GainSight: fine-grained profiler for designing heterogeneous on-chip memories for AI accelerators

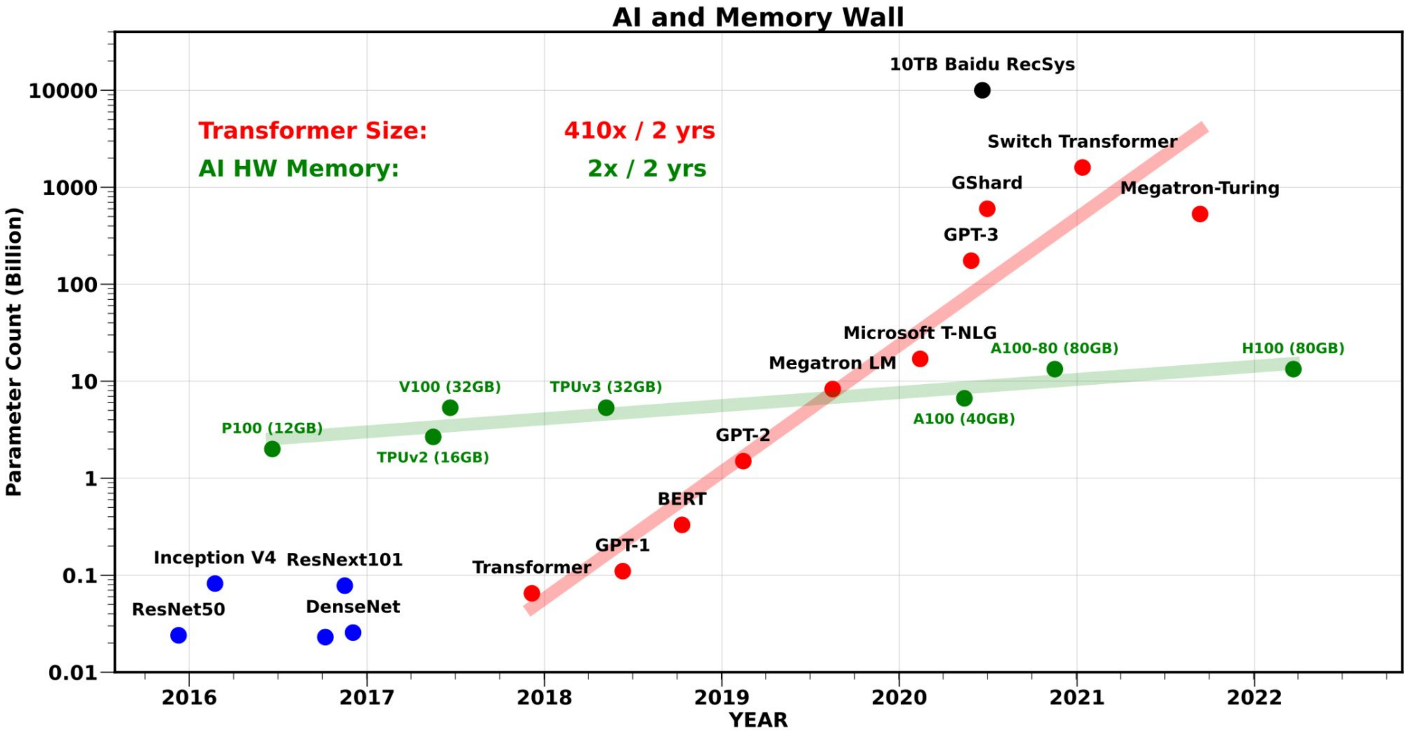
Peijing Li, peli@stanford.edu

Thierry Tambe, ttambe@stanford.edu

January 10, 2025

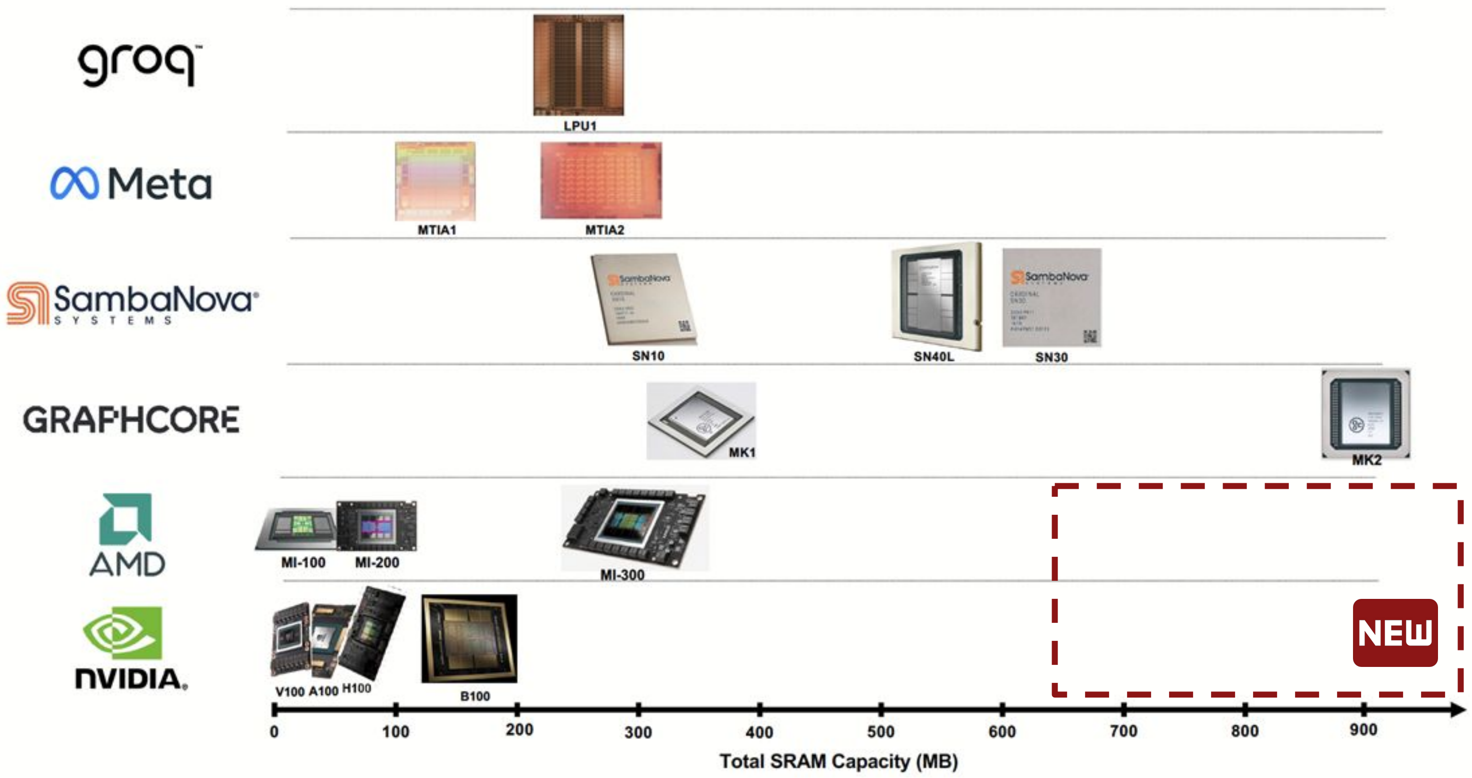
Stanford | ENGINEERING
Electrical Engineering

AI and the Memory Capacity Wall

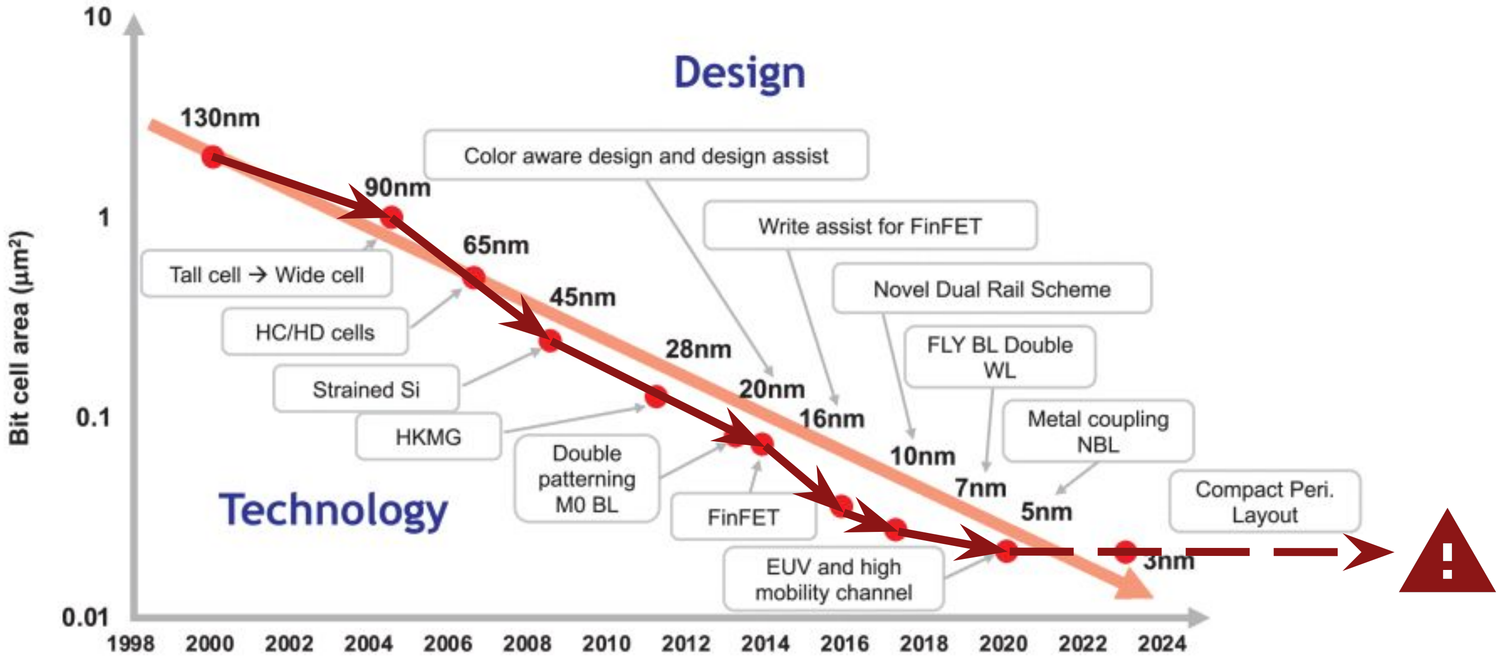


Source: [1] A. Gholami, Z. Yao, S. Kim, C. Hooper, M. W. Mahoney, and K. Keutzer, “AI and Memory Wall,” Mar. 21, 2024, arXiv: arXiv:2403.14123. doi: 10.48550/arXiv.2403.14123.

Trend towards Increasing On-Chip SRAM



SRAM Scaling is Ending



Source: [1] K. Zhang, "1.1 Semiconductor Industry: Present & Future," in 2024 IEEE International Solid-State Circuits Conference (ISSCC), Feb. 2024, pp. 10–15. doi: 10.1109/ISSCC49657.2024.10454358.

At a Glance

1 Motivation for non-SRAM On-Chip Memories and Fine-Grained Data Cache Access Pattern Profiling

2 Methodologies: Profiler with Retargetable Hardware Backend and Flexible Frontend

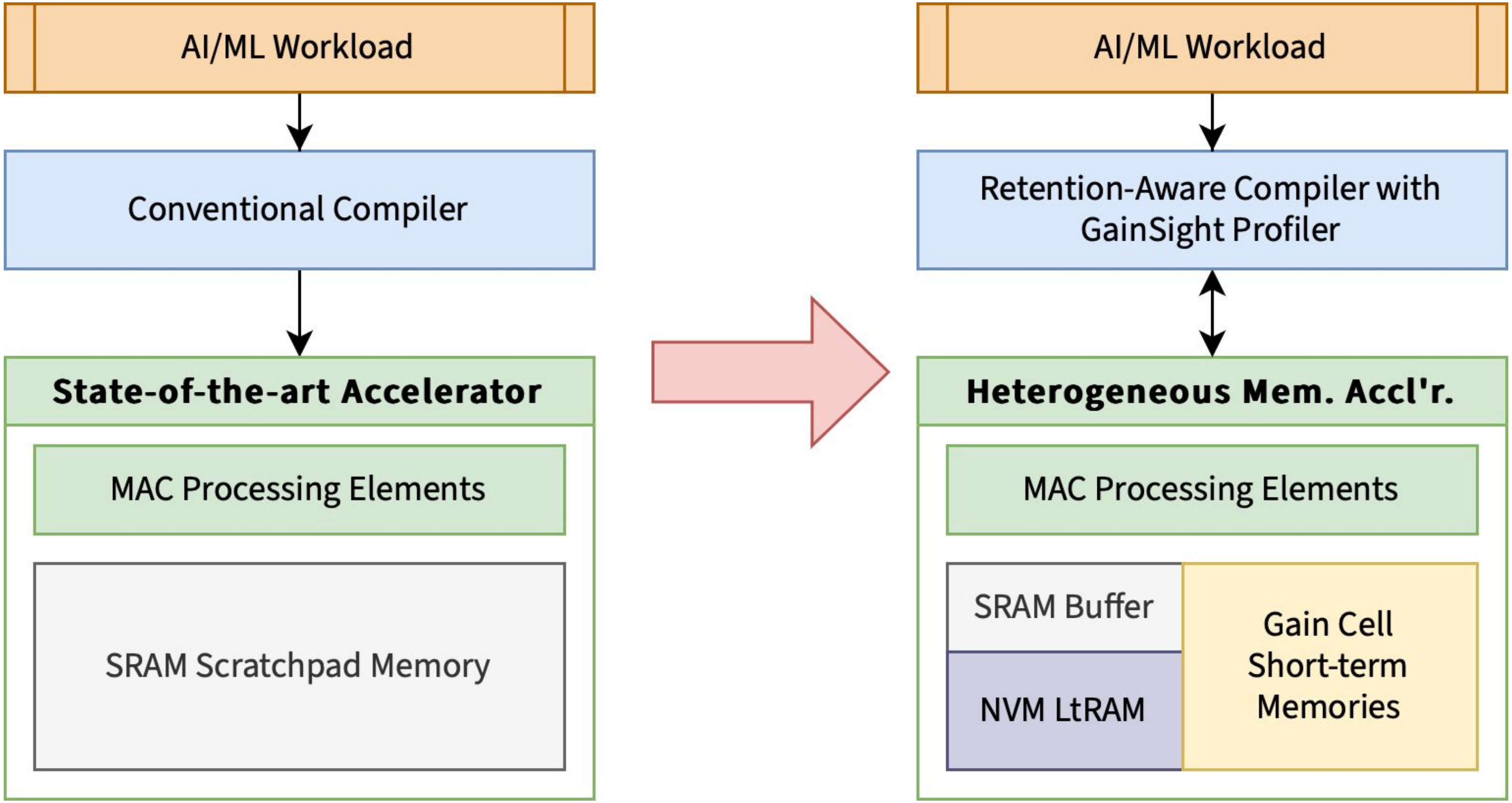
3 Preliminary Experiments and Next Steps

Motivations for Fine-Grained On-Chip Memory Profiling

Alternative On-Chip Memories

	SRAM	DRAM	Block Flash	Long-term RAM	Short-term RAM
Structure	6T	1T1C	1G	FeRAM, MRAM, RRAM, FRAM	2T or 3T gain cells
Benefits	Fast, easy to integrate, low static power	Dense	Huge capacity	Dense, low read energy	Dense, low energy
Drawbacks	Sparse	No logic, high power	No logic, low endurance, expensive & slow erases, block access only, low bandwidth	Expensive & slow writes, limited endurance	Short retention times, expensive refreshes, active research
Uses	Fast R/W caches	Large, random access R/W data	Large, mostly read data	Rare writes, static data caches	Fast write-and-read operations

Our Vision


AI/ML Workload
Conventional Compiler
State-of-the-art Accelerator
MAC Processing Elements
SRAM Scratchpad Memory
AI/ML Workload
Retention-Aware Compiler with GainSight Profiler
Heterogeneous Mem. Accl'r.
MAC Processing Elements
SRAM Buffer
NVM LtRAM
Gain Cell Short-term Memories

Our Vision

- **Breaking the memory wall** with alternative on-chip memory
  - Replace SRAM with devices 3x capacity and 0.3x leakage power
- The tuning of memory components affect performance
- **"Memory profile-guided HW-SW codesign"**
- Build a **profiler** that can measure **lifetimes** and other **fine-grained** memory access **patterns** to guide DSE
- Each DSA for each different memory-bound workload can be optimized and aligned with the ideal heterogeneous memory configuration and tuning

GainSight Organization

Requirements for Profiler Tool

The profiler tool should be able to work for different processing elements for DSE.

- Off-the-shelf GPGPUs
- Systolic arrays
- Dataflow architectures
- … & More!

“Retargetable backend”

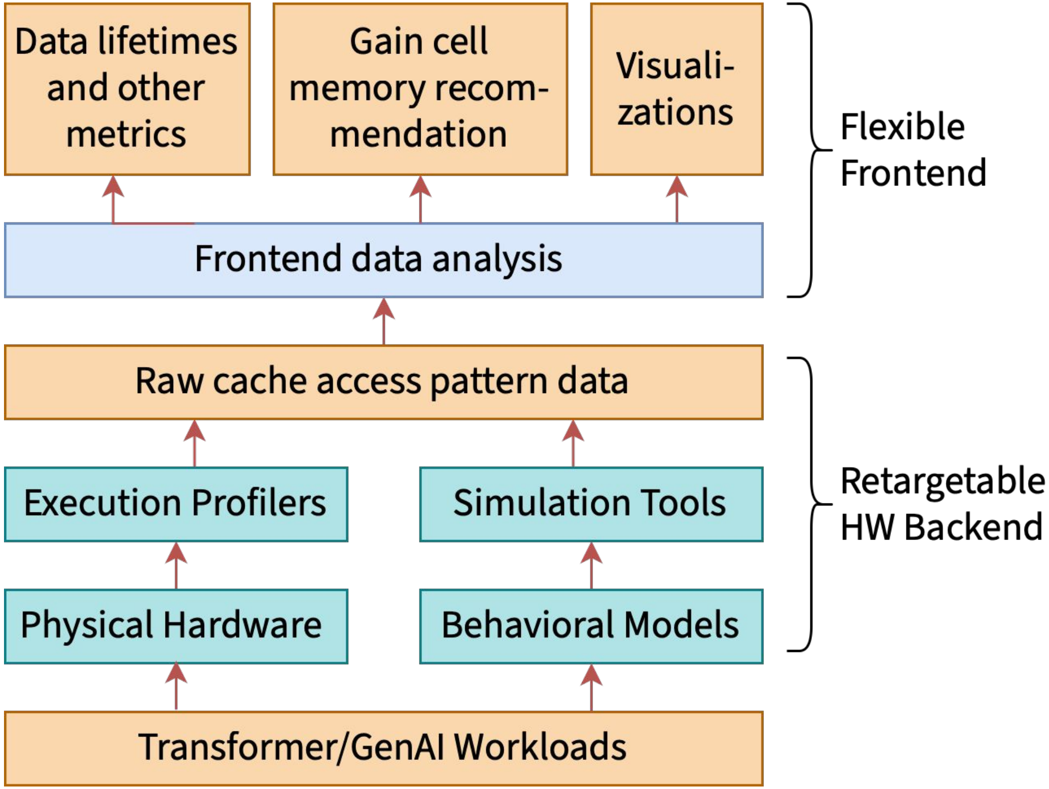
The profiler tool should be able to offer a variety of DSE insights based on domain knowledge.

- Raw lifetime and R/W freq
- What kind of gain cell to use
- Projected # of refreshes
- Comparison between gain cell design and SRAM

“Flexible frontend”

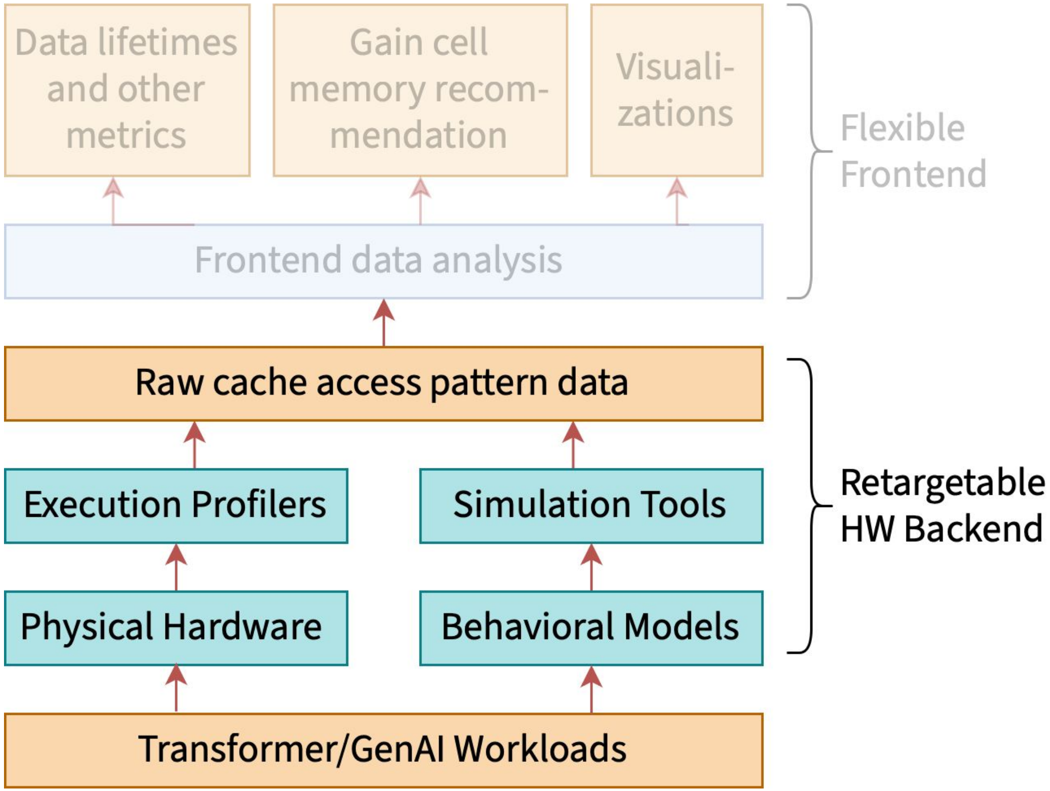
GainSight High-Level Organization

- **Retargetable Backend**
  - Work for different processing elements.
- **Flexible Frontend**
  - Offer a variety of DSE insights based on domain knowledge.



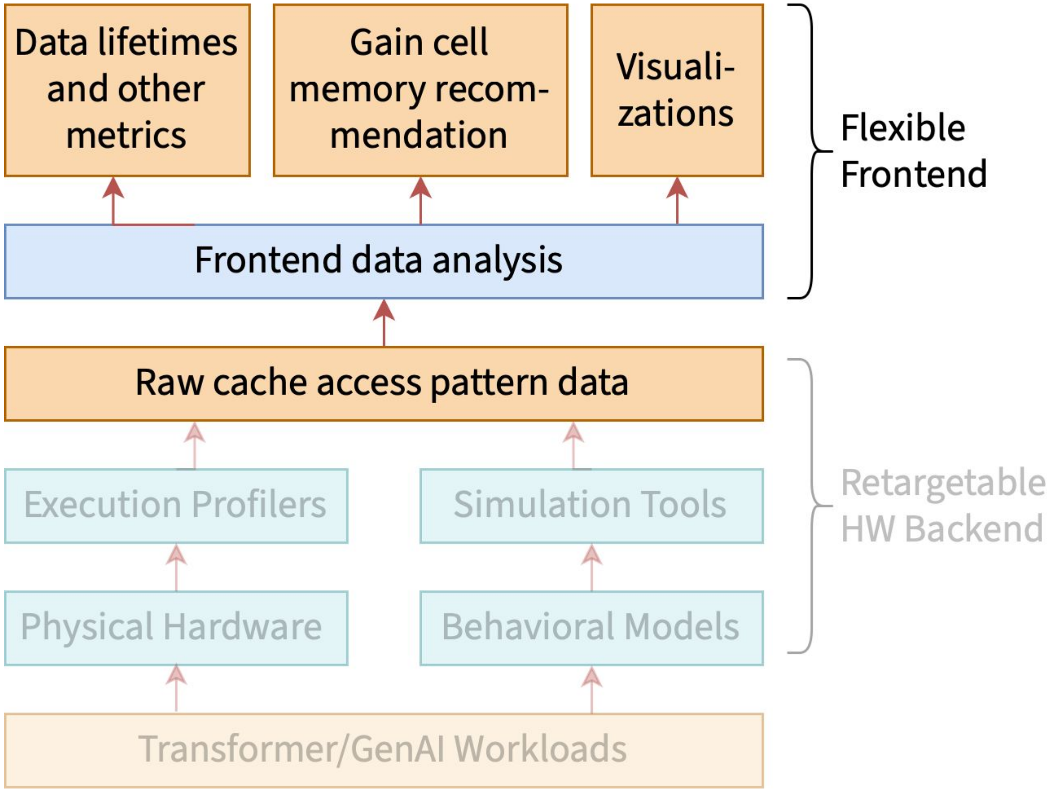
Retargetable Hardware Backends

- **Execution-based profilers** on physical hardware
  - *NVIDIA Hopper GPU*
  - Faster execution times for larger workloads
- **Simulators** for C++/SystemC/RTL models
  - *GPGPU-Sim/Accel-Sim*
  - *NVDLA*
  - *ESP Systolic Array*
  - *Gemmini*
  - More accurate results for smaller workloads



Flexible Frontend

- Numerical results for **data lifetimes and R/W freq.** and visualizations
- Analytical **models of memory arrays**, compare measured results with KPIs from model
- Recommend ideal heterogeneous config and tune
- Report KPI **improvements** over SRAM array



Preliminary Experiments

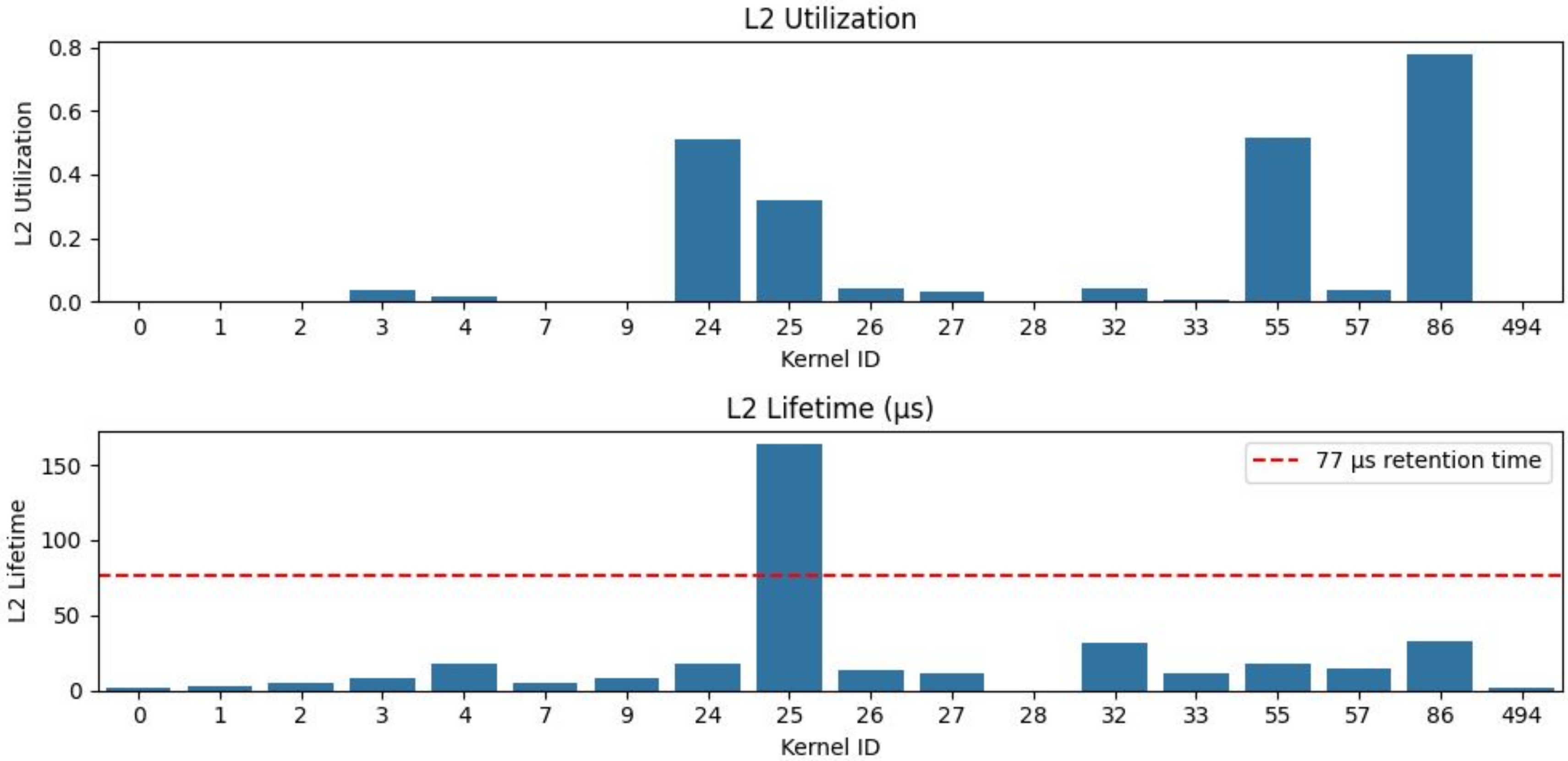
Understanding Workload Behavior

1. Task description
    a. Hypothetically replace cache in NVIDIA GPUs with gain cell RAM and other heterogenous memories
    b. Measure **lifetimes** and estimate number of **refreshes** needed
2. Methodology
    a. *Execution based backend* – run entire transformer workloads
    b. Inspect results and isolate "interesting" outlying kernels
    c. Rerun kernels in *simulation based backend* for more precise results
    d. Design choices on heterogeneous memory configurations
3. Demonstration on how this may be able to work…

Experiment 1: Transformer on NVIDIA GPU

- Workload: Simple **word prediction model** inference
    - Two **transformer** layers with two attention heads each
    - 13 million parameters
- Backend: Physical NVIDIA Hopper H100 GPU
    - Custom profiling **execution** with NVBit and NVIDIA Nsight Compute
    - Approximate per-kernel data lifetimes and R/W frequencies
- Frontend: *per-kernel* visualizations
    - Cache utilization: % of cache lines used by data in the kernel
    - L1/L2 read/write frequencies
    - L1/L2 lifetimes and required refreshes (based on 77 µs retention time from Giterman et al. (2020)'s CMOS gain cells)

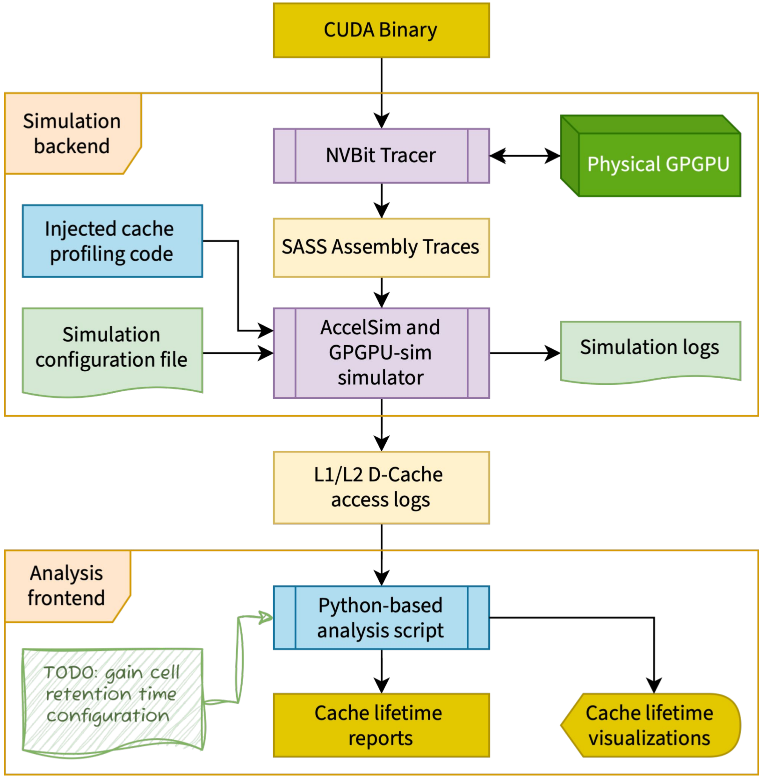
Experiment 1 L2 Results



Retention time reference: R. Giterman, A. Shalom, A. Burg, A. Fish, and A. Teman, "A 1-Mbit Fully Logic-Compatible 3T Gain-Cell Embedded DRAM in 16-nm FinFET," IEEE Solid-State Circuits Letters, vol. 3, pp. 110–113, 2020, doi: 10.1109/LSSC.2020.3006496.

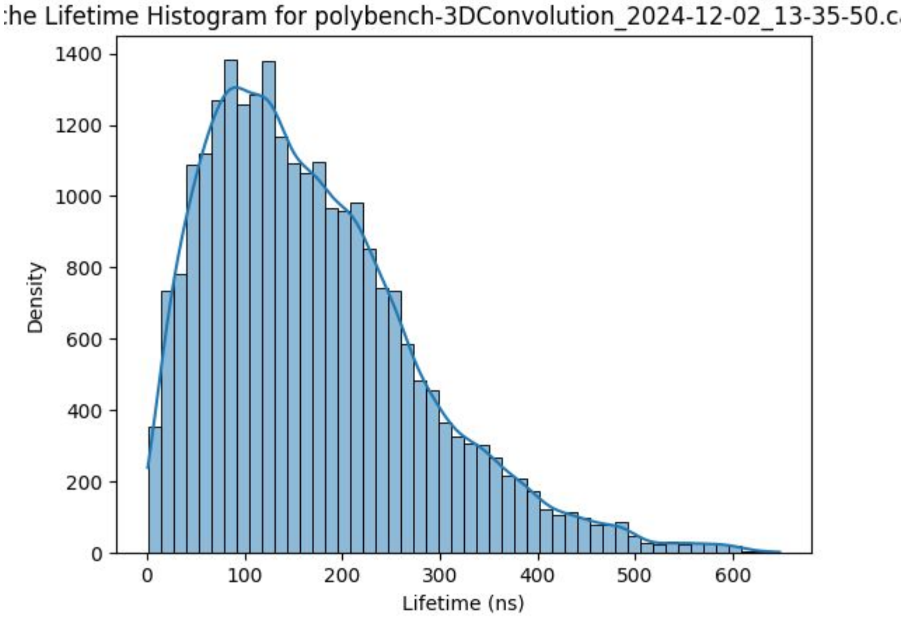
Experiment 2: PolyBench on GPGPU-Sim

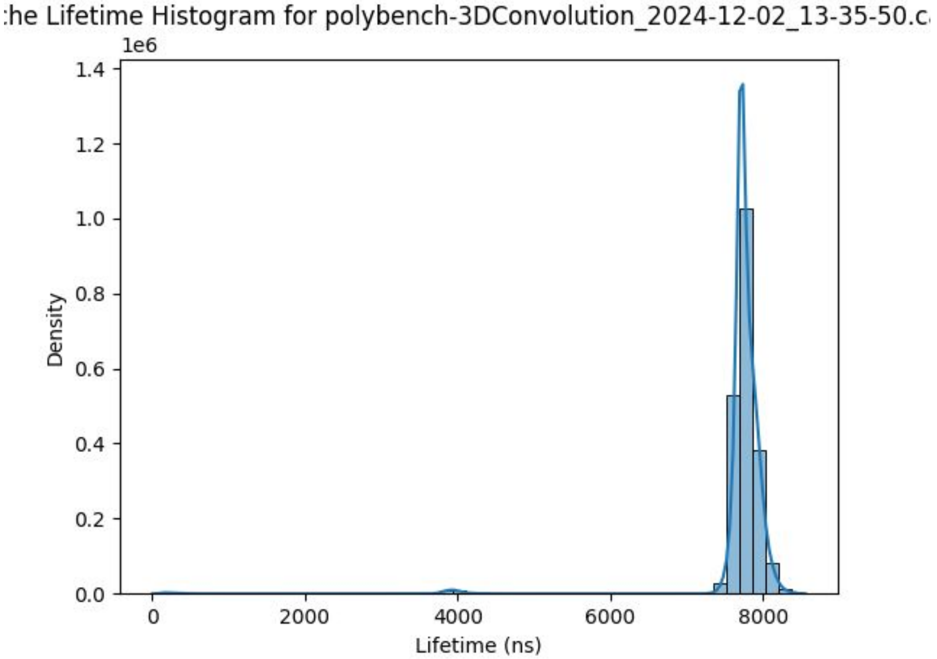
- Workload: benchmarks consisting of **single-kernel test programs**
  - 2D convolution kernel, 3D convolution kernel, GEMM kernel
- Backend: Accel-Sim and GPGPU-Sim
  - Modded **simulator** of NVIDIA GPUs
  - Captures cycle-accurate cache access information for each instruction
- Frontend: *per-address* visualizations
  - L1 and L2 lifetime distributions for *each cache line*



CUDA Binary
Simulation backend
NVBit Tracer
Physical GPGPU
Injected cache profiling code
SASS Assembly Traces
Simulation configuration file
AccelSim and GPGPU-sim simulator
Simulation logs
L1/L2 D-Cache access logs
Analysis frontend
Python-based analysis script
TODO: gain cell retention time configuration
Cache lifetime reports
Cache lifetime visualizations

Execution Results – 3D Convolution

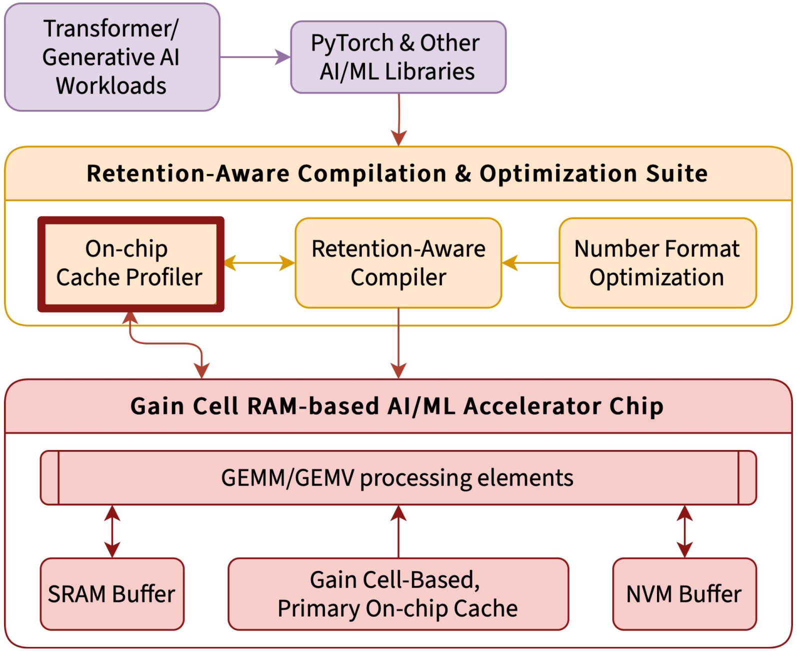




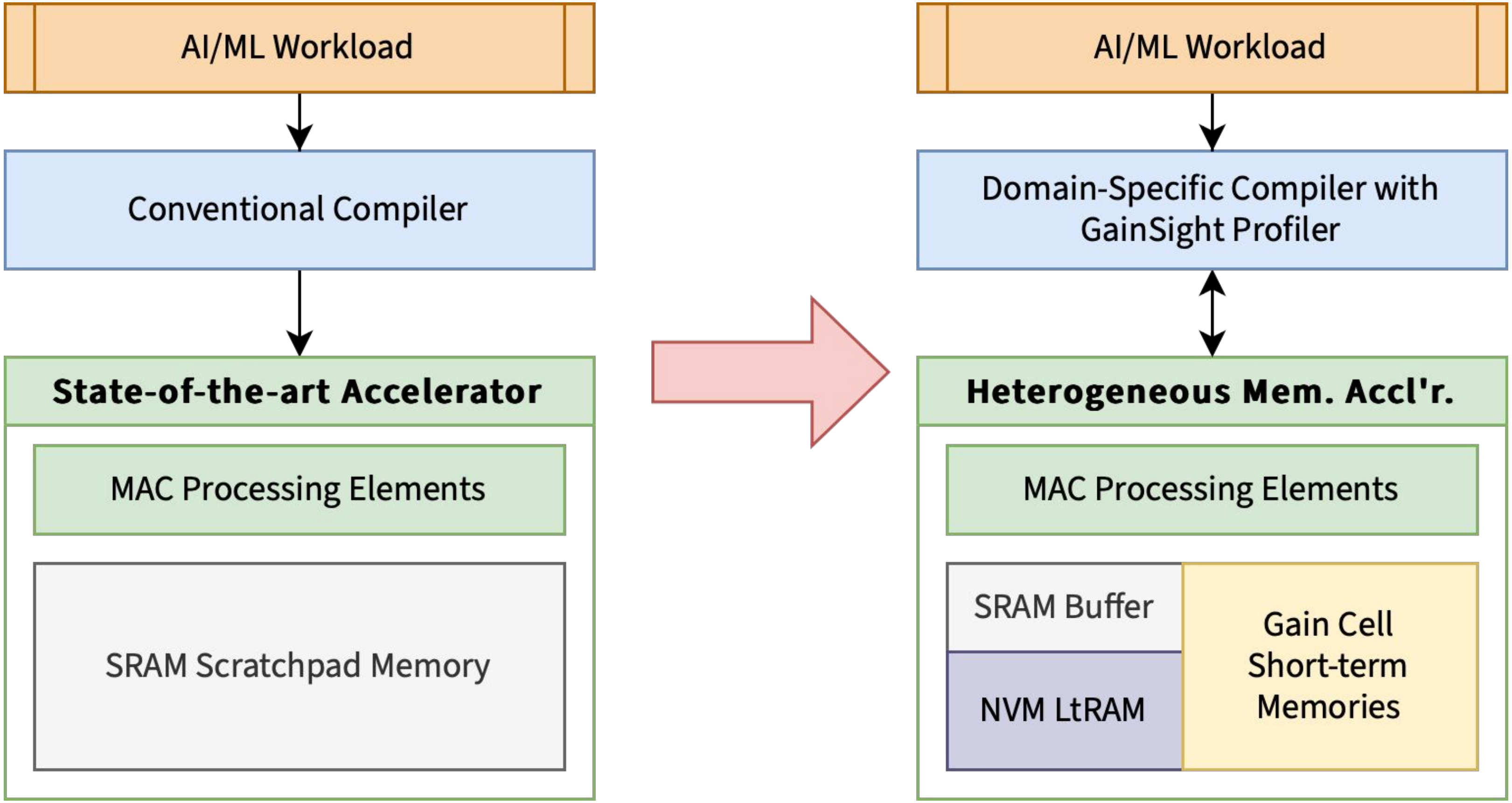
Next Steps

Next Steps for Proof-of-Concept Design

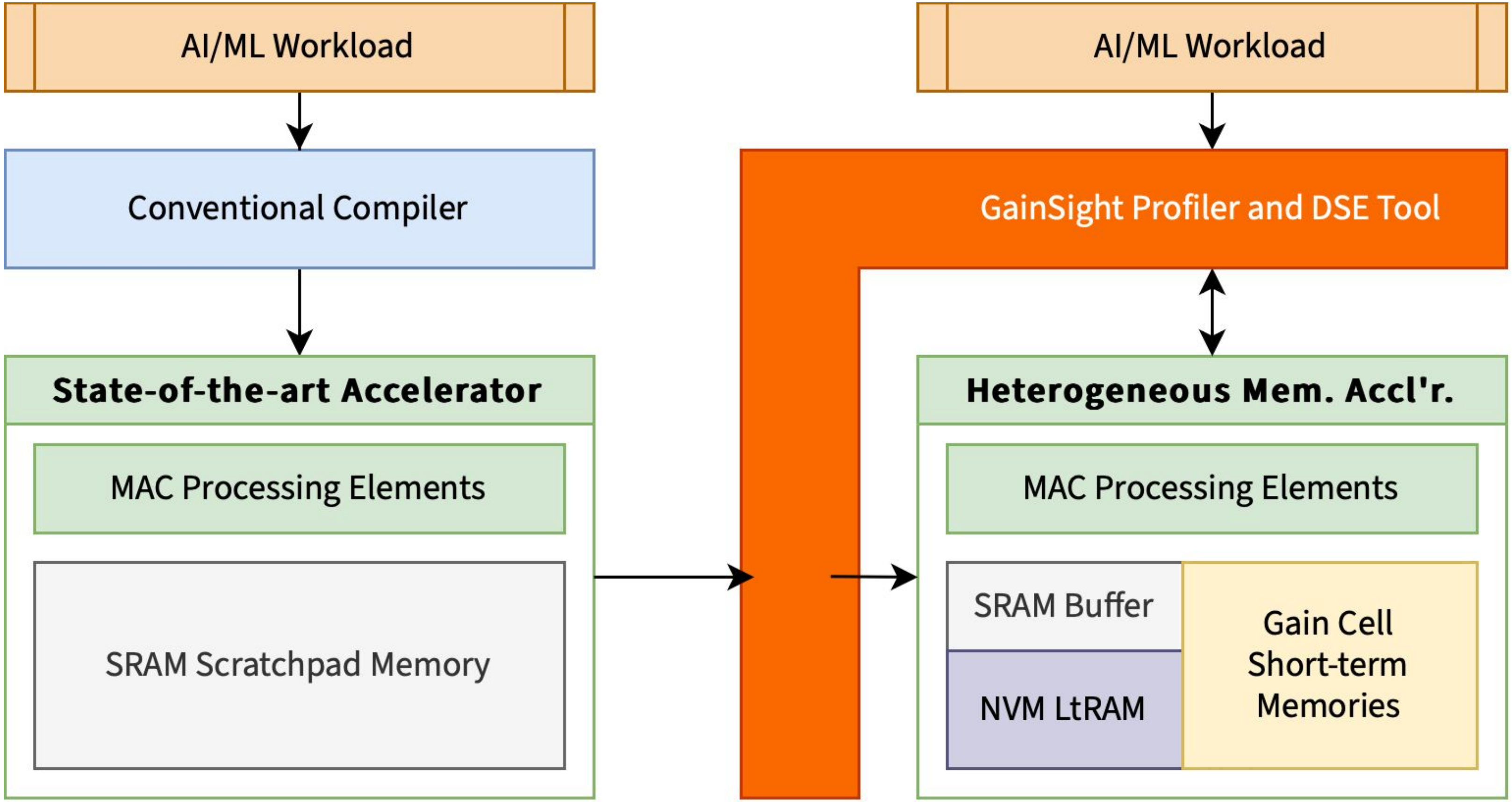
- Implement more HW backends (e.g., systolic array, dataflow accelerators)
- Frontend analytical models
- Goal: build a **proof of concept AI accelerator chip** using gain cells as primary on-chip, short term memory
  - 3x cell density, <0.3x leakage power
- First in a series of SW tools for retention aware compilation suite


Transformer/ Generative AI Workloads
PyTorch & Other AI/ML Libraries
Retention-Aware Compilation & Optimization Suite
On-chip Cache Profiler
Retention-Aware Compiler
Number Format Optimization
Gain Cell RAM-based AI/ML Accelerator Chip
GEMM/GEMV processing elements
SRAM Buffer
Gain Cell-Based, Primary On-chip Cache
NVM Buffer

Rounding Out Our Vision



Rounding Out Our Vision



Thank You!