Retention-Aware Computing

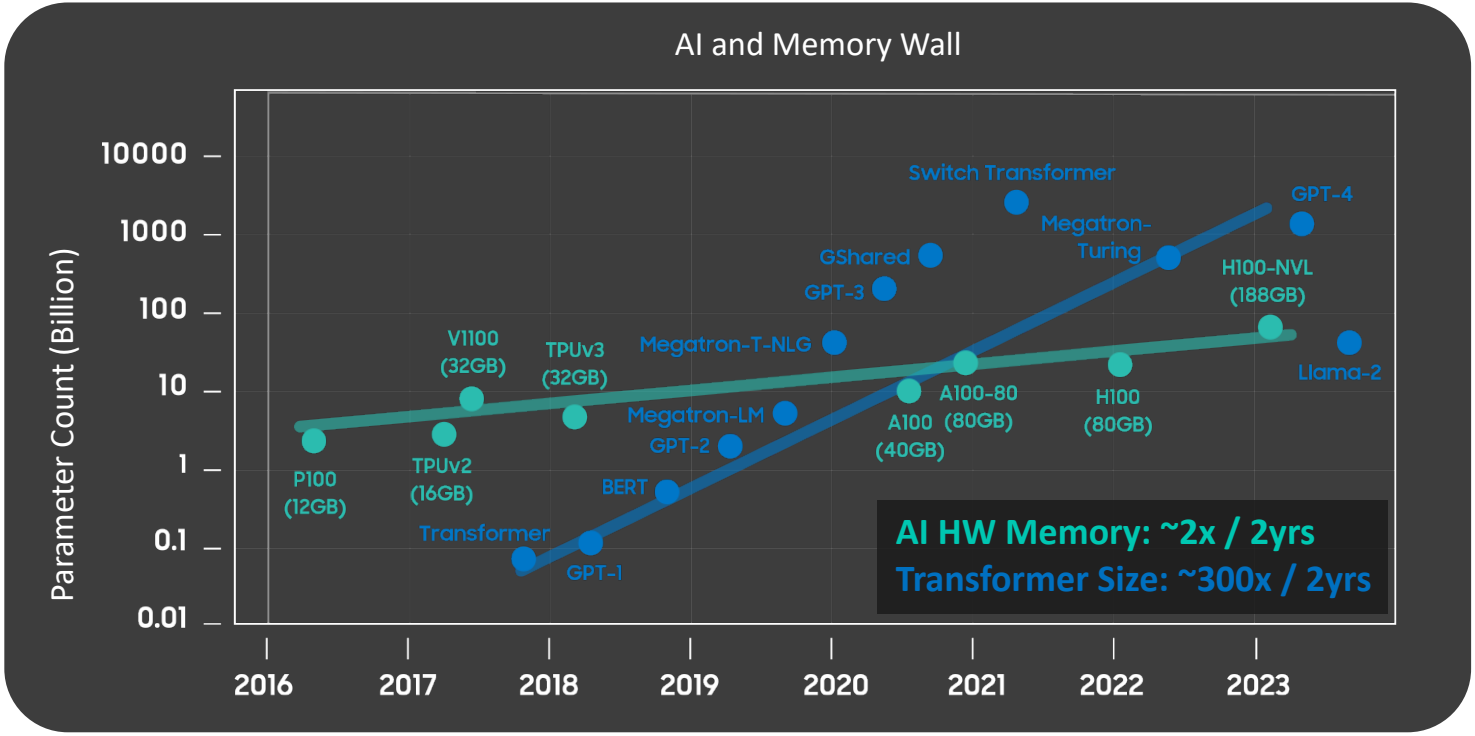
Thierry Tambe

MemoryDAX / DAM Workshop

January 21, 2026

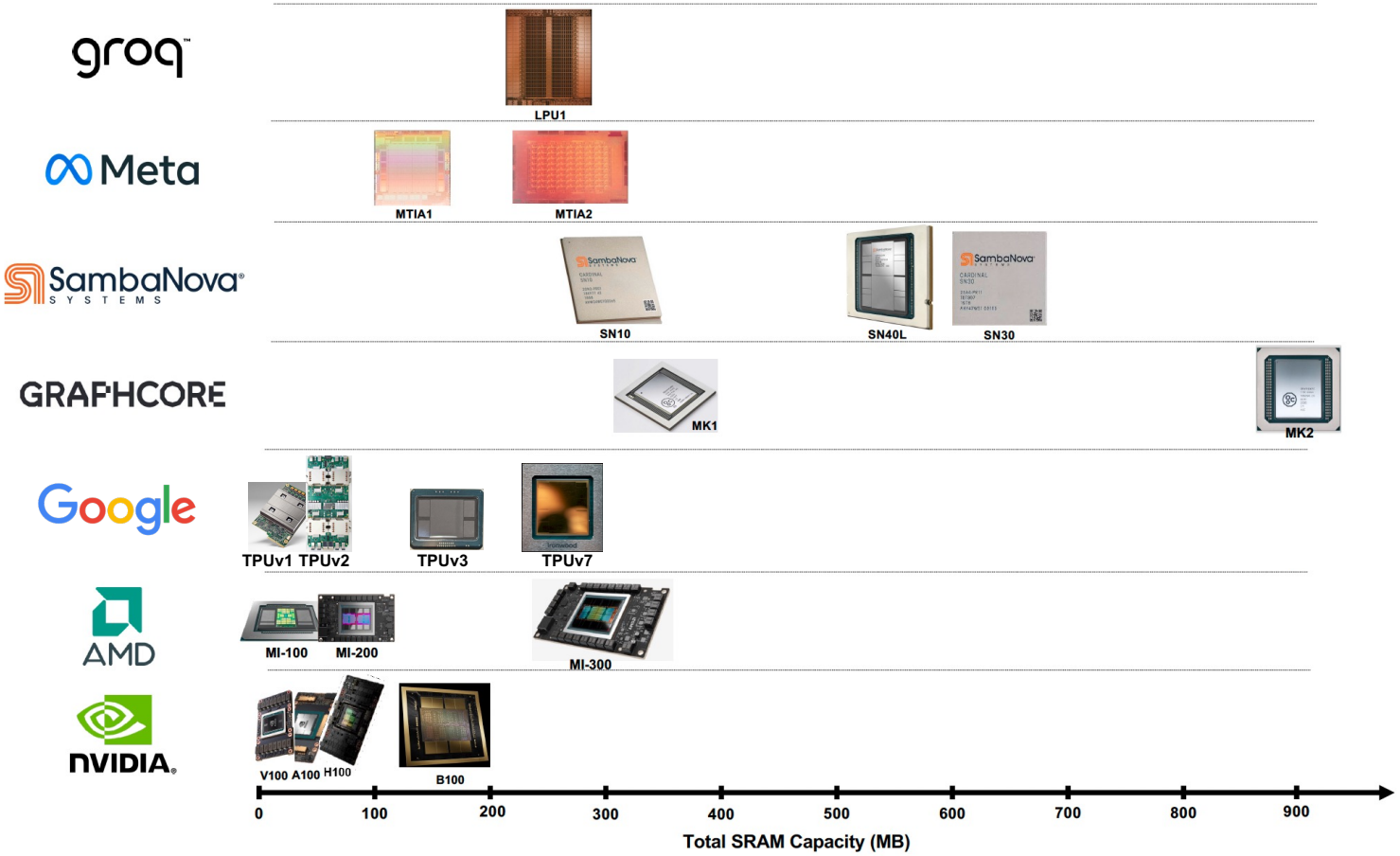
Stanford | ENGINEERING
Electrical Engineering

AI and the Memory Capacity Wall

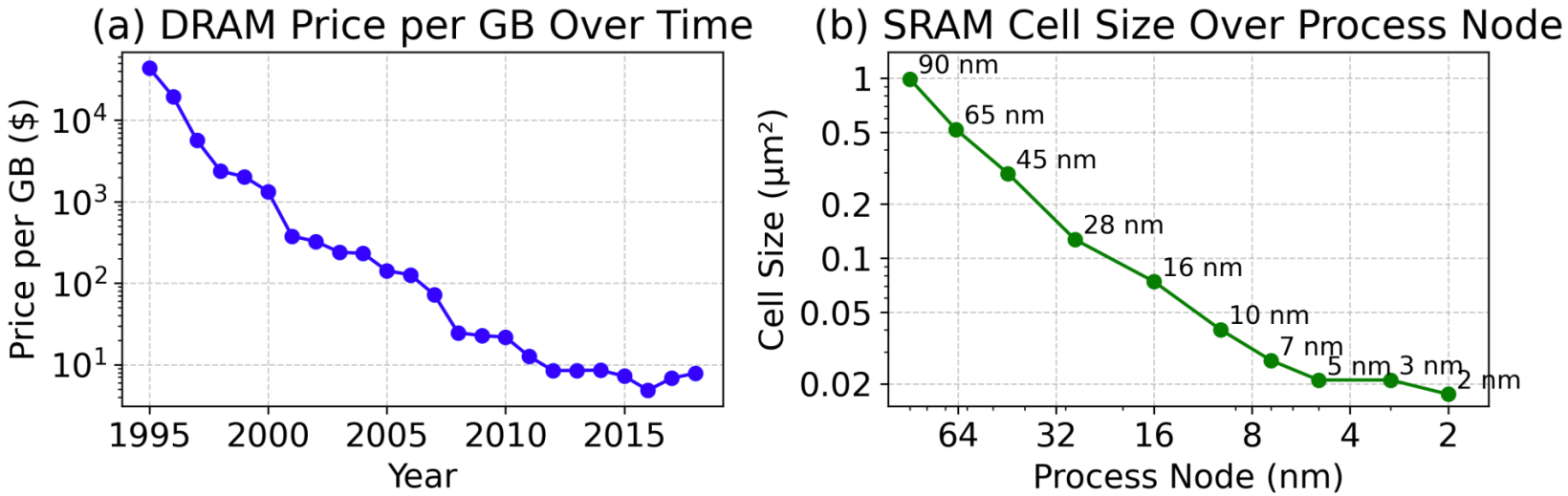


[A. Gholami, IEEE MICRO, 2024]

Towards higher on-chip capacity in AI HW



End of memory scaling



P. Li, “Towards Memory Specialization: A Case for Long-Term and Short-Term RAM” DIMES, 2025.

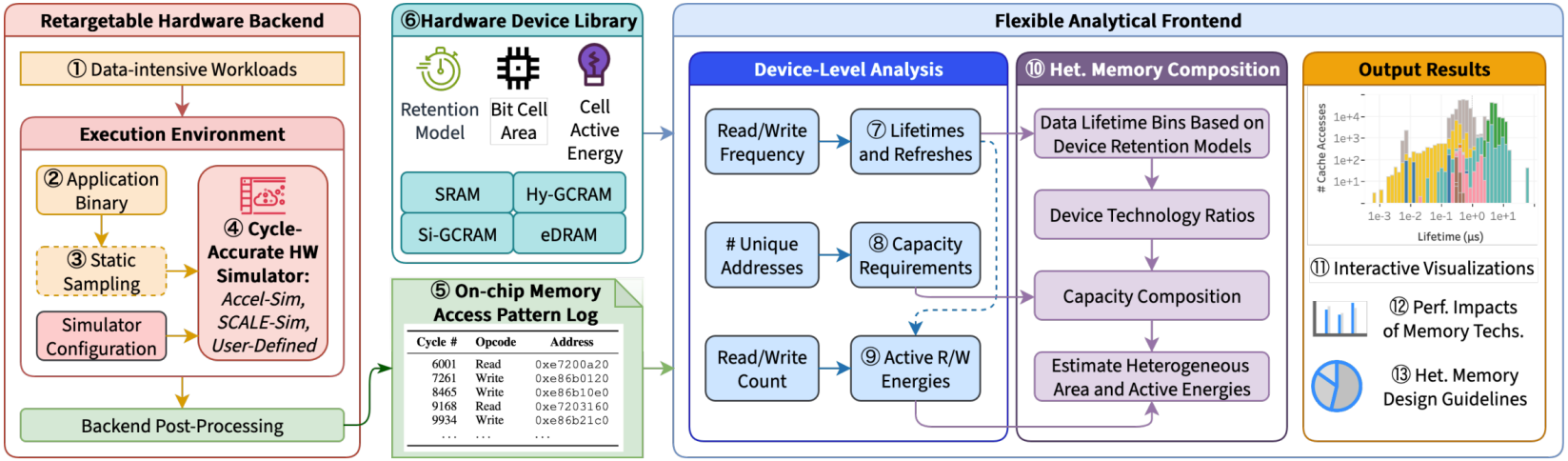
Is SRAM over-provisioned?

- Empirical observation: short- and long-lived data in AI models
  - Short-lived on-chip data: activations, KV cache
  - Long-lived on-chip data: weights
- SRAM offers great read/write latency, endurance, and data retention at the cost of area density and static power
- Its performance may be over-provisioned for both short- and long-lived AI/ML data
- Alternative devices can make better trade-offs to attain higher density and lower power

A Proposal for LtRAM and StRAM

	SRAM	DRAM	NAND	StRAM	LtRAM
Strengths	Low R/W latency, long retention, low static power	Very dense	Extremely dense, low static power	Dense, low write energy, low static power	Very dense, low read energy, low static power
Weaknesses	Low density	Off-chip only, high R/W energy, high static power, refresh overhead, destructive reads	Off-chip only, low bandwidth, limited endurance, expensive erases	Short retention, refresh overhead	High write energy, high write latency, limited endurance
Uses	Fast R/W caches	Large, random access, read/write data	Storage, rarely accessed data	Fast R/W caches, Write-and-read scratchpads	Read-mostly data, read-mostly caches

GainSight: Lifetime Profiler of On-Chip Data



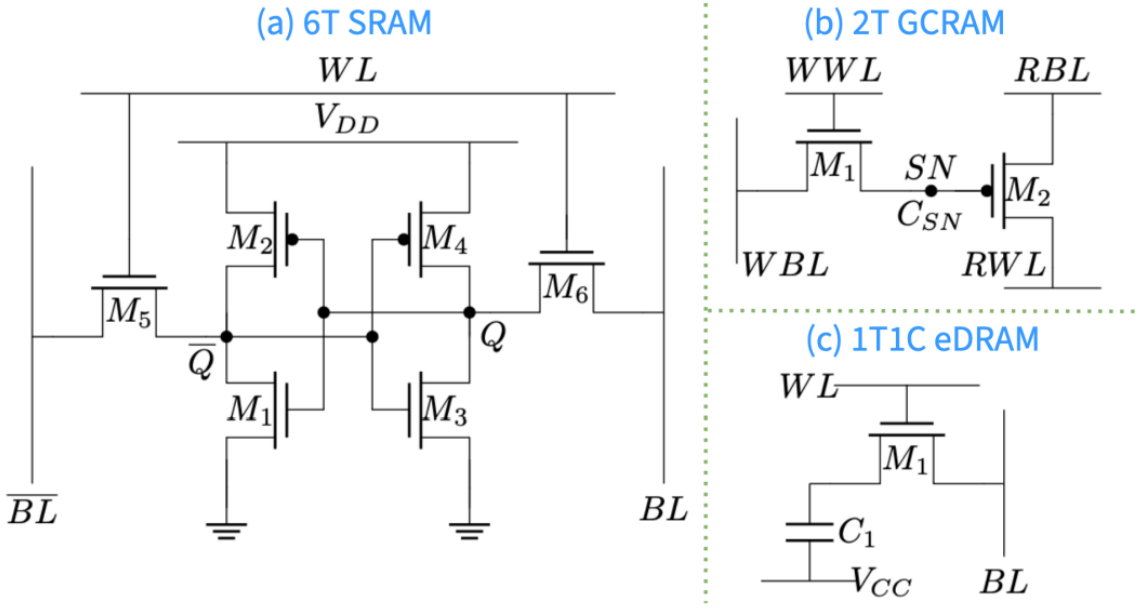
Retargetable Hardware Backend

- Measure fine-grained on-chip memory access patterns
- Variety of processing elements

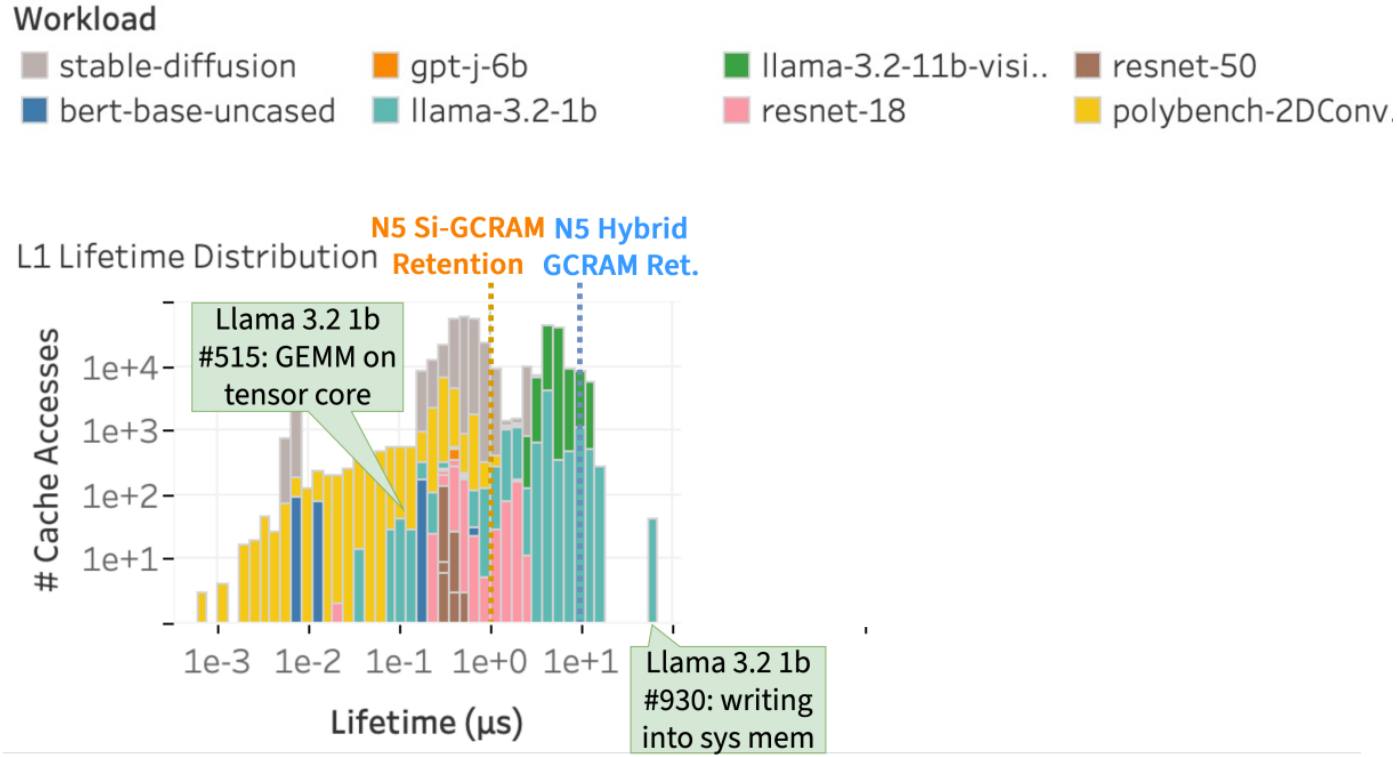
Flexible Analytical Frontend

- Calculate lifetimes & other memory statistics
- Utilize domain knowledge to project area/power/refresh of gain cell devices
- Align data lifetime to a generated composition of heterogeneous memories

StRAM devices

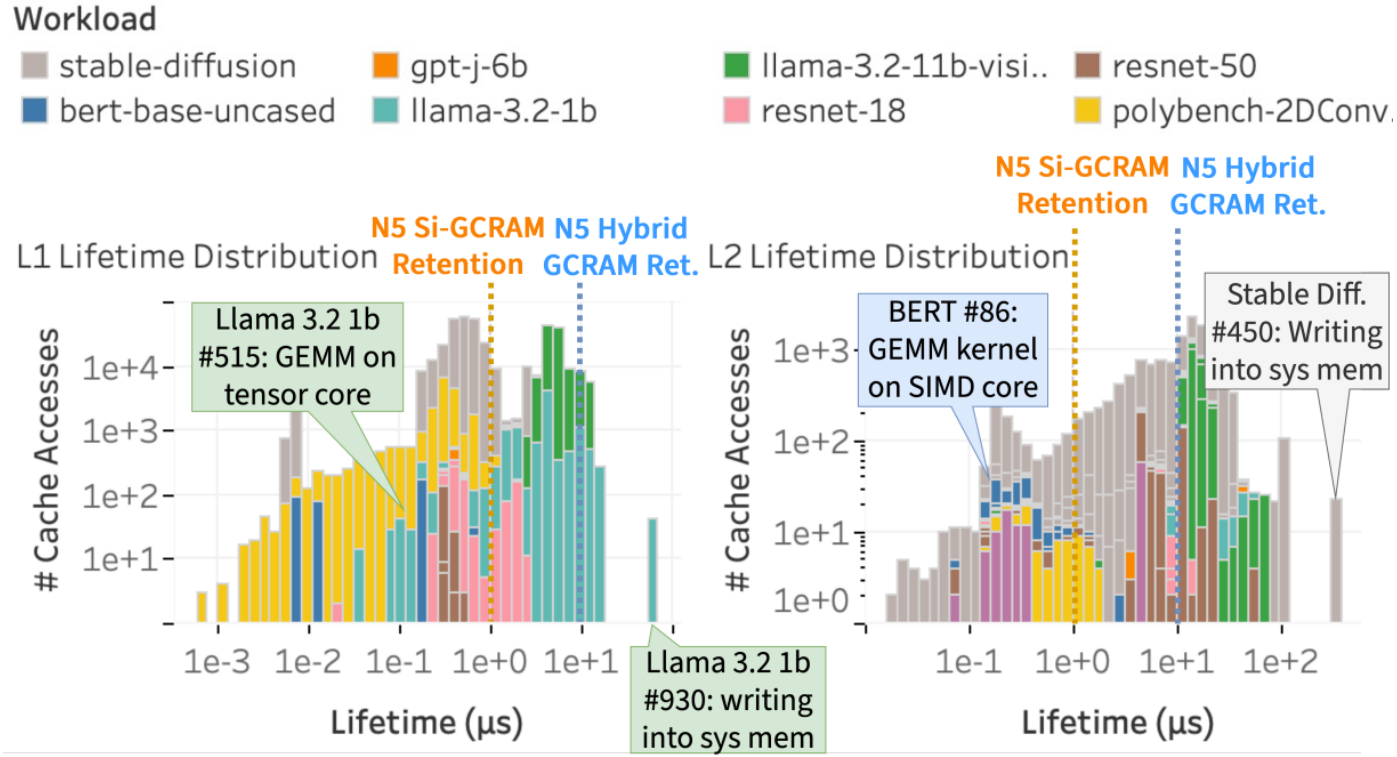


Case Study 1: H100-like GPU model Profiling



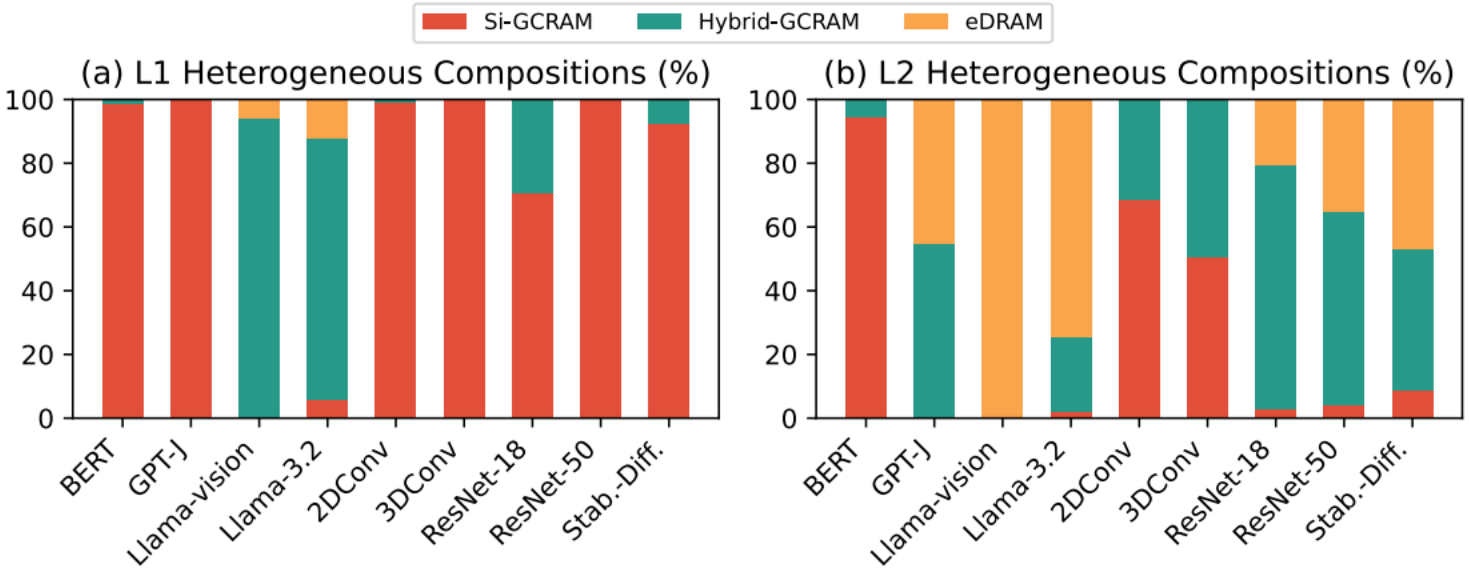
- Up to 98% of L1 accesses are below N5 Hybrid GCRAM retention time
- Most of GEMM kernels can be stored on N5 Si-GCRAM

Case Study 1: H100-like GPU model Profiling

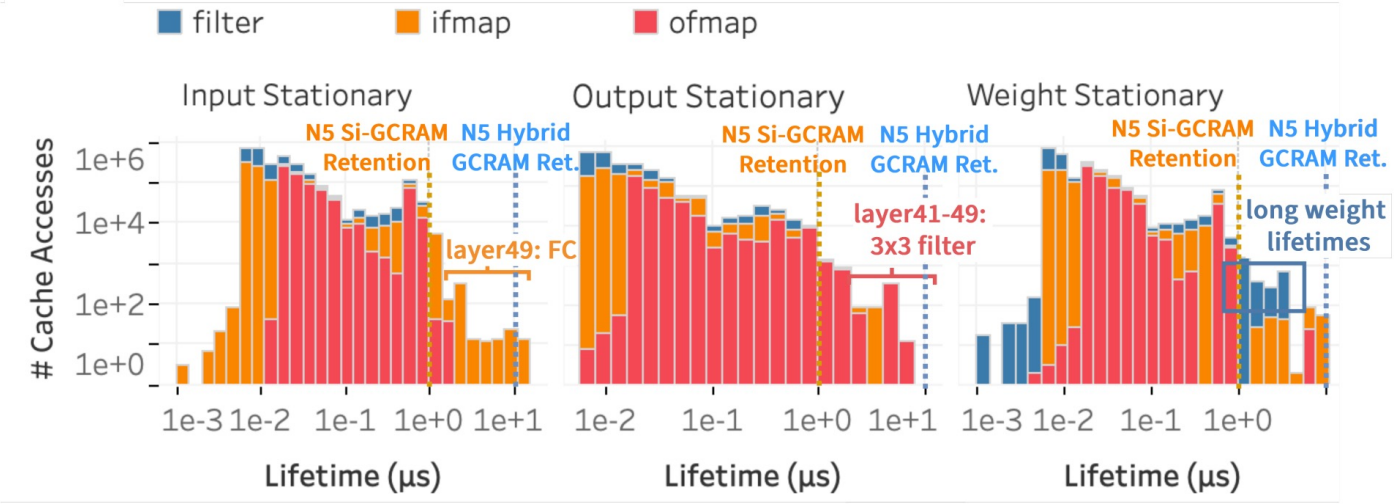


- Up to 52% of L2 access are below N5 Hybrid GCRAM retention time

Case Study 1: H100-like GPU model Profiling

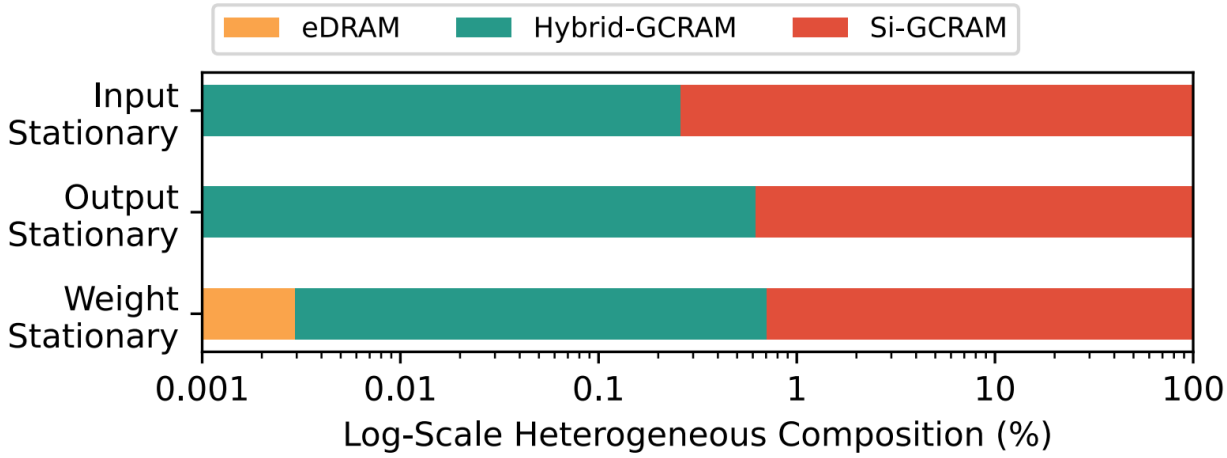


Case Study 2: 256 x 256 Systolic Array Results



- Output features exhibit lifetime amenable for Si-GCRAM storage regardless of dataflow

Case Study 2: 256 x 256 Systolic Array Results



GainSight is open source

https://gainsight.stanford.edu/

GainSite: Documentation and Artifacts for the GainSight Profiler Framework

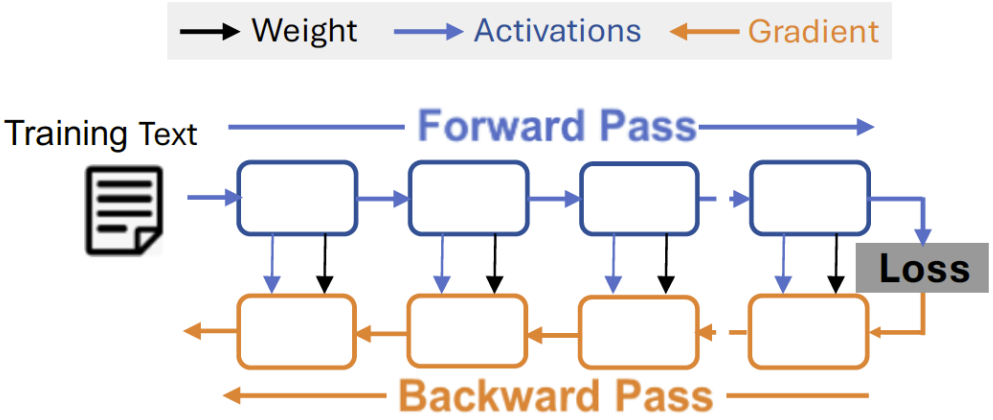
Essential Links

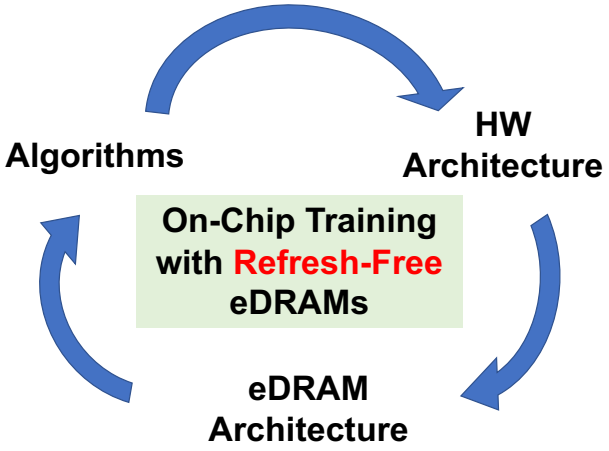
- **Source Code:** You can find the project's source code on the Stanford GitLab instance: https://code.stanford.edu/tambe-lab/gainsight
- **Preprint Paper:** Read the preprint of our research paper on arXiv: https://arxiv.org/abs/2504.14866
- **Website:** Source code for this website is available on the Stanford GitLab instance: https://code.stanford.edu/tambe-lab/gainsite

Latest Release v1.0.0-rc.

Leveraging StRAM for short-lived data storage

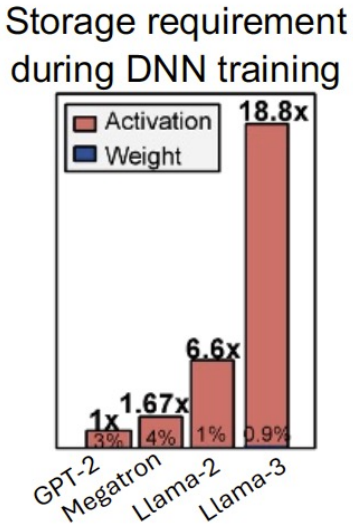
Algorithm-System Co-Design for eDRAM-based AI/ML Training

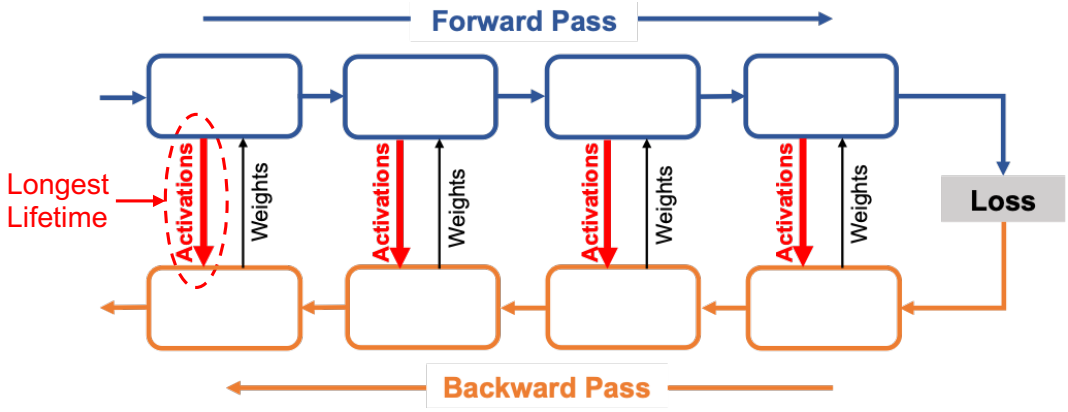




Zhang, Sai Qian, Thierry Tambe, et al. "*CAMEL: Co-Designing AI Models and eDRAMs for Efficient On-Device Learning.*" *HPCA, 2024.*

Challenges

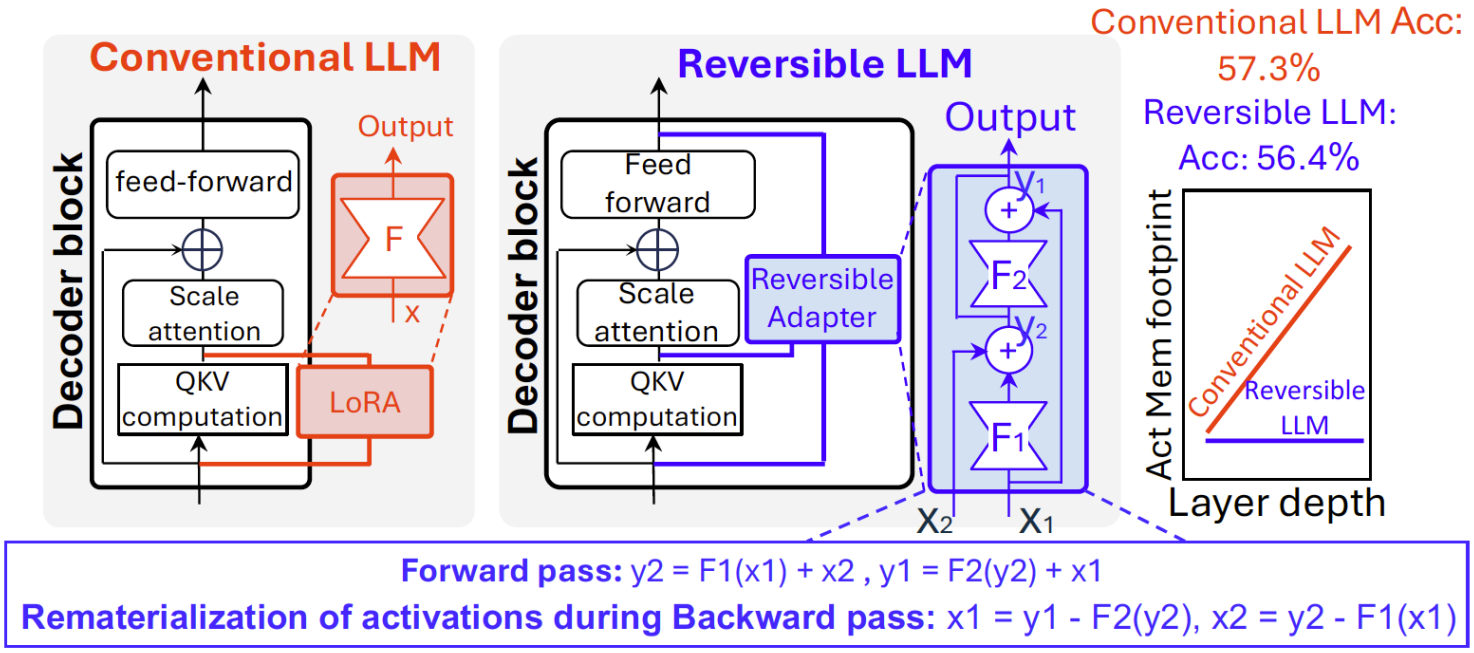




Growing dominance of activation storage as network depth, parameter count, or sequence length increases

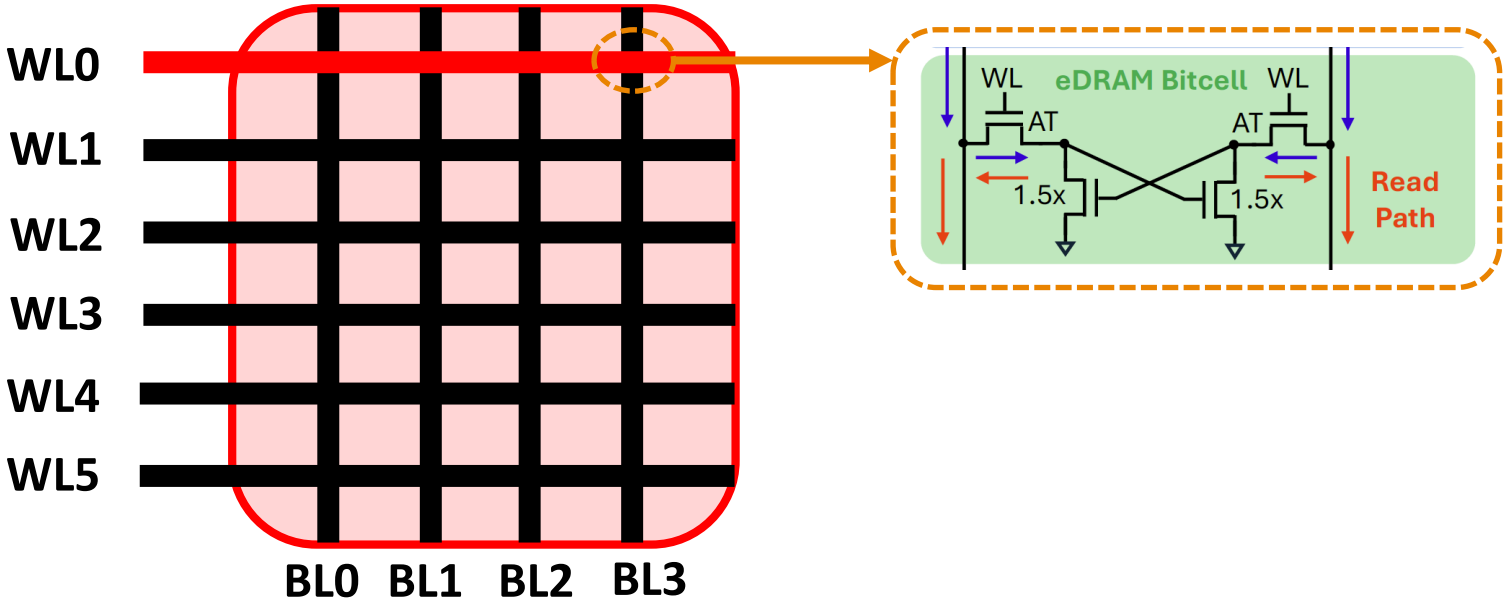
Nestor Cuevas, Thierry Tambe, et al. *"A 64.5 TFLOPS/W 16-nm AI Training Accelerator with 4T-eDRAM for Efficient Gradient and Activation Handling." CICC, 2026.*

Solution #1 Promoting Recomputation instead Memory Loads



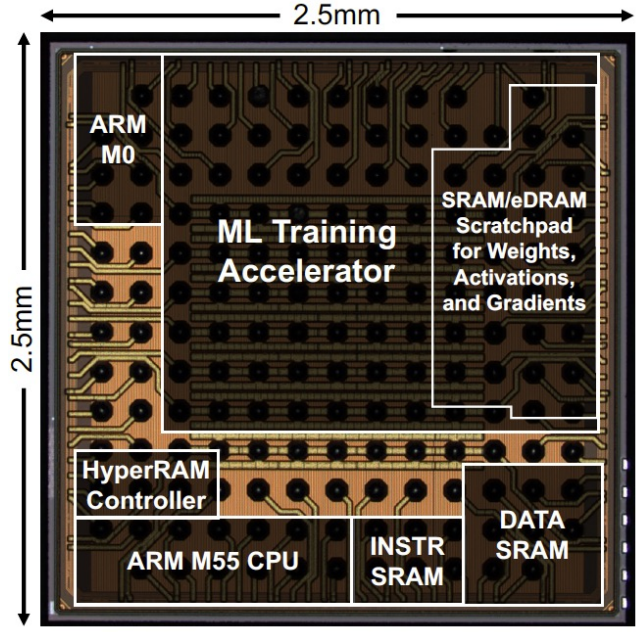
Nestor Cuevas, Thierry Tambe, et al. *"A 64.5 TFLOPS/W 16-nm AI Training Accelerator with 4T-eDRAM for Efficient Gradient and Activation Handling." CICC, 2026.*

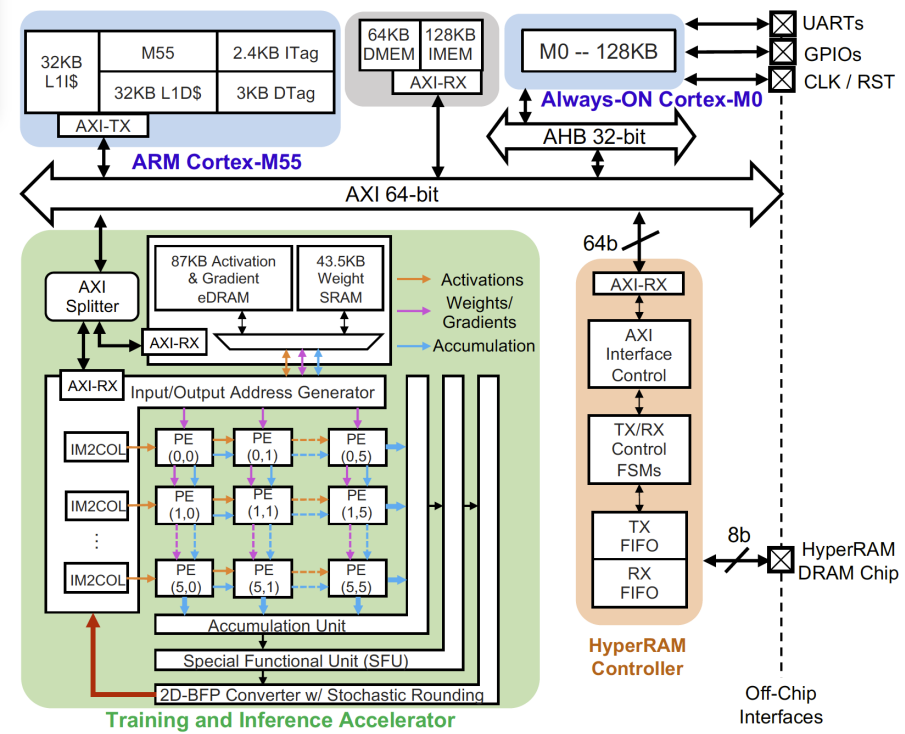
Solution #2: Leveraging Implicit Refresh



Accessing any cell refreshes other cells in the same row, because all the BLs are precharged to '1'

16nm Proof-of-Concept Tapeout



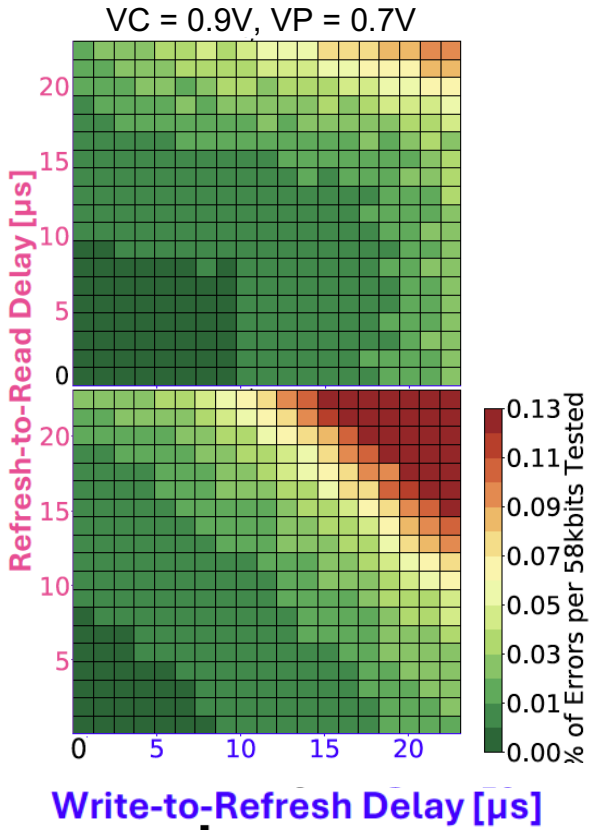


Nestor Cuevas, Thierry Tambe, et al. *"A 64.5 TFLOPS/W 16-nm AI Training Accelerator with 4T-eDRAM for Efficient Gradient and Activation Handling." CICC, 2026.*

Refresh profile

Explicit Refresh
(Read target bitcell)

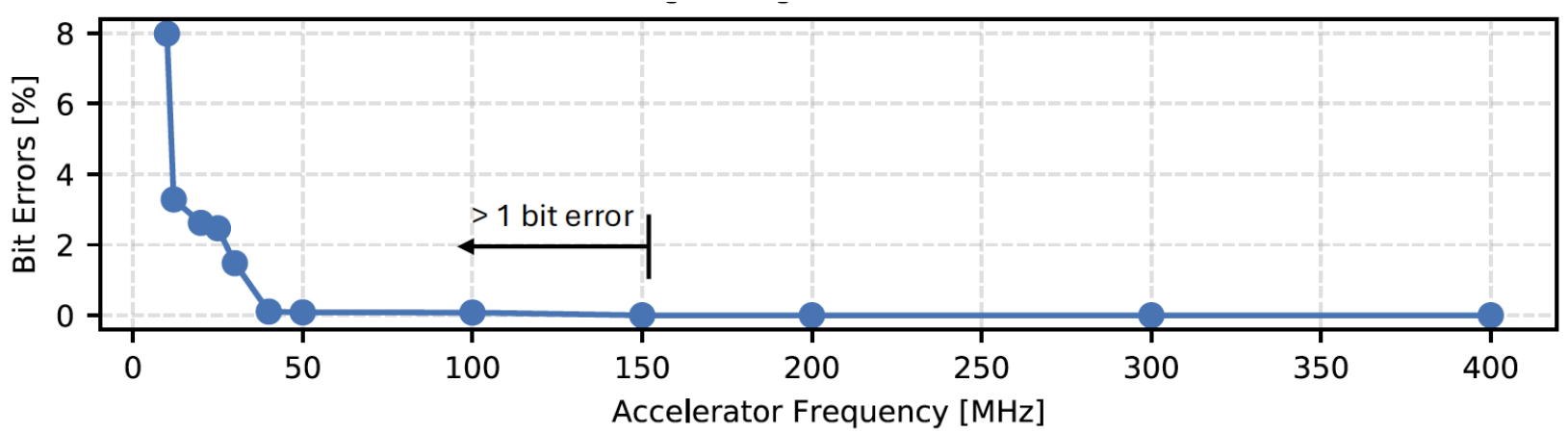
Implicit Refresh
(Read adjacent bitcell)



About 10us of retention time with 0% bit error

Nestor Cuevas, Thierry Tambe, et al. "*A 64.5 TFLOPS/W 16-nm AI Training Accelerator with 4T-eDRAM for Efficient Gradient and Activation Handling.*" *CICC, 2026.*

RevGPT-2 Bit Error vs. Accelerator Frequency after 10 finetuning iterations

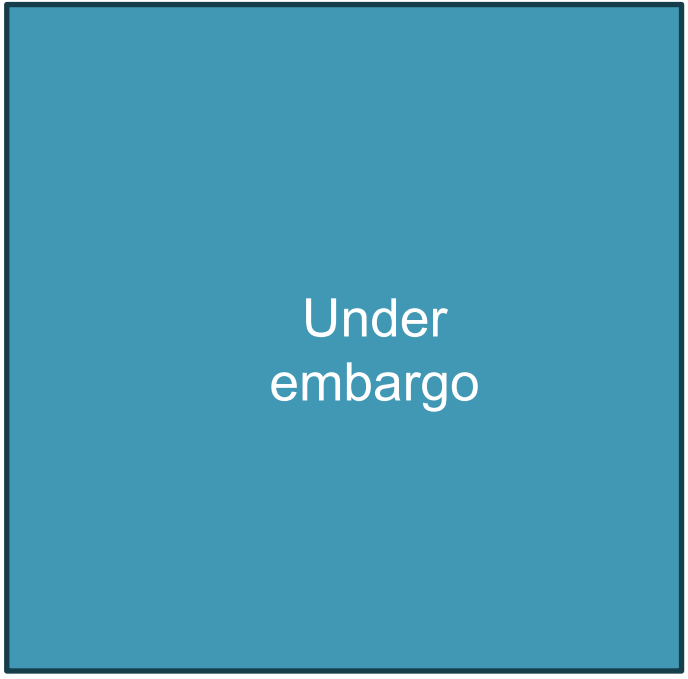


Zero bit flips once accelerator frequency > 150MHz

Nestor Cuevas, Thierry Tambe, et al. *"A 64.5 TFLOPS/W 16-nm AI Training Accelerator with 4T-eDRAM for Efficient Gradient and Activation Handling." CICC, 2026.*

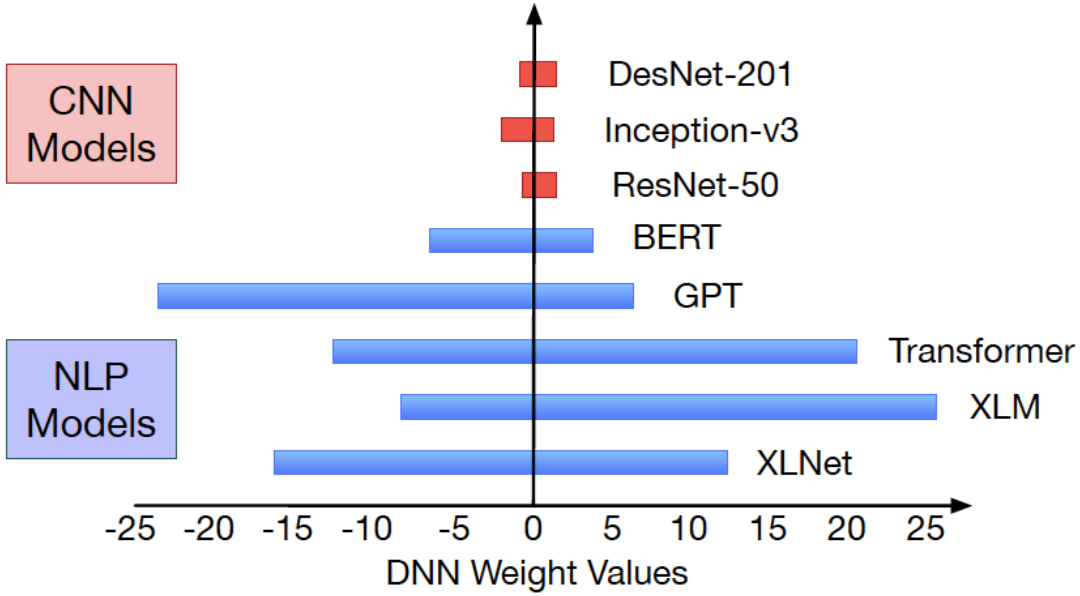
Accelerating Mamba in a 16nm Gain-cell based Chip Prototype

- We recently taped out a 16nm **gain-cell based inference accelerator chip**
  - SIMD-based GEMM/GEMV PEs
  - Gain-cell RAM as primary on-chip, short term memory
- Retention-aware refresh control that skips refresh on data no longer needed for future reads.
- First in a series of SW tools for retention aware compiler and optimizer
- Target model: Mamba state-space language model



Leveraging LtRAM for Efficient Number Format Creation

Large variance in the weight distribution of AI Models



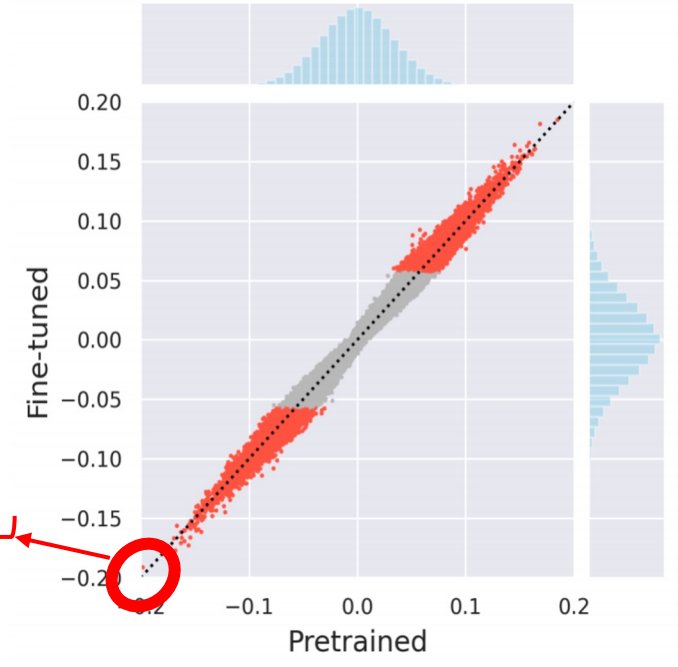
How can we quantize the NLP models to FP4 (E2M1) when their parameter distributions exceed E2M1 dynamic range?

Need for tensor scaling

$$-1^{sign} * mantissa * 2^{exponent}$$

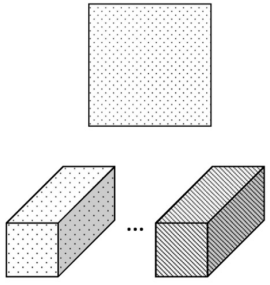
$$-1^{sign} * mantissa * 2^{exponent+bias}$$

Shift exponent range to accommodate largest magnitude



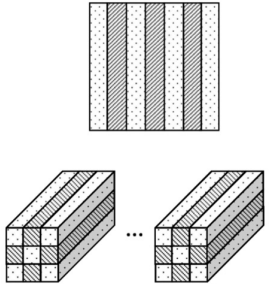
Towards finer-grained tensor scaling

- Per-tensor scaling



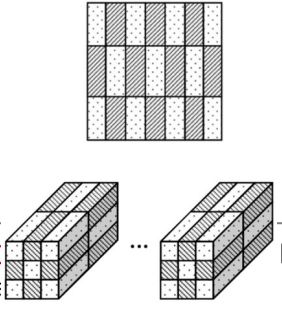
  - Scale factor is computed for the entire weight and activation tensor/matrix
  - Lowest overhead of hardware accelerator design
  - Model accuracy deteriorates at 8-bit, especially when dynamic range is different across channels

- Per-channel scaling



  - Scale factor is computed for each activation channel and weight kernel
  - Metadata overhead increases
  - Lossless model accuracy at 8-bit for small and large models

- Per-block scaling (e.g., VS-quant, microscaling (MX) quantization)

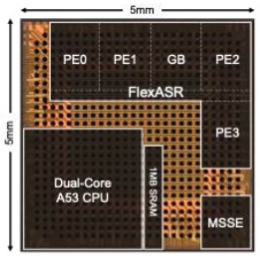


  - Scale factor is computed for each activation and weight vector or micro-tensor
  - Necessary for 4-bit precision and below
  - Popular for quantizing LLM weights

Image Credit: B. Rouhani et al. “Pushing the Limits of Narrow Precision Inferencing at Cloud Scale with Microsoft Floating Point”, NeurIPS 2020

Towards finer-grained tensor scaling

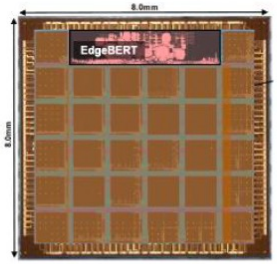
Adaptive FP8 w/ Per-Tensor Scaling



LSTM/RNN ASR Inference (2020)

7.8 TOPS/W

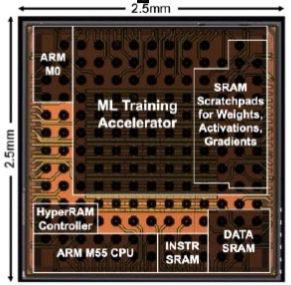
Adaptive FP4/FP8 w/ Per-Vector Scaling



Transformer Inference (2022)

18.1 TOPS/W

Block FP with Block Scaling

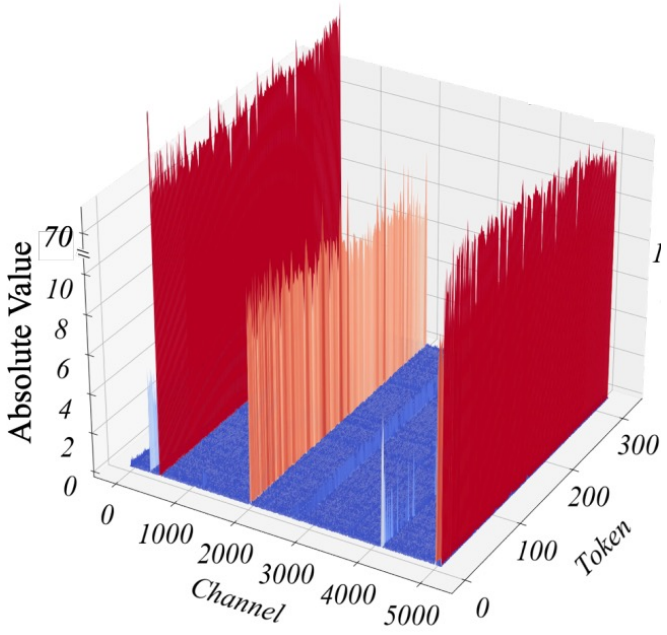


On-Device Training (2024)

64.5 TOPS/W

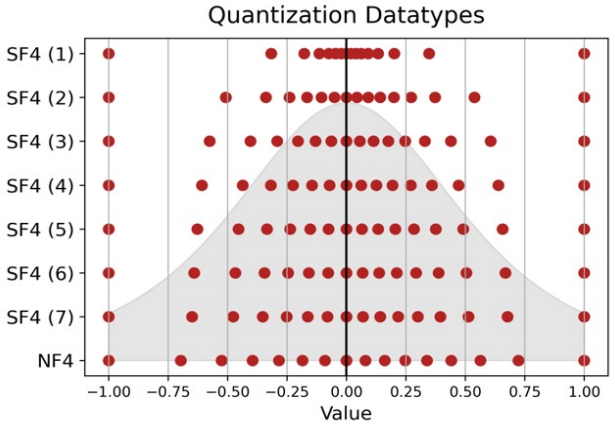
➢ **Higher performance obtained with number formats w/ finer representation granularity**

Generative AI models exhibit significant outliers especially in the activations!



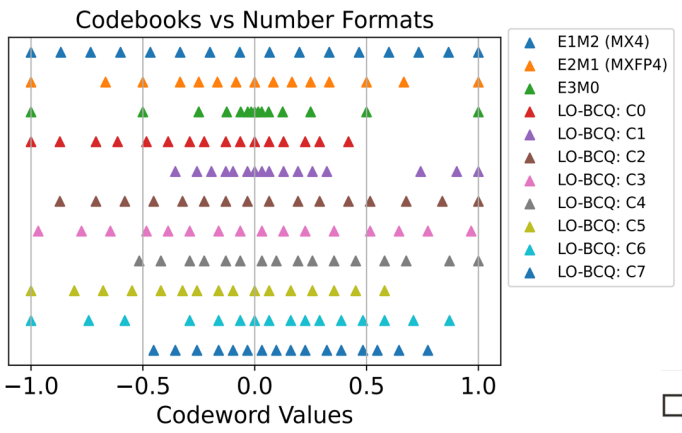
Trend towards custom codebooks / formatbooks

[Dettmers et al., NIPS'22], [Dotzel et al., ICML'24]



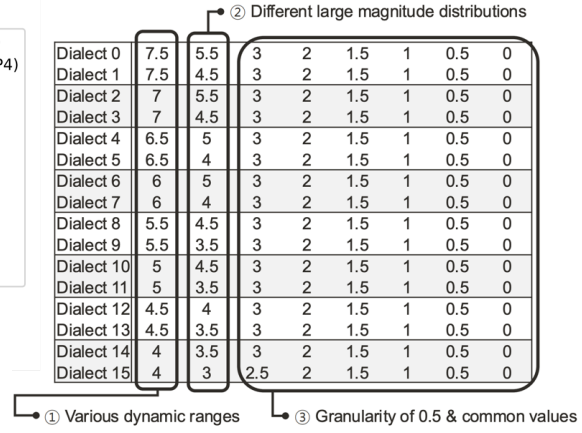
- 4-bit Normal-Float (NF4) and Student-Float (SF4) leverage the statistical quantile function to define 16 quantization values, which fit Normal Distribution and Student's t-Distribution.

[Elangovan et al., TMLR'25]



- LO-BCQ: Locally Optimal Block Clustered Quantization.
- Blocks are clustered based on their statistics, and a dedicated quantization codebook is optimized for each cluster.

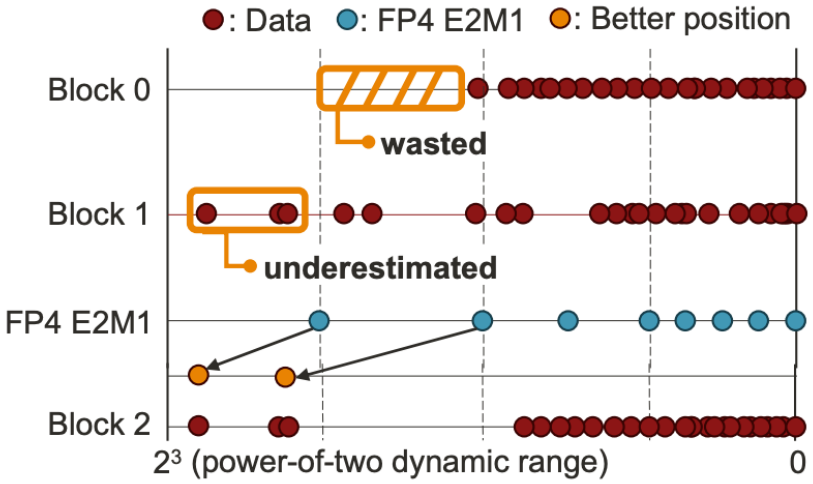
[Jang et al., ICML'25]



Dialect 0	7.5	5.5	3	2	1.5	1	0.5	0
Dialect 1	7.5	4.5	3	2	1.5	1	0.5	0
Dialect 2	7	5.5	3	2	1.5	1	0.5	0
Dialect 3	7	4.5	3	2	1.5	1	0.5	0
Dialect 4	6.5	5	3	2	1.5	1	0.5	0
Dialect 5	6.5	4	3	2	1.5	1	0.5	0
Dialect 6	6	5	3	2	1.5	1	0.5	0
Dialect 7	6	4	3	2	1.5	1	0.5	0
Dialect 8	5.5	4.5	3	2	1.5	1	0.5	0
Dialect 9	5.5	3.5	3	2	1.5	1	0.5	0
Dialect 10	5	4.5	3	2	1.5	1	0.5	0
Dialect 11	5	3.5	3	2	1.5	1	0.5	0
Dialect 12	4.5	4	3	2	1.5	1	0.5	0
Dialect 13	4.5	3.5	3	2	1.5	1	0.5	0
Dialect 14	4	3.5	3	2	1.5	1	0.5	0
Dialect 15	4	3	2.5	2	1.5	1	0.5	0

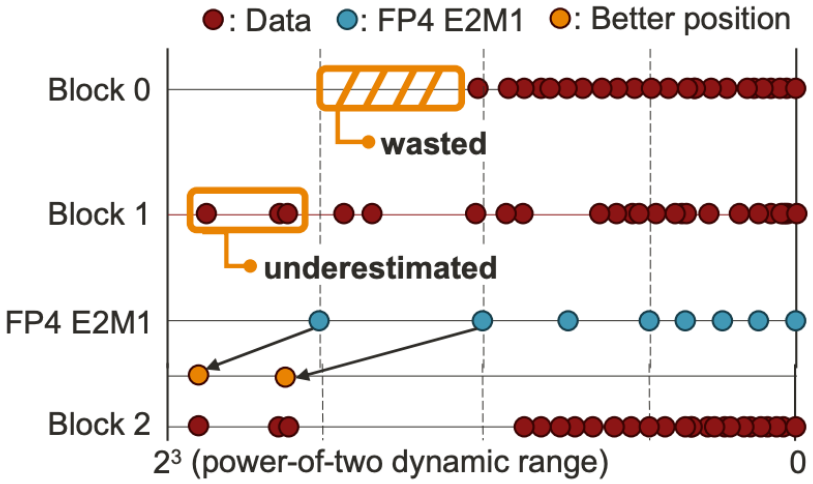
- BlockDialect: Block-wise Fine-grained Mixed Format Quantization.
- The unit of all dialects is 0.5 to align with the FP4 format.

Rationale for custom formatbook



Wonsuk Jang, Thierry Tambe. "*BlockDialect: Block-wise Fine-grained Mixed Format Quantization for Energy-Efficient LLM Inference.*" *ICML, 2025.*

FormatBook with 16 entries



② Different large magnitude distributions

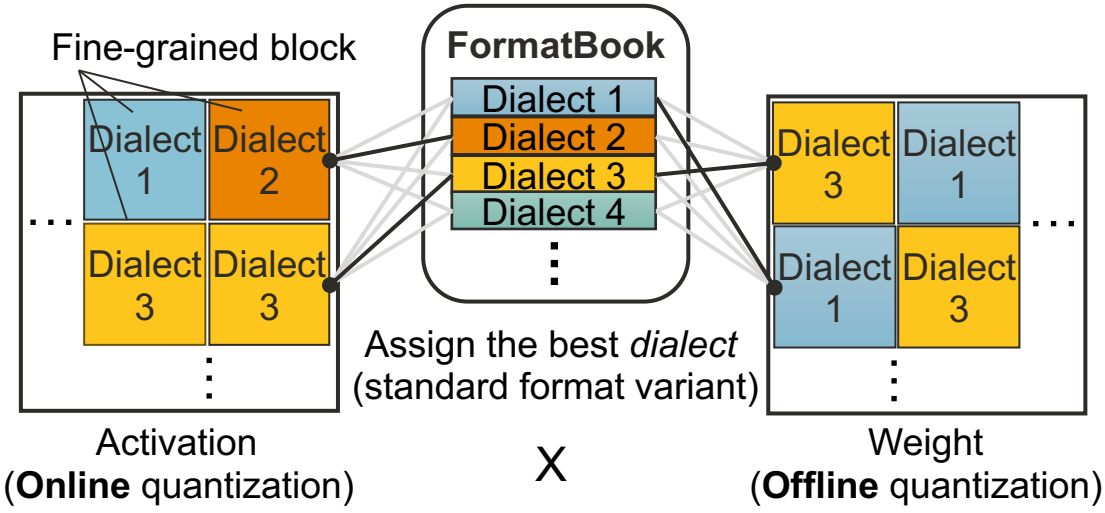
Dialect 0	7.5	5.5	3	2	1.5	1	0.5	0
Dialect 1	7.5	4.5	3	2	1.5	1	0.5	0
Dialect 2	7	5.5	3	2	1.5	1	0.5	0
Dialect 3	7	4.5	3	2	1.5	1	0.5	0
Dialect 4	6.5	5	3	2	1.5	1	0.5	0
Dialect 5	6.5	4	3	2	1.5	1	0.5	0
Dialect 6	6	5	3	2	1.5	1	0.5	0
Dialect 7	6	4	3	2	1.5	1	0.5	0
Dialect 8	5.5	4.5	3	2	1.5	1	0.5	0
Dialect 9	5.5	3.5	3	2	1.5	1	0.5	0
Dialect 10	5	4.5	3	2	1.5	1	0.5	0
Dialect 11	5	3.5	3	2	1.5	1	0.5	0
Dialect 12	4.5	4	3	2	1.5	1	0.5	0
Dialect 13	4.5	3.5	3	2	1.5	1	0.5	0
Dialect 14	4	3.5	3	2	1.5	1	0.5	0
Dialect 15	4	3	2.5	2	1.5	1	0.5	0

① Various dynamic ranges

③ Granularity of 0.5 & common values

Wonsuk Jang, Thierry Tambe. "*BlockDialect: Block-wise Fine-grained Mixed Format Quantization for Energy-Efficient LLM Inference.*" *ICML, 2025.*

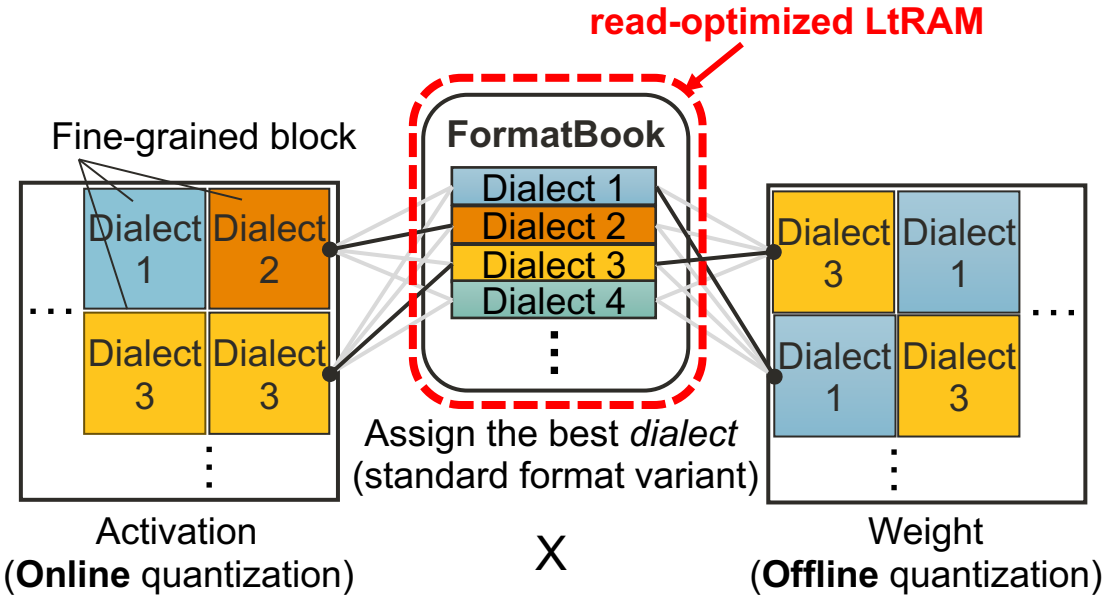
Block-Level Mixed-Format Representation (BlockDialect)



- Choose number format **candidates** - FP4 variants
- Split each matrix into blocks & assign the optimal **format per-block**
- Ensure **hardware-friendliness** to operation between candidates

Wonsuk Jang, Thierry Tambe. "*BlockDialect: Block-wise Fine-grained Mixed Format Quantization for Energy-Efficient LLM Inference.*" *ICML, 2025.*

Block-Level Mixed-Format Representation (BlockDialect)



- Choose number format **candidates** - FP4 variants
- Split each matrix into blocks & assign the optimal **format per-block**
- Ensure **hardware-friendliness** to operation between candidates

Wonsuk Jang, Thierry Tambe. "*BlockDialect: Block-wise Fine-grained Mixed Format Quantization for Energy-Efficient LLM Inference.*" *ICML, 2025.*

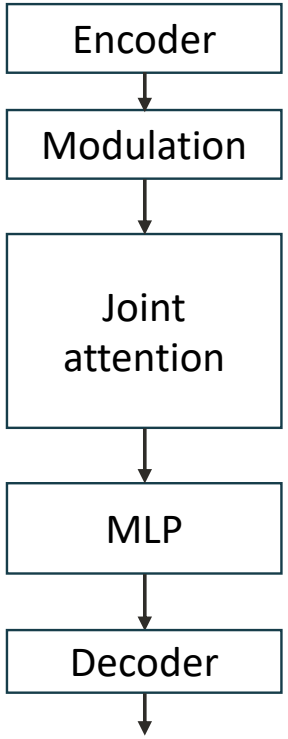
State-of-the-art encoding performance

Scope	Method	Block size (exception)	LLaMA3-8B			LLaMA2-7B			Mistral-7B		
			Eff. bit	Wiki↓	AVG.↑	Eff. bit	Wiki↓	AVG.↑	Eff. bit	Wiki↓	AVG.↑
-	FP16	-	16	6.14	74.45	16	5.47	70.94	16	5.32	74.92
Linear ($A \cdot W$)	MXFP4	16	4.31	8.20	68.53	4.31	7.07	66.86	4.31	6.49	70.33
		32	4.16	8.23	68.31	4.16	7.04	65.94	4.16	6.42	70.72
	BlockDialect (w/ DialectFP4)	32	4.28	**7.05**	72.24	4.28	**5.84**	**69.74**	4.28	**5.65**	**73.46**

Wonsuk Jang, Thierry Tambe. "*BlockDialect: Block-wise Fine-grained Mixed Format Quantization for Energy-Efficient LLM Inference.*" *ICML, 2025.*

Video generation is extremely compute- and data-intensive

- Massive output dimensionality
  - Must generate 4D tensors (H*W*channels*frames) which contain orders of magnitude more output elements than text tokens
- Quadratic spatiotemporal attention
  - 3D attention scales $O(n^2)$ where n is space-time patch
  - No KV cache reuse
- Large activation memory footprint
  - Intermediate multi-frame feature maps dominates memory
  - Highly likely to exceed edge SRAM capacity, forcing costly DRAM accesses
- Iterative denoising overhead
  - Diffusion model is run repeatedly (20 – 100 passes) to refine noise into a video clip



Open-Sora 2.0 (FLUX)

16-entry formatbook LUT is not enough for video generation!

as there is more variance in video than text…

② Different large magnitude distributions

Dialect 0	7.5	5.5	3	2	1.5	1	0.5	0
Dialect 1	7.5	4.5	3	2	1.5	1	0.5	0
Dialect 2	7	5.5	3	2	1.5	1	0.5	0
Dialect 3	7	4.5	3	2	1.5	1	0.5	0
Dialect 4	6.5	5	3	2	1.5	1	0.5	0
Dialect 5	6.5	4	3	2	1.5	1	0.5	0
Dialect 6	6	5	3	2	1.5	1	0.5	0
Dialect 7	6	4	3	2	1.5	1	0.5	0
Dialect 8	5.5	4.5	3	2	1.5	1	0.5	0
Dialect 9	5.5	3.5	3	2	1.5	1	0.5	0
Dialect 10	5	4.5	3	2	1.5	1	0.5	0
Dialect 11	5	3.5	3	2	1.5	1	0.5	0
Dialect 12	4.5	4	3	2	1.5	1	0.5	0
Dialect 13	4.5	3.5	3	2	1.5	1	0.5	0
Dialect 14	4	3.5	3	2	1.5	1	0.5	0
Dialect 15	4	3	2.5	2	1.5	1	0.5	0

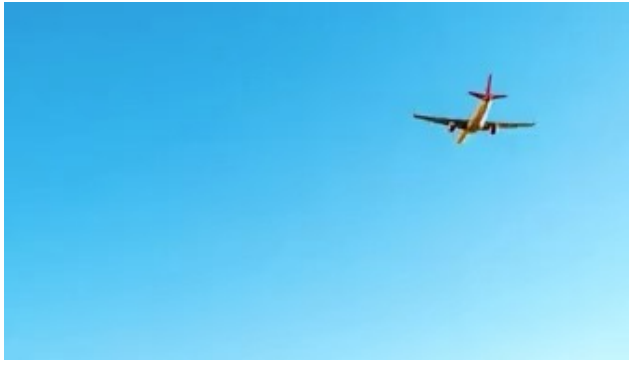
① Various dynamic ranges

③ Granularity of 0.5 & common values

An airplane soaring through a clear blue sky

FP16

BlockDialect 4.28 eff. bits



Proposed Formatbook Number System for Video Generation

- We came up w/ a 37-entry LUT formatbook
  - Cover all possible dynamic ranges: 8-15
  - Assign fewer candidates to narrower ranges
  - Focus on the high-value distribution
    - The second-largest value covers many possible distributions
  - The small-value region is more fine-grained, as more data is concentrated there

Cand #	Representable values								Cand #	Representable values							
0	8	6	5	4	3	2	1	0	22	14	7	5	4	3	2	1	0
1	8	7	5	4	3	2	1	0	23	14	8	6	4	3	2	1	0
2	9	7	5	4	3	2	1	0	24	14	9	7	5	3	2	1	0
3	9	8	7	5	3	2	1	0	25	14	10	8	6	4	2	1	0
4	10	7	5	4	3	2	1	0	26	14	11	9	7	5	3	1	0
5	10	8	6	4	3	2	1	0	27	14	12	10	8	6	4	2	0
6	10	9	7	5	3	2	1	0	28	14	13	11	8	6	4	2	0
7	11	7	5	4	3	2	1	0	29	15	7	5	4	3	2	1	0
8	11	8	6	4	3	2	1	0	30	15	8	6	4	3	2	1	0
9	11	9	7	5	3	2	1	0	31	15	9	7	5	3	2	1	0
10	11	10	8	6	4	2	1	0	32	15	10	8	6	4	2	1	0
11	12	7	5	4	3	2	1	0	33	15	11	9	7	5	3	1	0
12	12	8	6	4	3	2	1	0	34	15	12	10	8	6	4	2	0
13	12	9	7	5	3	2	1	0	35	15	13	10	8	6	4	2	0
14	12	10	8	6	4	2	1	0	36	15	14	10	8	6	4	2	0
15	12	11	9	7	5	3	1	0									
16	13	7	5	4	3	2	1	0									
17	13	8	6	4	3	2	1	0									
18	13	9	7	5	3	2	1	0									
19	13	10	8	6	4	2	1	0									
20	13	11	9	7	5	3	1	0									
21	13	12	10	8	6	4	2	0									

Video Samples

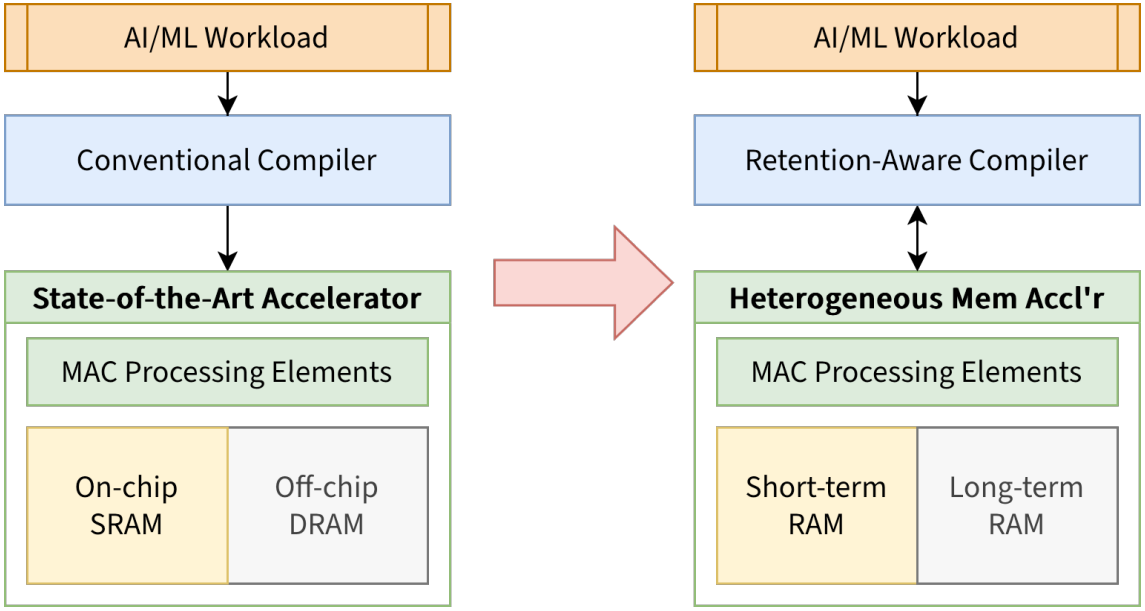
Input Prompt:	An airplane soaring through a clear blue sky	A person swimming in the ocean	Two pandas discussing a book
FP16			
NVFP4 ~4x lower memory footprint			
Ours w/ 37-entry LUT in LtRAM ~4x lower memory footprint			

Takeaway

- A read-optimized LtRAM storing custom number format book entries is promising for continuing to push the frontiers of quantized AI arithmetic performance
- High-bandwidth w/ fine-grained access
- Low access latency

Summary

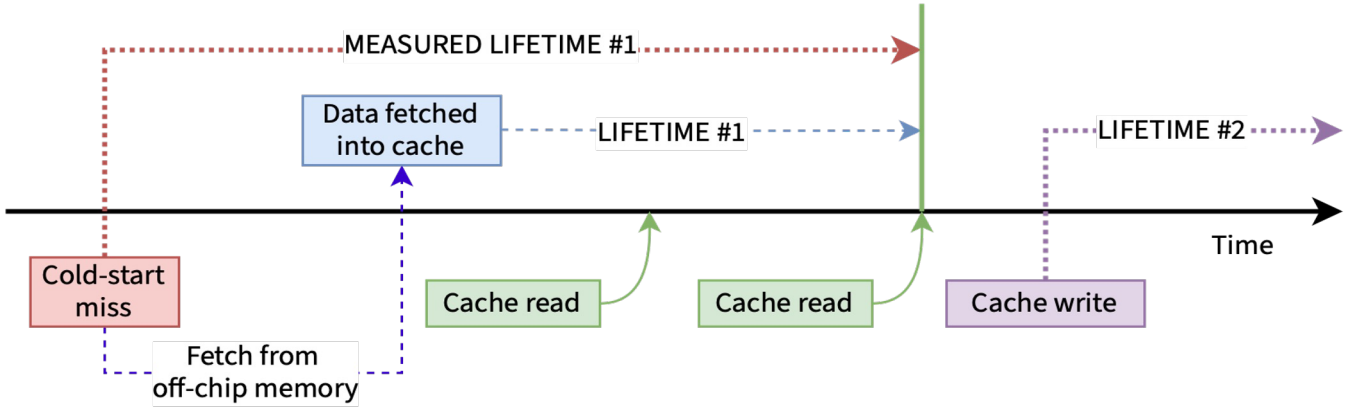
- Memory scaling is ending
- Huge opportunity to tame the memory wall by aligning application memory access patterns and lifetime metrics to differentiated memory systems.



Thanks!

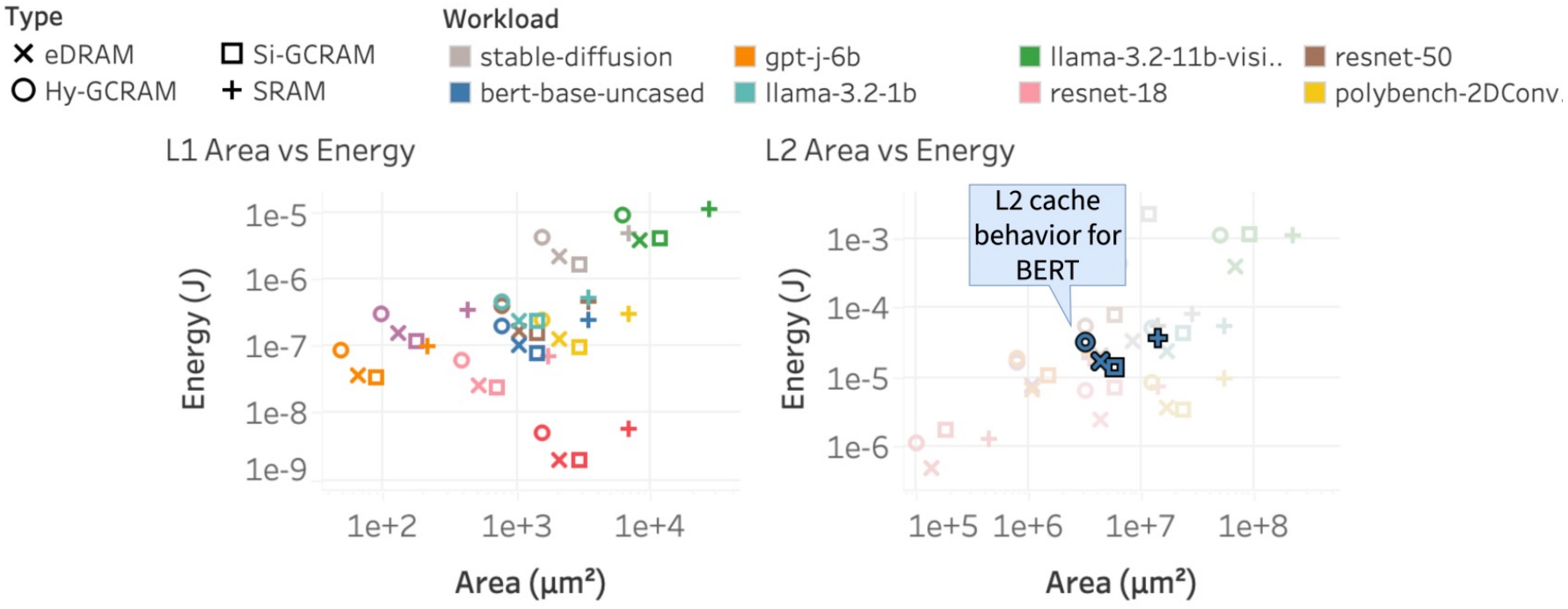
Appendix

Measuring Data Lifetimes



1. The time between a cold-start miss and when the fetched data is actually written into memory is harder to measure
2. The actual values we are measuring:
   a. Start of lifetime: cache miss on read operation, or write operation
   b. End of lifetime: last read before a write or before end of program

Case Study 1: H100-like GPU model Profiling



- Si-GCRAM have better access energy
- Hybrid-GCRAM offers better area efficiency