System Level Energy Comparison of DRAM and MRAM for Frame‑based MobileNetV3 Inference

Che-Ping Lin
Independent Researcher
Hsinchu City, Taiwan

ABSTRACT

Energy efficiency is a critical requirement for power-constrained inference workloads, especially in edge scenarios where data are often processed in a frame-oriented manner. While MRAM has been widely explored for low-standby-power systems, its system-level energy behavior under practical inference-driven memory access patterns still requires careful evaluation. A comparative energy analysis is presented for MRAM and DRAM memory access for frame-based MobileNetV3 inference under high-resolution (4K-class) input scenarios, where the memory access intensity is scaled to reflect high-resolution frame workloads rather than pixel-level convolutional dataflows. A counter-based, event-driven energy estimation framework is used to account for DRAM background activity, read/write traffic, and refresh overhead, as well as MRAM read/write energy using a charged-cycle abstraction. The evaluation explicitly incorporates frame gaps, enabling power-gating opportunities and a fair comparison between volatile and non-volatile memory domains under the same workload timeline. Experimental results show that DRAM energy is dominated by continuous background power and refresh overhead, whereas MRAM achieves substantially lower energy per frame due to low standby leakage and effective power gating during frame intervals. These findings highlight the importance of workload-aware memory energy evaluation and suggest that MRAM is a promising memory option for energy-efficient frame-based inference in power-limited systems.

General Terms

Performance, Measurement, Design, Algorithms

Keywords

System-level energy analysis, embedded MRAM, DRAM, MobileNetV3, frame-based inference, power-gating

1. INTRODUCTION

The rapid deployment of deep neural networks in power-constrained systems has intensified the demand for energy-efficient inference. In many practical scenarios, particularly in edge and embedded environments, inference workloads are executed in a frame-oriented manner rather than as continuous data streams. Each inference is triggered by an input frame, followed by an idle interval before the next frame arrives. Under such conditions, memory energy consumption plays a dominant role in overall system power, as memory subsystems remain active even when computation is intermittent. Conventional DRAM-based memory systems, such as DRAM, incur substantial background energy due to continuous standby power and periodic refresh operations, regardless of workload activity. This characteristic becomes increasingly inefficient for frame-based inference workloads, where memory access is bursty and interleaved with idle gaps. In contrast, emerging non-volatile memories such as Magnetoresistive Random Access Memory (MRAM) offer near-zero standby leakage and enable aggressive power gating during inactive periods. While MRAM has been widely discussed as a promising candidate for energy-constrained systems, a quantitative system-level comparison between MRAM and DRAM under realistic inference-driven memory access patterns remains limited [3]. Recent studies on neural network acceleration and edge inference have primarily focused on computational efficiency, model compression, or hardware acceleration techniques. However, memory energy behavior under inference workloads is often abstracted or treated as a secondary concern. Moreover, many existing analyses assume continuous data processing or pixel-level convolutional dataflows, which do not accurately reflect the execution characteristics of frame-based inference in practical systems. As a result, the energy implications of memory background activity, refresh overhead, and frame gaps are frequently overlooked [4].

A frame-based inference workload derived from MobileNetV3 is used to evaluate and compare the memory energy behavior of MRAM and DRAM under high-resolution, 4K-class input scenarios. In the following discussion, MRAM refers to embedded MRAM used as on-chip non-volatile memory for weight storage. Rather than modeling pixel-level operations, the analysis focuses on memory access intensity scaled to represent high-resolution frames and captures system-level memory activity over complete inference timelines. A counter-based, event-driven energy estimation framework is employed to account for read/write operations, background activity, and refresh overhead, while explicitly incorporating frame gaps that enable power-gating opportunities. Through this approach, the energy characteristics of volatile and non-volatile memory systems are compared under identical workload conditions [1]. The results demonstrate that DRAM energy consumption is largely dominated by background power and refresh activity, whereas MRAM benefits significantly from low standby leakage and effective power gating during frame intervals. These findings emphasize the importance of workload-aware memory energy evaluation and provide insight into memory selection for energy-efficient frame-based inference in power-limited systems [6].

2. METHODOLOGY

2.1 Workload Characterization

The evaluation is based on a frame-based inference workload derived from MobileNetV3. Instead of modeling pixel-level convolutional dataflows, the workload is abstracted at the memory access level to capture inference-driven read and write activity over discrete frames. The frame-based inference workload is executed on a cycle-level system model that emulates inference-driven memory access behavior and generates time-stamped memory read and write events, which are subsequently aggregated into activity counters for energy estimation. The overall evaluation flow is illustrated in Figure

1 [7]. The figure summarizes how frame-level inference activity is converted into memory access events, aggregated into activity counters, and finally mapped to DRAM and MRAM energy estimates. This visual flow clarifies the relationship between workload timing, counter accumulation, and the final system-level energy comparison.

Each inference frame generates a burst of memory accesses, followed by an idle interval before the arrival of the next frame, forming a frame-oriented execution timeline commonly observed in practical inference scenarios. This abstraction allows the temporal characteristics of inference execution to be preserved without relying on layer-level or hardware-specific modeling details [9].

To represent high-resolution input conditions, the memory access intensity is scaled to emulate 4K-class frame workloads. This scaling increases the volume of memory read and write operations per frame while preserving the temporal structure of frame execution. By focusing on access intensity and timing rather than detailed computation, the workload abstraction enables a system-level comparison of memory energy behavior under consistent inference conditions [10].

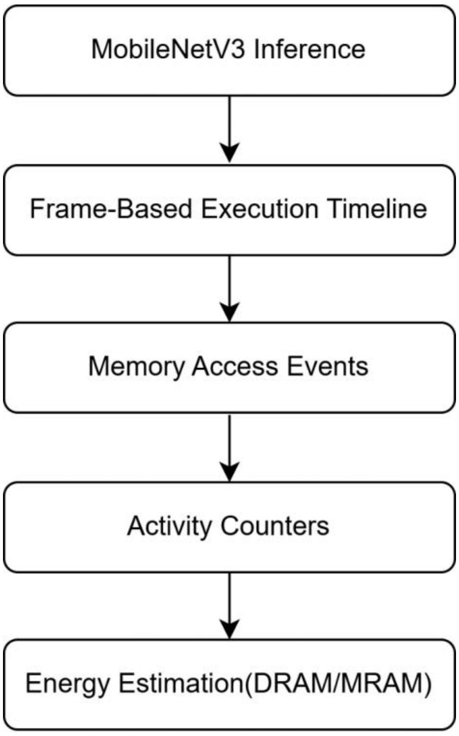

Figure 1. System-level energy evaluation flow for frame-based inference workloads

2.2 Model Scope and Assumptions

The proposed power model is a system-level, event-driven abstraction intended to compare memory energy trends across different memory mappings (DRAM versus MRAM) and power-gating policies. The model is not intended to replace circuit-level power signoff or device-level characterization. DRAM parameters are inspired by publicly available IDD-based power estimation methodologies, which decompose memory energy consumption into background, read, and write current components. MRAM parameters follow commonly reported trends in the literature, including near-zero standby leakage and higher write cost relative to read operations. To ensure robustness, sensitivity analysis is employed to confirm that the main conclusions are not dependent on specific parameter choices [8].

2.3 Event Counters and Timing Definitions

Let the system clock frequency be denoted as $Fclk = F_{(clk)}$, with a clock period of $T_{(clk)} = 1/F_{(clk)}$. Over a single inference frame, the following counters are accumulated:

- total number of cycles within the frame.
- number of cycles during which the memory domain is powered on.
- number of powered-on cycles without read or write activity.
- number of cycles with read beat handshakes.
- number of cycles with write beat handshakes.
- number of refresh busy cycles (if refresh is enabled).

For MRAM, additional operation-level counters are collected:

- number of MRAM read operations.
- number of MRAM write operations.
- $N^{m}_{(rd)}$: total charged cycles for MRAM read operations.
- $N^{m}_{(wr)}$: total charged cycles for MRAM write operations.

The duration of one inference frame is defined as:

$$t_{(frame)} = N_{(total)}T_{(clk)}$$

2.4 DRAM Energy Model

The DRAM memory domain is modeled using two supply rails, VDD = V_{DD} and VDDQ = V_{DDQ}. Energy is accumulated only when the memory domain is powered on, ensuring that reporting time does not affect the final energy result.

Core Rail Energy

The energy consumption on the core rail is expressed as:

$$E_{(DD)} = V_{(DD)}T_{(clk)}\left(N_{(on)}I^{(DD)}_{(leak)} + N_{(bg)}I^{(DD)}_{(bg)} + N_{(rd)}I^{(DD)}_{(rd)} + N_{(wr)}I^{(DD)}_{(wr)}\right)$$

I/O Rail Energy

Similarly, the energy consumption on the I/O rail is given by:

$$E_{(DDQ)} = V_{(DDQ)}T_{(clk)}\left(N_{(on)}I^{(DDQ)}_{(leak)} + N_{(bg)}I^{(DDQ)}_{(bg)} + N_{(rd)}I^{(DDQ)}_{(rd)} + N_{(wr)}I^{(DDQ)}_{(wr)}\right)$$

Total DRAM Energy

The total DRAM energy per frame is computed as:

$$E\_(DRAM) = E\_(DD) + E\_(DDQ)$$

Refresh operations are modeled as an energy overhead using a refresh busy-window abstraction. The associated energy is approximated using background current levels, rather than explicit command-level stalling.

2.5 MRAM Energy Model

The MRAM memory domain is modeled using a single supply rail, V_MRAM. Energy consumption is estimated using a charged-cycle abstraction to approximate internal read and write costs.

Charged Cycle Definition

The total charged cycles for MRAM operations are defined as:

$$N^{m}_{(rd)} = N_{(rd_op)}C_{(rd)}, N^{m}_{(wr)} = N_{(wr_op)}C_{(wr)}$$

where $C_{(rd)}$ and $C_{(wr)}$ denote the fixed number of charged cycles per read and write operation, respectively.

MRAM Energy

The MRAM energy per frame is calculated as:

$$E_{(MRAM)} = V_{(MRAM)}T_{(clk)}\left(N_{(on)}I^{m}_{(leak)} + N^{m}_{(rd)}I^{m}_{(rd)} + N^{m}_{(wr)}I^{m}_{(wr)}\right)$$

Consistent with prior studies, the model assumes $I^{m}_{(wr)} > I^{m}_{(rd)}$ and a very small $I^{m}_{(leak)}$, reflecting the low standby leakage and higher write cost characteristics of MRAM technology.

3. Parameter Settings

This section summarizes the parameter settings used in the energy evaluation framework. The selected parameters are intended to reflect representative system-level characteristics rather than device-specific implementations. Unless otherwise stated, all parameters are applied consistently across all experiments to ensure a fair comparison between DRAM and MRAM memory domains.

3.1 System and Timing Parameters

The system operates at a fixed clock frequency $F_{(clk)}$, which defines the temporal resolution of the cycle-level model. All timing-related counters and energy accumulation are derived from this clock. The frame duration is determined by the total number of cycles per inference frame, as defined in Section 2.3. Frame gaps are explicitly modeled as idle intervals between consecutive inference frames. During these intervals, the memory domain may remain powered on or enter a power-gated state, depending on the evaluated memory technology. This distinction is critical for capturing background and standby energy behavior under frame-based workloads.

3.2 DRAM Parameter Settings

DRAM energy parameters are inspired by publicly available IDD-based power estimation methodologies, which decompose memory current consumption into background, read, write, and leakage components. Two supply rails are modeled: the core rail ($V_{(DD)}$) and the I/O rail ($V_{(DDQ)}$).

Background current parameters represent powered-on cycles without active read or write transactions, while read and write currents capture the incremental energy associated with memory access activity. Refresh behavior is modeled using a fixed refresh interval and a refresh busy window, during which background current is assumed to dominate energy consumption. All DRAM parameters are selected to reflect relative energy trends rather than absolute device specifications. Sensitivity analysis is conducted to confirm that the comparative conclusions remain consistent across reasonable parameter variations.

3.3 MRAM Parameter Settings

The MRAM memory domain is modeled using a single supply rail ($V_{(MRAM)}$). Consistent with commonly reported characteristics of MRAM technologies, standby leakage current is assumed to be negligible compared to DRAM. Read and write operations incur energy costs proportional to their charged cycles, with write operations modeled as more energy-intensive than reads. The charged cycle parameters $C_{(rd)}$ and $C_{(wr)}$ are selected to approximate internal MRAM access behavior at a system level. These parameters are applied uniformly across all experiments. Similar to the DRAM case, parameter sensitivity is evaluated to ensure that the observed energy trends are not dependent on specific numerical choices.

3.4 Summary of Parameters

Table 1 summarizes the key parameters used in the evaluation, including supply voltages, current components, refresh settings, and charged-cycle definitions. The table serves as a reference for reproducibility and highlights the consistent parameter usage across all evaluated scenarios.

Table 1 summarizes the system-level parameters used in the proposed energy estimation framework. The values are selected to reflect representative trends reported in prior literature and publicly available power estimation methodologies, rather than device-specific implementations. All parameters are applied consistently across all evaluated scenarios to ensure a fair comparison between DRAM and MRAM.

Table 1. Summary of Energy Model Parameters

Category	Parameter	Symbol	Value	Unit	Description
System	Clock frequency	$F_{(clk)}$	200	MHz	System clock frequency
System	Clock period	$T_{(clk)}$	5	ns	Derived from F_clk
System	Frame duration	$N_{(total)}$	workload-dependent	cycles	Total cycles per inference frame
DRAM	Core supply voltage	$V_{(DD)}$	1.2	V	DRAM core rail
DRAM	I/O supply voltage	$V_{(DDQ)}$	1.2	V	DRAM I/O rail
DRAM	Core leakage current	$I^{(DD)}_{(leak)}$	representative	A	Standby leakage (powered-on)
DRAM	I/O leakage current	$I^{(DDQ)}_{(leak)}$	representative	A	I/O standby leakage
DRAM	Core background current	$I^{(DD)}_{(bg)}$	representative	A	Powered-on, no access
DRAM	I/O background current	$I^{(DDQ)}_{(bg)}$	representative	A	I/O background current

DRAM	Core read current	$I_{(rd)}^{(DD)}$	representative	A	Incremental read activity
DRAM	Core write current	$I_{(wr)}^{(DD)}$	representative	A	Incremental write activity
DRAM	Refresh period	$T_{(ref)}$	7.8	µs	Standard DRAM refresh interval
DRAM	Refresh busy cycles	$N_{(ref)}$	32	cycles	Refresh busy window
MRAM	Supply voltage	$V_{(MRAM)}$	0.8	V	MRAM single rail
MRAM	Standby leakage current	$I_{(leak)}^{m}$	negligible	A	Near-zero leakage
MRAM	Read current	$I_{(rd)}^{m}$	representative	A	Read access current
MRAM	Write current	$I_{(wr)}^{m}$	representative	A	Write access current
MRAM	Read charged cycles	$C_{(rd)}$	2	cycles	Charged cycles per read
MRAM	Write charged cycles	$C_{(wr)}$	20	cycles	Charged cycles per write

4. EXPERIMENTAL METHOD AND RESULTS

This hybrid placement is intentional and reflects a realistic SoC memory hierarchy for edge vision: (i) the input frame and intermediate activations are streamed through external DRAM (high capacity and bandwidth), while (ii) CNN weights are stored in embedded non-volatile MRAM to avoid repeated DRAM weight fetches and to enable aggressive power gating. Such a split is common in practice because weights are read-dominant and reused across frames, whereas frame/activation traffic is large and transient. Although MRAM may occupy larger on-die area than SRAM/DRAM-equivalent capacity, its non-volatility allows the MRAM domain to be completely turned off during VBLANK/idle intervals without losing model parameters, which directly reduces background (standby) energy. Therefore, comparing DRAM background against MRAM background under identical workload assumptions is meaningful: the system-level power difference is mainly driven by (a) DRAM refresh/standby current and (b) MRAM power-gating opportunity, while the weight-access-only energy remains comparable when normalized to the same effective weight bytes [5]. Interpretation: the absolute DRAM power is dominated by frame/activation traffic plus refresh/standby current; MRAM contributes mainly through weight traffic, and its standby power is minimized by power gating in VBLANK due to non-volatility. Hence a large DRAM-to-MRAM gap in background power is expected under this memory mapping.

4.1 Setup and Workload

The memory-domain energy of a frame-based inference workload is evaluated using a transaction-level AXI/NoC simulation with a lightweight power monitor attached to each memory slave. The workload includes (i) input-frame traffic and intermediate/output buffer updates (DRAM/SRAM), and (ii) NPU weight fetches whose placement ratio between DRAM and embedded MRAM is swept from 0% to 100%. The clock frequency is fixed at 200 MHz for all runs [2].

4.2 Power Monitor Model

Each monitor accumulates energy using event-driven cycle counters (BG/RD/WR/REF) and the rail-current parameters summarized in Table 2. DRAM uses two rails (VDD and VDDQ) and includes refresh, whereas MRAM uses a single VDD rail without refresh, with standby captured by the static leakage current parameter MRAM_I_LEAK_A (Table 2).

Table 2. Assumed voltage and current parameters for the DRAM/MRAM energy estimation model.

Parameter	**DRAM monitor**	**MRAM monitor**
FCLK_HZ	200 MHz	200 MHz
VDD / VDDQ	1.2 V / 1.2 V	0.8 V / 0 V

Leakage current	I_LEAK_VDD=0.020 A,
	I_LEAK_VDDQ=0.010 A
		MRAM_I_LEAK_A=0.0001 A

Background delta	I_BG_VDD=0.012 A,
	I_BG_VDDQ=0.006 A
		Not modeled (standby captured by MRAM_I_LEAK_A)

Read delta	I_RD_VDD=0.028 A,
	I_RD_VDDQ=0.012 A
		MRAM_I_RD_A=0.03 A

Write delta	I_WR_VDD=0.032 A,
	I_WR_VDDQ=0.014 A
		MRAM_I_WR_A=0.055 A

Refresh	REF_EN=1, REF_PERIOD=1560 cyc, REF_BUSY=32 cyc	REF_EN=0

Charged cycles	N/A (DRAM uses direct BG/RD/WR/REF counters)	MRAM_RD_CYCLES=2,
		MRAM_WR_CYCLES=20

4.3 Results

This subsection reports (i) isolated weight-access-only energy and (ii) background-domain energy. Weight-only numbers are extracted using the built-in weight-range tracking (WEIGHT_BASE / WEIGHT_BYTES), which counts only transactions whose addresses fall inside the configured weight region. Background energy corresponds to the remaining domain energy and is dominated by standby/leakage; for DRAM, it also includes refresh overhead. Table 3 lists the weight-write energy versus MRAM weight placement ratio. This is a one-time "model installation" cost (e.g., at boot time or when updating the model). Because MRAM writes are modeled with longer charged cycles and higher write current, MRAM write energy is higher than DRAM, so increasing MRAM placement increases the write component. As shown in Table 3, the weight-write energy increases with MRAM placement because more weights are programmed into MRAM. The 0% case is the DRAM-only reference and the 100% case fully programs the weights into MRAM. Table 3 is included to quantify this one-time provisioning cost, which occurs only during model installation or updates.

Quantitatively, the total weight-write energy increases from 18.088 µJ at 0% MRAM placement to 28.836 µJ at 100% MRAM placement, showing a monotonic increase as a larger fraction of weights is programmed into MRAM. At the same time, the DRAM write component decreases while the MRAM write component increases, reflecting the migration of one-time model provisioning traffic from DRAM to MRAM. Although this write cost is higher for MRAM, it is incurred only during model installation or update events and therefore does not dominate the steady-state energy behavior of frame-by-frame inference.

Table 3. Isolated weight-write-only energy versus MRAM weight placement.

MRAM weight (%)	DRAM weight-write (µJ)	MRAM weight-write (µJ)	Total weight-write (µJ)
0	18.088	0.000	18.088
20	14.470	5.767	20.237
40	10.853	11.534	22.387
60	7.235	17.302	24.537
80	3.618	23.069	26.686
100	0.000	28.836	28.836

Table 4 reports the weight-read energy per inference window in the frame-based workload. As the MRAM placement ratio increases, DRAM weight-read energy decreases while MRAM weight-read energy increases. The total weight-read energy decreases because DRAM read activity includes both core and I/O rails plus a larger activity delta, whereas MRAM read is modeled without an I/O rail and with a lower read current.

Quantitatively, the total weight-read energy decreases from 3.226 µJ at 0% MRAM placement to 2.903 µJ at 100% MRAM placement. The reduction is monotonic across all placement ratios, indicating a gradual benefit as a larger fraction of weight-read traffic is shifted from DRAM to MRAM. Although the reduction is modest in absolute magnitude, it is consistent across all cases and reflects the lower modeled per-read overhead of MRAM in the weight-access path.

Table 4. Isolated weight-read-only energy versus MRAM weight placement.

MRAM weight (%)	DRAM weight-read (µJ)	MRAM weight-read (µJ)	Total weight-read (µJ)
0	3.226	0.000	3.226
20	2.573	0.588	3.160
40	1.920	1.175	3.095
60	1.286	1.745	3.032
80	0.634	2.333	2.966
100	0.000	2.903	2.903

To avoid misinterpretation from total system energy alone, Table 5 compares background-domain energy. DRAM background is dominated by standby and refresh and is therefore only weakly sensitive to weight placement when frame traffic and timing are fixed. A small variation (≈1%) can still appear because background is computed as a residual after subtracting the weight-range component, and minor shifts in arbitration/refresh alignment slightly change the counted on-window cycles. MRAM has no refresh; its background is captured by MRAM_I_LEAK when powered on and can be further reduced by power-gating during VBLANK. Overall, full-system energy/power remains weakly sensitive to weight placement because input frames and intermediate feature traffic stay in DRAM in all configurations, so DRAM background/refresh dominates while MRAM background stays small.

Quantitatively, the DRAM background energy remains in the range of approximately 1.56–1.58 mJ across all placement ratios, whereas the MRAM background contribution stays below 0.36 µJ. The DRAM background term is nearly constant, while the MRAM background term increases slightly as more weights are placed in MRAM, but its magnitude remains negligible compared with the DRAM contribution. This result indicates that the dominant system-level energy burden is determined primarily by DRAM standby and refresh behavior rather than by the residual background cost of MRAM.

Table 5. Background-domain energy (non-weight). MRAM values include VBLANK power-gating (MRAM OFF).

MRAM weight (%)	DRAM background energy (mJ)	MRAM background energy (µJ)
0	1.578951	0
20	1.570485	0.199706
40	1.568643	0.239418
60	1.566868	0.278483
80	1.565063	0.319882
100	1.563281	0.358426

Frequency Scaling Trend

To avoid overlapping curves in absolute system energy, which is dominated by DRAM frame and feature traffic, the results are presented as relative energy saving. Distinct line styles and markers are used to ensure interpretability in grayscale print. As shown in Figure 2, the energy saving at each frequency is computed as

$$\Delta E_{(sys)}(x\%) = E_{(sys)}(0\%) - E_{(sys)}(x\%)$$

where $x\%$ denotes the MRAM weight placement ratio, $E_{(sys)}(0\%)$ and $E\_(sys)(x\%)$ are measured at the same operating frequency. Positive values indicate lower total system energy when a larger fraction of weights is placed in MRAM. The energy saving increases with the MRAM weight placement ratio at all three frequencies. The trend is strongest at 100 MHz and becomes smaller at 200 MHz and 400 MHz, indicating that lower-frequency operation provides a longer time window over which the background-energy difference between DRAM and MRAM can accumulate.

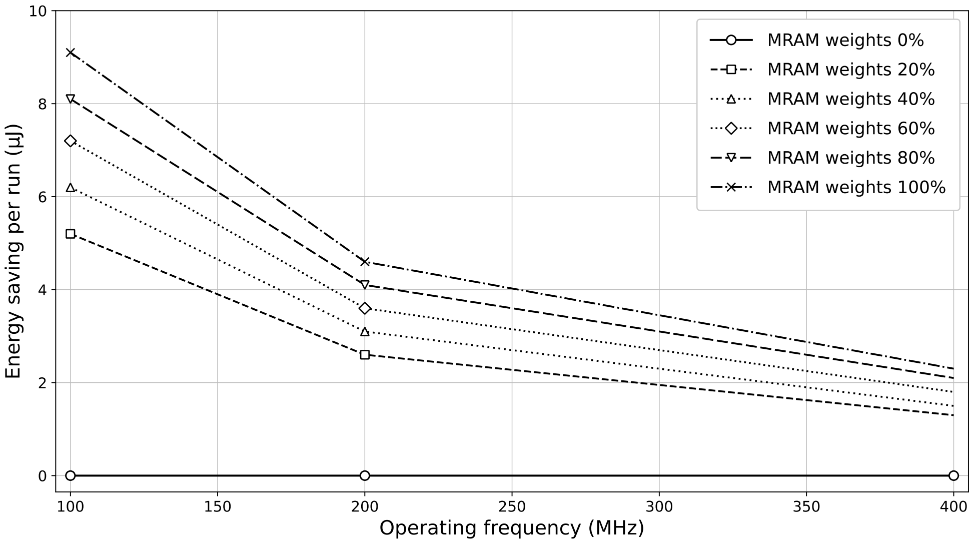

Figure 2. Energy saving per run versus operating frequency for different MRAM weight placement ratios (relative to the 0% weights-in-MRAM baseline at each frequency)

5. CONCLUSION

A system-level energy study is presented for frame-based CNN inference using two memory technologies: a conventional volatile DRAM domain for input frames and feature traffic, and an embedded MRAM (MRAM) domain for model weights. To enable a fair and interpretable comparison, the analysis separates (i) weight-access-only energy from full-system totals and (ii) active-window energy from time-averaged energy that includes power-gated intervals. Across 0–100% MRAM weight placement, the isolated weight-access energy behaves as expected: shifting weights from DRAM to MRAM transfers read/write energy from the DRAM domain to the MRAM domain, while the total weight-access energy remains within the same order of magnitude under the calibrated monitor parameters. These results support the feasibility of using MRAM as an on-chip weight store for frame-based inference without requiring a fundamentally different dataflow. At the full-system level, DRAM background energy (standby and refresh) together with non-weight frame/feature traffic dominates the budget because input frames and intermediate feature-map transfers reside in DRAM across all configurations. As a result, total system energy can appear only weakly sensitive to weight placement even when weight-access energy clearly shifts between domains. Finally, MRAM offers a key architectural advantage for frame-oriented workloads: aggressive power-gating during VBLANK (or other idle intervals) can reduce its time-averaged background contribution toward near zero without state loss, which is not practical for volatile DRAM. This indicates that the primary benefit of MRAM is not only relocating weights, but also reducing idle energy through non-volatile power-gating.

6. CODE AVAILABILITY

The RTL/testbench environment and scripts used in this study are available at GitHub:
https://github.com/acelin1981/MRAM-Based-AI-SOC

7. REFERENCES

[1] A. Howard et al., "Searching for MobileNetV3," in *Proc. IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, 2019, pp. 1314–1324, doi: 10.1109/ICCV.2019.00140.

[2] Arm, "AMBA Specifications (AXI/ACE, AXI4-Stream, APB, etc.)," Arm Developer. [Online]. Available: https://developer.arm.com/architectures/system-architectures/amba/amba-specifications. [Accessed: 13-Feb-2026].

[3] JEDEC Solid State Technology Association, "JESD79-4D: DDR4 SDRAM Standard," *JEDEC Standard.* https://store.accuristech.com/standards/jedec-jesd79-4d. [Accessed: 13-Feb-2026].

[4] C.-P. Lin, "Designing an AMBA-Compatible MRAM AXI Slave Controller for Modern SoCs," *Medium*, Nov. 19, 2025. https://medium.com/@ace.lin0121/designing-an-amba-compatible-mram-axi-slave-controller-for-modern-socs-3cade20bce41. [Accessed: 13-Feb-2026].

[5] D. Apalkov et al., "Spin-transfer torque magnetic random access memory (STT-MRAM)," ACM J. Emerg. Technol. Comput. Syst. (JETC), vol. 9, no. 2, Art. 13, pp. 1–35, 2013, doi: 10.1145/2463585.2463589.

[6] M. Sandler et al., "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 4510–4520.

[7] B. Jacob et al., "Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference," in Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR), 2018.

[8] Micron Technology, "TN-40-07: Calculating Memory Power for DDR4 SDRAM," Technical Note. [Online]. https://www.mouser.com/pdfdocs/tn4007_ddr4_power_calculation.pdf. [Accessed: 13-Feb-2026].

[9] R. Balasubramonian et al., "CACTI 7: New Tools for Interconnect Exploration in Innovative Off-Chip Memories," ACM Trans. Archit. Code Optim. (TACO), vol. 14, no. 2, Art. 14, 2017, doi: 10.1145/3085572.

[10] S. Yuasa and D. D. Djayaprawira, "Giant tunneling magnetoresistance in magnetic tunnel junctions with a crystalline barrier," *J. Phys. D: Appl. Phys.*, vol. 40, p. R337, 2007.