Self Adaptive Reconfigurable Arrays: Learning Flexible GEMM Accelerator Configuration & Mapping-space using ML

59th Design Automation Conference (DAC), 2022

Sang-Soo Park

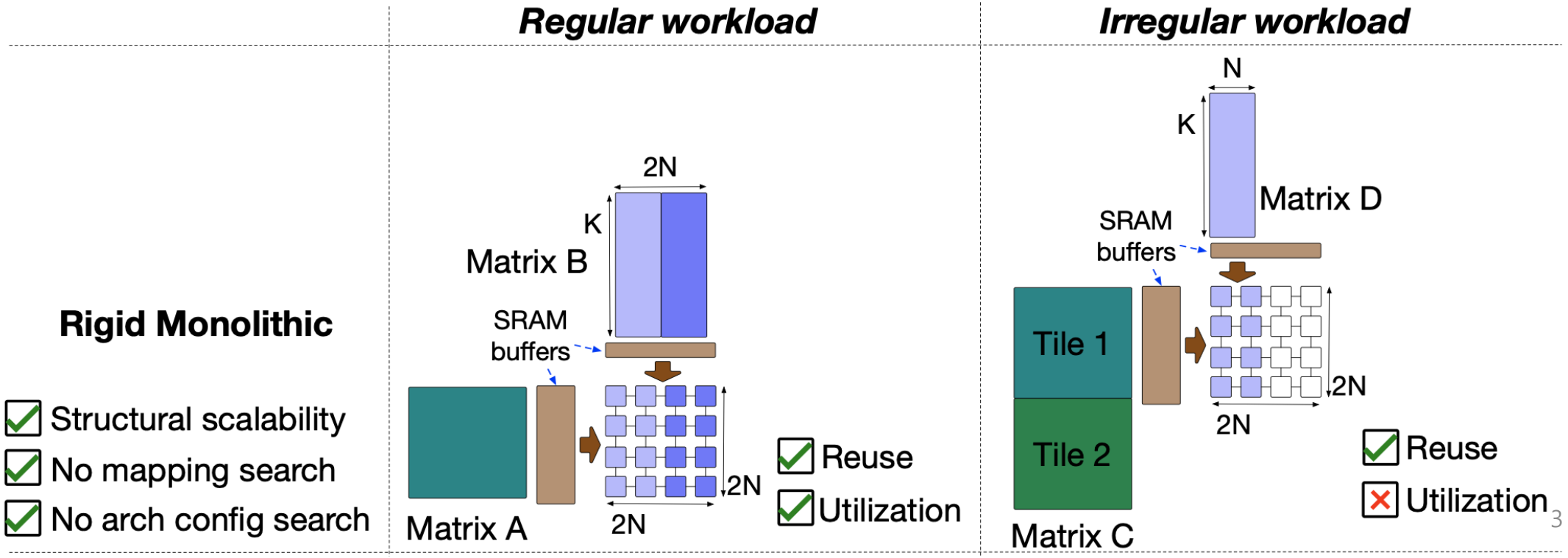
2022-09-15

Contents

- Systolic array (SA) architectures
  - Rigid, Flexible, Distributed
  - Motivation: Various GEMM operations
- SAGAR: Shape adaptive GEMM accelerator
  - Self adaptive unit (SA) and Reconfigurable array (RA) units
  - AdaptNet: Recommendation for GEMM operations
- SAGAR evaluations
  - Performance and hardware cost analysis

Monolithic & Distributed accelerators

- Rigid monolithic array
  - Simple to construct but no flexibility leaning to high under utilization
  - TPU's systolic array

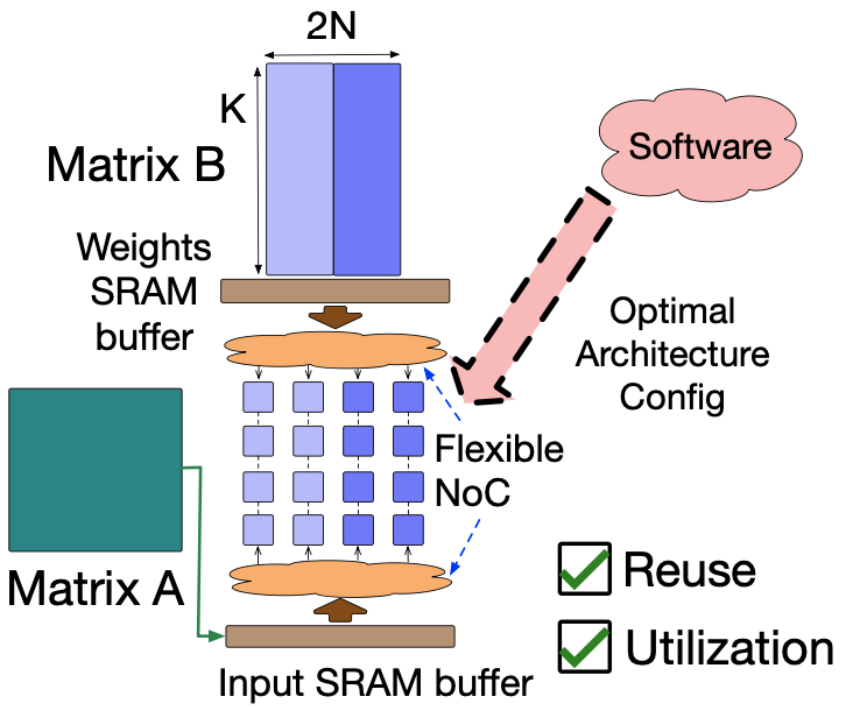


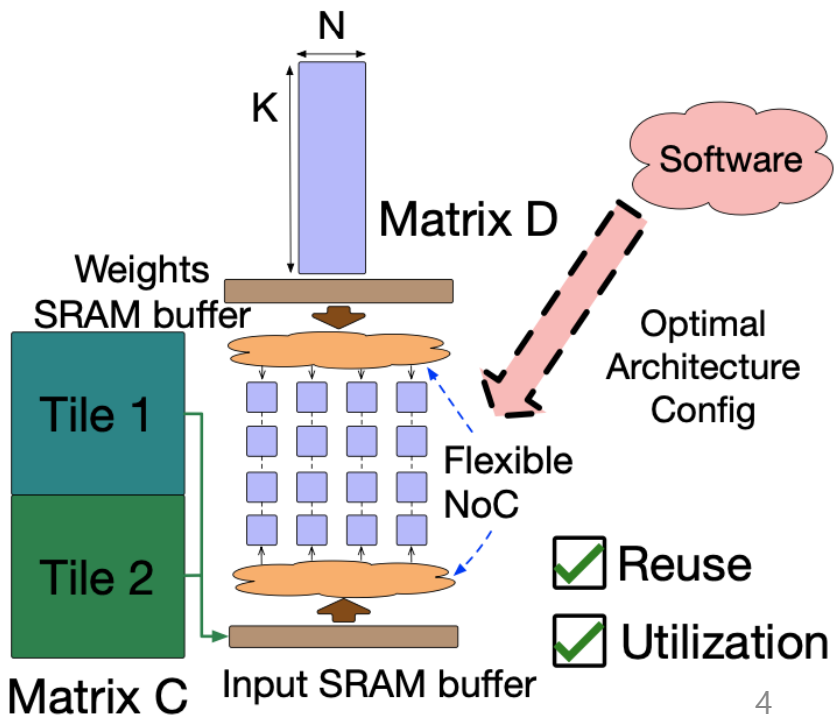
Monolithic & Distributed accelerators

- Flexible monolithic array
  - Flexibility via cluster of interconnects and configuration logics w/ SW
  - MAERI, Eyeriss v2, SIGMA

Flexible Monolithic

- ✗ Structural scalability
- ✓ No mapping search
- ✗ No arch config search



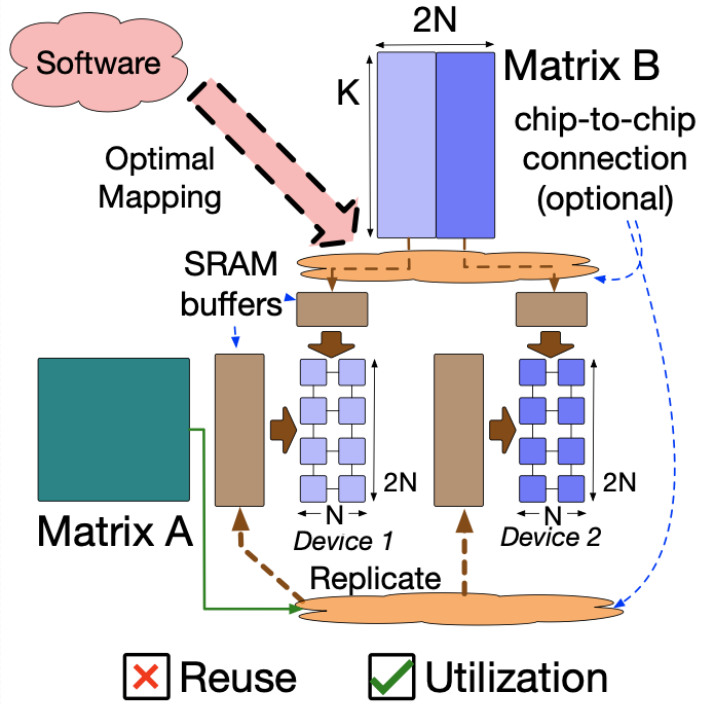


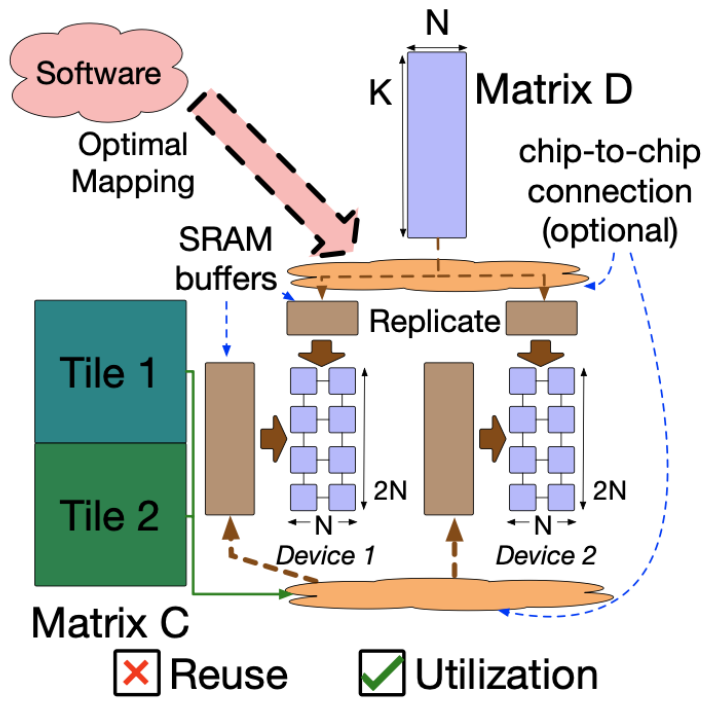
Monolithic & Distributed accelerators

- Distributed architecture
  - Exacerbating mapping search problem by distribute array
  - NoC architecture (Simba, Tangram)

Distributed

- ☑ Structural scalability
- ☒ No mapping search
- ☑ No arch config search





Monolithic & Distributed accelerators

- Distributed architecture
  - Exacerbating mapping search problem by distribute array
  - NoC architecture (Simba, Tangram)

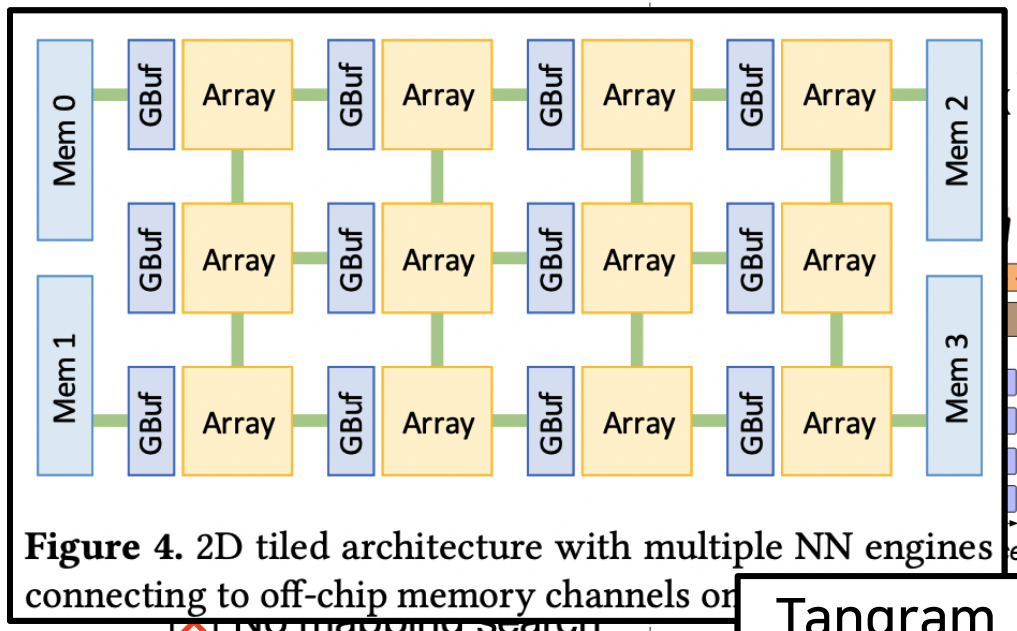
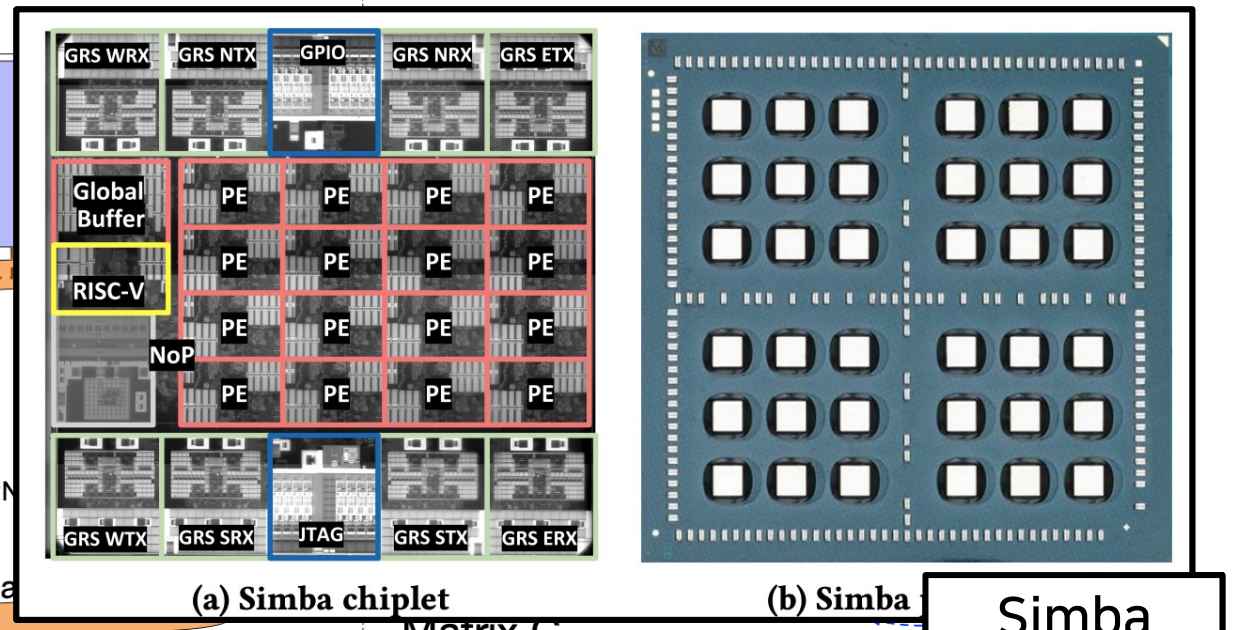


Figure 4. 2D tiled architecture with multiple NN engines connecting to off-chip memory channels or

Tangram



(a) Simba chiplet (b) Simba

Simba

No mapping search

No arch config search

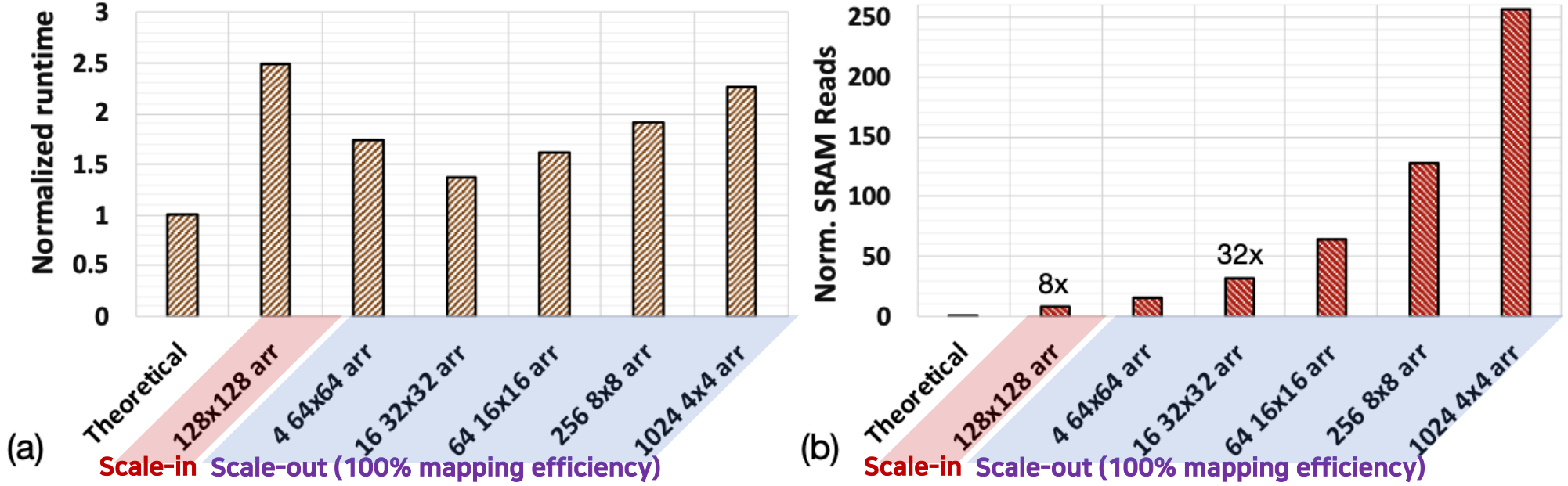
Matrix C

Reuse Utilization

Reuse Utilization

Motivation: Various GEMM operations

- Trade-off between performance and loss of reuse
  - Scale-Sim with16K PE array configurations
  - 16 32×32 array: 2× times faster, 4× memory access (energy eff. ↓)

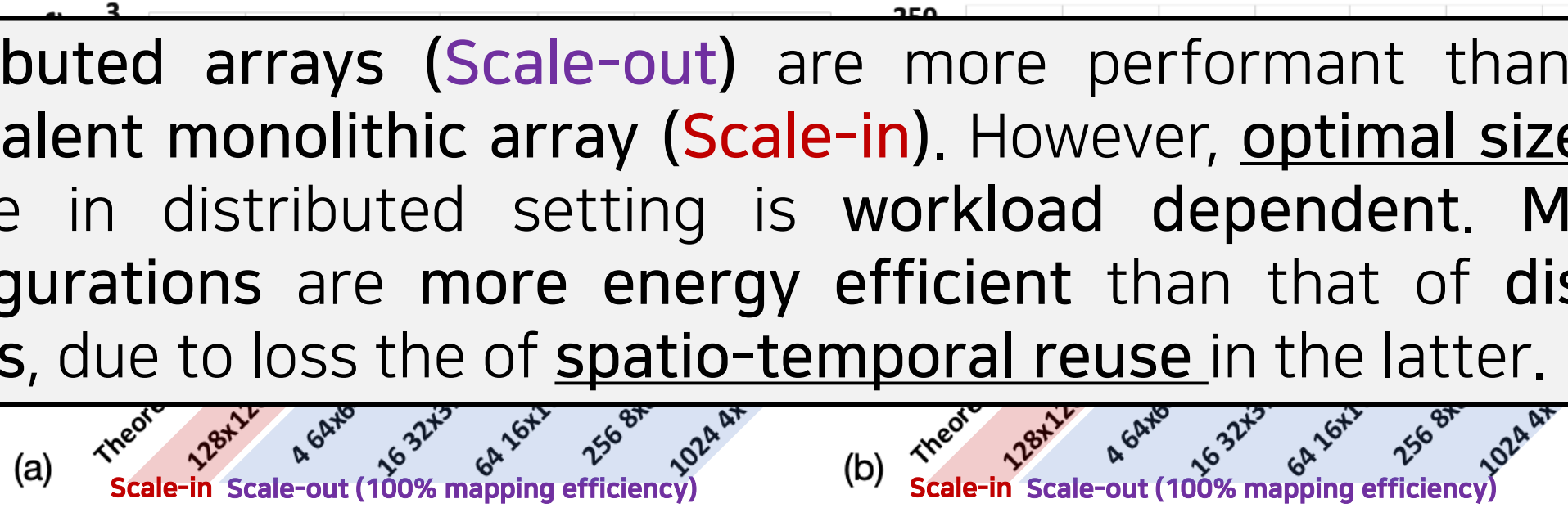


Trade-off between runtime and lost reuse in compute equivalent monolithic & distributed SA

Motivation: Various GEMM operations

- Trade-off between performance and loss of reuse
  - Scale-Sim with16K PE array configurations
  - 16 32×32 array: 2× times faster, 4× memory access (energy eff. ↓)

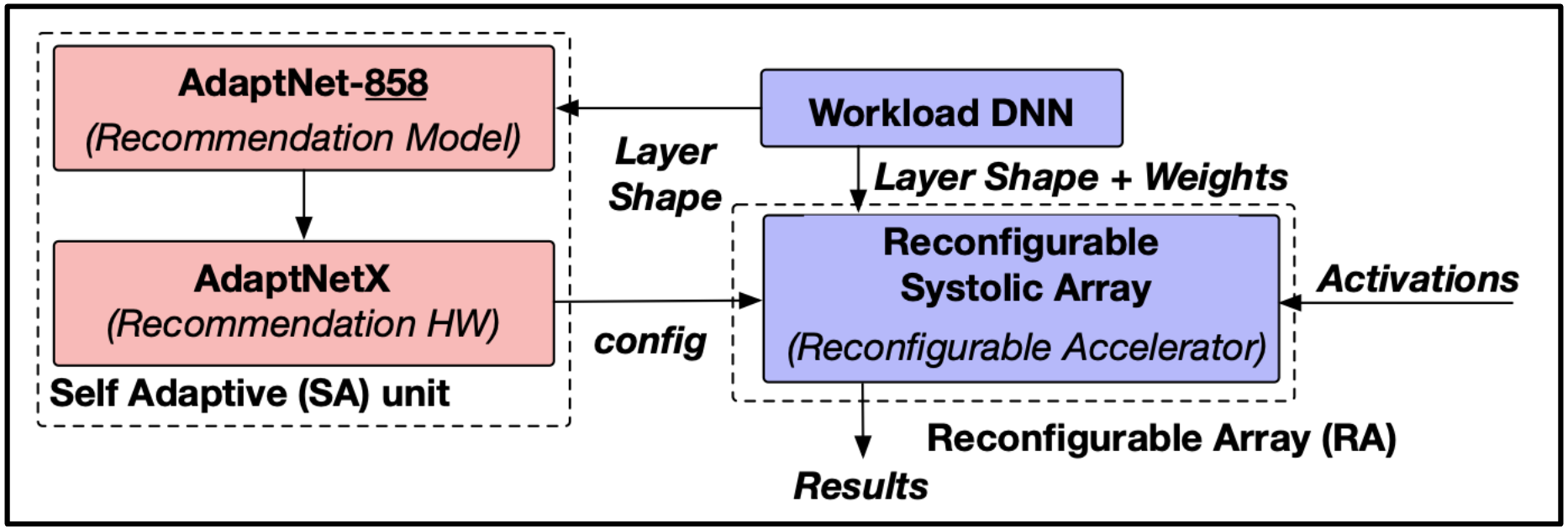
Distributed arrays (Scale-out) are more performant than that of **equivalent monolithic array (Scale-in)**. However, **optimal size** of each device in distributed setting is **workload dependent**. **Monolithic configurations** are **more energy efficient** than that of **distributed arrays**, due to loss the of **spatio-temporal reuse** in the latter.



Trade-off between runtime and lost reuse in compute equivalent monolithic & distributed SA

Shape adaptive GEMM accelerator

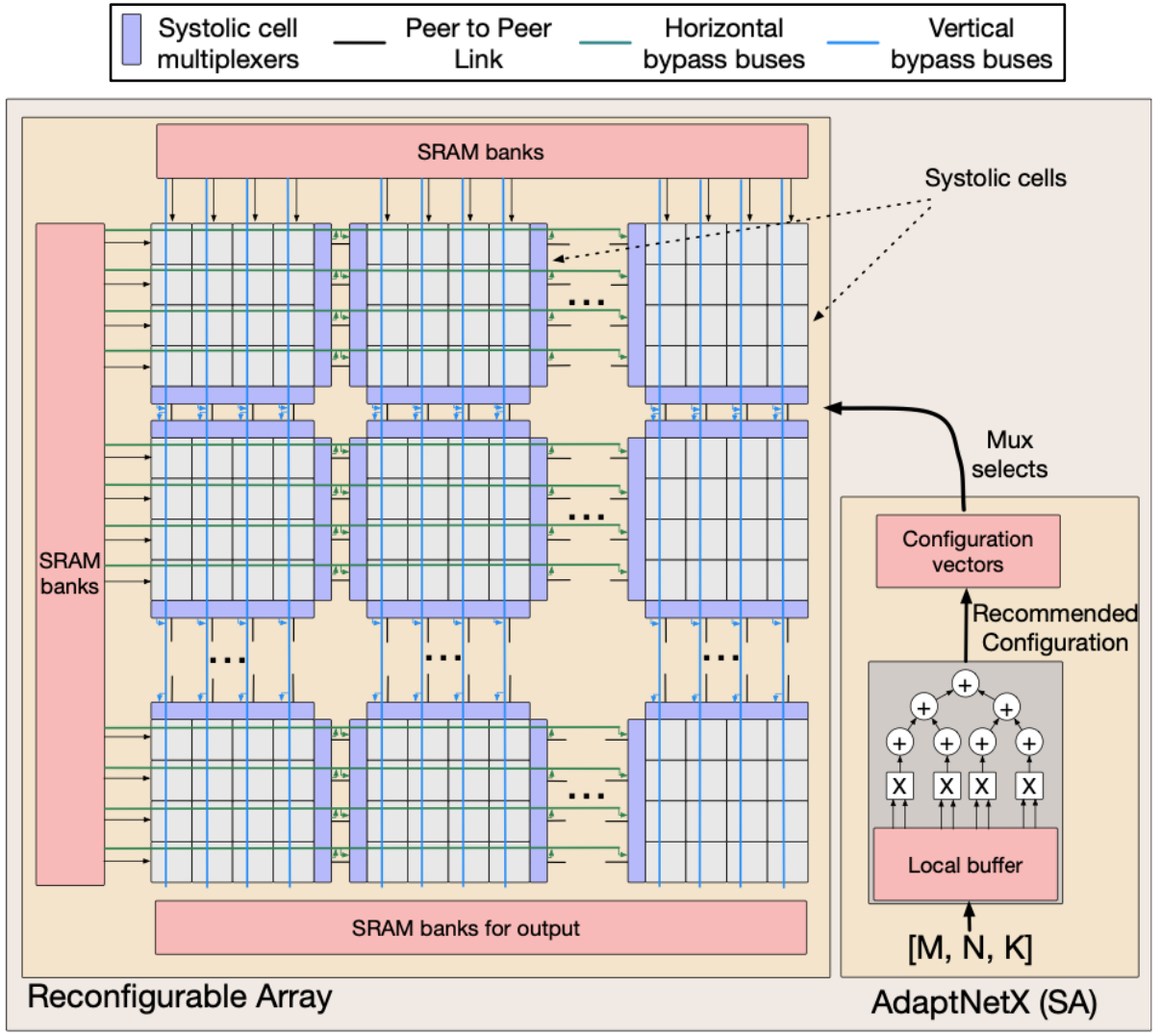
- Mapping & configuration space of reconfigurable accelerator
  - Reconfigurable array: Various dataflow, Mono/Distribute architecture
  - Self adaptive: Accelerator for ML model (**Optimal parameters**)



Constitution and interactions of self adaptive (SA) and reconfigurable array (RA)

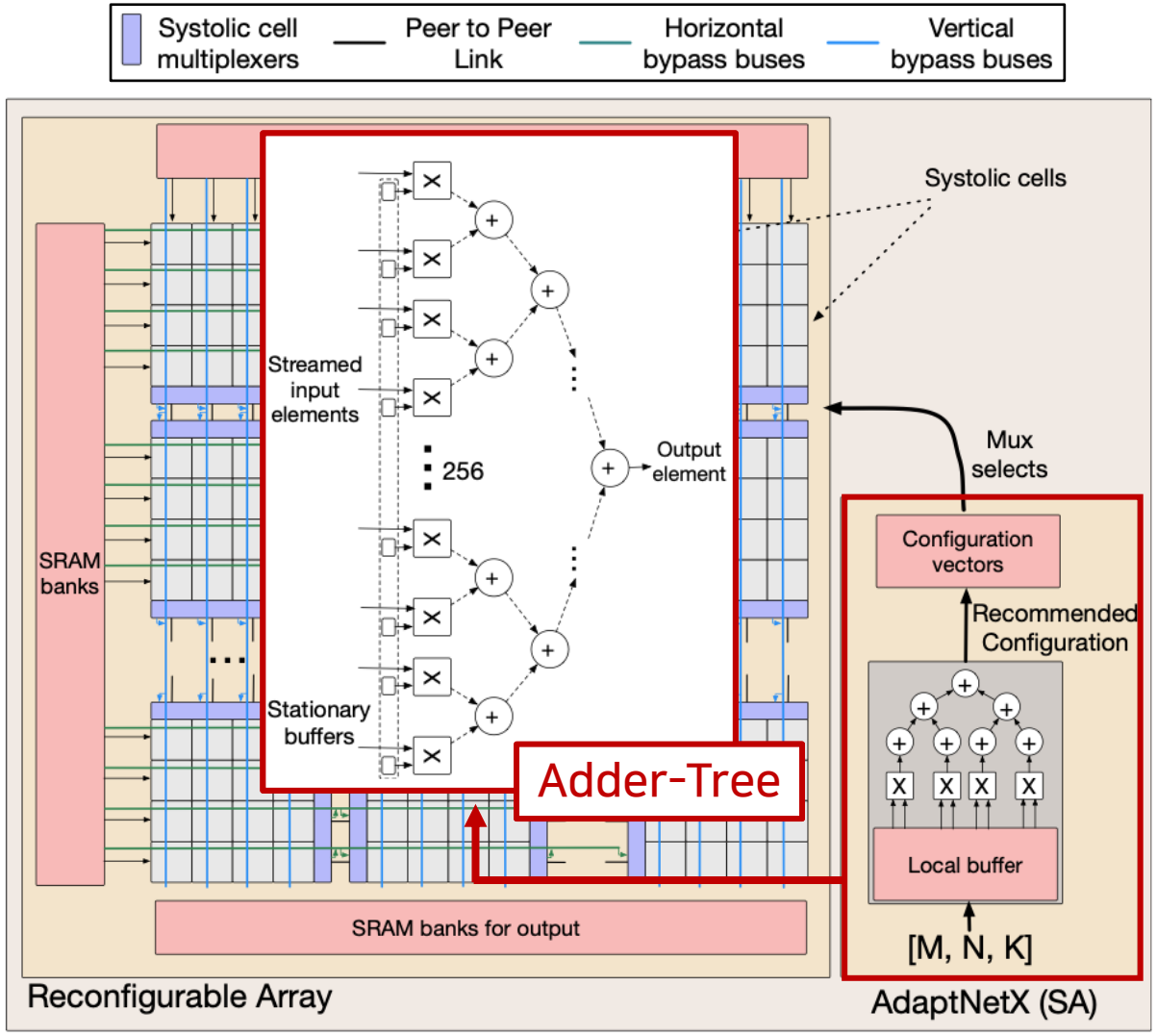
Shape adaptive GEMM accelerator

- SAGAR architecture
  - SA (1D adder-tree unit)
    - Streamed input & Stationary weight
    - Inference of AdaptNet
    - Choosing RA operations
  - RA (Reconfigurable SA)
    - Various dataflow (OS/WS/IS)
    - Monolithic, Distributed



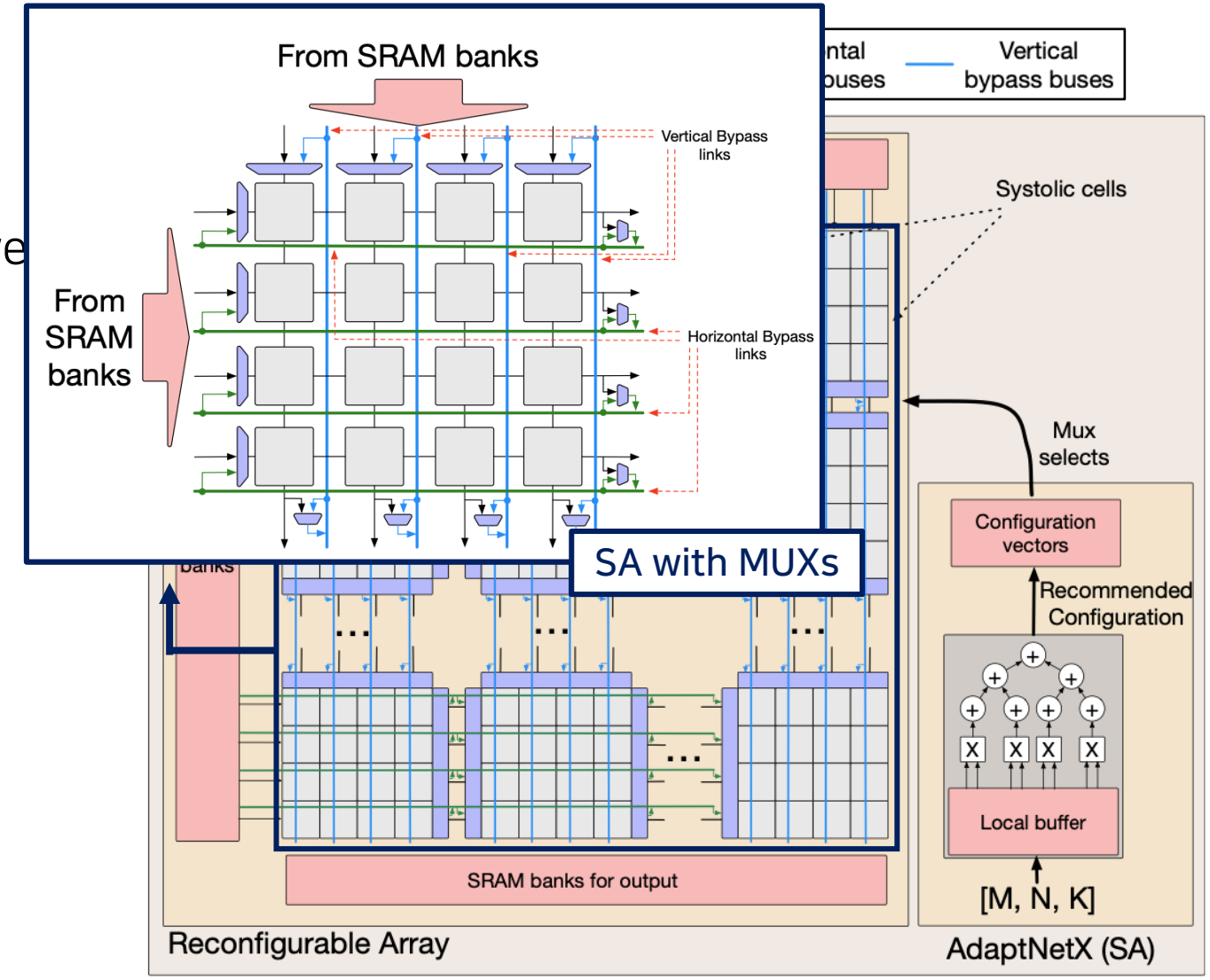
Shape adaptive GEMM accelerator

- SAGAR architecture
  - SA (1D adder-tree unit)
    - Streamed input & Stationary weight
    - Inference of AdaptNet
    - Choosing RA operations
  - RA (Reconfigurable SA)
    - Various dataflow (OS/WS/IS)
    - Monolithic, Distributed



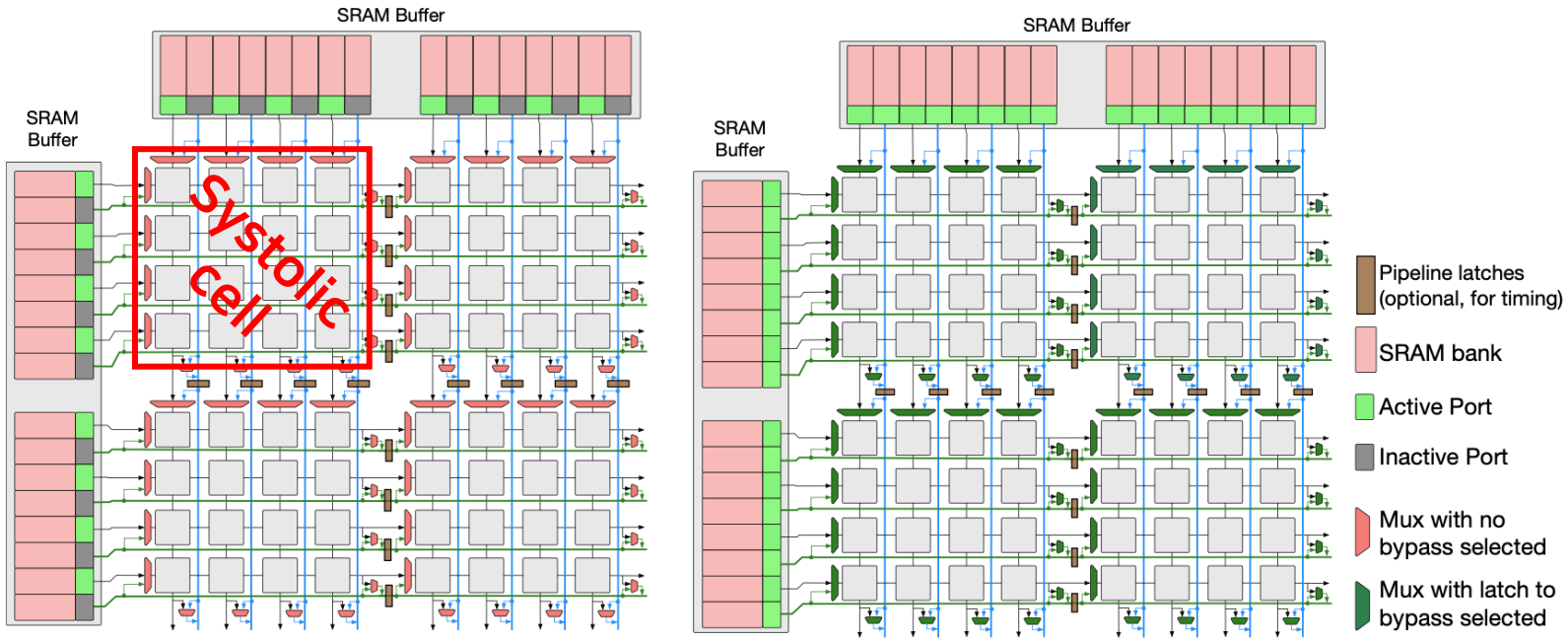
Shape adaptive GEMM accelerator

- SAGAR architecture
  - SA (1D adder-tree unit)
    - Streamed input & Stationary we
    - Inference of AdaptNet
    - Choosing RA operations
  - RA (Reconfigurable SA)
    - Various dataflow (OS/WS/IS)
    - Monolithic, Distributed



Shape adaptive GEMM accelerator

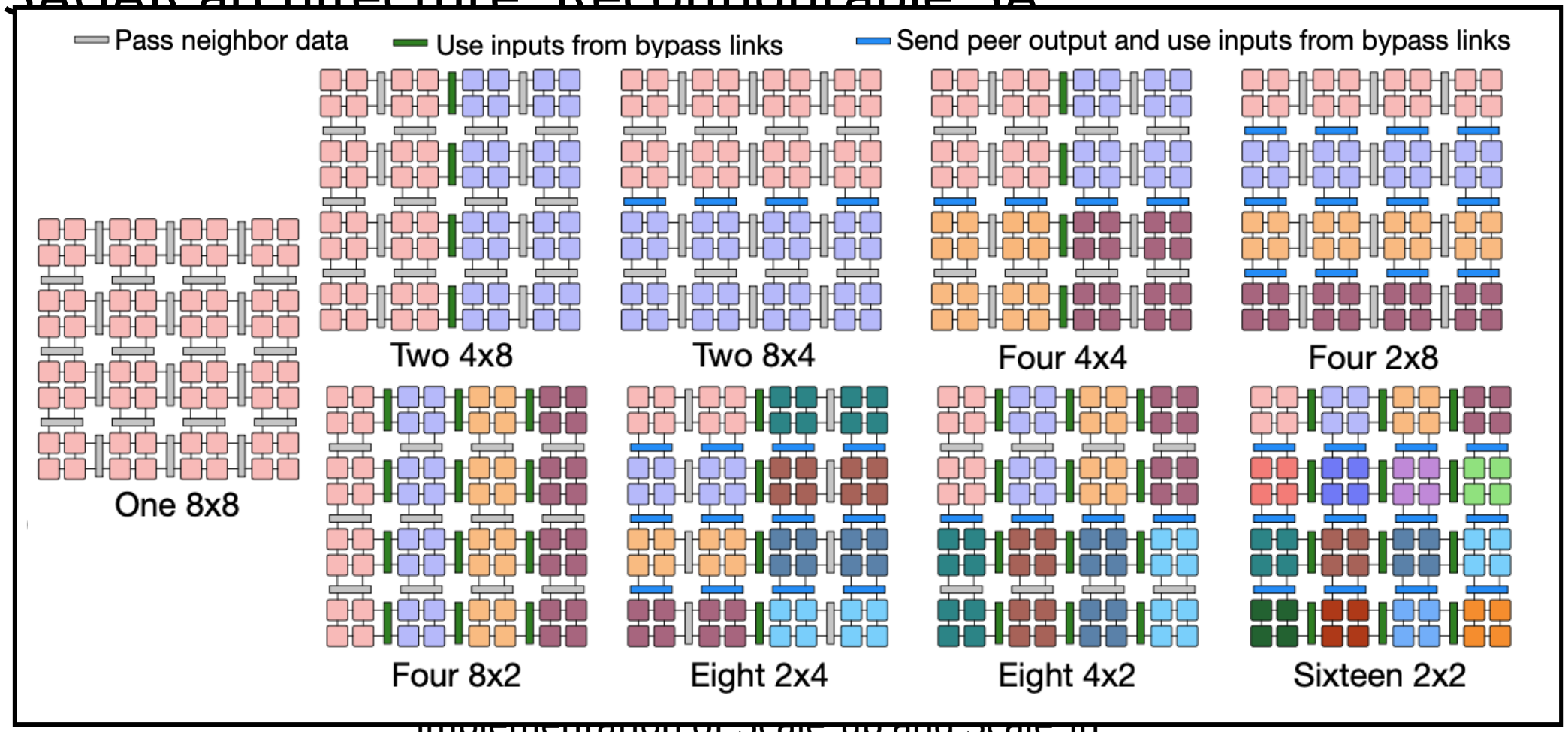
- SAGAR architecture: Reconfigurable SA
  - Mapping flexibility improved by work on different operations
    - Needs to provision for additional links from SRAM to PE units (Area, Energy ↑)
    - Systolic cell: Small grid of PE units augmented with MUXs at edges



Implementation of Scale-up and Scale-in

Shape adaptive GEMM accelerator

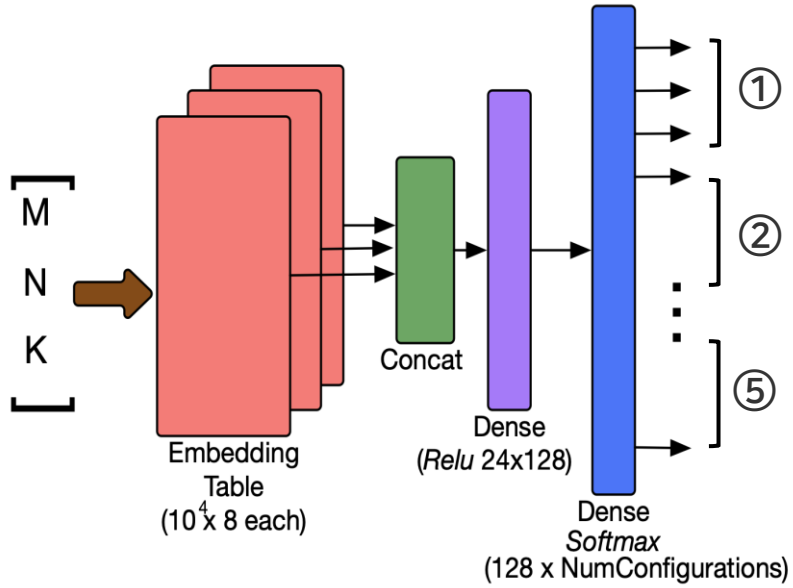
- SAGAR architecture: Reconfigurable SA



implementation of Scale-up and Scale-in

Recommendation for GEMM operations

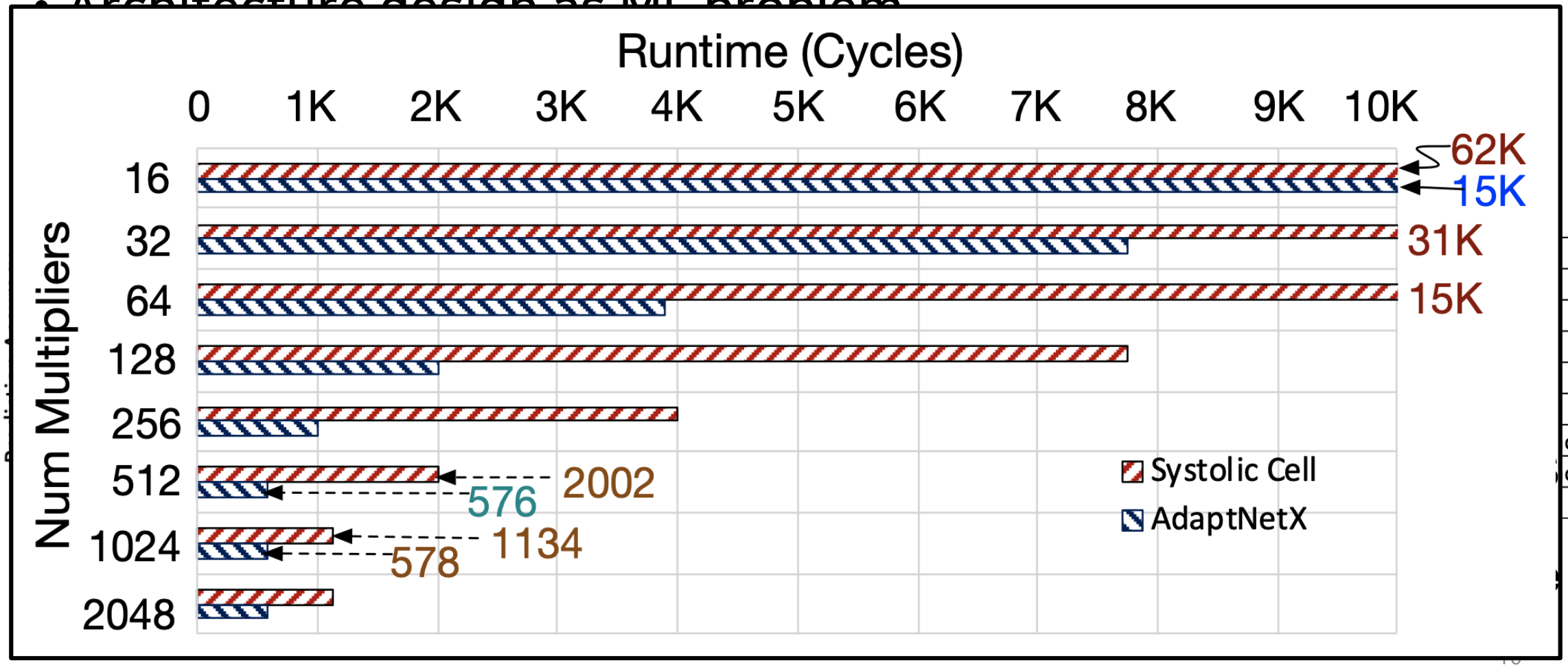
- Architecture design as ML problem
  - Framing as classification and recommendation task works best
    - Number and logical layout of partitions
    - Dimensions of array in each partition, mapping/dataflow (OS/WS/IS)



	Horizontal systolic cells ①	Vertical systolic cells ②	Systolic cell rows ③	Systolic cell cols ④	Dataflow ⑤
0	16	64	4	4	OS
1	32	32	4	4	OS
2	32	32	4	4	WS
3	16	16	8	8	IS
4	8	32	8	8	WS
⋮	⋮	⋮	⋮	⋮	⋮
N	2	2	32	32	IS

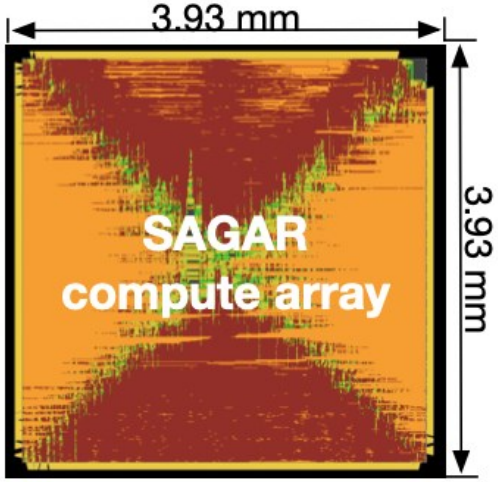
Overview of ADAPTNET

Recommendation for GEMM operations



SAGAR evaluations

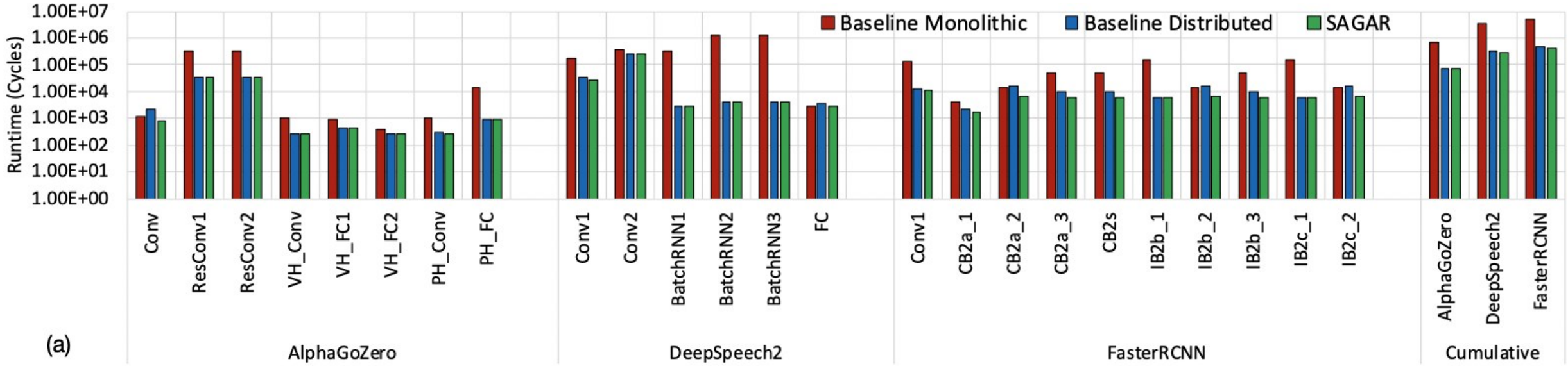
- Implementation, Methodology, and Workloads
  - RTL as 32×32 array of 4×4 systolic-cells, ASIC flow till PnR
    - 28nm library, SRAM buffers as collection of 1024 1KB cells (Synopsys)
    - Operating frequency of 1GHz, 32.768 TOPS, 81.90 mm^2, 13.01W
    - SA: 8.65% of area and 1.36% of power
  - In-house script to generate Scale-Sim to perform workload partitioning
    - Faster RCNN, DeepSpeech2, and AlphaGo Zero



SAGAR	
Systolic cell dims	4x4
Num systolic cells	1024
Max Throughput	32.768 TOPs
Frequency	1 GHz
Tech node	28nm
Area	81.90 mm2
Power	13.01 Watts

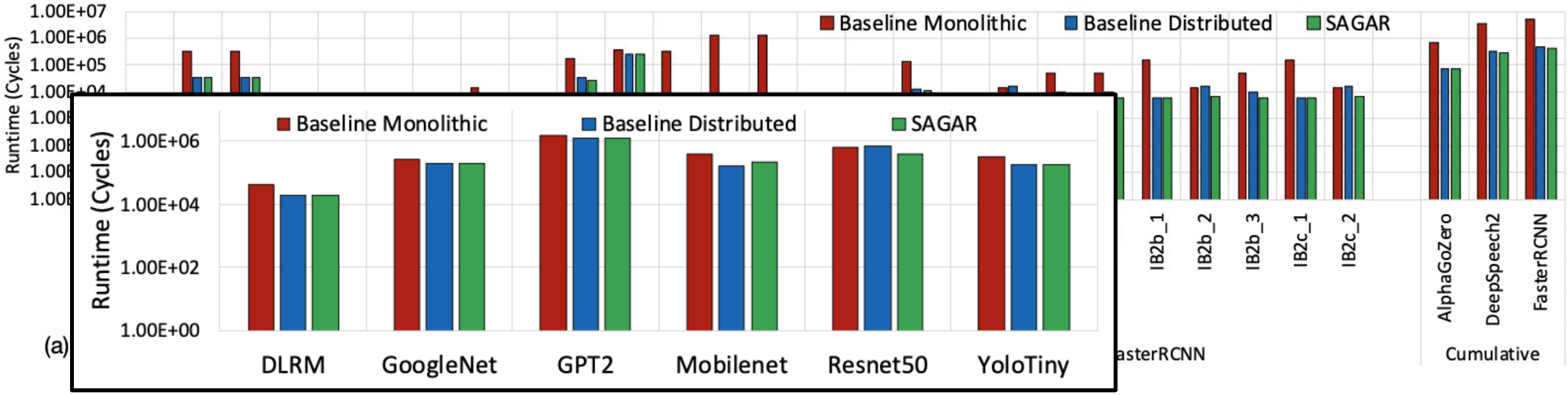
SAGAR evaluations

- Performance analysis: Runtime
  - Baseline/Distributed: 128×128 monolithic systolic and 1024 4×4 arrays
  - Flexibility leading to lower aggregated runtime for SAGAR



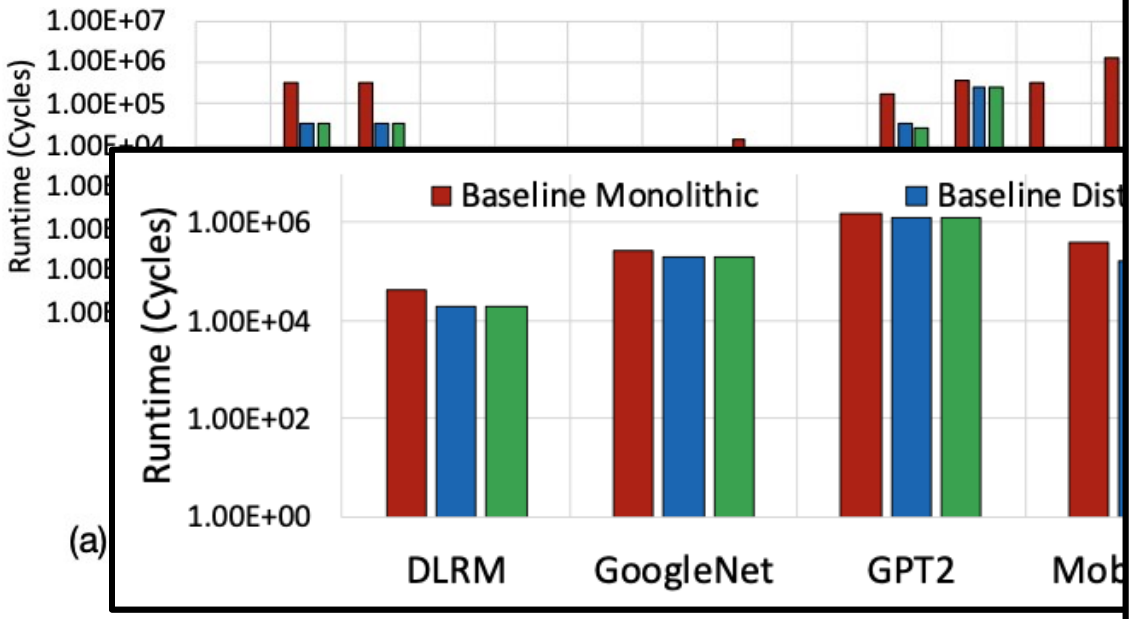
SAGAR evaluations

- Performance analysis: Runtime
  - Baseline/Distributed: 128×128 monolithic systolic and 1024 4×4 arrays
  - Flexibility leading to lower aggregated runtime for SAGAR



SAGAR evaluations

- Performance analysis: Runt
  - Baseline/Distributed: 128×12
  - Flexibility leading to lower ag



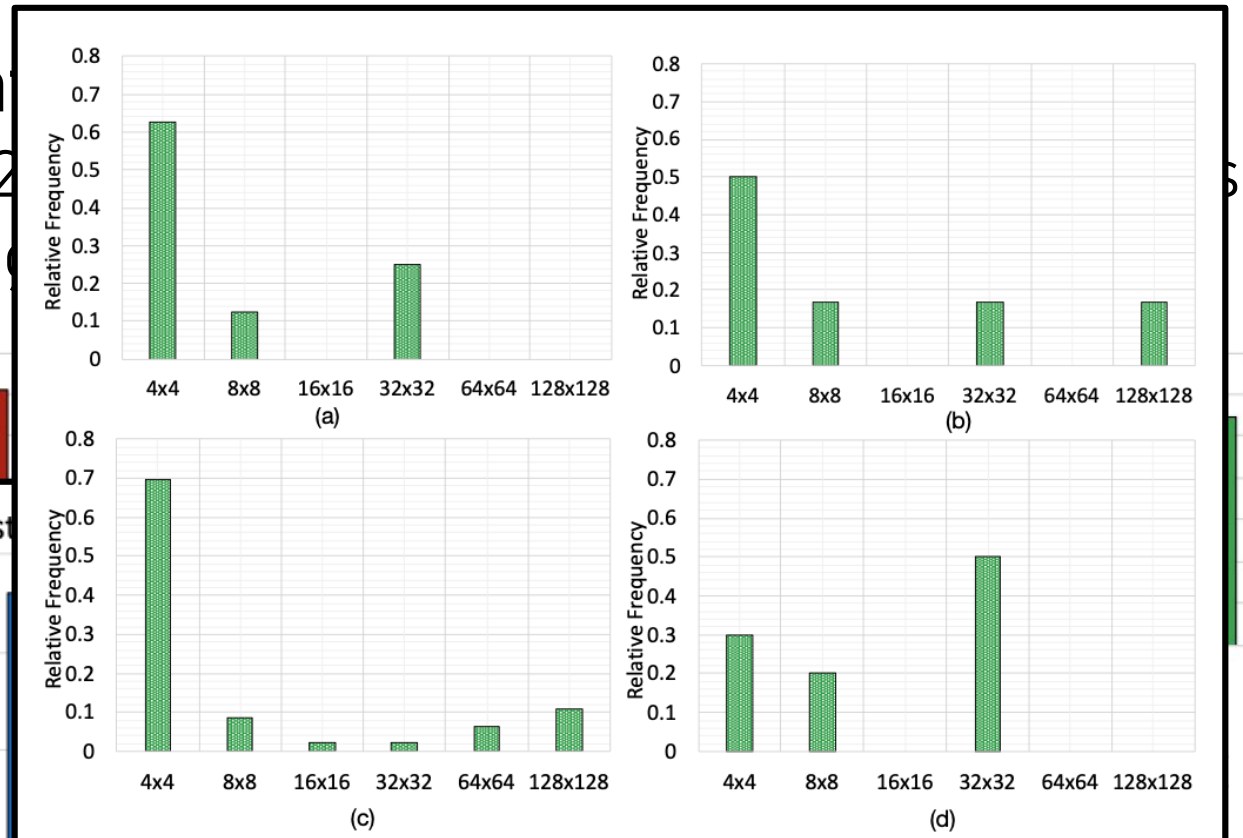
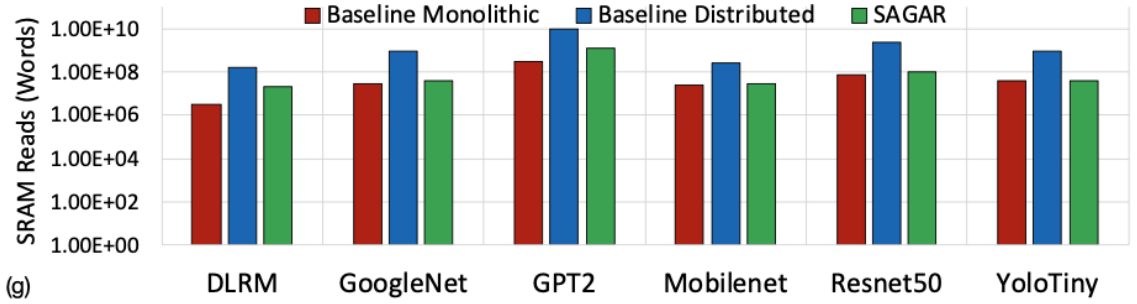
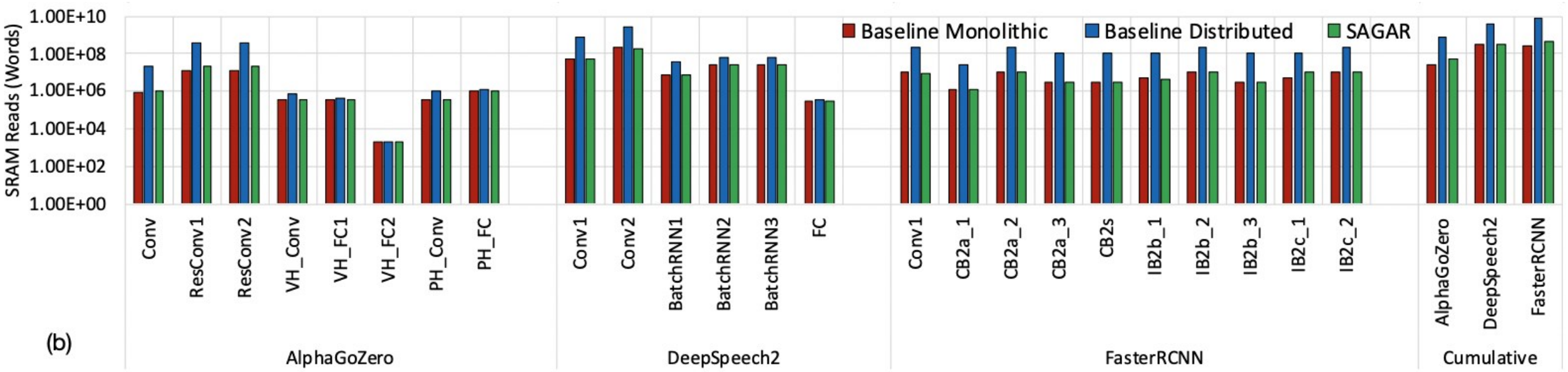


Fig. 12. Distribution of favorable array sizes for a 16384 MAC distributed system which attain the lowest runtime when run for each layer in (a) synthetic GEMM workloads (b) AlphaGoZero, (c) DeepSpeech2, and (d) FasterRCNN.

SAGAR evaluations

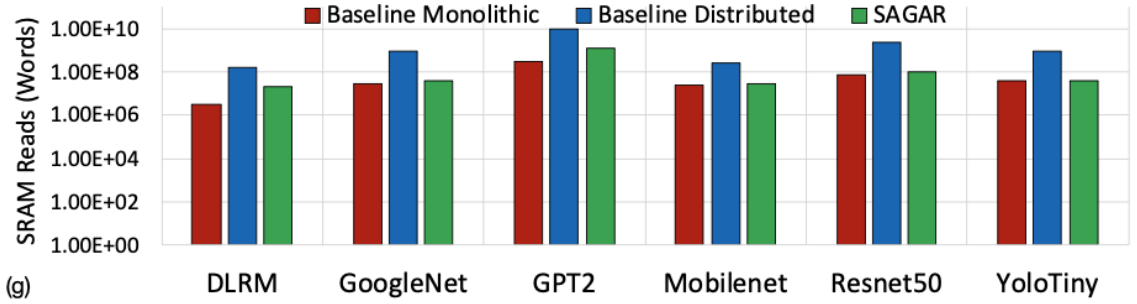


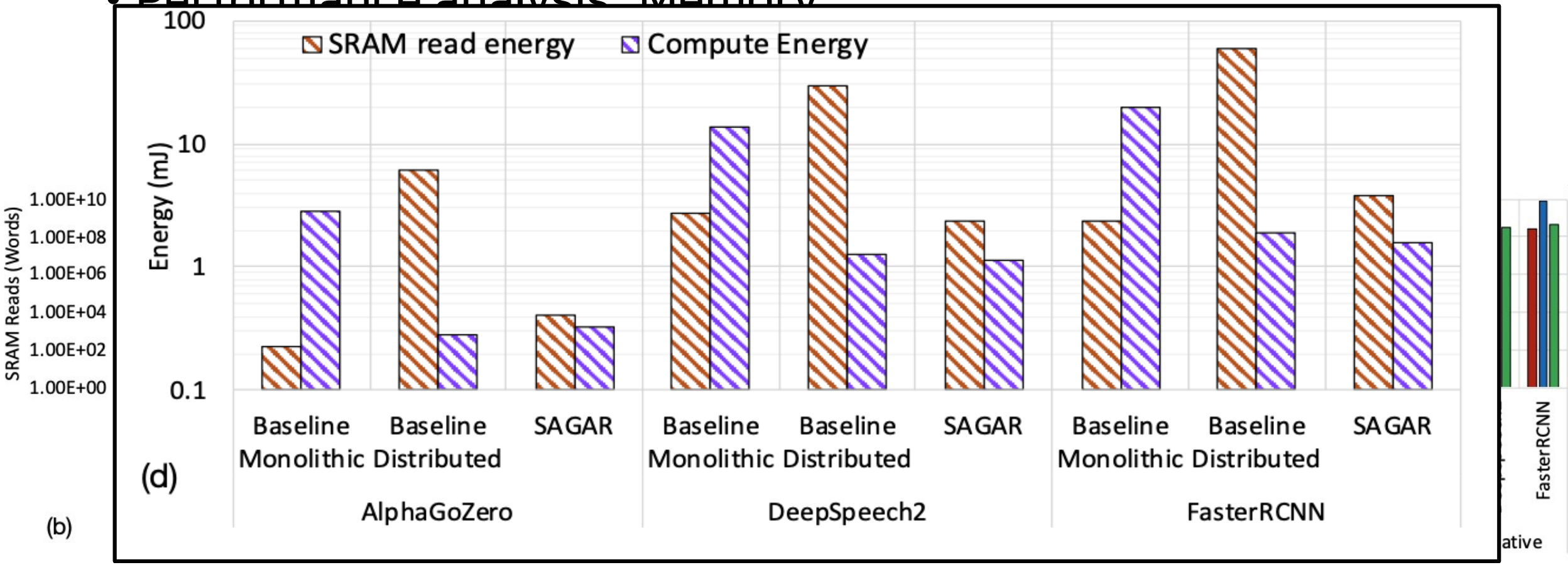
- Performance analysis: Memory
  - Baseline/Distributed: 128×128 monolithic systolic and 1024 4×4 arrays
  - Mitigated efficiency loss in reuse by bypassing links



SAGAR evaluations

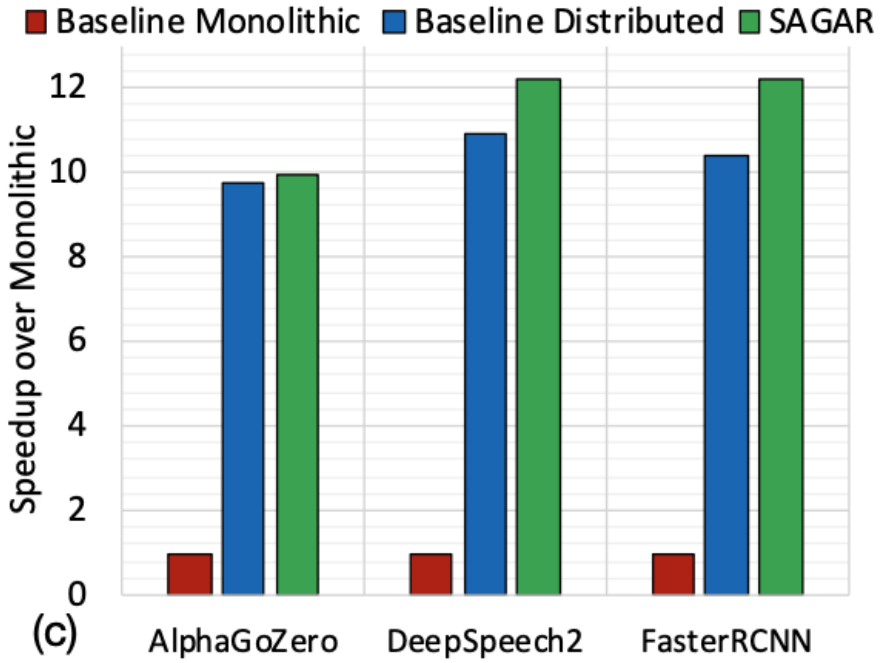
- Performance analysis: Memory

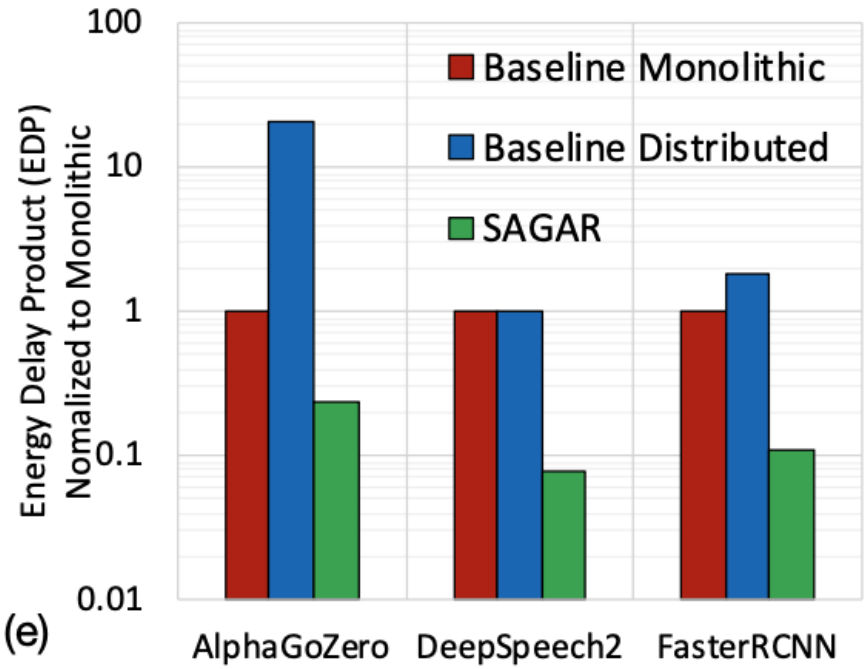




SAGAR evaluations

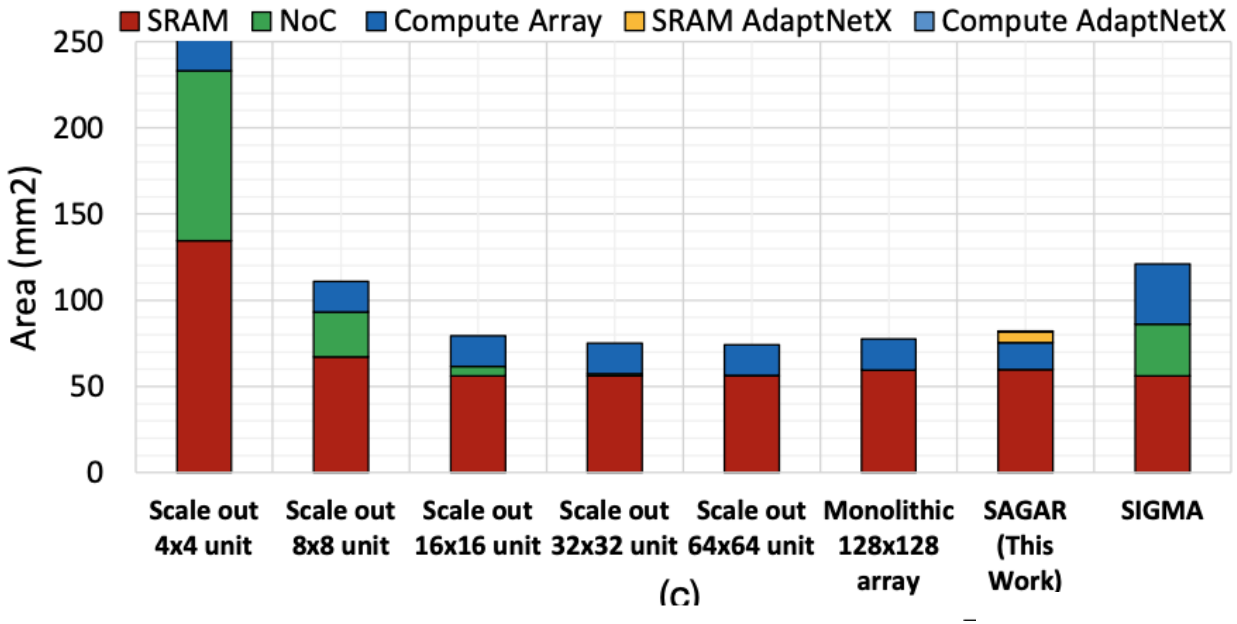
- Performance analysis: Overall
  - About > 10× speedup over monolithic baseline
  - 98% to 80% less EDP compared to monolithic baseline

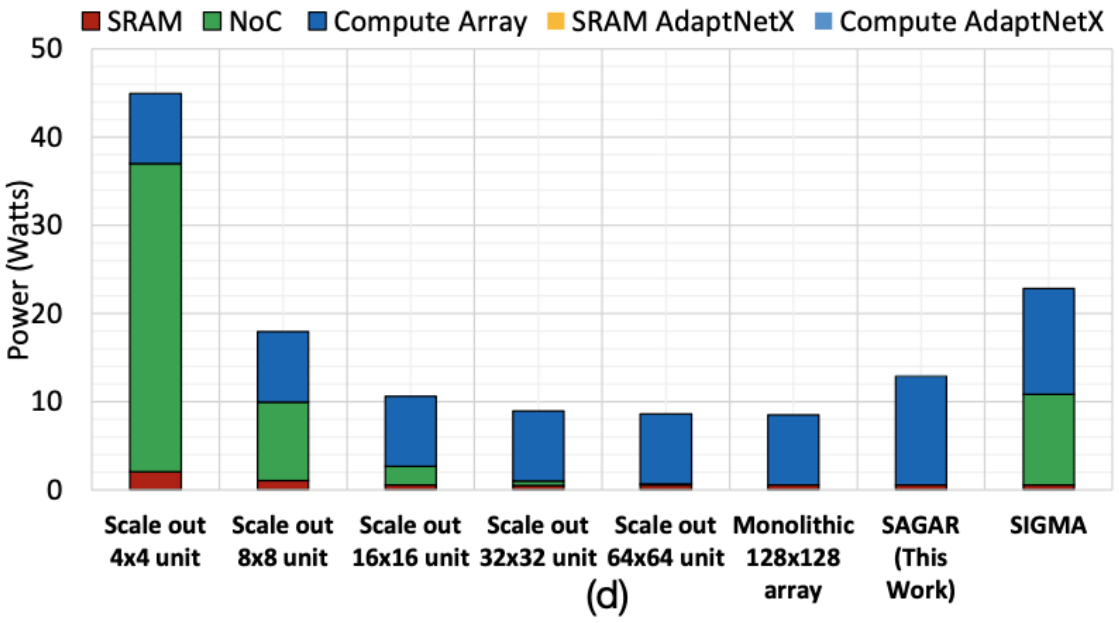




SAGAR evaluations

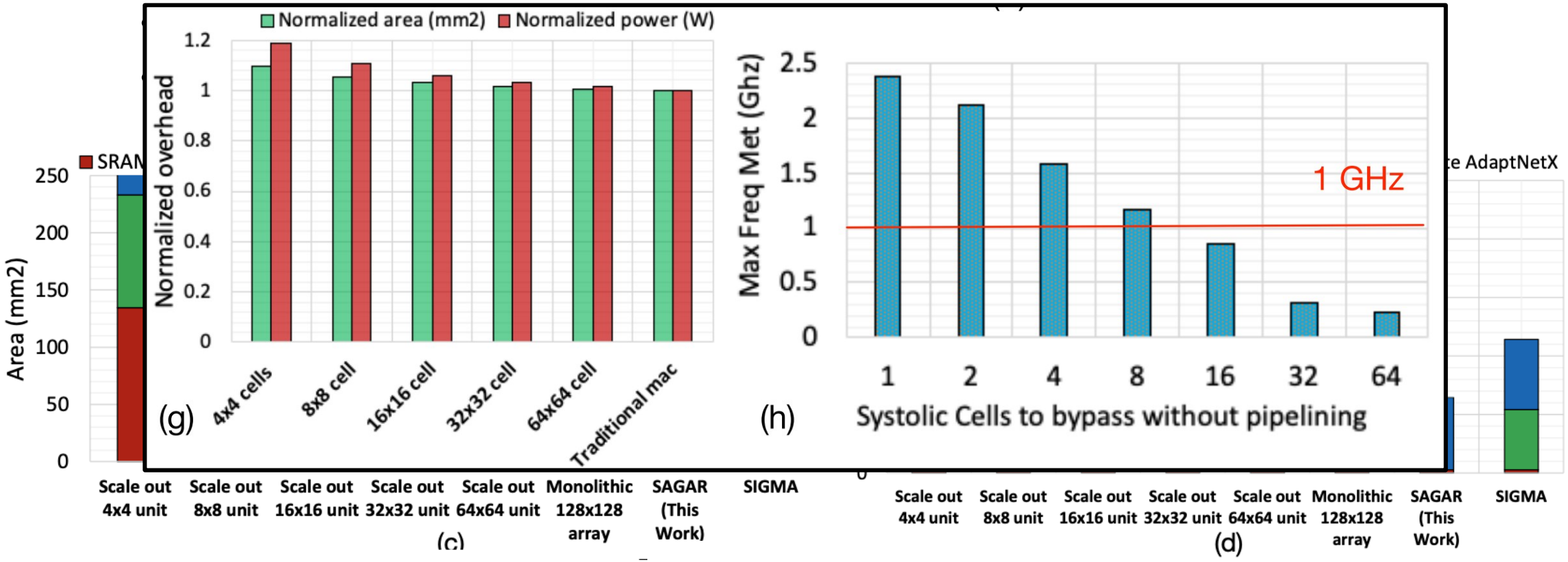
- Hardware cost analysis
  - Monolithic configuration (Best efficient in teams of area)
  - 50% more power than that of monolithic (3.5× expensive)





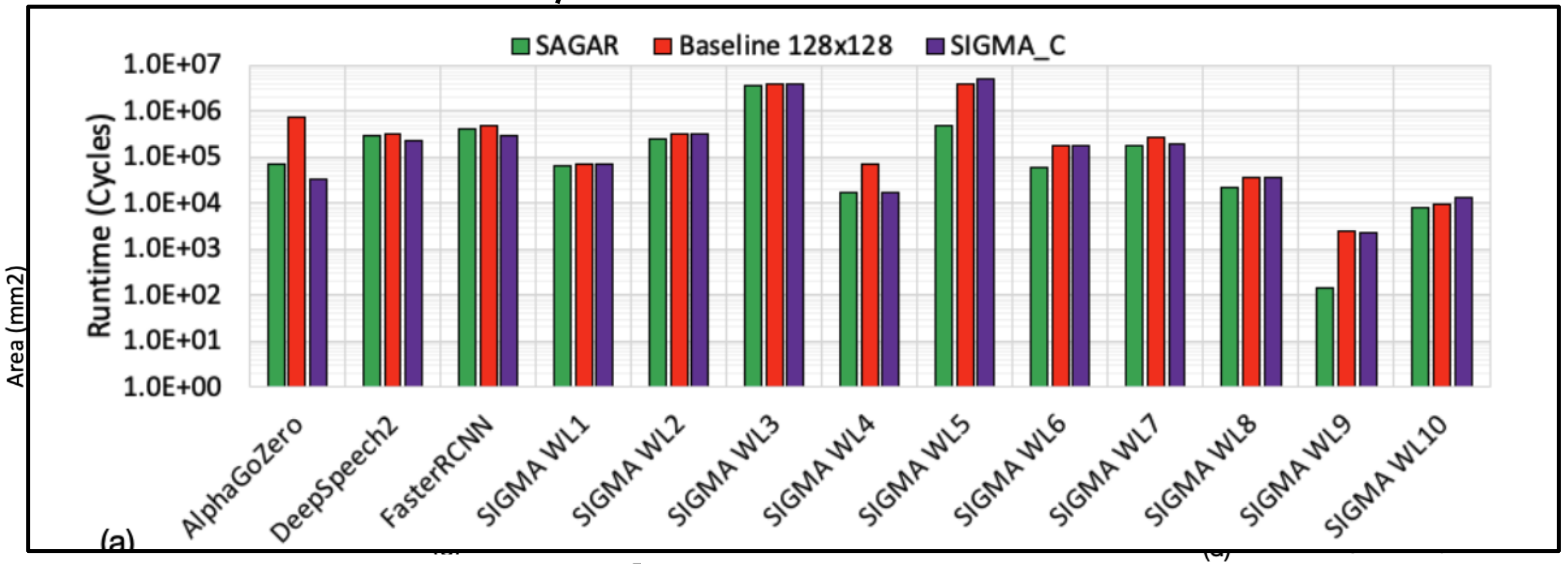
SAGAR evaluations

- Hardware cost analysis



SAGAR evaluations

- Hardware cost analysis



감사합니다