

Algorithm 1: EDAP-Optimal Cache Tuning Algorithm

```
$mem \in \mathcal{M} = \{SRAM, STT, SOT\}$;
$cap \in \mathcal{C} = \{1, 2, 4, 8, 16, 32\}$;
$opt \in \mathcal{O} = \{Read_{Latency}, Write_{Latency}, Read_{Energy},$
$\ldots Write_{Energy}, Read_{EDP}, Write_{EDP}, Area, Leakage\}$;
$acc \in \mathcal{A} = \{Normal, Fast, Sequential\}$;
for each $mem \in \mathcal{M}$ do
    for each $cap \in \mathcal{C}$ do
        $Q^{'} \leftarrow \infty$;
        for each $opt \in \mathcal{O}$ do
            for each $acc \in \mathcal{A}$ do
                $Q \leftarrow calculate(EDAP)$;
                if $Q < Q^{'}$ then
                    $Q^{'} \leftarrow Q$;
                end
            end
        end
        $TunedConfig.append(argv(Q))$;
    end
end
return $TunedConfig$;
```

TABLE II
LATENCY, ENERGY, AND AREA RESULTS FOR SRAM, STT-MRAM, AND SOT-MRAM CACHES FOR ISO-CAPACITY AND ISO-AREA

	SRAM	STT-MRAM		SOT-MRAM	
		Iso-Capacity	Iso-Area	Iso-Capacity	Iso-Area
Capacity (MB)	3	3	7	3	10
Read Latency (ns)	2.91	2.98	4.58	3.71	6.69
Write Latency (ns)	1.53	9.31	10.06	1.38	2.47
Read Energy (nJ)	0.35	0.81	0.93	0.49	0.51
Write Energy (nJ)	0.32	0.31	0.43	0.22	0.40
Leakage Power (mW)	6442	748	1706	527	1434
Area (mm^2)	5.53	2.34	5.12	1.95	5.64

Algorithm 1 depicts the EDAP-optimal cache tuning algorithm. Based on the optimization target used in *NVSim*, the cache PPA values vary substantially. Therefore, we independently choose the best configuration for each type of memory technology in terms of EDAP metric to perform a fair comparison that encompasses all and not just one of the design constraint dimensions.

As described in Section III-A, we use a commercial 16-nm bitcell design. To ensure a correct analysis, we modified the internal technology file of *NVSim* to the corresponding 16-nm technology parameters. Next, we compare SRAM, STT-MRAM, and SOT-MRAM for various cache capacities in terms of area, latency, and energy results. Based on these, we determine the EDAP for the cache (as denoted by *calculate(EDAP)* in Algorithm 1).

Table II shows the latency, energy, and area results that correspond to the cache capacity of 1080 Ti GPU (3 MB) and to the larger MRAM caches that fit into the same area of SRAM baseline. We convert read and write latencies to clock cycles based on 1080 Ti GPU's clock frequency for our calculations. For STT-MRAM and SOT-MRAM, we show parameters for both iso-capacity and iso-area when compared

TABLE III
CONFIGURATIONS FOR DNNS UNDER CONSIDERATION

	AlexNet	GoogLeNet	VGG-16	ResNet-18	SqueezeNet
Top-5 error	16.4	6.7	7.3	10.71	16.4
CONV Layers	5	57	13	17	26
FC Layers	3	1	3	1	0
Total Weights	61M	7M	138M	11.8M	1.2M
Total MACs	724M	1.43G	15.5G	2G	837M

to SRAM. We use these parameters to evaluate the workload-dependent impact of memory choices using DL workloads with diverse structures and multiply–accumulate operations (MACs) configurations.

The energy and latency benefits of STT-MRAM and SOT-MRAM depend on the data characteristics of a given workload. To account for differences in the data-related read/write characteristics, we used a simple model where we multiply the number of read and write transactions by the corresponding latency and energy values for those operations.

Implications in Architecture-Level Analysis: To gauge the benefits of using the MRAM technology, we consider two scenarios: 1) first, one could replace the SRAM cache in a GPU with the same capacity MRAM with a smaller area and 2) alternatively, by using the same area dedicated to the cache, one can increase the on-chip cache capacity, thereby reducing costly DRAM traffic. We analyze and discuss both approaches through platform profiling results for iso-capacity scenario and a set of architecture-level simulations for iso-area scenario.

C. Architecture-Level Iso-Capacity Analysis

As the platform target to demonstrate our work, we use a high-end 1080 Ti GPU which is fabricated in a commercial 16-nm technology node that also matches our bitcell and cache models. We use the *Caffe* [52] framework to run various DNNs, such as AlexNet [56], GoogLeNet [57], VGG-16 [58], ResNet-18 [59], and SqueezeNet [60] for the ImageNet [53] dataset as shown in Table III. Our analysis is generalizable to other types of neural network architectures since we cover a wide range of DNN configurations with various workload characteristics. We use the NVIDIA profiler [61] to obtain the device memory and L2 cache read and write transactions to better understand both on-chip and off-chip memory behavior of DNN workloads.

D. Architecture-Level Iso-Area Analysis

Since the iso-area larger capacities enabled by higher density NVM implementations do not exist in existing platforms, we use *GPGPU-Sim* [54] to explore the power and performance implications of having these larger L2 caches in GPU architectures for DNN workloads. For comparison, we model the high-end GTX 1080 Ti GPU. The configurations for 1080 Ti GPU are shown in Table IV. We extend the *GPGPU-Sim* simulator to support the cache capacity of GTX 1080 Ti GPU. This GPU is built using a commercial 16-nm technology node that matches our bitcell and cache models. In particular, for *GPGPU-Sim* compatibility, we set L2 cache capacity to 3 MB. We use this capacity for our analysis in the remainder of this article. We measure the number of DRAM transactions to quantify and better understand the relationship between larger L2 caches and the overall system power and performance. As a DNN benchmark, we use AlexNet [56] with the ImageNet [53] dataset which is provided by the *DarkNet* [62] framework. We extend the *DarkNet* source code to enable DL workloads on *GPGPU-Sim*.

TABLE IV
GPGPU-SIM CONFIGURATIONS

	GTX 1080 Ti
Number of Cores	28
Number of Threads/Core	2048
Number of Registers/Core	65536
L1 Data Cache	48 KB, 128 B line, 6-way LRU
L2 Data Cache	128 KB/channel, 128 B line, 16-way LRU
Instruction Cache	8 KB, 128 B line, 16-way LRU
Number of Schedulers / Core	4
Frequency (MHz): Core, Interconnect, L2, Memory	1481, 2962, 1481, 2750

IV. RESULTS

We analyze STT-MRAM and SOT-MRAM in terms of energy, performance, and area results by using GPU profiling results for both iso-capacity and iso-area cases in Sections IV-A and IV-B, respectively. In Section IV-B, we use iso-area cache parameters as shown in Table II and we use *GPGPU-Sim* to quantify the DRAM access reduction in the iso-area case at larger cache capacities. We include DRAM accesses in our performance and energy calculations for iso-area case. In Section IV-C, we perform a scalability analysis to project the implications of the current GPU trend shown in Fig. 1 on performance and energy results.

A. Performance and Energy Results for Iso-Capacity

By combining the actual technology-dependent latency and energy metrics from Table II, we can perform a performance and energy analysis for replacing conventional SRAM caches with MRAM caches. We choose batch size 4 for inference and 64 for training for our workloads as it is typically used in related work [63].

Fig. 3 shows normalized dynamic energy and leakage energy breakdown results for 1080 Ti GPU based on actual platform memory statistics and our MRAM cache models at the same cache capacity. We use our cache parameters and profiling results to calculate results for various DNNs for both inference and training workloads.

In Fig. 3, we observe that STT-MRAM has 2.1× dynamic energy whereas SOT-MRAM has 1.3× dynamic energy on average when compared to SRAM baseline. Furthermore, our results show that 83% of the total dynamic energy of SRAM comes from read operations whereas write operations only make for 17% of all transactions on average across all workloads. Our profiling results also support these findings as read operations dominate write operations in these DL workloads.

On the other hand, Fig. 3 also shows that STT-MRAM and SOT-MRAM provide 5.9× and 10× lower leakage energy

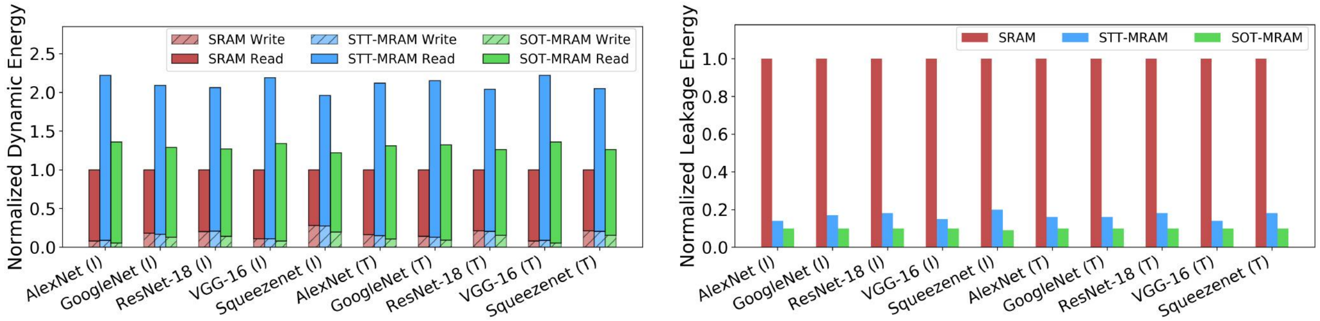


Fig. 3. Dynamic energy (left chart) and leakage energy (right chart) (lower is better) normalized with respect to SRAM by using NVMs with iso-capacity (3 MB) for inference (*I*) and training (*T*) stages.

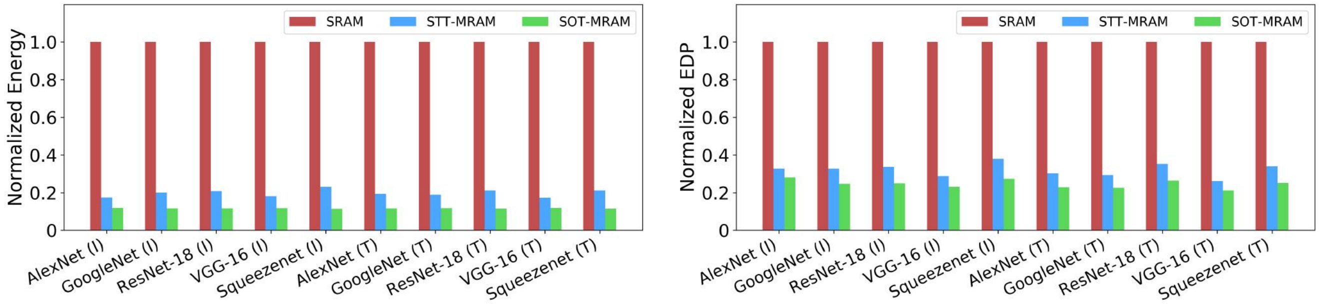


Fig. 4. Iso-capacity (3 MB) energy and EDP for NVM-based caches (lower is better) normalized with respect to SRAM-based caches for inference (*I*) and training (*T*) stages. DRAM energy and latency are also included in EDP results.

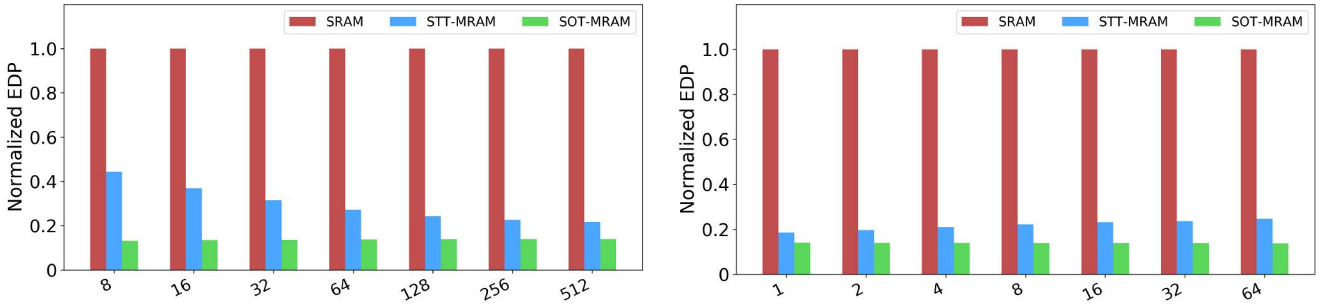


Fig. 5. Impact of batch size on EDP (lower is better) normalized with respect to SRAM by using NVMs with iso-capacity (3 MB) for AlexNet for training (left chart) and inference (right chart).

on average when compared to SRAM, respectively. Based on this result, Fig. 4 shows significant total normalized energy reduction of STT-MRAM and SOT-MRAM when compared to SRAM given that leakage energy dominates the total energy. In more detail, STT-MRAM and SOT-MRAM achieve 5.1× and 8.6× *energy reduction* on average across all workloads compared to SRAM baseline, respectively, due to their significantly low leakage energy. Moreover, Fig. 4 shows that STT-MRAM and SOT-MRAM provide up to 3.8× and 4.7× *EDP reduction* and 2.4× and 2.8× *area reduction*, respectively.

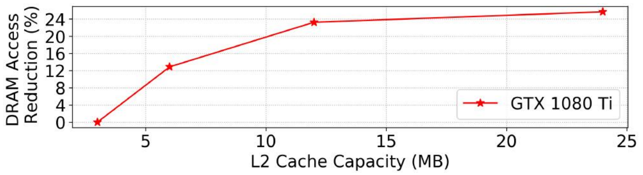


Fig. 6. Simulation results for the reduction in the total number of DRAM accesses in percentage (%).

Impact of Batch Size on EDP: We perform this study to better understand the relationship between batch size and its implications for performance and energy results of SRAM, STT-MRAM, and SOT-MRAM. Fig. 5 shows the impact of batch size on EDP results for AlexNet during training and inference stages based on 1080 Ti memory profiling statistics. We show that batch size significantly affects the improvement of STT-MRAM and SOT-MRAM for training. For training, STT-MRAM provides 2.3× to 4.6× EDP reduction as batch size increases. On the other hand, SOT-MRAM provides 7.2× to 7.6× EDP reduction when compared to the SRAM baseline. For inference, STT-MRAM and SOT-MRAM achieve 4.1× to 5.4× and 7.1× to 7.3× EDP reduction, respectively. These results also confirm the different workload characteristics of training and inference. STT-MRAM provides higher EDP reduction for training workloads as batch size increases.

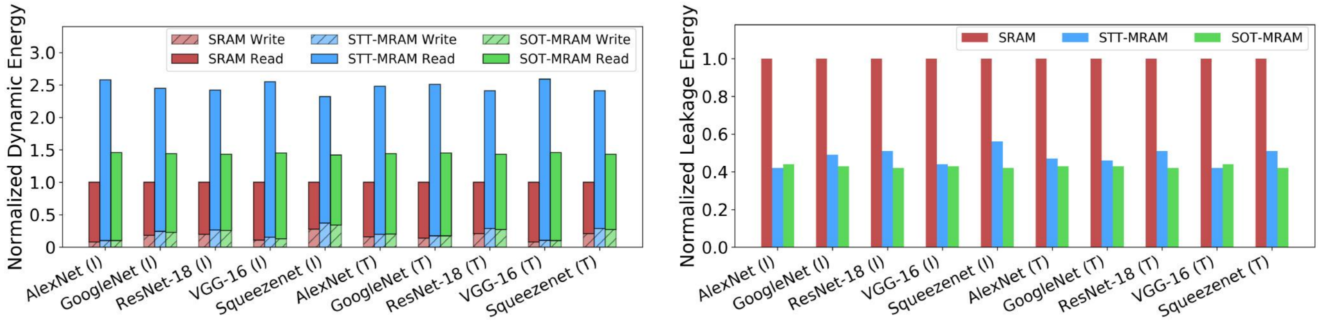


Fig. 7. Dynamic energy (left chart) and leakage energy (right chart) (lower is better) normalized with respect to SRAM by using STT-MRAM (7 MB) and SOT-MRAM (10 MB) with iso-area for inference (*I*) and training (*T*) stages.

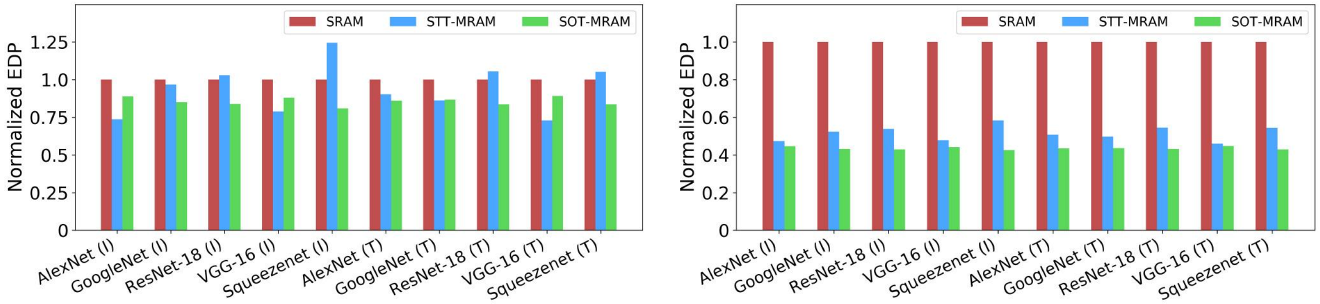


Fig. 8. Iso-area EDP results for STT-MRAM (7 MB) and SOT-MRAM (10 MB) (lower is better) normalized with respect to SRAM-based caches for inference (*I*) and training (*T*) stages without (left chart) and with (right chart) DRAM energy and latency.

On the other hand, SOT-MRAM follows the same pattern for inference workloads due to their different access characteristics as shown in Table II. We observe that training workloads become more read dominant whereas inference workloads have a lower read/write ratio as batch size increases.

B. Performance and Energy Results for Iso-Area

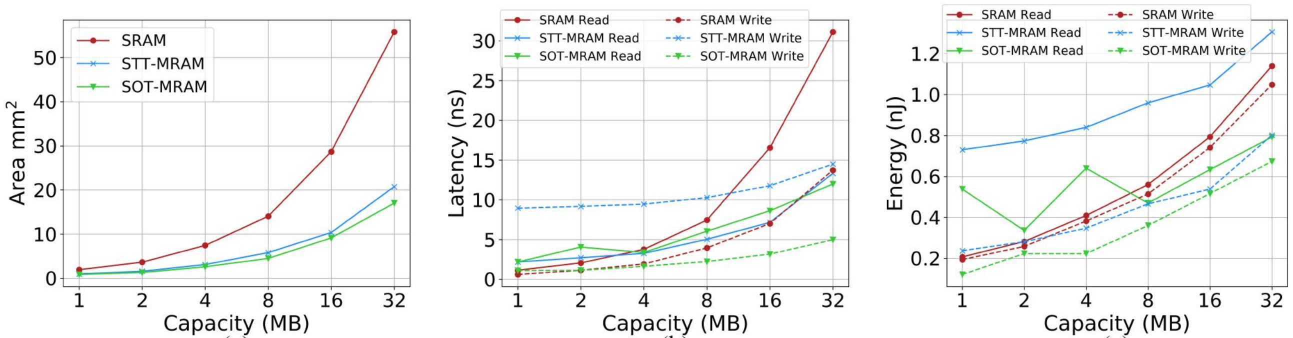
As in the iso-capacity study, for iso-area analysis, we use a batch size 4 for inference and 64 for training. Fig. 6 shows the reduction in the total number of DRAM accesses as L2 cache capacity increases. We use *GPGPU-Sim* and start with the baseline configuration which is 3 MB for GTX 1080 Ti and double its cache capacity up to 24 MB to quantify the percentage of DRAM access reduction for STT-MRAM and SOT-MRAM at larger cache capacities. Fig. 6 shows that replacing SRAM with STT-MRAM and SOT-MRAM equivalents that fit into the same area significantly reduces the total number of DRAM transactions by 14.6% and 19.8%, respectively for 1080 Ti GPU.

Fig. 7 shows normalized dynamic energy and leakage energy breakdown results for 1080 Ti GPU based on actual platform memory statistics and our MRAM cache models at the same area. We use our iso-area cache parameters in which STT-MRAM (7 MB) and SOT-MRAM (10 MB) have larger cache capacities for the same area budget with SRAM. We use these cache parameters and profiling results to calculate results for various DNNs for both inference and training workloads.

In Fig. 7, we observe that STT-MRAM has 2.5× dynamic energy whereas SOT-MRAM has 1.4× dynamic energy on average when compared to SRAM baseline. On the other hand, Fig. 7 also shows that STT-MRAM and SOT-MRAM provide 2.1× and 2.3× lower leakage energy on average when compared to SRAM, respectively. Based on this result, STT-MRAM and SOT-MRAM achieve 2× and 2.3× *lower energy* when compared to SRAM.

Furthermore, Fig. 8 shows that STT-MRAM and SOT-MRAM provide 1.1× and 1.2× *EDP reduction* and 2.3× and 3.3× *larger cache capacity* on average across all workloads when compared to SRAM and off-chip DRAM accesses are not included in the calculations, respectively. When DRAM accesses are included in determining EDP, as shown in Fig. 8, STT-MRAM and SOT-MRAM provide 2× and 2.3× *EDP reduction* on average across all workloads when compared to SRAM, respectively.

We show that although the cache latency and energy results for STT-MRAM and SOT-MRAM do not outperform SRAM results at larger cache capacities as shown in Table II, they do outperform SRAM when costly off-chip DRAM accesses are also considered in EDP calculations. To this end, Chen *et al.* [13] showed that the normalized energy cost of a global buffer access relative to a MAC operation is 6×, whereas a DRAM access is 200× for a machine learning hardware accelerator. By the same token, the higher cell density of NVM can be exploited to shift the memory traffic from DRAM to L2 cache to further improve the power and performance of the overall system. This approach can dramatically reduce the total number of costly DRAM accesses and reduce data movement, which is a daunting impediment for achieving energy-efficient machine learning hardware [13], [63]–[66].



(a) (b) (c)

Fig. 9. Cache capacity scaling results for SRAM, STT-MRAM, and SOT-MRAM for (a) area, (b) latency, and (c) energy metrics.

C. Scalability Analysis

As shown in Fig. 1, the current trend for NVIDIA GPUs is toward increasing L2 size with each new GPU generation. The most recent high-end NVIDIA GPUs have even up to 6-MB L2 cache to further improve the performance of the system by reducing costly off-chip memory accesses. However, SRAM has a scalability problem due to its high leakage and large bitcell area, which poses a significant challenge to further continue the current GPU trend. To this end, NVM technologies come to the rescue of future GPU architectures since their PPA scale better as cache capacity increases. Therefore, there is a need for a scalability analysis to project and quantify performance and energy gains that can be achieved by using more scalable memory solutions.

To this end, we perform a scalability analysis by first comparing SRAM, STT-MRAM, and SOT-MRAM for various cache capacities in terms of area, latency, energy results following the *DeepNVM++* framework methodology as described in Section III. Therefore, each memory technology is optimized for EDAP objective at each cache capacity independently to make a fair comparison among SRAM, STT-MRAM, and SOT-MRAM. Next, we evaluate and show how NVM-based caches behave in terms of performance and energy when compared to conventional SRAM-based caches for DL workloads in a scalability analysis.

Area: Fig. 9(a) demonstrates the impact of higher cell density of MRAMs on the area of caches compared to SRAM. The area difference between SRAM and the MRAM variants grows significantly as the cache capacity increases. The main reason for this difference comes from the bitcell area difference between SRAM and MRAMs as shown in the last row of Table I. Particularly for deeply scaled technology nodes wherein interconnects account for a significant portion of parasitics, bigger bitcells translate to longer wires, bigger buffers, and peripheral logic. Therefore, STT-MRAM and SOT-MRAM caches become more area efficient when compared to SRAM caches as cache capacity increases.

Latency: Fig. 9(b) shows that for capacities smaller than 3 MB SRAM offers lower read latency, whereas both MRAM variants have lower read latency than SRAM beyond 4 MB. In terms of write latency, STT-MRAM has always the highest among all memory technologies due to its inherent device characteristic. In contrast, the write latency of SOT-MRAM becomes increasingly smaller than that of SRAM. Moreover, the write latency of SRAM almost matches that of STT-MRAM at 32 MB.

Energy: In terms of read access energy, Fig. 9(c) shows that 7 MB is a break even point where SOT-MRAM becomes more efficient than SRAM whereas STT-MRAM clearly has the highest read energy among all memories. Regarding write access energy, SOT-MRAM is the most efficient option whereas SRAM consumes the most energy for a write operation beyond 3 MB.

Based on these PPA results, we perform a detailed scalability analysis for SRAM, STT-MRAM, and SOT-MRAM. In Fig. 10, we show the normalized energy, latency, and EDP results with respect to SRAM for STT-MRAM and SOT-MRAM for various cache capacities. As it can be seen, STT-MRAM and SOT-MRAM provide lower energy and latency results as cache capacity increases.

In terms of energy, STT-MRAM and SOT-MRAM provide lower energy as cache capacity increases. Specifically, STT-MRAM and SOT-MRAM caches achieve up to 31.2× and 36.4× *energy reduction* as cache capacity increases, respectively. In terms of latency, STT-MRAM and SOT-MRAM have higher latency results for cache capacities up to 4 MB, whereas both MRAM variants have lower latency results when compared to SRAM beyond that point. In more detail, SRAM provides up to 3.2× and 2× *latency reduction* for small cache capacities when compared to STT-MRAM and SOT-MRAM, respectively. However, STT-MRAM and SOT-MRAM achieve up to 2.1× and 2.6× *latency reduction* as cache capacity increases, respectively. In terms of EDP, we show that STT-MRAM and SOT-MRAM provide up to 65× and 95× *EDP reduction* when compared to SRAM, respectively. Therefore, we conclude that for latency-critical applications, SRAM-based caches become a more suitable option when compared to MRAM variants for small cache capacities whereas MRAMs provide more energy-efficient solutions. Although SRAM provides lower EDP results for smaller cache capacities, STT-MRAM and SOT-MRAM outperform SRAM by orders of magnitude for larger cache capacities due to their better PPA scalability when compared to SRAM. These results show that a significant portion of the overall system energy or latency

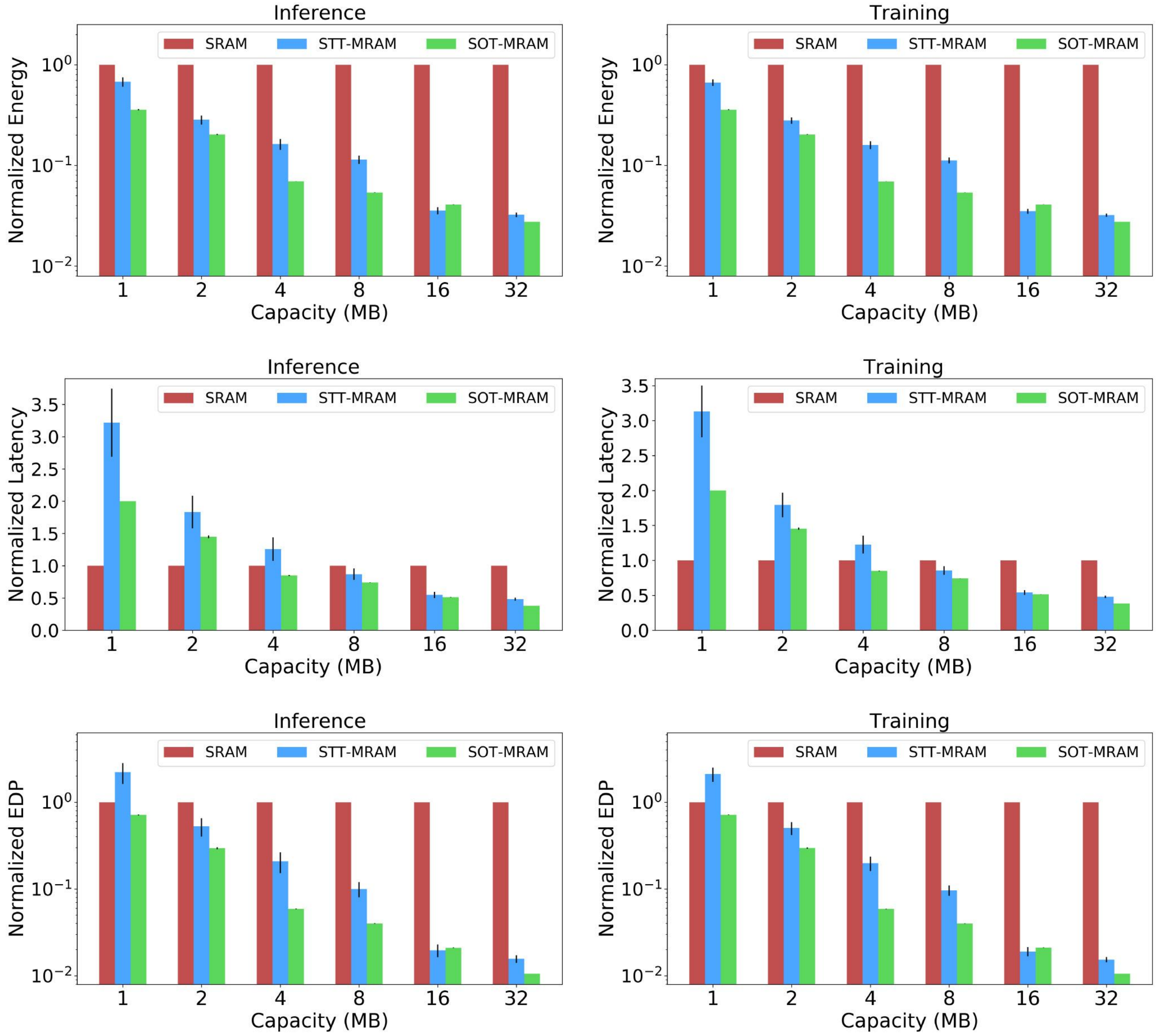


Fig. 10. Mean energy (top), latency (middle), and EDP (bottom) across all workloads (lower is better) normalized with respect to SRAM for various cache capacities for inference (left) and training (right) stages. Error bars show standard deviation across workloads.

is saved and can be used for additional on-chip resources or capabilities that are not available now.

V. Discussion

In this section, we discuss the implications of the results shown in this article. We also share the potential future directions to guide our community to better explore the use of nonvolatile memories for DL workloads in different design spaces.

Scalability is a Major Problem for SRAM: As we show in Fig. 9 and Section IV-C, one of the key challenges for the current GPU architectures is the scalability problem of SRAM due to its significantly high leakage energy and large area when compared to STT-MRAM and SOT-MRAM. We observe that there is a current trend in GPU architectures toward increasing L2 cache capacity and we show that SRAM has significant scalability problems in terms of area, latency, and energy. We show that STT-MRAM and SOT-MRAM have promising solutions for larger cache capacities which can maintain the current trend shown in Fig. 1 with increasing performance and energy benefits.

Implications of Dense NVM Caches on Logic Usage: Fig. 9(a) shows the area results for SRAM, STT-MRAM, and SOT-MRAM for various cache capacities. We note that STT-MRAM and SOT-MRAM provide an increasingly smaller area than SRAM as cache capacity increases. For the same cache capacity, STT-MRAM and SOT-MRAM provide 58% and 65% area reduction on average, respectively. Therefore, the remaining whitespace can be utilized by cramming more processing elements, register files, or L2 cache on the die. This analysis is left for future work.

As CMOS scaling issues limit the affordable improvement of computing systems, our results from device-level simulations to actual GPU profiling show that MRAMs are extremely promising candidates. Particularly, as STT-MRAM and SOT-MRAM fabrication processes become more mature, system-level benefits of STT-MRAM and SOT-MRAM can be maximized, enabling faster and more energy-efficient computation.

Mobile Design Space Exploration for NVM: In this work, we explore the GPU architecture design space to unveil the potential of nonvolatile memories for DL workloads. Having said that, we note that inference at the edge devices also becomes a common practice for many service providers, such as Google [67], Amazon [68], and Facebook [69] to improve user experience by reducing latency and preserving the private user data on device [70]. To this end, Wu *et al.* [69] shows that the majority of mobile inference for Facebook workloads run on mobile CPUs. Mobile platforms have various resource constraints, such as energy, memory, and computing capabilities. Thus, last-level caches of mobile CPUs or hardware accelerators can also be replaced by STT-MRAM and SOT-MRAM to improve performance and energy by reducing leakage energy and costly off-chip memory accesses due to their nonvolatility and higher cell density [71]–[74]. Therefore, the design space exploration of STT-MRAM and SOT-MRAM for mobile CPUs and hardware accelerators for inference workloads merits further research.

VI. Conclusion

In this article, we present the first cross-layer analysis framework to characterize, model, and analyze various NVM technologies in GPU architectures for DL workloads. Our novel framework can be used to further explore the feasibility of emerging NVM technologies for DL applications for different design choices, such as technology nodes, bitcell models, DL workloads, cache configurations, optimization targets, and target platforms.

Our results show that in the iso-capacity case, STT-MRAM and SOT-MRAM provide up to 3.8× and 4.7× *EDP reduction* and 2.4× and 2.8× *area reduction* when compared to SRAM, respectively. In the iso-area case, STT-MRAM and SOT-MRAM achieve up to 2×and 2.3× *EDP reduction* and accommodate 2.3× and 3.3× *cache capacity* when compared to SRAM, respectively. Finally, we perform a scalability analysis and show that STT-MRAM and SOT-MRAM outperform their SRAM counterpart by orders of magnitude in terms of EDP for large cache capacities. The newly created energy or latency slack can be used for additional on-chip resources or capabilities that are currently not possible.

References

[1] W. A. Wulf and S. A. McKee, “Hitting the memory wall: Implications of the obvious,” *SIGARCH Comput. Archit. News*, vol. 23, no. 1, pp. 20–24, Mar. 1995. [Online]. Available: https://doi.org/10.1145/216585.216588

[2] R. H. Dennard, F. H. Gaensslen, H. Yu, V. L. Rideout, E. Bassous, and A. R. LeBlanc, “Design of ion-implanted MOSFET’s with very small physical dimensions,” *IEEE J. Solid-State Circuits*, vol. 9, no. 5, pp. 256–268, Oct. 1974.

[3] S. Murali *et al.*, “Temperature control of high-performance multi-core platforms using convex optimization,” in *Proc. Conf. Design Autom. Test Europe*, Mar. 2008, pp. 110–115.

[4] A. K. Coskun, T. S. Rosing, and K. Whisnant, “Temperature aware task scheduling in MPSoCs,” in *Proc. Design Autom. Test Europe Conf. Exhibition*, 2007, pp. 1–6.

[5] A. K. Coskun, T. S. Rosing, K. A. Whisnant, and K. C. Gross, “Static and dynamic temperature-aware scheduling for multiprocessor SoCs,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 16, no. 9, pp. 1127–1140, Sep. 2008. [Online]. Available: https://doi.org/10.1109/TVLSI.2008.2000726

[6] M. Tan and Q. Le, “EfficientNet: Rethinking model scaling for convolutional neural networks,” in *Proc. Mach. Learn. Res.*, Jun. 2019, pp. 6105–6114. [Online]. Available: http://proceedings.mlr.press/v97/tan19a.html

[7] H. Touvron, A. Vedaldi, M. Douze, and H. Jégou, “Fixing the train-test resolution discrepancy,” in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2019, pp. 8250–8260.

[8] M. Tan, R. Pang, and Q. V. Le, “EfficientDet: Scalable and efficient object detection,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020, pp. 10778–10787.

[9] A. Mirhoseini *et al.*, “Chip placement with deep reinforcement learning,” 2020, *arXiv:2004.10746*.

[10] S. Han, H. Mao, and W. J. Dally, “Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding,” 2016, *arXiv:1510.00149*.

[11] R. Ding, Z. Liu, R. D. S. Blanton, and D. Marculescu, “Lightening the load with highly accurate storage- and energy-efficient lightNNs,” *ACM Trans. Reconfigurable Technol. Syst.*, vol. 11, no. 3, pp. 1–24, Dec. 2018. [Online]. Available: https://doi.org/10.1145/3270689

[12] T.-W. Chin, R. Ding, C. Zhang, and D. Marculescu, “Towards efficient model compression via learned global ranking,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1515–1525.

[13] Y. Chen, J. Emer, and V. Sze, “Eyeriss: A spatial architecture for energy-efficient dataflow for convolutional neural networks,” in *Proc. 43rd Int. Symp. Comput. Archit.*, 2016, pp. 367–379. [Online]. Available: https://doi.org/10.1109/ISCA.2016.40

[14] M. M. S. Aly *et al.*, “Energy-efficient abundant-data computing: The N3XT 1,000x,” *Computer*, vol. 48, no. 12, pp. 24–33, Dec. 2015.

[15] Y.-H. Chen, J. Emer, and V. Sze, “Using dataflow to optimize energy efficiency of deep neural network accelerators,” *IEEE Micro*, vol. 37, no. 3, pp. 12–21, Jun. 2017.

[16] Y. Shao *et al.*, “Simba: Scaling deep-learning inference with multi-chip-module-based architecture,” in *Proc. 52nd Annu. IEEE/ACM Int. Symp. Microarchit.*, 2019, pp. 14–27.

[17] A. F. Inci, M. M. Isgenc, and D. Marculescu, “DeepNVM: A framework for modeling and analysis of non-volatile memory technologies for deep learning applications,” in *Proc. 23rd Conf. Design Autom. Test Europe*, 2020, pp. 1295–1298.

[18] A. Inci *et al.*, “The architectural implications of distributed reinforcement learning on CPU-GPU systems,” 2020, *arXiv:2012.04210*.

[19] A. Inci, M. M. Isgenc, and D. Marculescu, “Cross-layer design space exploration of NVM-based caches for deep learning,” in *Proc. NVMW*, 2021, pp. 1–2.

[20] A. Inci, S. G. Virupaksha, A. Jain, V. V. Thallam, R. Ding, and D. Marculescu, “QAPPA: Quantization-aware power, performance, and area modeling of DNN accelerators,” in *Proc. On-Device Intell. Workshop Conjunction MLSys*, 2021.

[21] M.-T. Chang, P. Rosenfeld, S.-L. Lu, and B. Jacob, “Technology comparison for large last-level caches (L3Cs): Low-leakage SRAM, low write-energy STT-RAM, and refresh-optimized eDRAM,” in *Proc. IEEE 19th Int. Symp. High Perform. Comput. Archit. (HPCA)*, 2013, pp. 143–154.

[22] H. Homayoun and A. Veidenbaum, “Reducing leakage power in peripheral circuits of l2 caches,” in *Proc. 25th Int. Conf. Comput. Design*, 2007, pp. 230–237.

[23] W. Xu, H. Sun, X. Wang, Y. Chen, and T. Zhang, “Design of last-level on-chip cache using spin-torque transfer RAM (STT RAM),” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 19, no. 3, pp. 483–493, Mar. 2011.

[24] X. Dong, X. Wu, G. Sun, Y. Xie, H. Li, and Y. Chen, “Circuit and microarchitecture evaluation of 3D stacking magnetic RAM (MRAM) as a universal memory replacement,” in *Proc. 45th ACM/IEEE Design Autom. Conf.*, 2008, pp. 554–559.

[25] “List of NVIDIA GPUs.” [Online]. Available: https://en.wikipedia.org/wiki/List-of-Nvidia-graphics-processing-units (Accessed: Sep. 1, 2019).

[26] R. E. Scheuerlein, “Magneto-resistive IC memory limitations and architecture implications,” in *Proc. 7th Biennial IEEE Int. Nonvolatile Memory Technol. Conf.*, Jun. 1998, pp. 47–50.

[27] W. Zhao *et al.*, "Macro-model of spin-transfer torque based magnetic tunnel junction device for hybrid magnetic-CMOS design," in *Proc. IEEE Int. Behav. Model. Simulat. Workshop*, Sep. 2006, pp. 40–43.

[28] J. J. Kan *et al.*, "A study on practically unlimited endurance of STT-MRAM," *IEEE Trans. Electron Devices*, vol. 64, no. 9, pp. 3639–3646, Sep. 2017.

[29] M. Hosomi *et al.*, "A novel nonvolatile memory with spin torque transfer magnetization switching: Spin-RAM," in *Proc. IEEE Int. Electron Devices Meeting IEDM Tech. Dig.*, 2005, pp. 459–462.

[30] P. Chi, S. Li, Y. Cheng, Y. Lu, S. H. Kang, and Y. Xie, "Architecture design with STT-RAM: Opportunities and challenges," in *Proc. 21st Asia South Pac. Design Autom. Conf. (ASP-DAC)*, 2016, pp. 109–114.

[31] M. Rasquinha, D. Choudhary, S. Chatterjee, S. Mukhopadhyay, and S. Yalamanchili, "An energy efficient cache design using spin torque transfer (STT) RAM," in *Proc. ACM/IEEE Int. Symp. Low-Power Electron. Design (ISLPED)*, 2010, pp. 389–394.

[32] G. Prenat *et al.*, "Ultra-fast and high-reliability SOT-MRAM: From cache replacement to normally-off computing," *IEEE Trans. Multi-Scale Comput. Syst.*, vol. 2, no. 1, pp. 49–60, Jan.–Mar. 2016.

[33] R. Bishnoi, M. Ebrahimi, F. Oboril, and M. B. Tahoori, "Architectural aspects in design and analysis of SOT-based memories," in *Proc. 19th Asia South Pac. Design Autom. Conf. (ASP-DAC)*, 2014, pp. 700–707.

[34] F. Oboril, R. Bishnoi, M. Ebrahimi, and M. B. Tahoori, "Evaluation of hybrid memory technologies using SOT-MRAM for on-chip cache hierarchy," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 34, no. 3, pp. 367–380, Mar. 2015.

[35] G. Li *et al.*, "A STT-RAM-based low-power hybrid register file for GPGPUs," in *Proc. 52nd ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, 2015, pp. 1–6.

[36] X. Wu, J. Li, L. Zhang, E. Speight, R. Rajamony, and Y. Xie, "Hybrid cache architecture with disparate memory technologies," in *Proc. ISCA*, 2009, pp. 34–45. [Online]. Available: https://doi.org/10.1145/1555754.1555761

[37] M. Imani, S. Patil, and T. Rosing, "Low power data-aware STT-RAM based hybrid cache architecture," in *Proc. 17th Int. Symp. Qual. Electron. Design (ISQED)*, 2016, pp. 88–94.

[38] M. V. Beigi and G. Memik, "TAPAS: Temperature-aware adaptive placement for 3D stacked hybrid caches," in *Proc. 2nd Int. Symp. Memory Syst.*, 2016, pp. 415–426. [Online]. Available: https://doi.org/10.1145/2989081.2989085

[39] C. W. Smullen, V. Mohan, A. Nigam, S. Gurumurthi, and M. R. Stan, "Relaxing non-volatility for fast and energy-efficient STT-RAM caches," in *Proc. IEEE 17th Int. Symp. High Perform. Comput. Archit.*, Feb. 2011, pp. 50–61.

[40] K. Kuan and T. Adegbija, "Energy-efficient runtime adaptable L1 STT-RAM cache design," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 6, pp. 1328–1339, Jun. 2020.

[41] A. Jog *et al.*, "Cache revive: Architecting volatile STT-RAM caches for enhanced performance in CMPs," in *Proc. DAC Design Autom. Conf.*, 2012, pp. 243–252.

[42] Z. Sun *et al.*, "Multi retention level STT-RAM cache designs with a dynamic refresh scheme," in *Proc. 44th Annu. IEEE/ACM Int. Symp. Microarchit. (MICRO)*, 2011, pp. 329–338.

[43] J. Wang, X. Dong, and Y. Xie, "OAP: An obstruction-aware cache management policy for STT-RAM last-level caches," in *Proc. Design Autom. Test Europe Conf. Exhibition (DATE)*, 2013, pp. 847–852.

[44] G. Sun, X. Dong, Y. Xie, J. Li, and Y. Chen, "A novel architecture of the 3D stacked MRAM l2 cache for CMPs," in *Proc. IEEE 15th Int. Symp. High Perform. Comput. Archit.*, 2009, pp. 239–249.

[45] M. Imani, A. Rahimi, Y. Kim, and T. Rosing, "A low-power hybrid magnetic cache architecture exploiting narrow-width values," in *Proc. 5th Non-Volatile Memory Syst. Appl. Symp. (NVMSA)*, 2016, pp. 1–6.

[46] X. Dong, C. Xu, Y. Xie, and N. P. Jouppi, "NVSIM: A circuit-level performance, energy, and area model for emerging nonvolatile memory," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 31, no. 7, pp. 994–1007, Jul. 2012. [Online]. Available: https://doi.org/10.1109/TCAD.2012.2185930

[47] S. Angizi *et al.*, "Accelerating deep neural networks in processing-in-memory platforms: Analog or digital approach?" in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI (ISVLSI)*, 2019, pp. 197–202.

[48] D. Reis *et al.*, "Modeling and benchmarking computing-in-memory for design space exploration," in *Proc. Great Lakes Symp. VLSI*, 2020, pp. 39–44.

[49] S. Angizi, N. Khoshavi, A. Marshall, P. Dowben, and D. Fan, "MERAM: Non-volatile cache memory based on magneto-electric FETs," 2020, *arXiv:2009.06119*.

[50] J. Kim, A. Chen, B. Behin-Aein, S. Kumar, J. Wang, and C. H. Kim, "A technology-agnostic MTJ SPICE model with user-defined dimensions for STT-MRAM scalability studies," in *Proc. IEEE Custom Integr. Circuits Conf. (CICC)*, Sep. 2015, pp. 1–4.

[51] M. Kazemi, G. E. Rowlands, E. Ipek, R. A. Buhrman, and E. G. Friedman, "Compact model for spin-orbit magnetic tunnel junctions," *IEEE Trans. Electron Devices*, vol. 63, no. 2, pp. 848–855, Feb. 2016.

[52] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. 22nd ACM Int. Conf. Multimedia*, 2014, pp. 675–678. [Online]. Available: http://doi.acm.org/10.1145/2647868.2654889

[53] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[54] A. Bakhoda, G. L. Yuan, W. W. L. Fung, H. Wong, and T. M. Aamodt, "Analyzing CUDA workloads using a detailed GPU simulator," in *Proc. IEEE Int. Symp. Perform. Anal. Syst. Softw.*, Apr. 2009, pp. 163–174.

[55] Y. Seo and K. Roy, "High-density SOT-MRAM based on shared bitline structure," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 26, no. 8, pp. 1600–1603, Aug. 2018.

[56] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. 25th Int. Conf. Neural Inf. Process. Syst.*, vol. 1, 2012, pp. 1097–1105.

[57] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2015, pp. 1–9.

[58] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[59] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 770–778.

[60] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size," 2016, *arXiv:1602.07360*.

[61] "NVIDIA CUDA Profiler." [Online]. Available: https://docs.nvidia.com/cuda/profiler-users-guide/nvprof-overview (Accessed: Sep. 1, 2019).

[62] J. Redmon. "Darknet: Open Source Neural Networks in C." 2016. [Online]. Available: http://pjreddie.com/darknet/

[63] Y. Chen, T. Krishna, J. S. Emer, and V. Sze, "Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," *IEEE J. Solid-State Circuits*, vol. 52, no. 1, pp. 127–138, Jan. 2017.

[64] M. Gao, J. Pu, X. Yang, M. Horowitz, and C. Kozyrakis, "TETRIS: Scalable and efficient neural network acceleration with 3D memory," *SIGARCH Comput. Archit. News*, vol. 45, no. 1, pp. 751–764, 2017. [Online]. Available: http://doi.acm.org/10.1145/3093337.3037702

[65] A. Boroumand *et al.*, "Google workloads for consumer devices: Mitigating data movement bottlenecks," in *Proc. 23rd Int. Conf. Archit. Support Program. Lang. Oper. Syst.*, 2018, pp. 316–331. [Online]. Available: https://doi.org/10.1145/3173162.3173177

[66] M. Donato, B. Reagen, L. Pentecost, U. Gupta, D. Brooks, and G. Wei, "On-chip deep neural network storage with multi-level ENVM," in *Proc. 55th ACM/ESDA/IEEE Design Autom. Conf. (DAC)*, 2018, pp. 1–6.

[67] A. Kannan *et al.*, "Smart reply: Automated response suggestion for email," 2016, *arXiv:1606.04870*.

[68] G. Tucker, M. Wu, M. Sun, S. Panchapagesan, G. Fu, and S. Vitaladevuni, "Model compression applied to small-footprint keyword spotting," in *Proc. Interspeech*, 2016, pp. 1878–1882. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2016-1393

[69] C. Wu *et al.*, "Machine learning at facebook: Understanding inference at the edge," in *Proc. IEEE Int. Symp. High Perform. Comput. Archit. (HPCA)*, Feb. 2019, pp. 331–344.

[70] Z. Ghodsi, A. Veldanda, B. Reagen, and S. Garg, "CryptoNAS: Private inference on a RELU budget," 2020, *arXiv:2006.08733*.

[71] K. Korgaonkar *et al.*, "Density tradeoffs of non-volatile memory as a replacement for SRAM based last level cache," in *Proc. ACM/IEEE 45th Annu. Int. Symp. Comput. Archit. (ISCA)*, 2018, pp. 315–327.

[72] A. Hankin, T. Shapira, K. Sangaiah, M. Lui, and M. Hempstead, "Evaluation of non-volatile memory based last level cache given modern use case behavior," in *Proc. IEEE Int. Symp. Workload Characterization (IISWC)*, 2019, pp. 143–154.

[73] L. Pentecost *et al.*, "MaxNVM: Maximizing DNN storage density and inference efficiency with sparse encoding and error mitigation," in *Proc. 52nd Annu. IEEE/ACM Int. Symp. Microarchit.*, 2019, pp. 769–781. [Online]. Available: https://doi.org/10.1145/3352460.3358258

[74] H. Li, M. Bhargav, P. N. Whatmough, and H.-S. P. Wong, "On-chip memory technology design space explorations for mobile deep neural network accelerators," in *Proc. 56th ACM/IEEE Design Autom. Conf. (DAC)*, 2019, pp. 1–6.

Ahmet Inci received the B.Sc. degree in electronics engineering from Sabanci University, Istanbul, Turkey, in 2017. He is currently pursuing the Ph.D. degree with Carnegie Mellon University, Pittsburgh, PA, USA.

His research interests include systems for ML, computer architecture, hardware-efficient deep learning, and HW/ML model co-design.

Mehmet Meric Isgenc was born in Izmir, Turkey, in 1992. He received the B.S. degree in microelectronics engineering from Sabanci University, Istanbul, Turkey, in 2014, and the M.Sc. and Ph.D. degrees in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, USA, in 2016 and 2019, respectively.

He is currently with Apple Inc., Cupertino, CA, USA. His research interests are low-power SoC accelerator design implementation and automation.

Diana Marculescu (Fellow, IEEE) received the Dipl.Ing. degree in computer science from the Polytechnic University of Bucharest, Bucharest, Romania, in 1991, and the Ph.D. degree in computer engineering from the University of Southern California, Los Angeles, CA, USA, in 1998.

She is the Department Chair, the Cockrell Family Chair for Engineering Leadership #5, a Professor, and a Motorola Regents Chair of Electrical and Computer Engineering #2 with the University of Texas at Austin, Austin, TX, USA. Prior to joining UT Austin in December 2019, she was the David Edward Schramm Professor of Electrical and Computer Engineering with Carnegie Mellon University, Pittsburgh, PA, USA, where she was the Founding Director of the College of Engineering Center for Faculty Success from 2015 to 2019, and an Associate Department Head of the Academic Affairs in Electrical and Computer Engineering from 2014 to 2018. Her research interests include energy- and reliability-aware computing, hardware-aware machine learning, and computing for sustainability and natural science applications.

Prof. Marculescu was the recipient of the National Science Foundation Faculty Career Award from 2000 to 2004, the ACM SIGDA Technical Leadership Award in 2003, the Carnegie Institute of Technology George Tallman Ladd Research Award in 2004, and several best paper awards. She was the IEEE Circuits and Systems Society Distinguished Lecturer from 2004 to 2005 and the Chair of the Association for Computing Machinery Special Interest Group on Design Automation from 2005 to 2009. She chaired several conferences and symposia in her area and is currently an Associate Editor of the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS. She was selected as an ELATE Fellow from 2013 to 2014, and was a recipient of an Australian Research Council Future Fellowship from 2013 to 2017, the Marie R. Pistilli Women in EDA Achievement Award in 2014, and the Barbara Lazarus Award from Carnegie Mellon University in 2018. She is a Fellow of ACM.