# DeepNVM: A Framework for Modeling and Analysis of Non-Volatile Memory Technologies for Deep Learning Applications

Ahmet Inci, Mehmet Meric Isgenc, Diana Marculescu
Carnegie Mellon University
Department of Electrical and Computer Engineering, Pittsburgh, PA, USA
*ainci@andrew.cmu.edu, misgenc@andrew.cmu.edu, dianam@cmu.edu*

***Abstract*—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 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 4.2× and 5× energy-delay product (EDP) reduction and 2.4× and 3× area reduction compared to conventional SRAM, respectively. Under iso-area assumptions, STT-MRAM and SOT-MRAM provide 2.3× EDP reduction on average across all workloads when compared to SRAM. 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 GPU platforms for deep learning applications.**

## I. Introduction

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 classification and speech recognition 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. Non-volatile memory (NVM) technology is one of the most promising solutions to tackle memory bottleneck problem for data-intensive applications. 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 deep learning (DL) workloads.

In this work, we present *DeepNVM*, a framework to characterize, model, and analyze 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 (STT-MRAM) and spin-orbit torque magnetic random access memory (SOT-MRAM), keeping in mind that it can be used for analyzing 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* (PPA) 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.

## II. Related Work and Paper Contributions

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. To move beyond the computing limitations imposed by staggering CMOS scaling trends, MRAM has emerged as a promising candidate.

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 [1] and relaxing non-volatility of NVM to reduce its high write latency [2]. While these studies have 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 for DL workloads.

The most commonly used modeling tool for emerging NVM technologies is *NVSim* [3], 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. In this paper, we incorporate *NVSim* with our novel architecture-level iso-capacity and iso-area

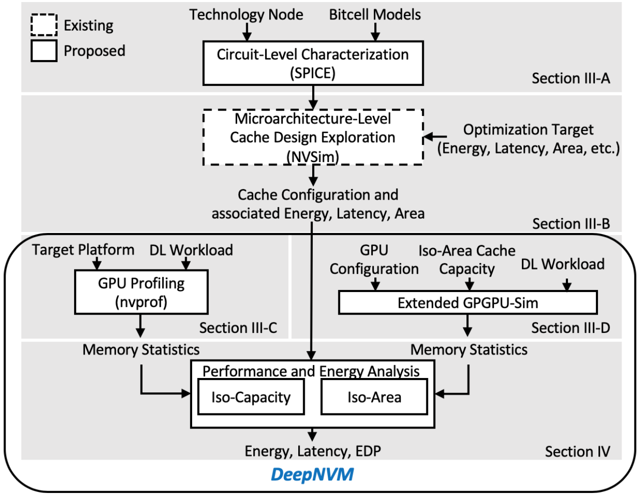

Fig. 1: Overview of the cross-layer analysis flow

analysis flow to perform design space exploration for conventional SRAM and emerging NVM caches for DL workloads. This paper makes the following contributions:

1) **Circuit-level bitcell characterization.** We perform detailed circuit-level characterization combining a commercial 16nm CMOS technology and prominent STT [4] and SOT [5] models from the literature to iterate through our framework in an end-to-end manner to demonstrate the flexibility of our framework for future studies.
2) **Microarchitecture-level cache design exploration.** We use *NVSim* [3] to perform a fair comparison between SRAM, STT-MRAM, and SOT-MRAM by incorporating 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 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 [6] on a high-end NVIDIA 1080 Ti GPU (implemented in 16nm technology) for the ImageNet dataset [7].
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 [8] simulator to run DL workloads and support larger cache capacities for STT-MRAM and SOT-MRAM.

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.

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

The rest of the paper 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 for STT-MRAM, SOT-MRAM, and conventional SRAM. Finally, Section V concludes the paper by summarizing the results.

## III. Methodology

### A. Circuit-level NVM Characterization

A vast majority of work in the literature uses simple bitcell models [9] 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[1] 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. For our simulations, we used perpendicular to the plane STT [4] and SOT [5] models and a commercial 16nm FinFET model that takes post-layout effects into account. To 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. We summarize the obtained bitcell parameters in Table I. We use these bitcell parameters for cache design exploration as discussed in Section III-B.

### B. Microarchitecture-level Cache Design Exploration

In order to demonstrate the impact of using STT and SOT bitcells in L2 caches, we use *NVSim* [3], 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 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.

Based on the optimization target used in *NVSim*, the cache PPA values vary substantially. Therefore, we choose the best configuration for each type of memory technology in terms of energy-delay-area product (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 16nm bitcell design. Next, we compare SRAM, STT-MRAM, and SOT-MRAM for various cache capacities in terms of area, latency, and energy results.

[1] Details about the commercial bitcell design cannot be shared due to non-disclosure agreement terms.

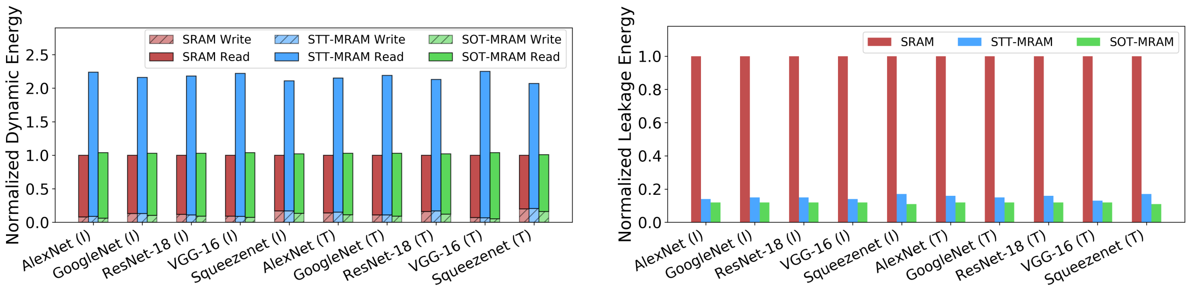

Fig. 2: Dynamic energy (left chart) and leakage energy (right chart) (lower is better) normalized with respect to SRAM by using NVMs with iso-capacity (3MB) for inference (I) and training (T) stages

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 | 4.47 | 6.68 |
| Write Latency (ns) | 1.53 | 9.31 | 10.06 | 1.34 | 2.46 |
| Read Energy (nJ) | 0.35 | 0.81 | 0.93 | 0.37 | 0.51 |
| Write Energy (nJ) | 0.32 | 0.31 | 0.43 | 0.25 | 0.39 |
| Leakage Power (mW) | 6442 | 748 | 1463 | 563 | 1434 |
| Area ($mm^2$) | 5.53 | 2.34 | 5.12 | 1.83 | 5.5 |

Table II shows the latency, energy, and area results that correspond to the cache capacity of 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.

**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.

### 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 16nm technology node which also matches our bitcell and cache models. We use the *Caffe* [6] framework to run various DNNs such as AlexNet, GoogLeNet, VGG-16, ResNet-18, and SqueezeNet for the ImageNet [7] dataset. We use the NVIDIA profiler [10] to obtain the device memory and L2 cache read and write transactions to better understand both on-chip and off-chip memory behavior.

### 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* [8] to explore power and performance implications of having these larger L2 caches in GPU architectures for DL workloads. For comparison, we model the high-end GTX 1080 Ti GPU. The configurations for 1080 Ti GPU are shown in Table III. 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 paper. 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 DL benchmark, we use AlexNet with the ImageNet dataset which is provided by the *DarkNet* [11] framework. We extend *DarkNet* source code to enable DL workloads on *GPGPU-Sim*.

TABLE III: 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 Section IV-A and Section IV-B, respectively.

### 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. Figure 2 shows normalized leakage energy and dynamic 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.

In Figure 2, we observe that STT-MRAM has 2.17× dynamic energy whereas SOT-MRAM has 1.02× dynamic

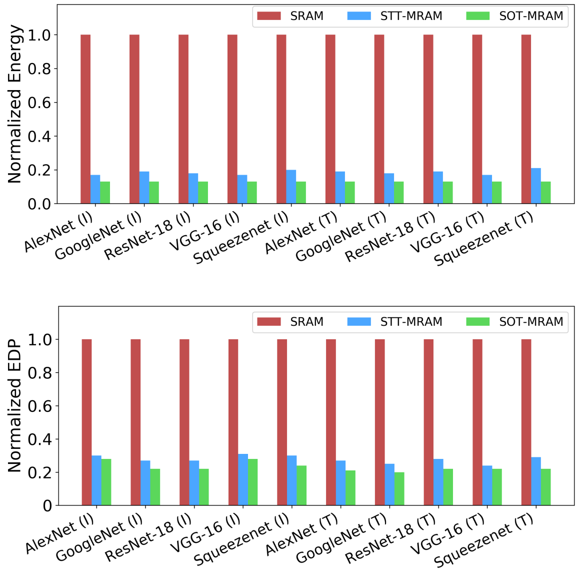

Fig. 3: Iso-capacity (3MB) energy and energy-delay product for NVM-based caches (lower is better) normalized with respect to SRAM-based caches for inference (I) and training (T) stages. DRAM energy and latency are also included in EDP results.

energy on average when compared to SRAM baseline. Furthermore, our results show that 87% of the total dynamic energy comes from read operations whereas write operations only make for 13% of all transactions on average across all workloads. Our profiling results also support these findings as read operations dominate write operations in DL workloads.

On the other hand, Figure 2 also shows that STT-MRAM and SOT-MRAM provide 6.6× and 8.5× lower leakage energy on average when compared to SRAM, respectively. Based on this, Figure 3 shows significant total normalized energy reduction of STT-MRAM and SOT-MRAM compared to SRAM given that leakage energy dominates the total energy. In more detail, STT-MRAM and SOT-MRAM achieve 5.6× *and* 7.7× *energy reduction* on average across all workloads compared to SRAM baseline, respectively. Moreover, Figure 3 shows that STT-MRAM and SOT-MRAM provide up to 4.2× *and* 5× *EDP reduction and* 2.4× *and* 3× *area reduction*, respectively.

### B. Performance and Energy Results for Iso-Area

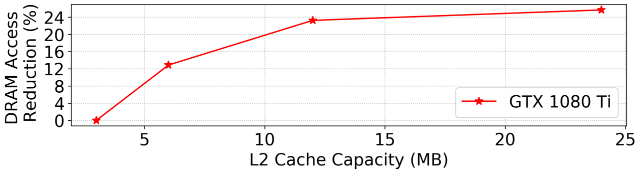

Fig. 4: Simulation results for the reduction in the total number of DRAM accesses in percentage (%)

As in the iso-capacity study, for iso-area analysis we use a batch size 4 for inference and 64 for training. Figure 4 shows the reduction in the total number of DRAM accesses as L2 cache capacity increases. We use *GPGPU-Sim* and start

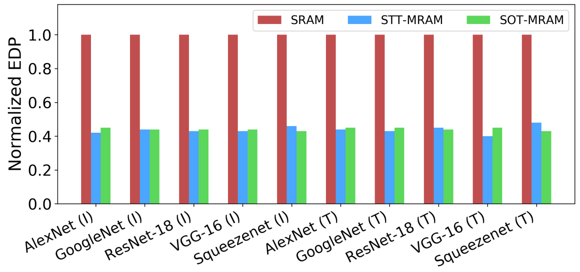

Fig. 5: Iso-area energy-delay product for NVM-based caches (lower is better) normalized with respect to SRAM-based caches for inference (I) and training (T) stages. DRAM energy and latency are also included in the results.

with the baseline configuration which is 3MB for GTX 1080 Ti and double its cache capacity up to 24MB to quantify the percentage of DRAM access reduction for STT-MRAM and SOT-MRAM at larger cache capacities. Figure 4 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. When DRAM accesses are included in determining EDP, as shown in Figure 5, STT-MRAM and SOT-MRAM provide 2.3× *EDP reduction* on average across all workloads when compared to SRAM.

## V. Conclusion

In this paper, 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 deep learning applications for different design choices such as technology nodes, bitcell models, deep learning workloads, cache configurations, optimization targets, and target platforms.

## VI. Acknowledgements

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

## References

[1] G. Li, *et al,*, "A stt-ram-based low-power hybrid register file for gpgpus," in *DAC*, 2015, pp. 1–6.

[2] C. W. Smullen, *et al.*, "Relaxing non-volatility for fast and energy-efficient stt-ram caches," in *HPCA*, 2011, pp. 50–61.

[3] X. Dong, *et al.*, "Nvsim: A circuit-level performance, energy, and area model for emerging nonvolatile memory," *TCAD*, 31(7):994-1007, 2012.

[4] J. Kim, *et al.*, "A technology-agnostic mtj spice model with user-defined dimensions for stt-mram scalability studies," in *CICC*, 2015, pp. 1–4.

[5] M. Kazemi, *et al.*, "Compact model for spin–orbit magnetic tunnel junctions," *IEEE Transactions on Electron Devices*, 63(2), 2016.

[6] Y. Jia, *et al.*, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[7] J. Deng *et al.*, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009.

[8] A. Bakhoda *et al.*, "Analyzing cuda workloads using a detailed gpu simulator," in *ISPASS*, 2009, pp. 163–174.

[9] R. Bishnoi, *et al.*, "Architectural aspects in design and analysis of sot-based memories," *ASP-DAC*, pp. 700–707, 2014.

[10] *NVIDIA CUDA Profiler, https://docs.nvidia.com/cuda/profiler-users-guide/nvprof-overview.*

[11] J. Redmon, "Darknet: Open source neural networks in c," http://pjreddie.com/darknet/, 2013–2016.