

Efficient Deep Learning Using Non-Volatile Memory Technology

Ahmet Inci, Mehmet Meric Isgenc, and Diana Marculescu

Abstract Embedded machine learning (ML) systems have now become the dominant platform for deploying ML serving tasks and are projected to become of equal importance for training ML models. With this comes the challenge of overall efficient deployment, in particular low power and high throughput implementations, under stringent memory constraints. In this context, non-volatile 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 non-volatility, higher cell density, and scalability features. While prior work has investigated several architectural implications of NVM for generic applications, in this work we present *DeepNVM++*, a comprehensive *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. *DeepNVM++* relies on *iso-capacity* and *iso-area* performance and energy models for last-level caches implemented using 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 \times and 4.7 \times energy-delay product (EDP) reduction and 2.4 \times and 2.8 \times area reduction compared to conventional SRAM, respectively. Under iso-area assumptions, STT-MRAM and

Ahmet Inci
Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA
e-mail: ainci@andrew.cmu.edu

Mehmet Meric Isgenc
Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA
e-mail: misgenc@andrew.cmu.edu
(Work done while at Carnegie Mellon University; currently with Apple Inc.)

Diana Marculescu
The University of Texas at Austin, Austin, TX 78712, USA
Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA
e-mail: dianam@utexas.edu, dianam@cmu.edu

SOT-MRAM provide up to $2.2\times$ and $2.4\times$ EDP reduction and accommodate $2.3\times$ and $3.3\times$ 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. *DeepNVM++* 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.

1 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, 4, 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, 11, 12], system architects investigate hardware architectures to overcome the memory bottleneck problem and improve the overall system performance [13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24].

We note the current trend of GPU architectures is towards increasing last-level cache capacity as shown in Figure 1. Our analysis shows that conventional SRAM technology incurs scalability problems as far as power, performance, and area (PPA) is concerned [22, 25, 26, 27]. Non-volatile memory (NVM) technology is one of the most promising solutions to tackle memory bottleneck problem for data-intensive applications [28]. 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++* [20], an extended and improved framework [19] to characterize, model, and optimize NVM-based caches in GPU architectures for deep learning workloads. Without loss of generality, we demonstrate our framework for spin-transfer torque magnetic random access memory (STT-MRAM) and spin-orbit torque magnetic random access memory (SOT-MRAM), keeping in

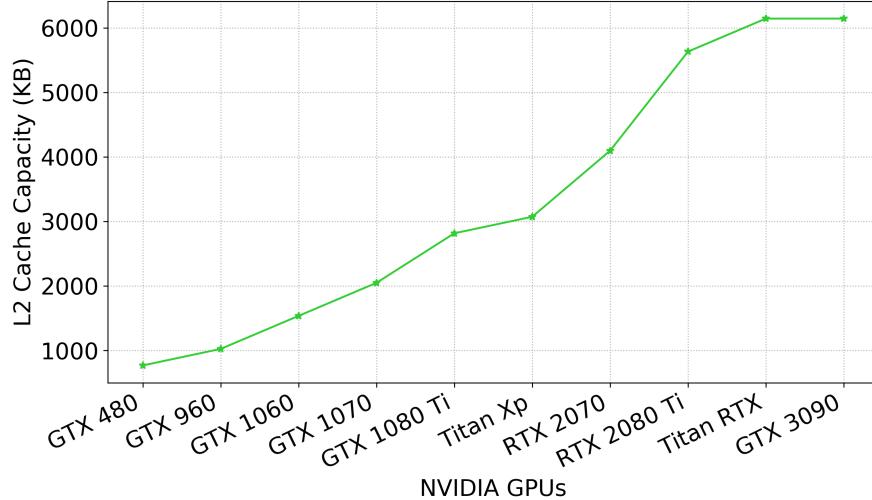


Fig. 1: L2 cache capacity in recent NVIDIA GPUs [29]

mind that it can be used for any NVM technology, GPU platform, or deep learning 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 deep learning applications.

To perform *iso-capacity* analysis, we carry out extensive memory profiling of various deep learning 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.

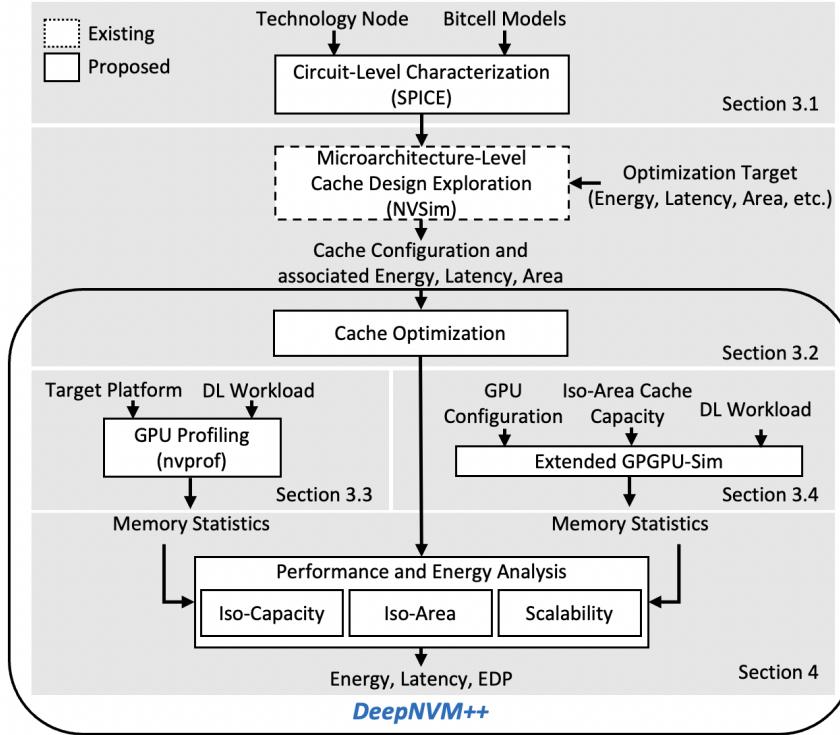
We make the following contributions:

1. **Circuit-level bitcell characterization.** We perform detailed circuit-level characterization combining a commercial 16nm CMOS technology and prominent STT [30] and SOT [31] models from the literature to iterate through our framework in an end-to-end manner to demonstrate the flexibility of *DeepNVM++* [20] for future studies.
2. **Microarchitecture-level cache design exploration.** We use *NVSim* [32] to perform a fair comparison between SRAM, STT-MRAM, and SOT-MRAM by in-

- corporating the circuit-level models developed in 1) using 16nm technology and choosing the best cache configuration for each of them.
3. **Iso-capacity analysis.** To compare the efficacy of magnetic random access memory (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 [33] on a high-end NVIDIA 1080 Ti GPU (implemented in 16nm technology) for the ImageNet dataset [34].
 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 [35] 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 power, performance, and area 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++* [20] is the *first comprehensive framework* for cross-layer characterization, modeling, and analysis of emerging NVM technologies for deep learning 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 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 *2.2× and 2.4× energy-delay product reduction* and accommodate *2.3× and 3.3× larger cache capacity* compared to SRAM, respectively.

Next, we present our cross-layer analysis framework, as shown in Figure 2. First, we present the background and related work on non-volatile memory technologies (Section 2). Next, we show our detailed circuit-level characterization analysis using CMOS, STT, and SOT device models (Section 3.1). After developing bitcell models, we present our microarchitecture-level cache design methodology to obtain cache area, latency, and energy results (Section 3.2). Next, we describe our iso-capacity analysis flow in which we gather actual memory statistics through GPU profiling (Section 3.3). Furthermore, we detail our iso-area analysis in which we extend GPGPU-Sim to run deep learning workloads and support larger cache capacities for STT-MRAM and SOT-MRAM (Section 3.4). Next, we present experimental results demonstrating the efficiency of STT-MRAM and SOT-MRAM over the conventional SRAM for iso-capacity and iso-area cases (Section 4). We then discuss the implications of the results shown in this chapter (Section 5). Finally, we conclude this chapter by summarizing the results (Section 6).

Fig. 2: Overview of the *DeepNVM++* [20] cross-layer analysis flow

2 Related Work

Although 16nm 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 [36, 37]. To move beyond the computing limitations imposed by staggering CMOS scaling trends, MRAM has emerged as a promising candidate [28].

The enabling technology of MRAM consists of magnetic tunnel junction (MTJ) pillars that can store data as a resistive state [38]. 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 are 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 [39].

STT bitcells [40] use an MTJ pillar as their core storage element and an additional access transistor to enable read and write operations. Although STT bitcells offer non-volatility, low read latency, and high endurance [41], the write current is also high [42, 43, 44], 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 [45]. 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 [46, 47]. 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 take 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 [48, 49, 50, 51], relaxing non-volatility properties to reduce the high write latency and energy [52, 53, 54, 55], and implementing cache replacement policies [56, 57, 58] for higher level caches such as L1 caches and register files. However, NVM technology appear 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 power, performance, and area 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* [32], 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 non-traditional architectures such as processing-in-memory based analog and digital architectures [59, 60, 61]. 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 work, 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.

Table 1: 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

3 Methodology

3.1 Circuit-level NVM Characterization

A vast majority of work in the literature uses simple bitcell models [46] 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 16nm 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 3.3.

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 [30] and SOT [31] models and a commercial 16nm FinFET model that takes post-layout effects into account. To find the latency and energy parameters, we used parameterized 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 16nm layout design rules, we used the bitcell area formulations provided in prior work [62].

We summarize the obtained bitcell parameters in Table 1. The sensing delay is measured from wordline activation to the point where the bitline voltage difference reaches 25mV. The sense energy is the integration of the power consumed over

Algorithm 1: EDAP-Optimal Cache Tuning Algorithm

Input: Memory type mem , Cache capacity cap , Optimization target opt , ...
... Access type acc

Output: EDAP-tuned cache configuration

```

1  $mem \in \mathcal{M} = \{SRAM, STT, SOT\}$ ;
2  $cap \in \mathcal{C} = \{1, 2, 4, 8, 16, 32\}$ ;
3  $opt \in \mathcal{O} =$ 
     $\{Read_{Latency}, Write_{Latency}, Read_{Energy}, Write_{Energy}, Read_{EDP}, \dots$ 
4  $\dots Write_{EDP}, Area, Leakage\}$ ;
5  $acc \in \mathcal{A} = \{Normal, Fast, Sequential\}$ ;
6 for each  $mem \in \mathcal{M}$  do
7   for each  $cap \in \mathcal{C}$  do
8      $Q' \leftarrow \infty$ ;
9     for each  $opt \in \mathcal{O}$  do
10       for each  $acc \in \mathcal{A}$  do
11          $Q \leftarrow calculate(EDAP)$ ;
12         if  $Q < Q'$  then
13            $Q' \leftarrow Q$ ;
14         end
15       end
16     end
17      $TunedConfig.append(argv(Q))$ ;
18   end
19 end
20 return  $TunedConfig$ ;
```

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 3.2. We use these bitcell parameters for energy-delay-area product (EDAP) optimized cache design exploration as discussed in the next section.

Table 2: 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

3.2 Microarchitecture-level Cache Design Exploration

In order to demonstrate the impact of using STT and SOT bitcells in L2 caches, we use *NVSim* [32], a circuit-level analysis framework that delivers energy, latency, and area results. After developing *NVSim*-compatible bitcell models as described in Section 3.1, we analyzed a range of cache capacities (1MB to 32MB) 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 Figure 1.

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 3.1, we use a commercial 16nm bitcell design. To ensure a correct analysis, we modified the internal technology file of *NVSim* to the corresponding 16nm 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 2 shows the latency, energy, and area results that correspond to the cache capacity of NVIDIA GTX 1080 Ti GPU (3MB) 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 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

Table 3: Configurations for DNNs under consideration

	AlexNet [63]	GoogLeNet [64]	VGG-16 [65]	ResNet-18 [66]	SqueezeNet [67]
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

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 MRAM technology, we consider two scenarios: (i) First, one could replace the SRAM cache in a GPU with the same capacity MRAM with a smaller area. (ii) 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.

3.3 Architecture-level Iso-Capacity Analysis

As the platform target to demonstrate our work, we use a high-end NVIDIA GTX 1080 Ti GPU which is fabricated in a commercial 16nm technology node which also matches our bitcell and cache models. We use the *Caffe* [33] framework to run various DNNs such as AlexNet [63], GoogLeNet [64], VGG-16 [65], ResNet-18 [66], and SqueezeNet [67] for the ImageNet [34] dataset as shown in Table 3. Our analysis is generalizable to other types of neural network architectures since we cover a wide range of DNN configurations with various workload characteristics. Furthermore, we also use the high performance conjugate gradients (HPCG) [68] benchmark, a widely used high performance computing (HPC) workload, to demonstrate the generalizability of our analysis to different workloads besides deep learning applications.

We use the NVIDIA profiler [69] to obtain the device memory and L2 cache read and write transactions to better understand both on-chip and off-chip memory behavior of various deep learning and HPC workloads. To this end, Figure 3 shows the profiling results for L2 cache read/write ratio for various deep learning and HPC workloads. In particular, we run the HPCG benchmark with different input local subgrid dimensions such as 4x4x4, 8x8x8, 16x16x16, 32x32x32, 64x64x64, and 128x128x128. We show that the ratio of the total number of read transactions to the total number of write transactions in L2 cache varies significantly from 2 to 26. Therefore, these profiling results also show that we cover a wide range of workloads with different workload characteristics in our analysis. To this end, we use 128x128x128, 32x32x32, and 8x8x8 workload configurations for our analysis

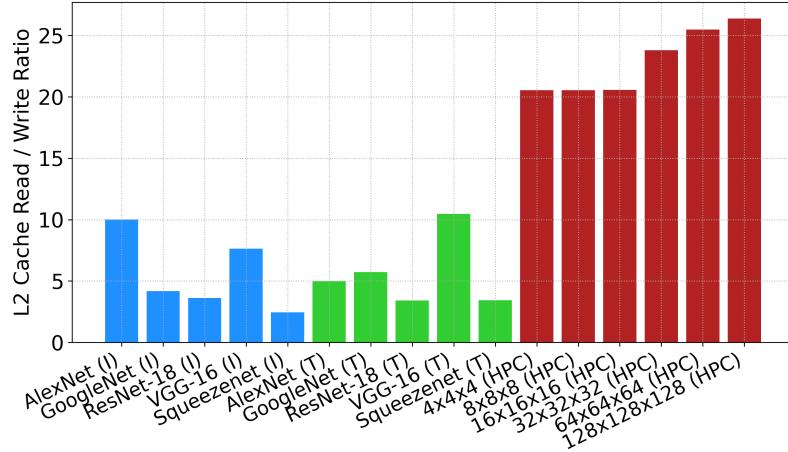


Fig. 3: Profiling results for L2 cache read/write ratio for various workloads

in the rest of the chapter which we refer to as HPCG-L, HPCG-M, and HPCG-S, respectively.

3.4 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* [35] to explore power and performance implications of having these larger L2 caches in GPU architectures for DNN workloads. For comparison, we model the high-end NVIDIA GTX 1080 Ti GPU. The configurations for NVIDIA GTX 1080 Ti GPU are shown in Table 4. We extend the *GPGPU-Sim* simulator to support the cache capacity of NVIDIA GTX 1080 Ti GPU. This GPU is built using a commercial 16nm technology node which matches our bitcell and cache models. In particular, for *GPGPU-Sim* compatibility, we set L2 cache capacity to 3MB. We use this capacity for our analysis in the rest of the chapter. 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 [63] with the ImageNet [34] dataset which is provided by the *DarkNet* [70] framework. We extend *DarkNet* source code to enable deep learning workloads on *GPGPU-Sim*.

Table 4: GPGPU-Sim Configurations

	NVIDIA 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
Core Frequency:	1481 MHz
Interconnect Frequency:	2962 MHz
L2 Cache Frequency:	1481 MHz
Memory Frequency:	2750 MHz

4 Experimental 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 Section 4.1 and Section 4.2, respectively. In Section 4.2, we use iso-area cache parameters as shown in Table 2 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 4.3, we perform a scalability analysis to project the implications of the current GPU trend shown in Figure 1 on performance and energy results.

4.1 Performance and Energy Results for Iso-Capacity

By combining the actual technology-dependent latency and energy metrics from Table 2, 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 [71].

Figure 4 shows normalized dynamic energy and leakage energy breakdown results for NVIDIA GTX 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 as well as HPCG workloads with different input sizes.

In Figure 4, we observe that STT-MRAM consumes 2.2× more dynamic energy whereas SOT-MRAM has 1.3× more dynamic energy on average when compared to the SRAM baseline. Furthermore, our results show that 83% of the total dynamic

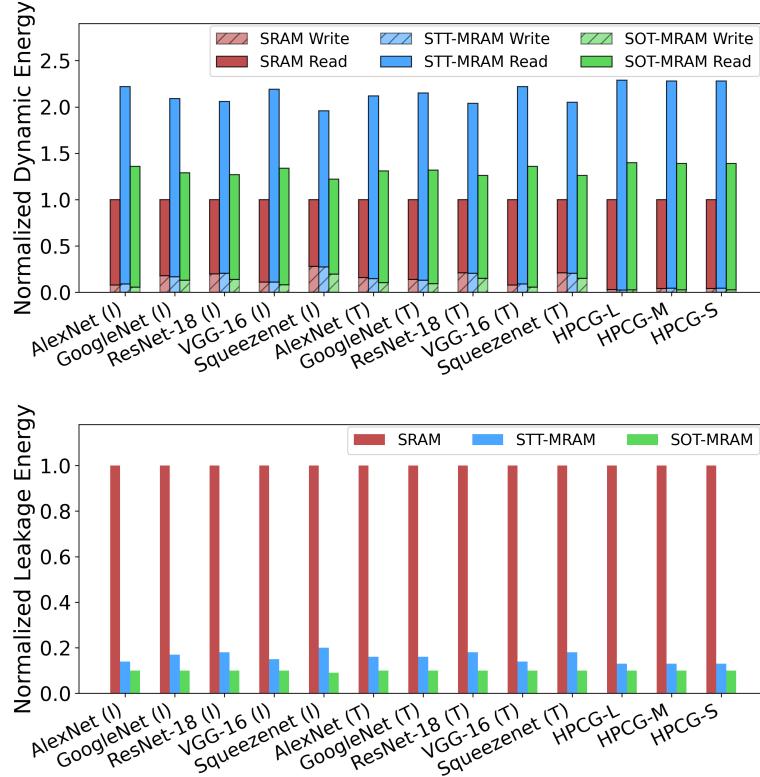


Fig. 4: Dynamic energy (top chart) leakage energy (bottom chart) (lower is better) normalized with respect to SRAM by using NVMs with iso-capacity (3MB) for inference (I) and training (T) stages

energy of SRAM comes from read operations whereas write operations only make for 17% of all transactions on average across deep learning workloads. For HPCG workloads, read operations take 96% of the total dynamic energy of SRAM and write operations only make for 4% of the total energy. Our profiling results also support these findings as read operations dominate write operations in these DL and HPCG workloads.

On the other hand, Figure 4 also shows that STT-MRAM and SOT-MRAM provide 6.3× and 10× lower leakage energy on average when compared to SRAM, respectively. Based on this result, Figure 5 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 5.3× and 8.6× energy reduction on average across all workloads compared to SRAM baseline, respectively, due to their significantly low leakage

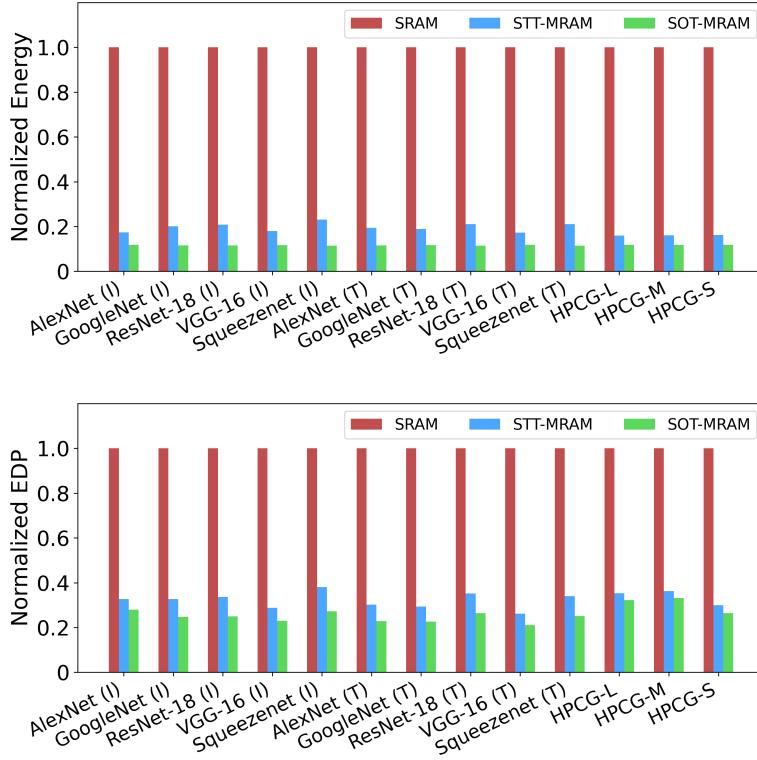


Fig. 5: Iso-capacity (3MB) energy (top chart) and energy-delay product (bottom chart) for NVM-based caches (lower is better) normalized with respect to SRAM-based caches for inference (I) and training (T) stages. DRAM energy and latency are also included in EDP results.

energy. Moreover, Figure 5 shows that STT-MRAM and SOT-MRAM provide up to $3.8\times$ and $4.7\times$ EDP reduction and $2.4\times$ and $2.8\times$ area reduction, respectively.

The 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. Figure 6 shows the impact of batch size on EDP results for AlexNet during training and inference stages based on NVIDIA GTX 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 $2.3\times$ to $4.6\times$ EDP reduction as batch size increases. On the other hand, SOT-MRAM provides $7.2\times$ to $7.6\times$ EDP reduction when compared to SRAM baseline. For inference, STT-MRAM and SOT-MRAM achieve $4.1\times$ to $5.4\times$ and $7.1\times$ to $7.3\times$ EDP reduction, respectively. These results also confirm the different workload characteristics of training and inference. STT-MRAM

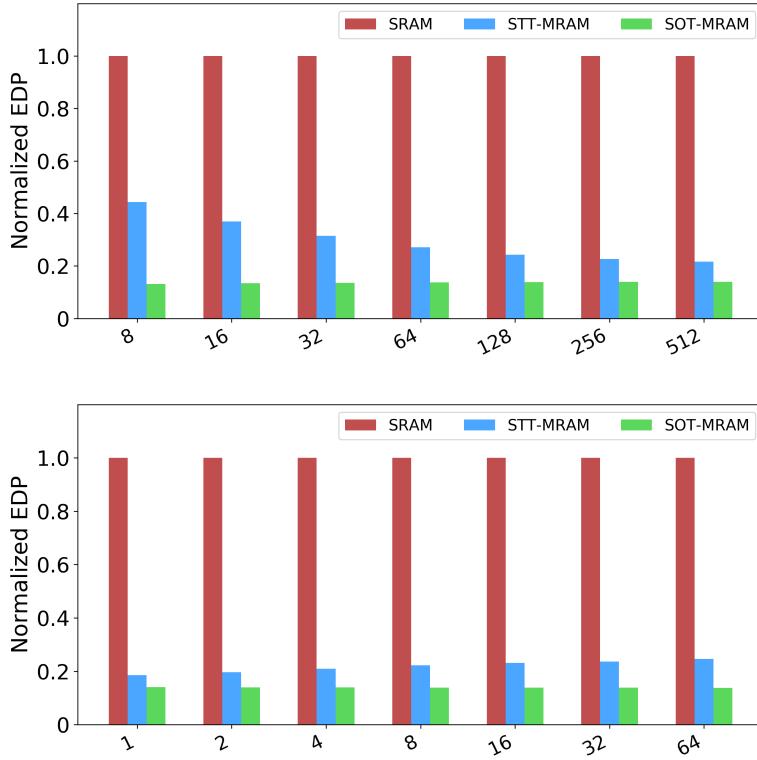


Fig. 6: Impact of batch size on energy-delay product (lower is better) normalized with respect to SRAM by using NVMs with iso-capacity (3MB) for AlexNet for training (top chart) and inference (bottom chart)

provides higher EDP reduction for training workloads as batch size increases. On the other hand, SOT-MRAM follows the same pattern for inference workloads due to their different access characteristics as shown in Table 2. We observe that training workloads become more read dominant whereas inference workloads have lower read/write ratio as batch size increases.

4.2 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. Figure 7 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 3MB for NVIDIA GTX 1080 Ti and double its

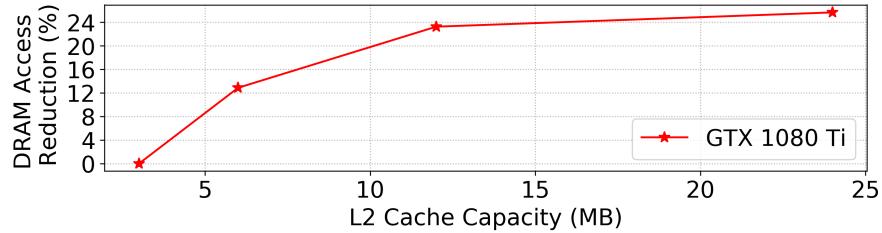


Fig. 7: Simulation results for the reduction in the total number of DRAM accesses in percentage

cache capacity up to 24MB to quantify the percentage of DRAM access reduction for STT-MRAM and SOT-MRAM at larger cache capacities. Figure 7 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.

Figure 8 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 (7MB) and SOT-MRAM (10MB) 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 and HPCG workloads with various input sizes.

In Figure 8, we observe that STT-MRAM has $2.5\times$ dynamic energy whereas SOT-MRAM has $1.5\times$ dynamic energy on average when compared to SRAM baseline. On the other hand, Figure 8 also shows that STT-MRAM and SOT-MRAM provide $2.2\times$ and $2.3\times$ lower leakage energy on average when compared to SRAM, respectively. Based on this result, STT-MRAM and SOT-MRAM achieve $2\times$ and $2.2\times$ lower energy when compared to SRAM.

Furthermore, Figure 9 shows that STT-MRAM and SOT-MRAM provide $1.2\times$ *EDP reduction* and $2.3\times$ and $3.3\times$ *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 Figure 9, STT-MRAM and SOT-MRAM provide $2\times$ and $2.3\times$ *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 2, 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 $6\times$, whereas a DRAM access is $200\times$ 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 power and performance of the

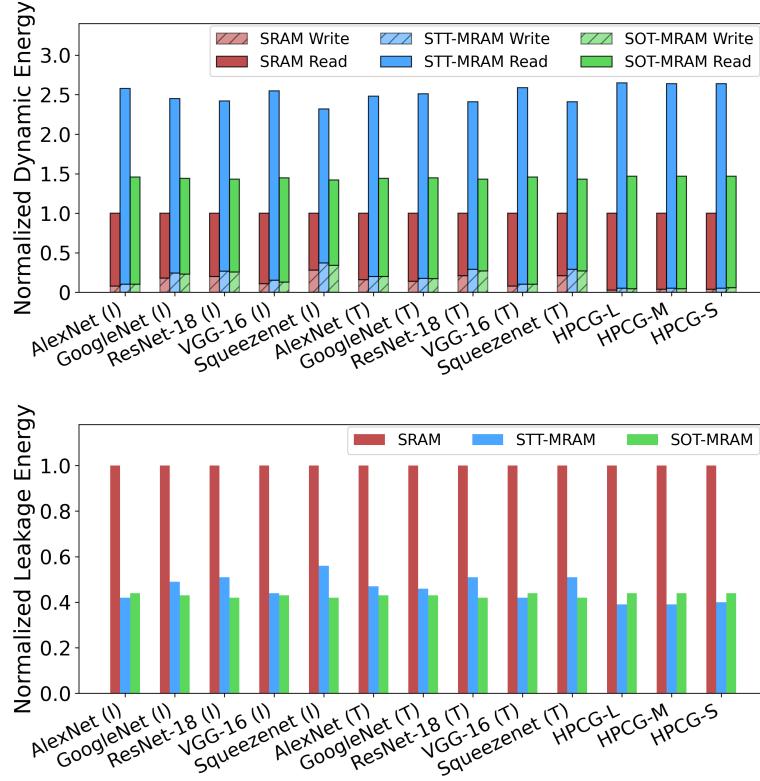


Fig. 8: Dynamic energy (top chart) and leakage energy (bottom chart) (lower is better) normalized with respect to SRAM by using STT-MRAM (7MB) and SOT-MRAM (10MB) with iso-area for inference (I) and training (T) stages

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, 71, 72, 73, 74].

4.3 Scalability Analysis

As shown in Figure 1, the current trend for NVIDIA GPUs is towards increasing L2 size with each new GPU generation. The most recent high-end NVIDIA GPUs have even up to 6MB L2 cache to further improve 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, non-volatile memory technologies

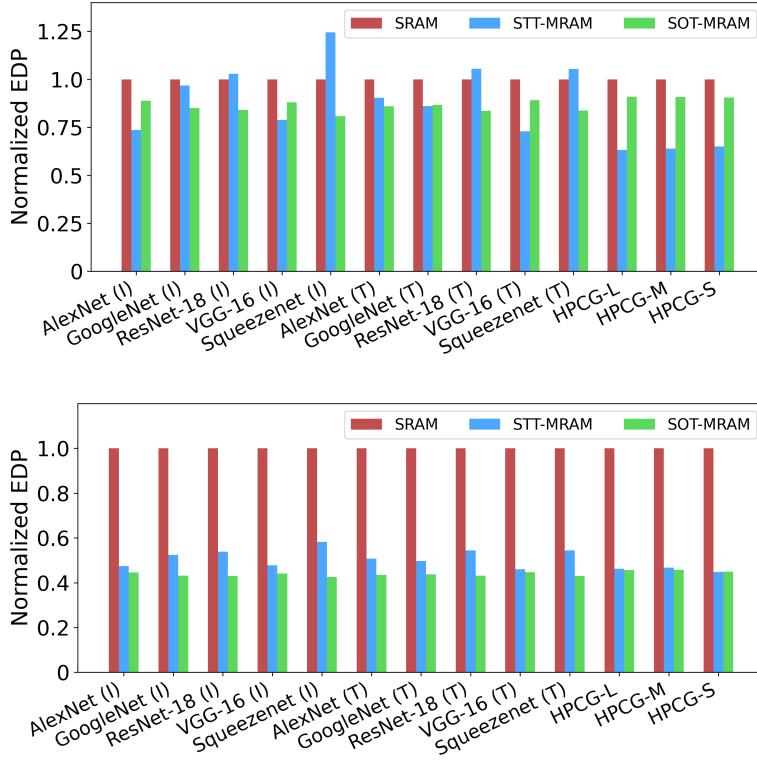


Fig. 9: Iso-area energy-delay product results for STT-MRAM (7MB) and SOT-MRAM (10MB) (lower is better) normalized with respect to SRAM-based caches for inference (I) and training (T) stages without (top chart) and with (bottom chart) DRAM energy and latency.

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 Figure 2. Therefore, each memory technology is optimized for EDAP objective at each cache capacity independently to perform 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 deep learning workloads in a scalability analysis.

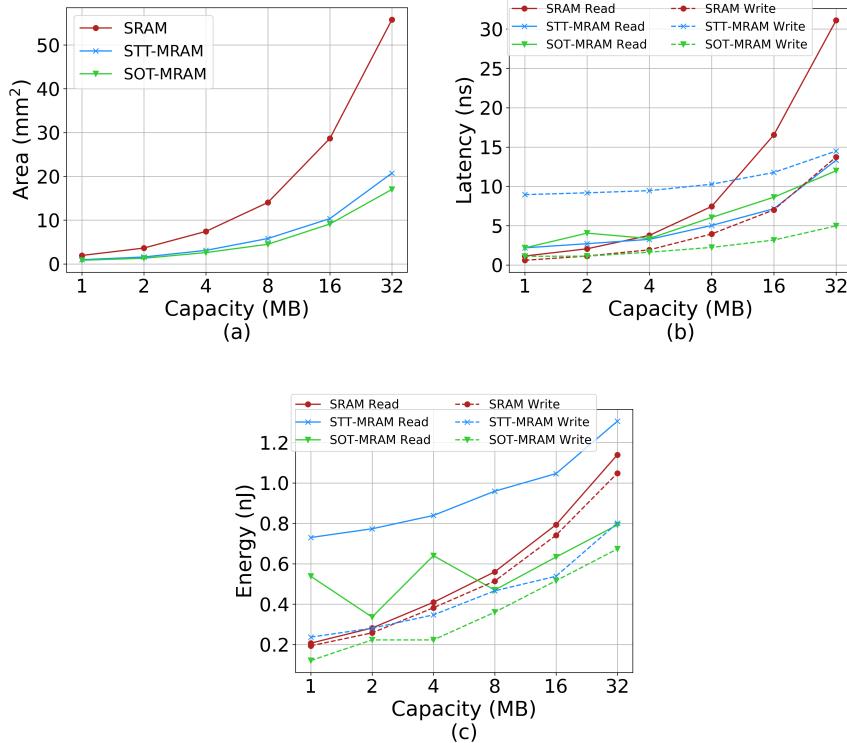


Fig. 10: Cache capacity scaling results for SRAM, STT-MRAM, and SOT-MRAM for (a) area, (b) latency, and (c) energy metrics

Area Figure 10(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 grow significantly as the cache capacity increases. The main reason of this difference comes from the bitcell area difference between SRAM and MRAMs as shown in the last row of Table 1. 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 Figure 10(b) shows that for capacities smaller than 3MB SRAM offers lower read latency, whereas both MRAM variants have lower read latency than SRAM beyond 4MB. 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 32MB.

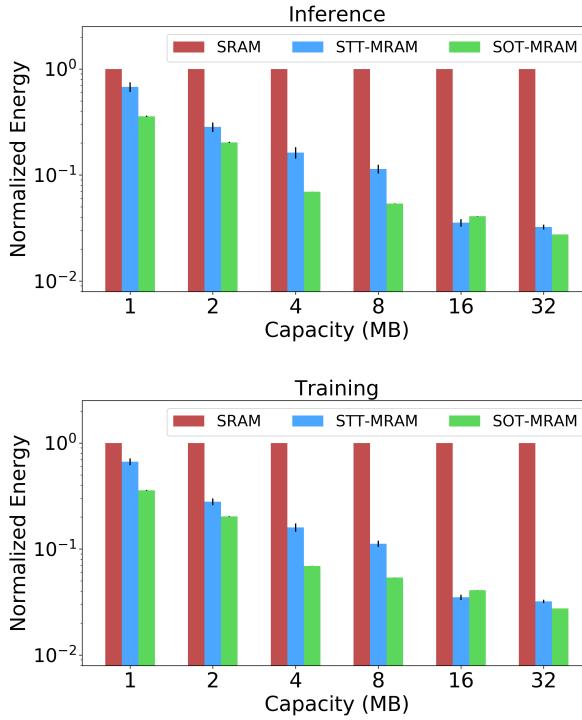


Fig. 11: Mean energy results across all workloads (lower is better) normalized with respect to SRAM for various cache capacities for inference (top chart) and training (bottom chart) stages. Error bars show standard deviation across workloads.

Energy In terms of read access energy, Figure 10(c) shows that 7MB 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 3MB.

Based on these PPA results, we perform a detailed scalability analysis for SRAM, STT-MRAM, and SOT-MRAM. In Figure 11-13, we show the normalized energy, latency, and EDP results with respect to SRAM for STT-MRAM and SOT-MRAM for various cache capacities, respectively. 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 $31.2\times$ and $36.4\times$ 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 4MB, whereas both MRAM variants have lower latency results when compared to SRAM beyond that point. In more detail, SRAM provides up to $3.2\times$

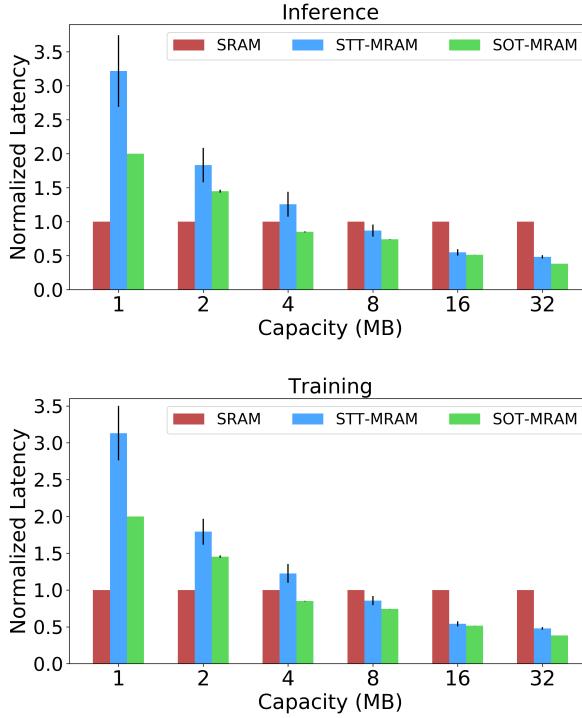


Fig. 12: Mean latency results across all workloads (lower is better) normalized with respect to SRAM for various cache capacities for inference (top chart) and training (bottom chart) stages. Error bars show standard deviation across workloads.

and $2\times$ latency reduction for small cache capacities when compared to STT-MRAM and SOT-MRAM, respectively. However, STT-MRAM and SOT-MRAM achieve up to $2.1\times$ and $2.6\times$ latency reduction as cache capacity increases, respectively. In terms of EDP, we show that STT-MRAM and SOT-MRAM provide up to $65\times$ and $95\times$ 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 provide 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 is saved and can be used for additional on-chip resources or capabilities that are not available now.

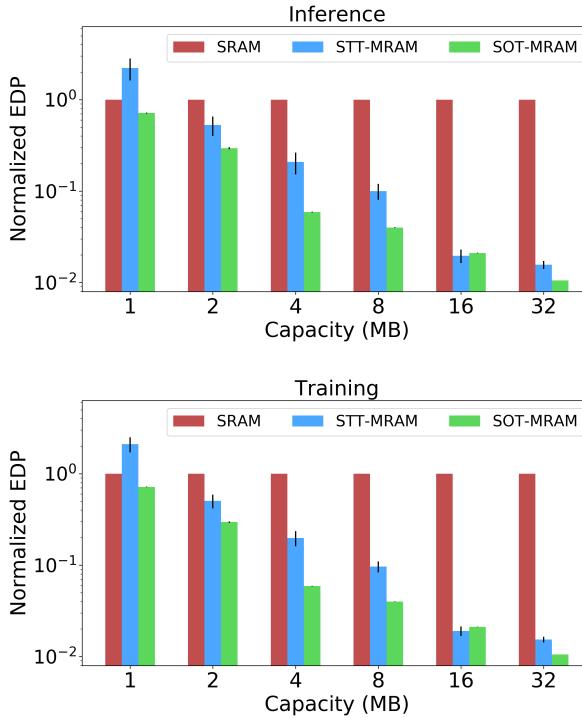


Fig. 13: Mean energy-delay product results across all workloads (lower is better) normalized with respect to SRAM for various cache capacities for inference (top chart) and training (bottom chart) stages. Error bars show standard deviation across workloads.

5 Discussion

In this section, we discuss the implications of the results shown in this chapter. We also share the potential future directions to guide our community to better explore the use of non-volatile memories for deep learning workloads in different design spaces.

Scalability is a major problem for SRAM. As we show in Figure 10 and Section 4.3, 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 towards 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 Figure 1 with increasing performance and energy benefits.

Implications of dense NVM caches on logic usage. Figure 10(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 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 non-volatile memories for deep learning workloads. Having said that, we note that inference at the edge devices also becomes a common practice for many service providers such as Google [75], Amazon [76], and Facebook [77] to improve user experience by reducing latency and preserving the private user data on device [78]. To this end, Wu *et al.* [77] shows that 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 non-volatility and higher cell density [79, 80, 81, 82]. Therefore, the design space exploration of STT-MRAM and SOT-MRAM for mobile CPUs and hardware accelerators for inference workloads merits further research.

6 Conclusion

In this chapter, we present the first cross-layer analysis framework to characterize, model, and analyze various NVM technologies in GPU architectures for deep learning 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 $3.8\times$ and $4.7\times$ *EDP reduction* and $2.4\times$ and $2.8\times$ *area reduction* when compared to SRAM, respectively. In the iso-area case, STT-MRAM and SOT-MRAM achieve up to $2.2\times$ and $2.4\times$ *EDP reduction* and accommodate $2.3\times$ and $3.3\times$ *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 energy-delay product 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.

Acknowledgements This research was supported in part by NSF CCF Grant No. 1815899 and NSF CSR Grant No. 1815780.

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, p. 20–24, Mar. 1995. [Online]. Available: <https://doi.org/10.1145/216583.216588>
2. R. H. Dennard, F. H. Gaenslen, H. Yu, V. L. Rideout, E. Bassous, and A. R. LeBlanc, “Design of ion-implanted mosfet’s with very small physical dimensions,” *IEEE Journal of Solid-State Circuits*, vol. 9, no. 5, pp. 256–268, Oct 9(5):256–268, 1974.
3. S. Murali, A. Mutapcic, D. Atienza, R. Gupta, S. Boyd, L. Benini, and G. De Micheli, “Temperature control of high-performance multi-core platforms using convex optimization,” in *Proceedings of the Conference on Design, Automation and Test in Europe*, March 2008, pp. 110–115.
4. A. K. Coskun, T. S. Rosing, and K. Whisnant, “Temperature aware task scheduling in mpsocs,” in *2007 Design, Automation Test in Europe Conference 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. Syst.*, vol. 16, no. 9, p. 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 *Proceedings of Machine Learning Research*, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. Long Beach, California, USA: PMLR, 09–15 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 *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.
8. M. Tan, R. Pang, and Q. V. Le, “Efficientdet: Scalable and efficient object detection,” *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10 778–10 787, 2020.
9. A. Mirhoseini, A. Goldie, M. Yazgan, J. Jiang, E. Songhori, S. Wang, Y.-J. Lee, E. Johnson, O. Pathak, S. Bae, A. Nazi, J. Pak, A. Tong, K. Srinivasa, W. Hang, E. Tuncer, A. Babu, Q. V. Le, J. Laudon, R. Ho, R. Carpenter, and J. Dean, “Chip placement with deep reinforcement learning,” 2020.
10. S. Han, H. Mao, and W. J. Dally, “Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding,” 2016.
11. R. Ding, Z. Liu, R. D. S. Blanton, and D. Marculescu, “Lightening the load with highly accurate storage- and energy-efficient lighttns,” *ACM Trans. Reconfigurable Technol. Syst.*, vol. 11, no. 3, 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
13. Y. Chen, J. Emer, and V. Sze, “Eyeriss: A spatial architecture for energy-efficient dataflow for convolutional neural networks,” in *Proceedings of the 43rd International Symposium on Computer Architecture*. Piscataway, NJ, USA: IEEE Press, 2016, pp. 367–379. [Online]. Available: <https://doi.org/10.1109/ISCA.2016.40>

14. M. M. Sabry Aly, M. Gao, G. Hills, C.-S. Lee, G. Pitner, M. M. Shulaker, T. F. Wu, M. Asheghi, J. Bokor, F. Franchetti, K. E. Goodson, C. Kozyrakis, I. Markov, K. Olukotun, L. Pileggi, E. Pop, J. Rabaey, C. Ré, H.-S. P. Wong, and S. Mitra, “Energy-efficient abundant-data computing: The n3xt 1,000x,” *Computer*, vol. 48, no. 12, pp. 24–33, 2015.
15. S. Han, X. Liu, H. Mao, J. Pu, A. Pedram, M. A. Horowitz, and W. J. Dally, “Eie: Efficient inference engine on compressed deep neural network,” *International Conference on Computer Architecture (ISCA)*, 2016.
16. 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, 2017.
17. Y. Shao, J. Clemons, R. Venkatesan, B. Zimmer, M. R. Fojtik, N. Jiang, B. Keller, A. Klinefelter, N. Pinckney, P. Raina, S. Tell, Y. Zhang, W. Dally, J. Emer, C. T. Gray, B. Khailany, and S. Keckler, “Simba: Scaling deep-learning inference with multi-chip-module-based architecture,” *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*, 2019.
18. A. Inci and D. Marculescu, “Solving the non-volatile memory conundrum for deep learning workloads,” *Architectures and Systems for Big Data Workshop in conjunction with ISCA*, 2018.
19. 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 *Proceedings of the 23rd Conference on Design, Automation and Test in Europe*, ser. DATE ’20, 2020, p. 1295–1298.
20. A. Inci, M. M. Isgenc, and D. Marculescu, “Deepnvm++: Cross-layer modeling and optimization framework of non-volatile memories for deep learning,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 1–1, 2021.
21. A. Inci, E. Bolotin, Y. Fu, G. Dalal, S. Mannor, D. Nellans, and D. Marculescu, “The architectural implications of distributed reinforcement learning on cpu-gpu systems,” *arXiv preprint arXiv:2012.04210*, 2020.
22. A. Inci, M. M. Isgenc, and D. Marculescu, “Cross-layer design space exploration of nvm-based caches for deep learning,” *NVMW*, 2021.
23. 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,” *arXiv preprint arXiv:2205.08648*, 2022.
24. ——, “Qadam: Quantization-aware dnn accelerator modeling for pareto-optimality,” *arXiv preprint arXiv:2205.13045*, 2022.
25. M. Chang, P. Rosenfeld, S. 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 *2013 IEEE 19th International Symposium on High Performance Computer Architecture (HPCA)*, 2013, pp. 143–154.
26. H. Homayoun and A. Veidenbaum, “Reducing leakage power in peripheral circuits of l2 caches,” in *2007 25th International Conference on Computer Design*, 2007, pp. 230–237.
27. 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 Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 19, no. 3, pp. 483–493, 2011.
28. Xiangyu Dong, Xiaoxia Wu, Guangyu Sun, Yuan Xie, H. Li, and Yiran Chen, “Circuit and microarchitecture evaluation of 3d stacking magnetic ram (mram) as a universal memory replacement,” in *2008 45th ACM/IEEE Design Automation Conference*, 2008, pp. 554–559.
29. *List of NVIDIA GPUs*, <https://en.wikipedia.org/wiki/List-of-Nvidia-graphics-processing-units>.
30. 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 *2015 IEEE Custom Integrated Circuits Conference (CICC)*, Sept 2015, pp. 1–4.
31. M. Kazemi, G. E. Rowlands, E. Ipek, R. A. Buhrman, and E. G. Friedman, “Compact model for spin-orbit magnetic tunnel junctions,” *IEEE Transactions on Electron Devices*, vol. 63, no. 2, pp. 848–855, Feb 63(2):848-855, 2016.
32. X. Dong, C. Xu, Y. Xie, and N. P. Jouppi, “Nvsm: A circuit-level performance, energy, and area model for emerging nonvolatile memory,” *IEEE Transactions on Computer-Aided*

- Design of Integrated Circuits and Systems*, vol. 31(7):994–1007, no. 7, pp. 994–1007, Jul. 31(7):994–1007, 2012. [Online]. Available: <https://doi.org/10.1109/TCAD.2012.2185930>
33. Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, “Caffe: Convolutional architecture for fast feature embedding,” in *Proceedings of the 22Nd ACM International Conference on Multimedia*, ser. MM ’14. New York, NY, USA: ACM, 2014, pp. 675–678. [Online]. Available: <http://doi.acm.org/10.1145/2647868.2654889>
 34. J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.
 35. A. Bakhoda, G. L. Yuan, W. W. L. Fung, H. Wong, and T. M. Aamodt, “Analyzing cuda workloads using a detailed gpu simulator,” in *2009 IEEE International Symposium on Performance Analysis of Systems and Software*, April 2009, pp. 163–174.
 36. M. M. Isgenc, “Enabling design of low-volume high-performance ics,” Ph.D. dissertation, Carnegie Mellon University, 2019.
 37. M. M. Isgenc, M. G. A. Martins, V. M. Zackriya, S. N. Pagliarini, and L. Pileggi, “Logic ip for low-cost ic design in advanced cmos nodes,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 2, pp. 585–595, 2020.
 38. S. N. Pagliarini, S. Bhui, M. M. Isgenc, A. K. Biswas, and L. Pileggi, “A probabilistic synapse with strained mtjs for spiking neural networks,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 4, pp. 1113–1123, 2020.
 39. R. E. Scheuerlein, “Magneto-resistive ic memory limitations and architecture implications,” in *Seventh Biennial IEEE International Nonvolatile Memory Technology Conference. Proceedings (Cat. No.98EX141)*, June 1998, pp. 47–50.
 40. W. Zhao, E. Belaire, Q. Mistral, C. Chappert, V. Javerliac, B. Dieny, and E. Nicolle, “Macro-model of spin-transfer torque based magnetic tunnel junction device for hybrid magnetic-cmos design,” in *2006 IEEE International Behavioral Modeling and Simulation Workshop*, Sept 2006, pp. 40–43.
 41. J. J. Kan, C. Park, C. Ching, J. Ahn, Y. Xie, M. Pakala, and S. H. Kang, “A study on practically unlimited endurance of stt-mram,” *IEEE Transactions on Electron Devices*, vol. 64, no. 9, pp. 3639–3646, Sept 64(9):3639-3646, 2017.
 42. M. Hosomi, H. Yamagishi, T. Yamamoto, K. Bessho, Y. Higo, K. Yamane, H. Yamada, M. Shoji, H. Hachino, C. Fukumoto, H. Nagao, and H. Kano, “A novel nonvolatile memory with spin torque transfer magnetization switching: spin-ram,” in *IEEE International Electron Devices Meeting, 2005. IEDM Technical Digest.*, 2005, pp. 459–462.
 43. P. Chi, S. Li, Yuanqing Cheng, Yu Lu, S. H. Kang, and Y. Xie, “Architecture design with stt-mram: Opportunities and challenges,” in *2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2016, pp. 109–114.
 44. M. Rasquinha, D. Choudhary, S. Chatterjee, S. Mukhopadhyay, and S. Yalamanchili, “An energy efficient cache design using spin torque transfer (stt) ram,” in *2010 ACM/IEEE International Symposium on Low-Power Electronics and Design (ISLPED)*, 2010, pp. 389–394.
 45. G. Prenat, K. Jabeur, P. Vanhauwaert, G. D. Pendina, F. Oboril, R. Bishnoi, M. Ebrahimi, N. Lamard, O. Boulle, K. Garello, J. Langer, B. Ocker, M. Cyrille, P. Gambardella, M. Tahoori, and G. Gaudin, “Ultra-fast and high-reliability sot-mram: From cache replacement to normally-off computing,” *IEEE Transactions on Multi-Scale Computing Systems*, vol. 2, no. 1, pp. 49–60, 2016.
 46. R. Bishnoi, M. Ebrahimi, F. Oboril, and M. B. Tahoori, “Architectural aspects in design and analysis of sot-based memories,” in *2014 19th Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2014, pp. 700–707.
 47. F. Oboril, R. Bishnoi, M. Ebrahimi, and M. B. Tahoori, “Evaluation of hybrid memory technologies using sot-mram for on-chip cache hierarchy,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 3, pp. 367–380, 2015.
 48. Gushu Li, Xiaoming Chen, Guangyu Sun, H. Hoffmann, Yongpan Liu, Yu Wang, and Huazhong Yang, “A stt-ram-based low-power hybrid register file for gpgpus,” in *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2015, pp. 1–6.

49. X. Wu, J. Li, L. Zhang, E. Speight, R. Rajamony, and Y. Xie, "Hybrid cache architecture with disparate memory technologies," ser. ISCA '09. New York, NY, USA: Association for Computing Machinery, 2009, p. 34–45. [Online]. Available: <https://doi.org/10.1145/1555754.1555761>
50. M. Imani, S. Patil, and T. Rosing, "Low power data-aware stt-ram based hybrid cache architecture," in *2016 17th International Symposium on Quality Electronic Design (ISQED)*, 2016, pp. 88–94.
51. M. V. Beigi and G. Memik, "Tapas: Temperature-aware adaptive placement for 3d stacked hybrid caches," in *Proceedings of the Second International Symposium on Memory Systems*, ser. MEMSYS '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 415–426. [Online]. Available: <https://doi.org/10.1145/2989081.2989085>
52. 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 *2011 IEEE 17th International Symposium on High Performance Computer Architecture*, Feb 2011, pp. 50–61.
53. K. Kuan and T. Adegbija, "Energy-efficient runtime adaptable l1 stt-ram cache design," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 6, pp. 1328–1339, 2020.
54. A. Jog, A. K. Mishra, C. Xu, Y. Xie, V. Narayanan, R. Iyer, and C. R. Das, "Cache revive: Architecting volatile stt-ram caches for enhanced performance in cmps," in *DAC Design Automation Conference 2012*, 2012, pp. 243–252.
55. Z. Sun, X. Bi, H. Li, W. Wong, Z. Ong, X. Zhu, and W. Wu, "Multi retention level stt-ram cache designs with a dynamic refresh scheme," in *2011 44th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2011, pp. 329–338.
56. J. Wang, X. Dong, and Y. Xie, "Oap: An obstruction-aware cache management policy for stt-ram last-level caches," in *2013 Design, Automation Test in Europe Conference Exhibition (DATE)*, 2013, pp. 847–852.
57. G. Sun, X. Dong, Y. Xie, J. Li, and Y. Chen, "A novel architecture of the 3d stacked mram l2 cache for cmps," in *2009 IEEE 15th International Symposium on High Performance Computer Architecture*, 2009, pp. 239–249.
58. M. Imani, A. Rahimi, Y. Kim, and T. Rosing, "A low-power hybrid magnetic cache architecture exploiting narrow-width values," in *2016 5th Non-Volatile Memory Systems and Applications Symposium (NVMSA)*, 2016, pp. 1–6.
59. S. Angizi, Z. He, D. Reis, X. Hu, W. Tsai, S. J. Lin, and D. Fan, "Accelerating deep neural networks in processing-in-memory platforms: Analog or digital approach?" *2019 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pp. 197–202, 2019.
60. D. Reis, D. Gao, S. Angizi, X. Yin, D. Fan, M. Niemier, C. Zhuo, and X. S. Hu, "Modeling and benchmarking computing-in-memory for design space exploration," *Proceedings of the 2020 on Great Lakes Symposium on VLSI*, 2020.
61. S. Angizi, N. Khoshavi, A. Marshall, P. Dowben, and D. Fan, "Meram: Non-volatile cache memory based on magneto-electric fets," 2020.
62. Y. Seo and K. Roy, "High-density sot-mram based on shared bitline structure," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 8, pp. 1600–1603, Aug 26(8):1600–1603, 2018.
63. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'12. Red Hook, NY, USA: Curran Associates Inc., 2012, p. 1097–1105.
64. C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.
65. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014. [Online]. Available: <http://arxiv.org/abs/1409.1556>
66. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

67. 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.
68. J. J. Dongarra, M. Heroux, and P. Luszczek, “Hpcg benchmark: a new metric for ranking high performance computing systems,” 2015.
69. NVIDIA CUDA Profiler, <https://docs.nvidia.com/cuda/profiler-users-guide/nvprof-overview>.
70. J. Redmon, “Darknet: Open source neural networks in c,” <http://pjreddie.com/darknet/>, 2013–2016.
71. Y. Chen, T. Krishna, J. S. Emer, and V. Sze, “Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks,” *IEEE Journal of Solid-State Circuits*, vol. 52, no. 1, pp. 127–138, Jan 52(1):127-138, 2017.
72. M. Gao, J. Pu, X. Yang, M. Horowitz, and C. Kozyrakis, “Tetris: Scalable and efficient neural network acceleration with 3d memory,” *SIGARCH Computer Architecture News*, vol. 45, no. 1, pp. 751–764, 2017. [Online]. Available: <http://doi.acm.org/10.1145/3093337.3037702>
73. A. Boroumand, S. Ghose, Y. Kim, R. Ausavarungnirun, E. Shiu, R. Thakur, D. Kim, A. Kuusela, A. Knies, P. Ranganathan, and O. Mutlu, “Google workloads for consumer devices: Mitigating data movement bottlenecks,” in *Proceedings of the Twenty-Third International Conference on Architectural Support for Programming Languages and Operating Systems*, ser. ASPLOS ’18. New York, NY, USA: Association for Computing Machinery, 2018, p. 316–331. [Online]. Available: <https://doi.org/10.1145/3173162.3173177>
74. M. Donato, B. Reagen, L. Pentecost, U. Gupta, D. Brooks, and G. Wei, “On-chip deep neural network storage with multi-level envm,” in *2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC)*, 2018, pp. 1–6.
75. A. Kannan, K. Kurach, S. Ravi, T. Kaufmann, A. Tomkins, B. Miklos, G. Corrado, L. Lukács, M. Ganea, P. Young, and V. Ramavajjala, “Smart reply: Automated response suggestion for email,” *CoRR*, vol. abs/1606.04870, 2016. [Online]. Available: <http://arxiv.org/abs/1606.04870>
76. G. Tucker, M. Wu, M. Sun, S. Panchapagesan, G. Fu, and S. Vitaladevuni, “Model compression applied to small-footprint keyword spotting,” in *Interspeech 2016*, 2016, pp. 1878–1882. [Online]. Available: <http://dx.doi.org/10.21437/Interspeech.2016-1393>
77. C. Wu, D. Brooks, K. Chen, D. Chen, S. Choudhury, M. Dukhan, K. Hazelwood, E. Isaac, Y. Jia, B. Jia, T. Leyvand, H. Lu, Y. Lu, L. Qiao, B. Reagen, J. Spisak, F. Sun, A. Tulloch, P. Vajda, X. Wang, Y. Wang, B. Wasti, Y. Wu, R. Xian, S. Yoo, and P. Zhang, “Machine learning at facebook: Understanding inference at the edge,” in *2019 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, Feb 2019, pp. 331–344.
78. Z. Ghodsi, A. Veldanda, B. Reagen, and S. Garg, “Cryptonas: Private inference on a relu budget,” 2020.
79. K. Korgaonkar, I. Bhati, H. Liu, J. Gaur, S. Manipatruni, S. Subramoney, T. Karnik, S. Swanson, I. Young, and H. Wang, “Density tradeoffs of non-volatile memory as a replacement for sram based last level cache,” in *2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA)*, 2018, pp. 315–327.
80. 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 *2019 IEEE International Symposium on Workload Characterization (IISWC)*, 2019, pp. 143–154.
81. L. Pentecost, M. Donato, B. Reagen, U. Gupta, S. Ma, G.-Y. Wei, and D. Brooks, “Maxnv: Maximizing dnn storage density and inference efficiency with sparse encoding and error mitigation,” in *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO ’52. New York, NY, USA: Association for Computing Machinery, 2019, p. 769–781. [Online]. Available: <https://doi.org/10.1145/3352460.3358258>
82. H. Li, M. Bhargav, P. N. Whatmough, and H. . Philip Wong, “On-chip memory technology design space explorations for mobile deep neural network accelerators,” in *2019 56th ACM/IEEE Design Automation Conference (DAC)*, 2019, pp. 1–6.