3D Long-Term Memories (LtRAMs) for Model-Resident Inference Architectures

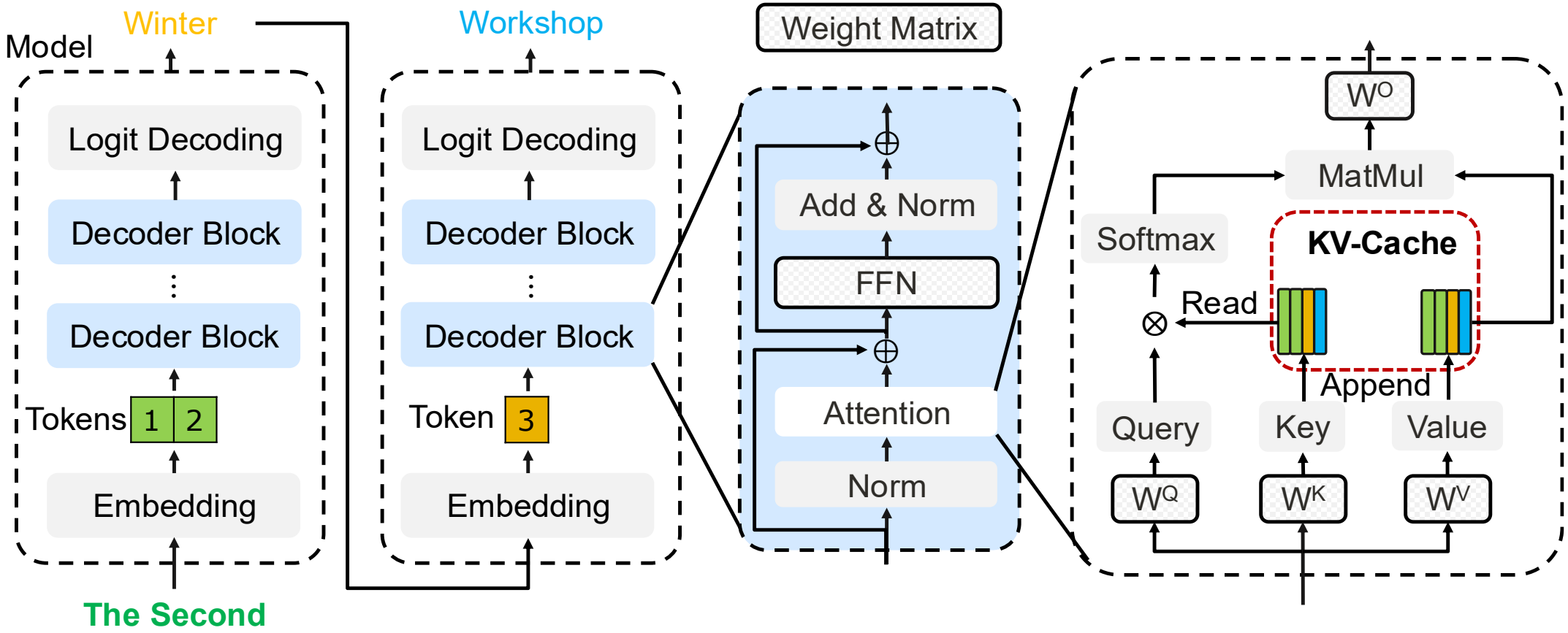
A Framework for Quantitative Analysis

Samuel A. Dayo[1], Colin H. Drewes[1] , Alex Aranburu*, Victor J.B. Jung[2], Shridhar Mukund*
Prof. Robert M. Radway[3], Prof. Christos Kozyrakis[1,4], Prof. Luca Benini[2], Prof. H.-S Philip Wong[1],
Prof. Subhasish Mitra[1]

[1]Stanford University [2]ETH Zurich [3]University of Pennsylvania [4]NVIDIA Research *EMD Electronics

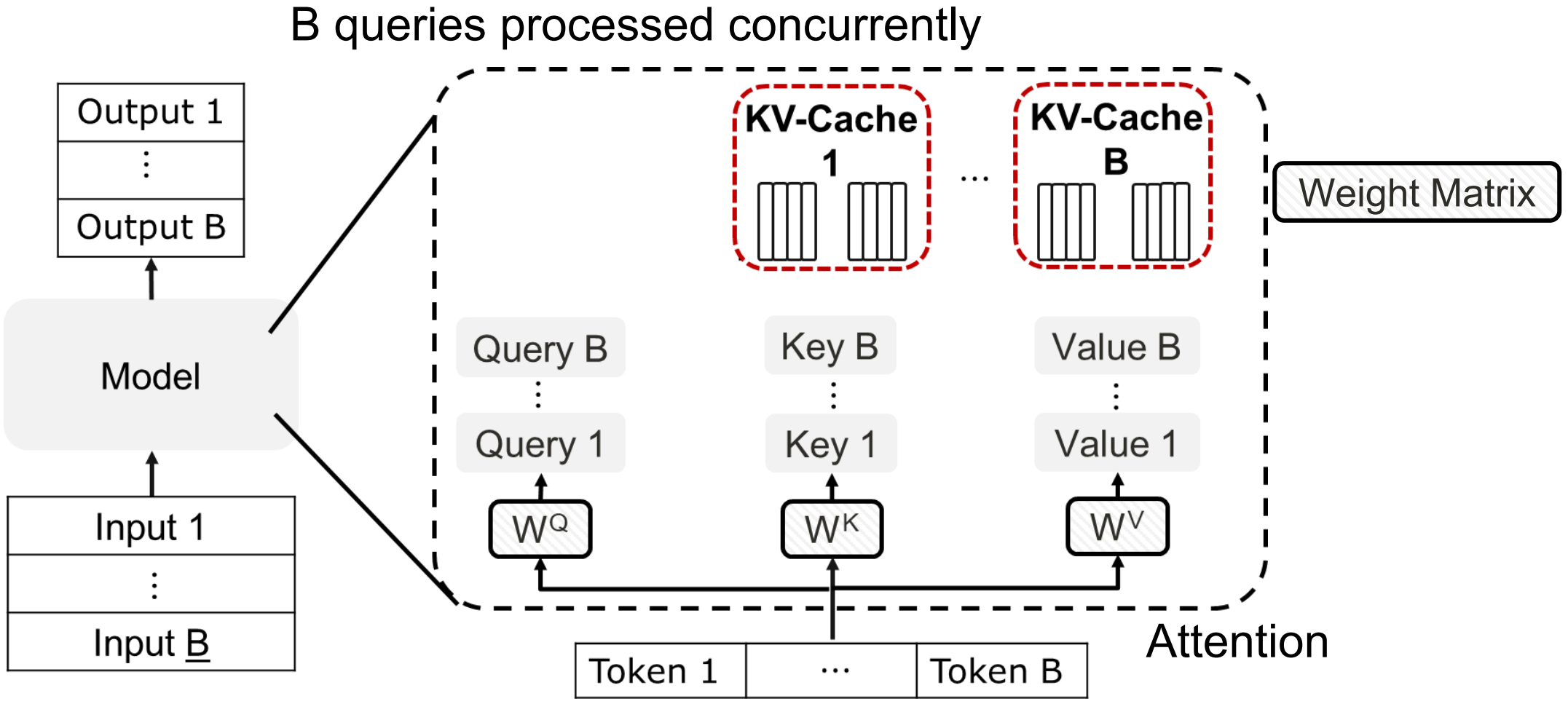
Background: LLM Inference

Weights and KV-Cache read every pass



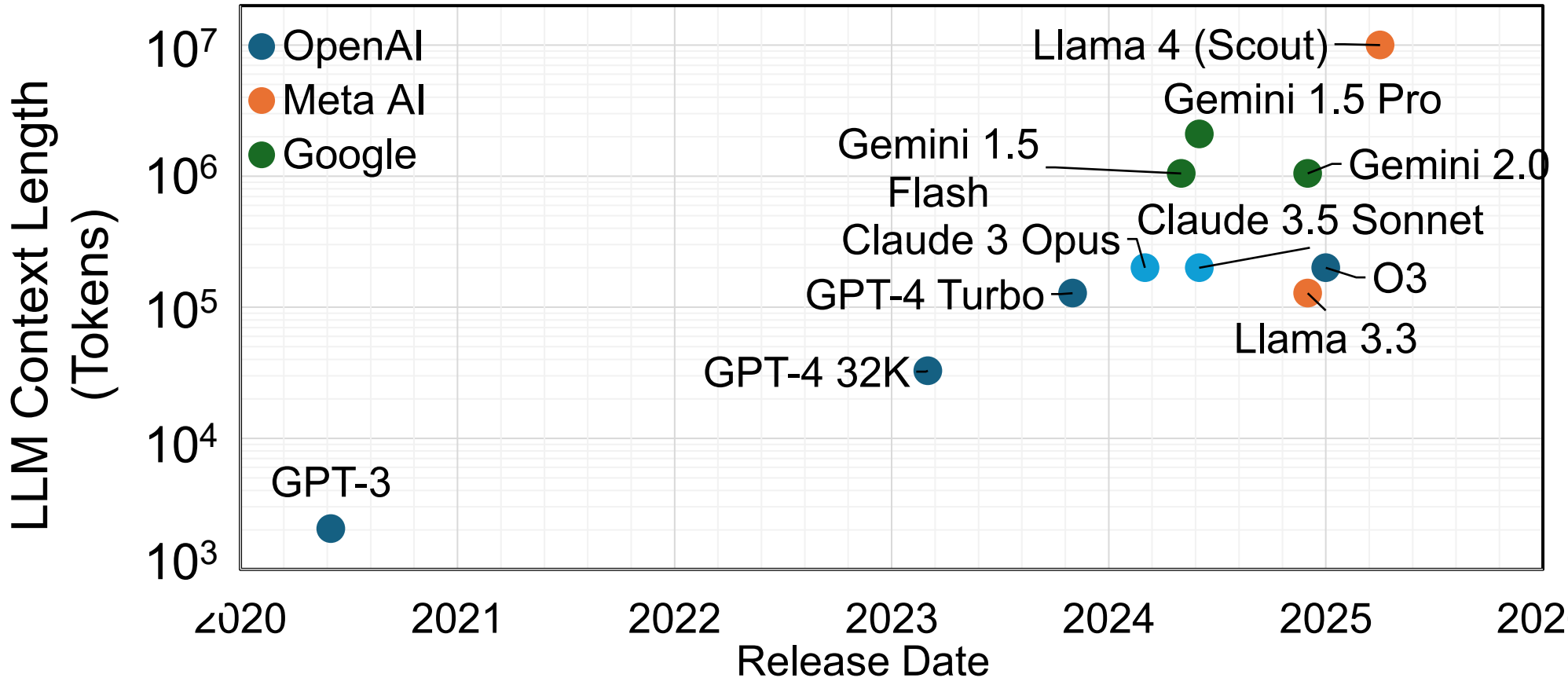
KV-Cache grows linearly with total tokens (context length)

Background: Batch Inference



Weight access but not KV$ access amortized

Background: Explosive Growth of Context Lengths



Larger context → more weight + KV-Cache reads

Memory needs: high-bandwidth, low-energy access

Model-Resident Architectures

Model-Streaming	Model-Resident
Weight/KV-cache materialized during execution	Weight/KV-cache allocated during model load
Continuous re-allocation of weight/KV-cache	Allocation persist through model inference
Memory is remote from compute node	Memory co-located with compute node

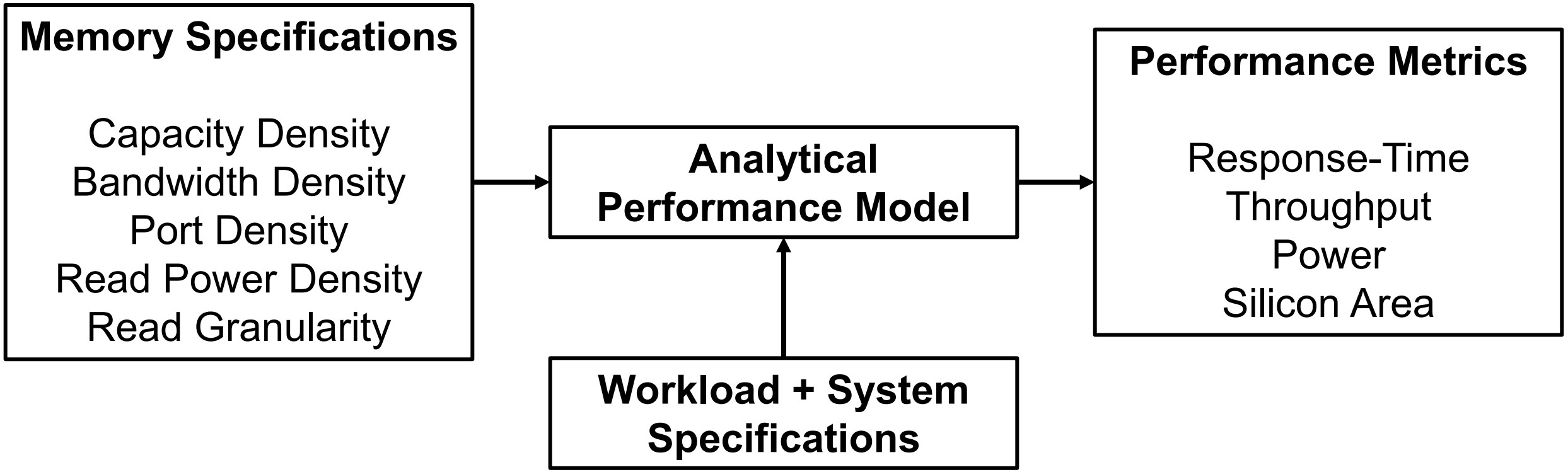
Minimize: Weight/KV-Cache data movement

Long-Term RAM (LtRAM)

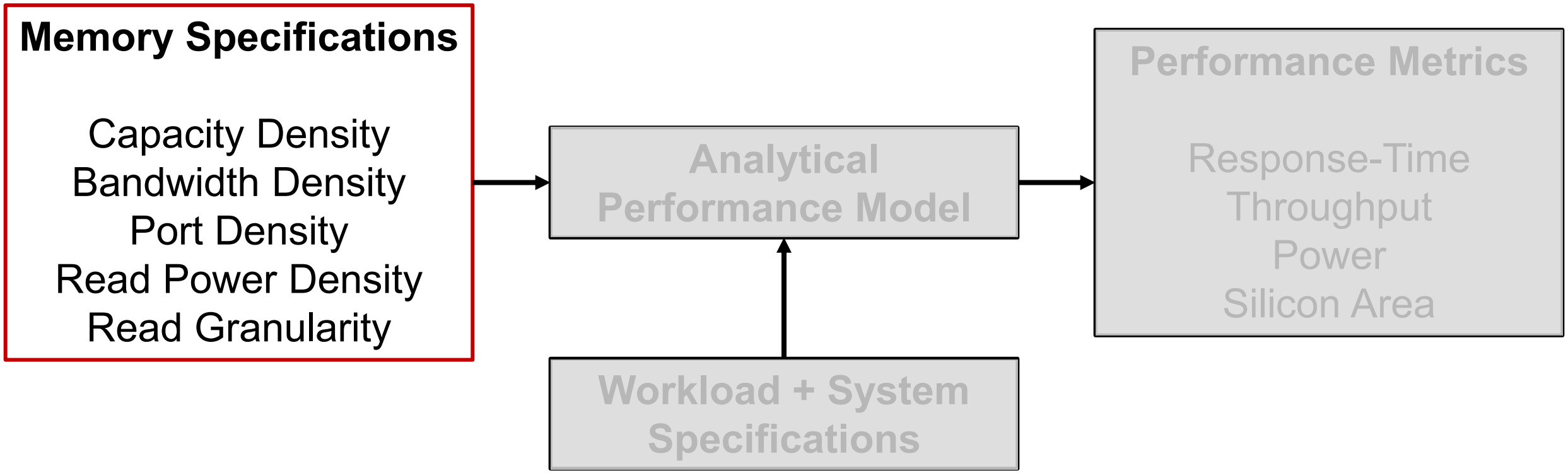
	Long-Term RAM	Short-Term RAM
Desired Properties	High density Low read energy	High density Low read/write energy
Acceptable Compromise	Higher write energy Higher write latency Limited write endurance	Retention Refresh overhead
Use Case	Read-mostly weights Read-mostly KV-Cache	Read/write caches Read/write scratchpads

Read-optimized memory weights and KV-Cache

Quantitative Performance Evaluation Framework

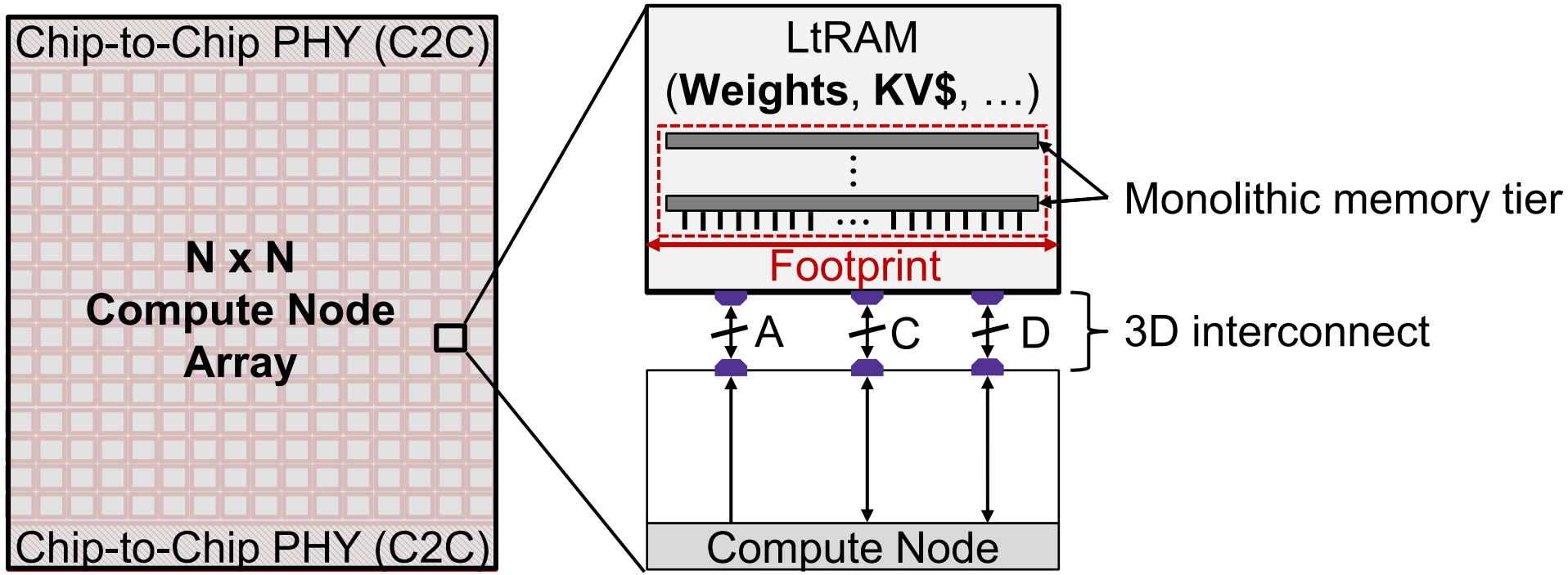


Quantitative Performance Evaluation Framework



Memory Capacity Density

Memory capacity per unit area (GB/mm^2)

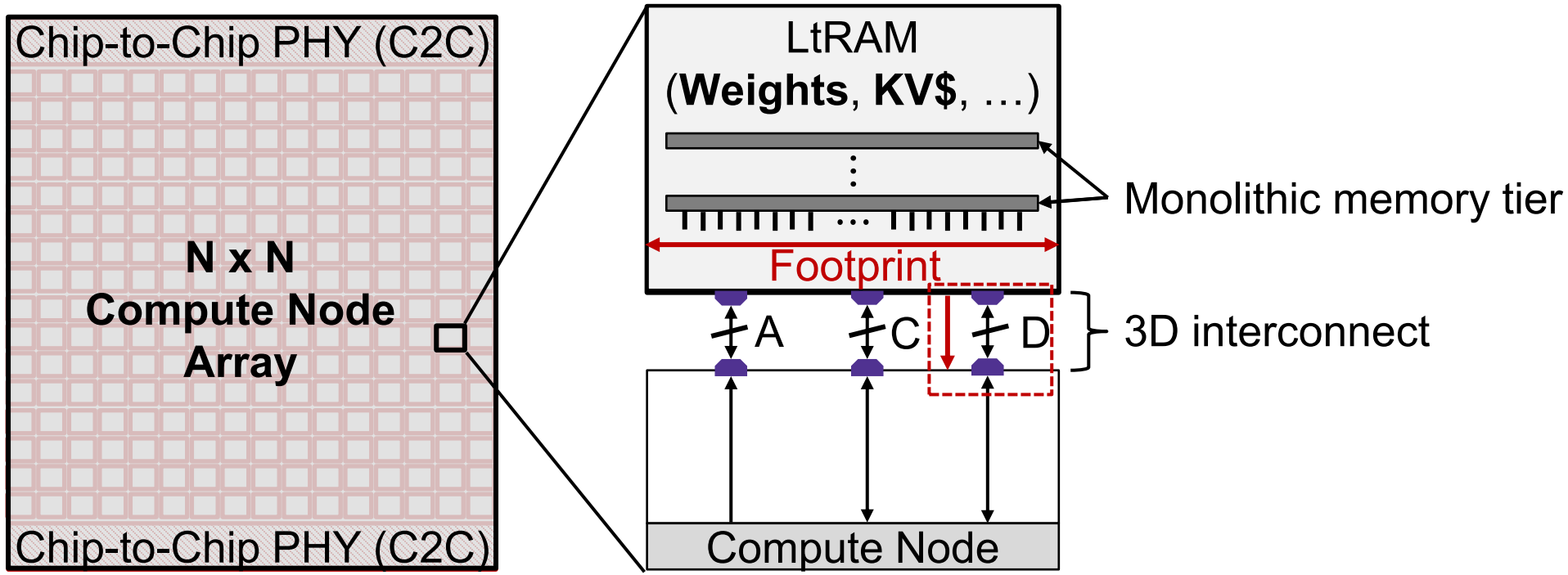


A: Address width; C: Control width; D: Data width

(Per-layer capacity × Monolithic Layers) / Footprint

Memory Bandwidth Density

Peak data transfer throughput per unit area (TB/s/mm^2)

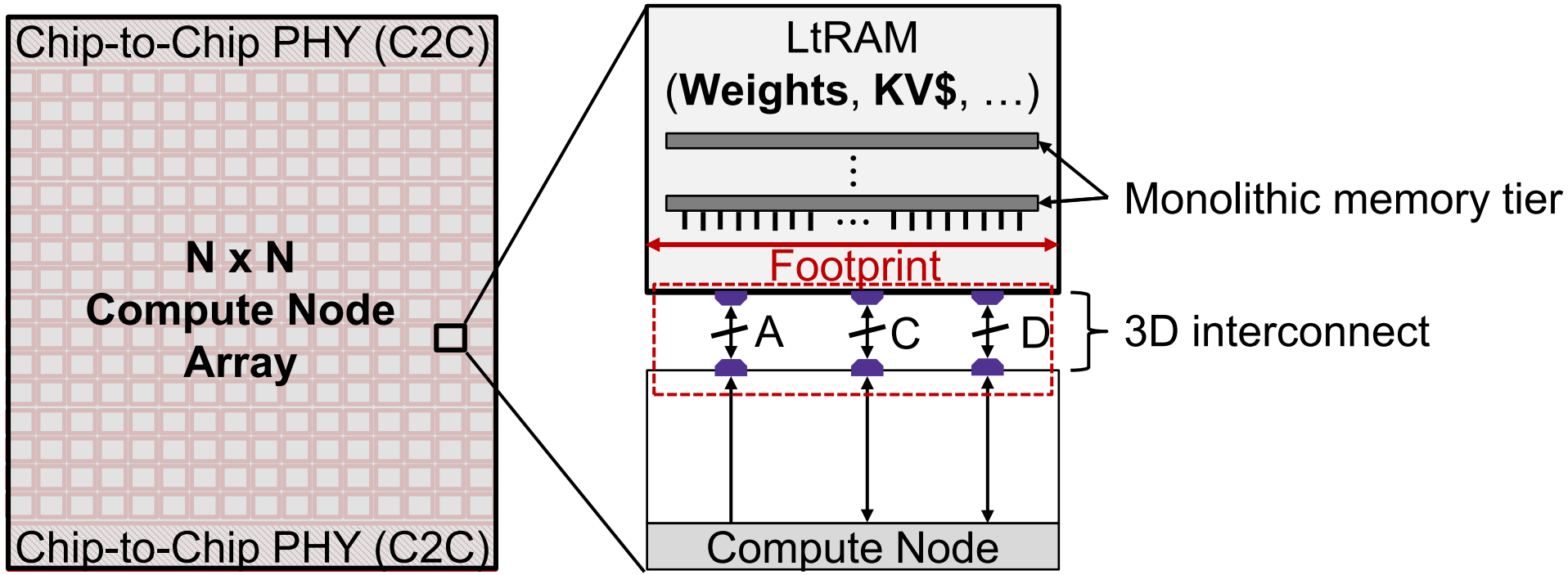


A: Address width; C: Control width; D: Data width

(Data width × clock frequency) / Footprint

Memory Port Density

Independent memory access ports per unit area (ports/mm^2)

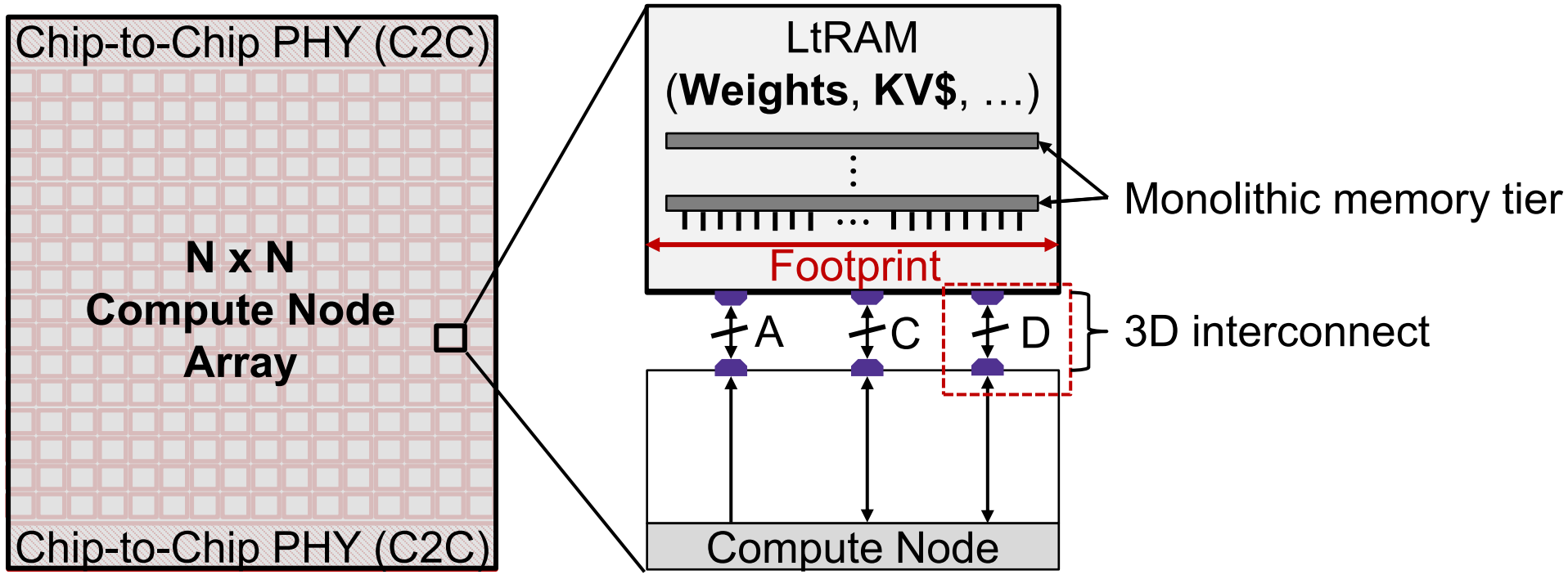


A: Address width; C: Control width; D: Data width

1/ Footprint

Memory Read Power Density

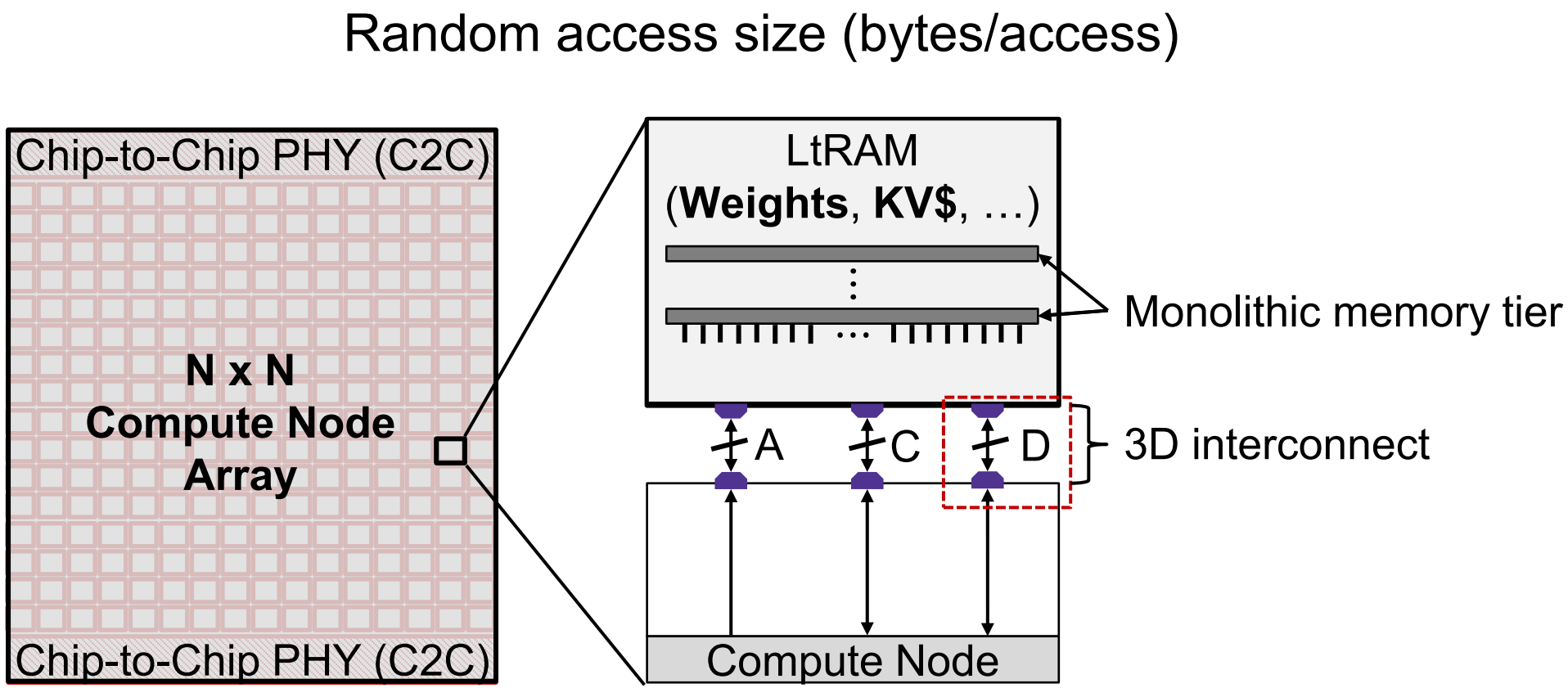
Memory read power per unit area (W/mm^2)



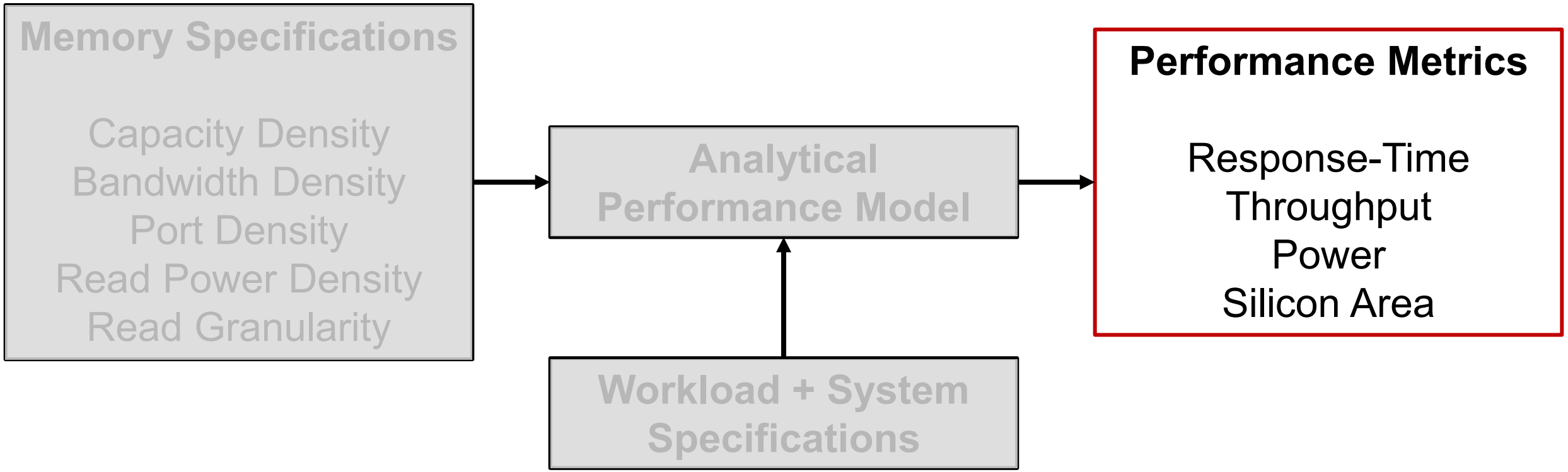
A: Address width; C: Control width; D: Data width

(Data width × clock frequency × energy-per-bit) / Footprint

Memory Read Granularity



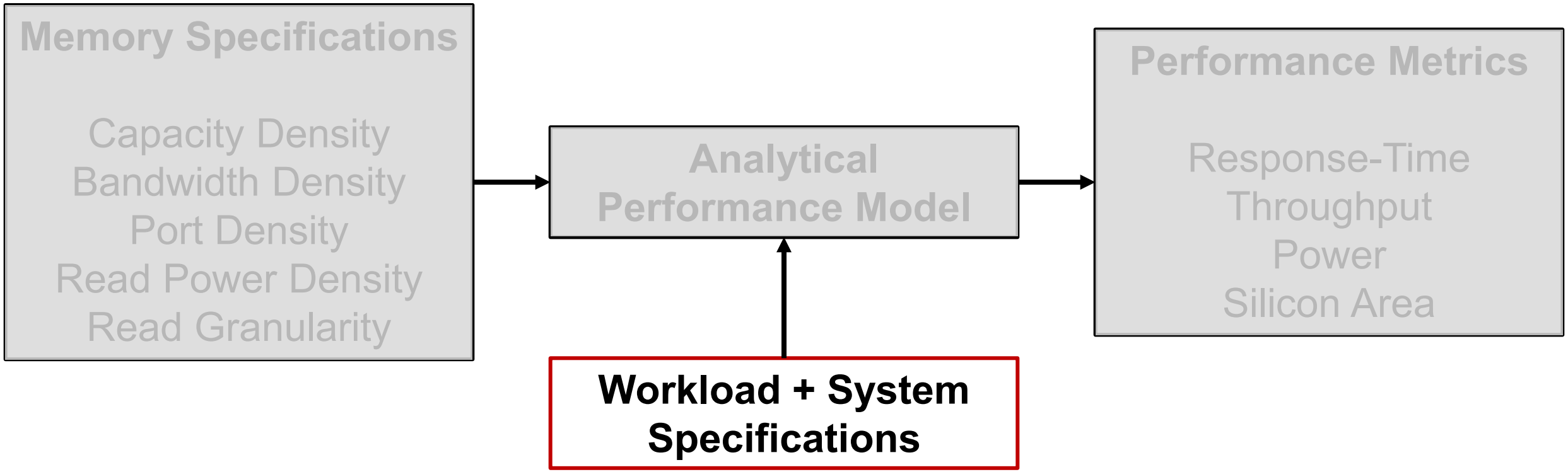
Quantitative Performance Evaluation Framework



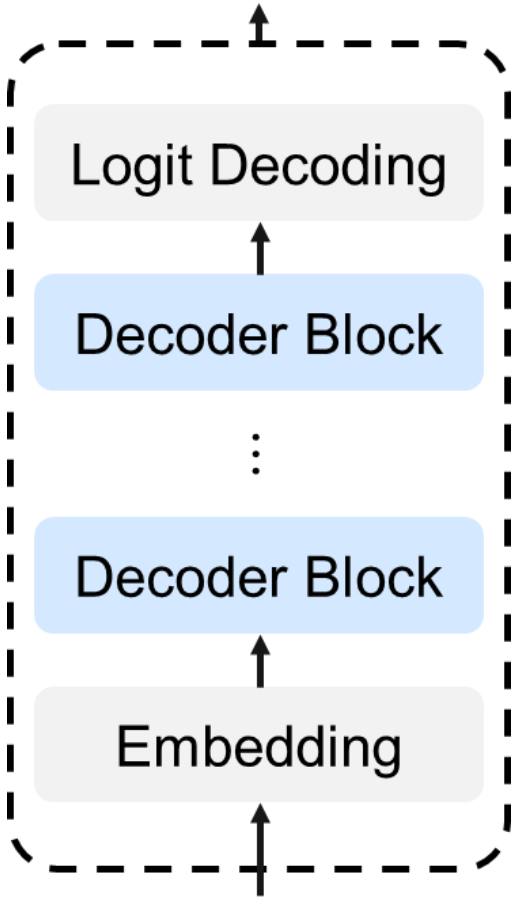
Review: System Performance Metrics

System Performance Metric	Definition	Better
User Query Response Time (s)	End-to-end latency per user query	↓
System Token Response Time (s)	Average time per generated token	↓
User Tokens Throughput (token/s)	Token generation rate per user	↑
System Tokens Throughput(token/s)	Aggregate token generation rate across all users	↑
System Power-Efficiency (token/s/watt)	Represents energy cost to generate token (OPEX)	↑
Silicon Area-Efficiency (token/s/mm^2)	Silicon area utilization efficiency (CAPEX)	↑

Quantitative Performance Evaluation Framework



Workload Parameterization: LLM Decode



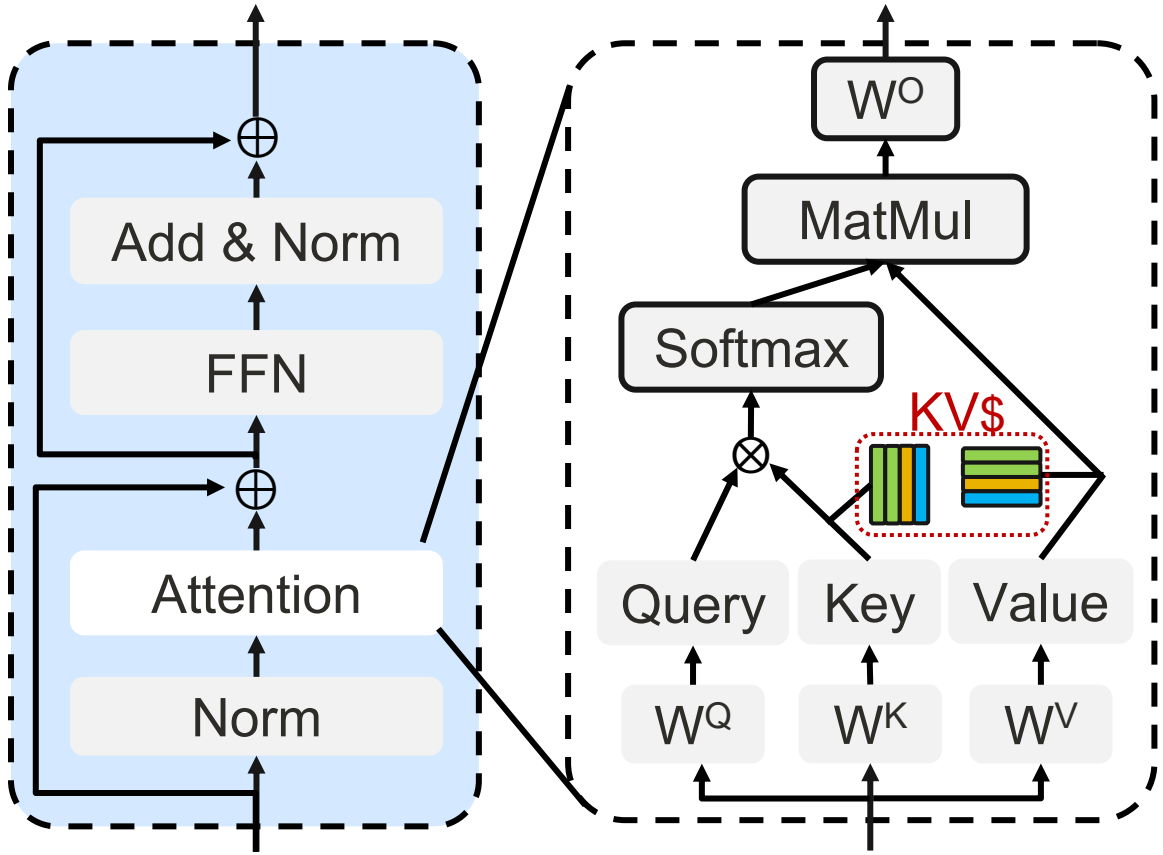
Component	Compute FLOPS(%)	Memory Capacity (%)
Embedding	0.48	1.28
Decoder Blocks	99.47	97.43
Output Logits	0.04	1.28

Dominates workload compute and memory capacity

Workload Parameterization

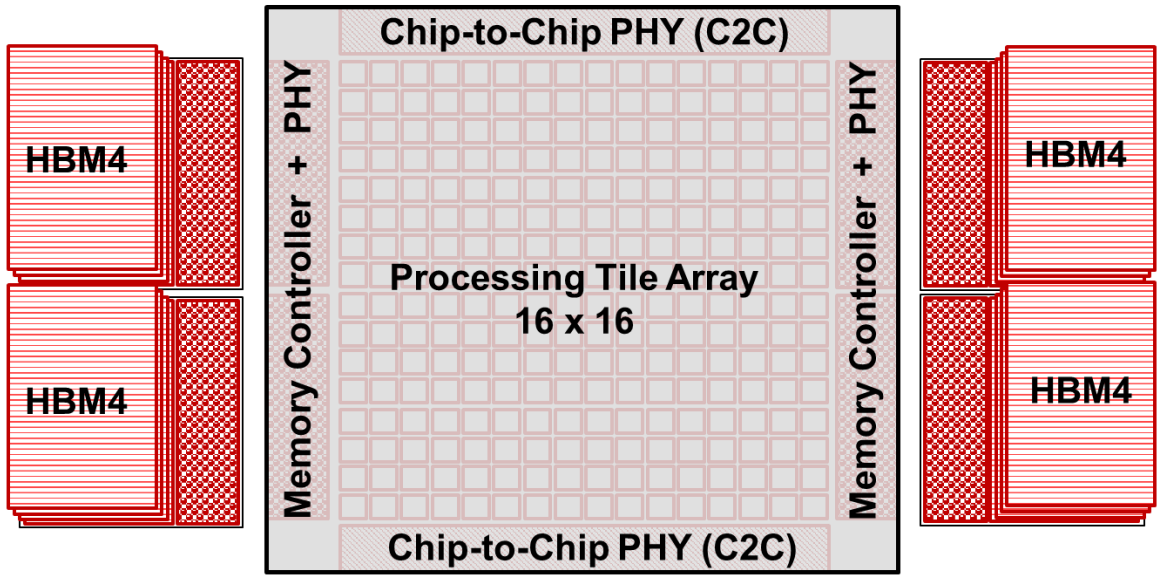
Focus only on performance significant components

Variables	Variable
Emb. Dimension	d_{model}
Head Dimension	d_h
# Query Heads	H_q
# KV Heads	H_{kv}
# Decoder Layers	L
# Experts	E
Top-K (Active Experts)	K
Expert Dimension	d_{ffn}
Batched Users	B
Context Length	C

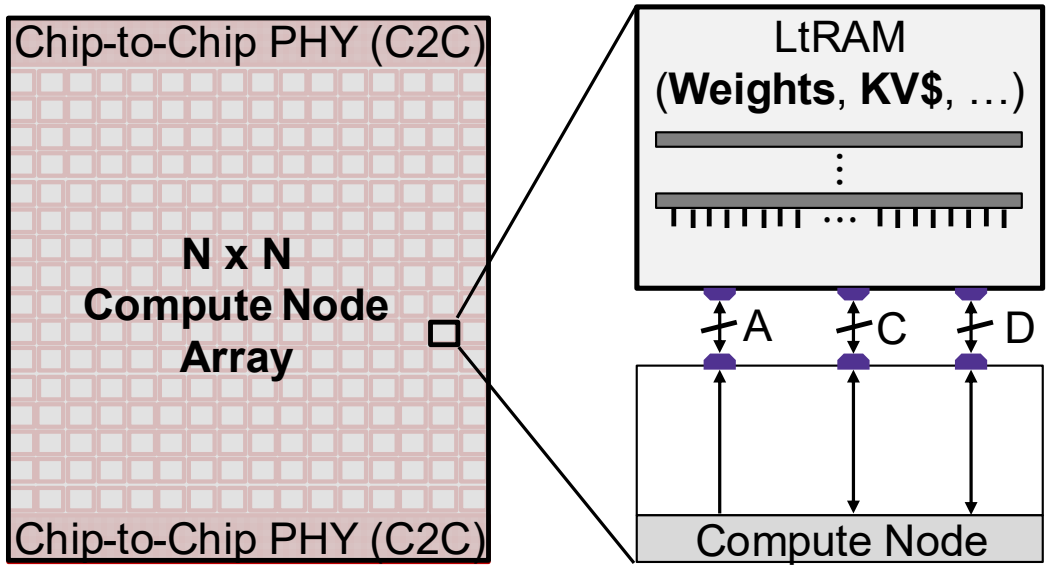


System Parameterization Assumptions

Baseline



Model-Resident



Tile-based spatiotemporal computation

2D Mesh NoC with dimension-ordered routing

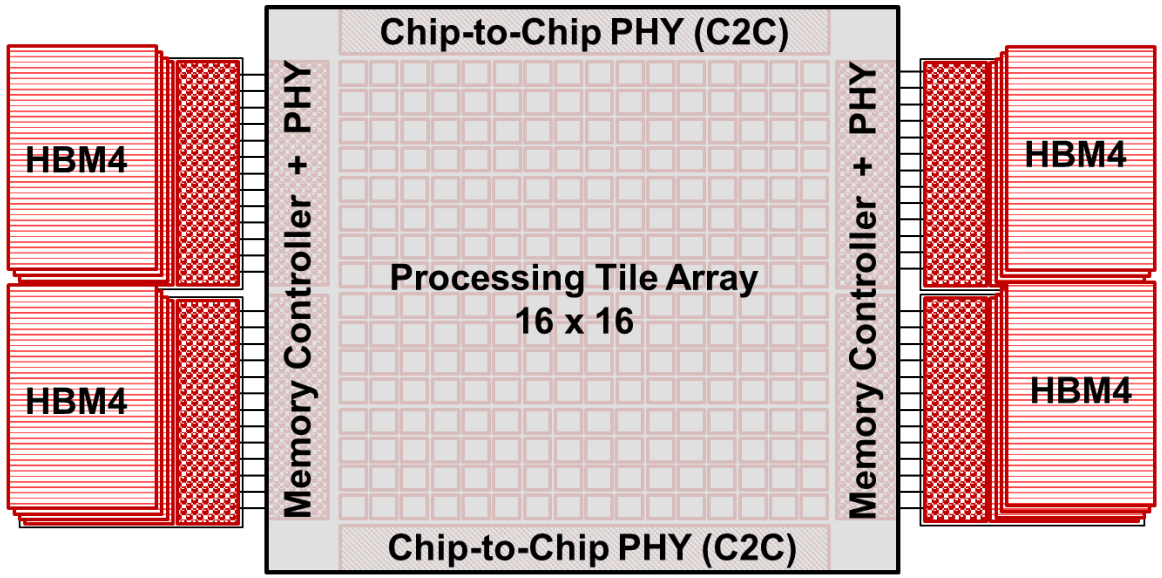
Inter-chip pipeline parallelism

Intra-chip tensor parallelism

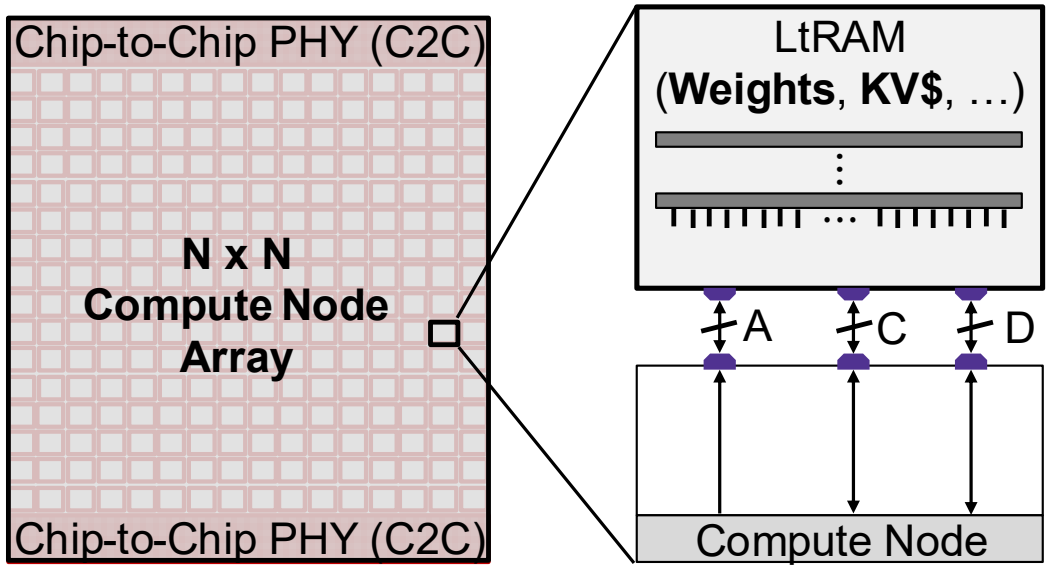
Several industry chips makers employ similar architecture (e.g., Tenstorrent, Cerebras, Meta, …)

System Parameters

Baseline



Model-Resident



Compute

Parameter	Unit
Area	mm^2
Energy	pJ/Op
Throughput	TFLOPS

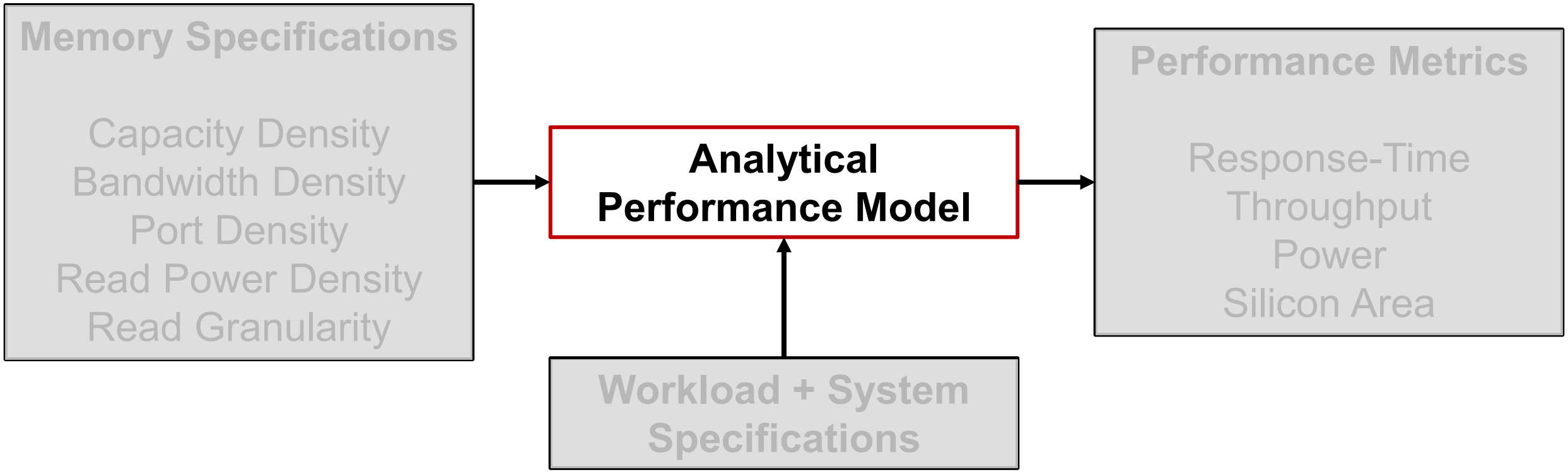
NoC

Parameter	Unit
Hop Latency	ps
Hop Energy	pJ/bit
Bandwidth	GB/s

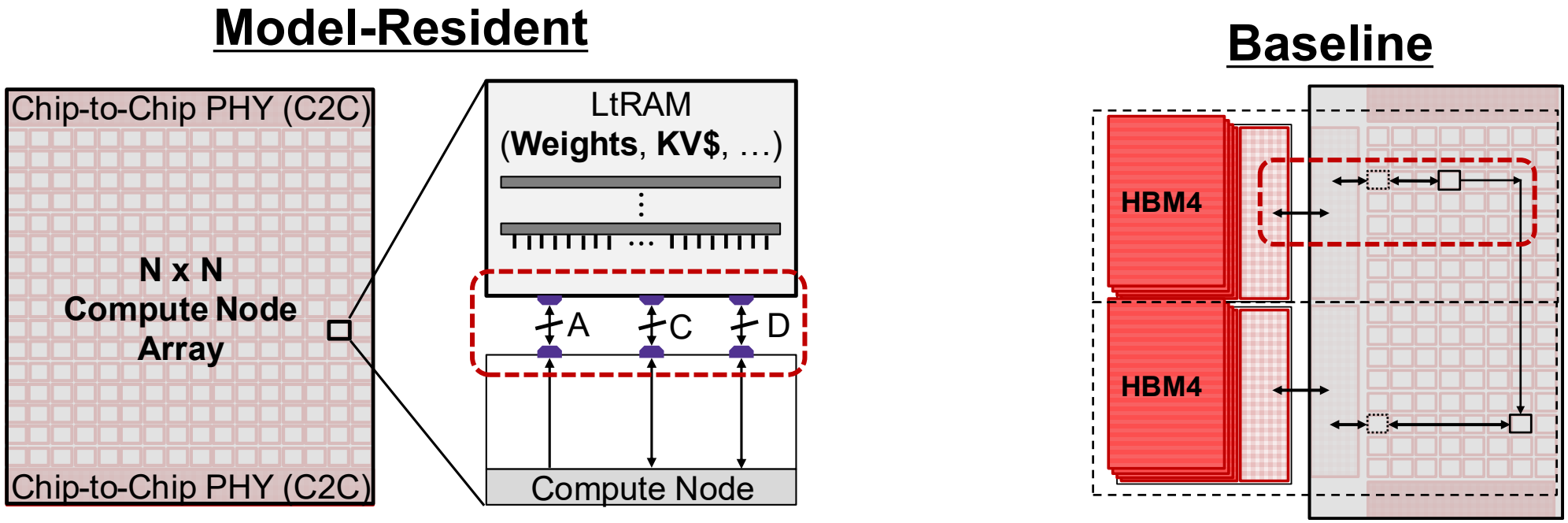
Chip-to-Chip

Parameter	Unit
Hop Latency	ps
Hop Energy	pJ/bit
Bandwidth	GB/s

Quantitative Performance Evaluation Framework



System Modeling: Memory Access Assumptions

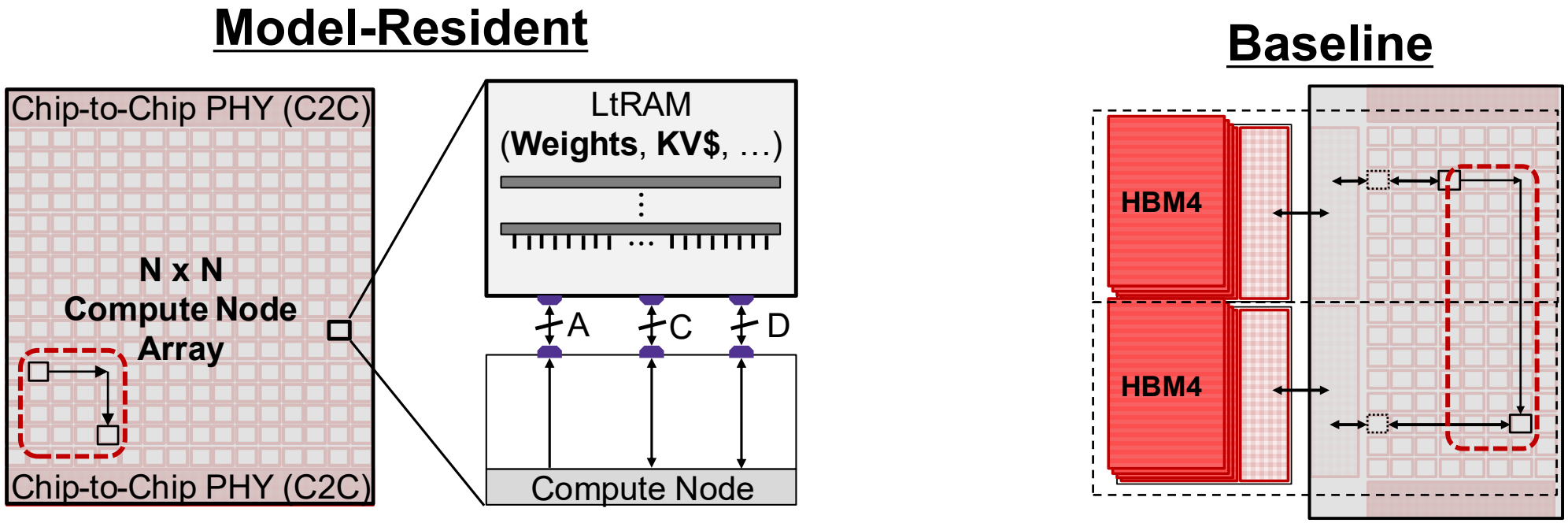


Alpha-beta access model: fixed + streaming

Fixed latency & energy cost: memory access + transport

Streaming latency & energy cost: memory + network access

System Modeling: Network-on-Chip Assumptions



Alpha-beta model of NoC transfers: fixed + streaming

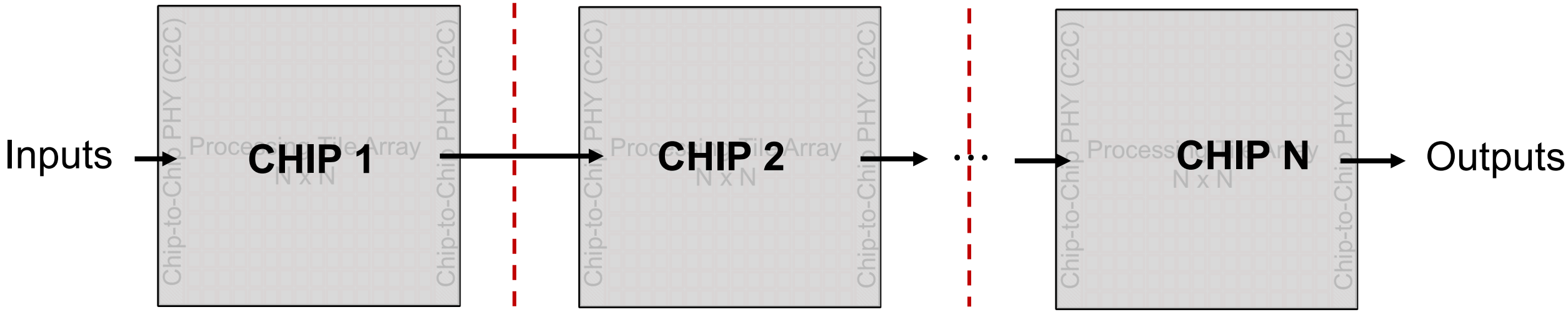
Fixed latency & energy cost: router + hop

Streaming latency & energy cost: NoC bisection

Average-hop approximation

NoC: Network-on-Chip; Model-resident compute node may include cache/register-file

System Modeling: Multi-Chip Parallelism



Inter-chip pipeline parallelism

Workload split by per-chip memory capacity

Integer number of layers per chip

Intra-chip tensor parallelism

Analytical Performance Model

- Layers processed as set of sequential stages
  - Input projection → Self-Attention → MoE
- Overlapped compute/ memory + exposed network synchronization
- Total latency & energy aggregated across stages & layers
  - $\mathrm{T}_{\mathrm{Total}} = \mathrm{Layers} \times \left(\sum_{\mathrm{stage}} \max \left(\mathrm{T}^{\mathrm{comp}}_{\mathrm{stage}}, \mathrm{T}^{\mathrm{mem}}_{\mathrm{stage}} \right) + \mathrm{T}^{\mathrm{net}}_{\mathrm{layer}} \right) + T_{C2C}$
  - $\mathrm{E}_{\mathrm{Tot}al} = \mathrm{Layers} \times \left(\sum_{\mathrm{stage}} \mathrm{E}^{\mathrm{comp}}_{\mathrm{stage}} + \mathrm{E}^{\mathrm{mem}}_{\mathrm{stage}} + \mathrm{E}^{\mathrm{net}}_{\mathrm{layer}} \right) + E_{C2C}$
- Validate vs. *LIMINAL*[1]

Framework Simplifications

- Single face-to-face bonded memory die
- Uniform per-layer execution
- Mean hop-count NoC approximations
- Deterministic MoE routing
- Simplified roofline compute model

Capture dominant **first-order** architectural trade-offs

Case Study: Memory Assumptions

	Capacity Density (GB/mm^2)	**Bandwidth Density** (TB/s/mm^2)	**Port Density** (ports/mm^2)	**Read Granularity** (bytes/access)	**Read Power Density** (W/mm^2)
HBM4[1]	2.56	0.01	0.16	8	0.017
6T Static[2] (SRAM)	0.05	0.35	12.3	256	0.035
OS 1T1C[3] (TSMC)	0.07	4.10	15.9	256	2.050
FeNOR[4] (EMD)	0.32	2.32	1250.0	4	0.023

[1] HBM4 JESD270-4A; [2] DESTINY simulation (512 KB RAM macro)
[3] TSMC. [Chiang, Symp. VLSI Circuits 25]; [4] EMD Electronics Projections

Case Study: System Parameter Assumptions

System Parameter		Value
NoC[1]	Bandwidth	$128\ \mathrm{GB/s}$
	Latency	$500\ \mathrm{ps}$
	Energy	$0.1\ \mathrm{pJ/bit}$
C2C[2]	Bandwidth	$320\ \mathrm{GB/s}$
	Latency	$4\ \mathrm{ns}$
	Energy	$1.17\ \mathrm{pJ/bit}$
Compute	Area	$800\ \mathrm{mm}^2$
	Energy[3]	$0.160\ \mathrm{pJ/op}$
	Throughput[4]	$2250\ \mathrm{TFLOPS/s}$

[1] Orenes-Vera *et. al*, “Massive Data-Centric Parallelism in the Chiplet Era.”
[2] Open Compute Group, “Bunch of wires PHY specification.”
[5] B. Dally *et. al*, “On the model of computation.”
[6] M. Davies *et.al, LIMINAL.* [NVIDIA Research].

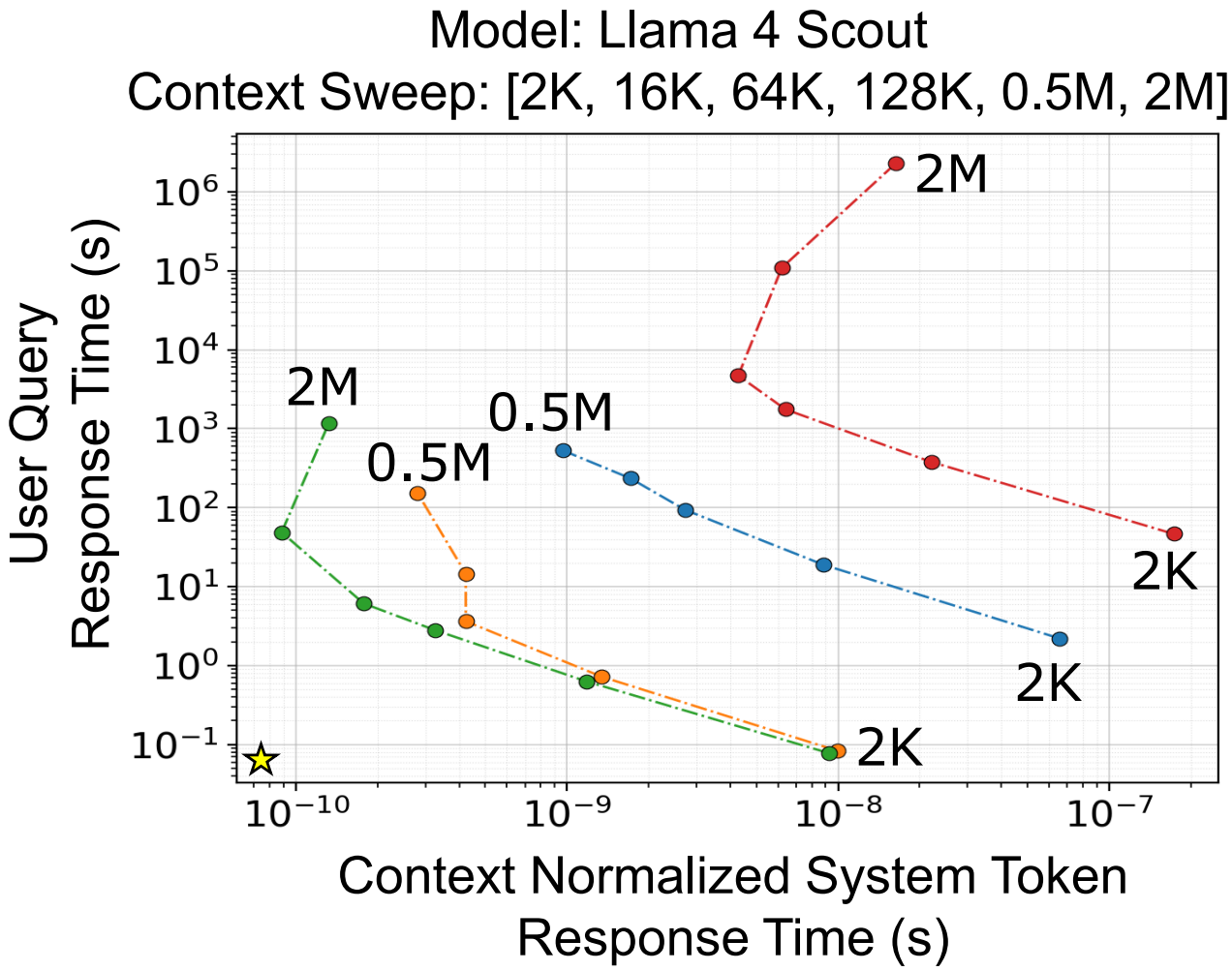
Case Study: Representative Workload Assumptions

Workload Variable	Llama 4 Scout	GPT-OSS-120B	Llama 3 405B
d_{model}	4096	4096	16384
d_h	128	128	128
H_q	32	32	128
H_{kv}	8	8	8
L	32	36	126
E	16	128	1
K	2	4	–
d_{ffn}	14336	14336	53248

Review: System Performance Metrics

System Performance Metric	Definition	Better
User Query Response Time (s)	End-to-end latency per user query	↓
System Token Response Time (s)	Average time per generated token	↓
User Tokens Throughput (token/s)	Token generation rate per user	↑
System Tokens Throughput(token/s)	Aggregate token generation rate across all users	↑
System Power-Efficiency (token/s/watt)	Represents energy cost to generate token (OPEX)	↑
Silicon Area-Efficiency (token/s/mm^2)	Silicon area utilization efficiency (CAPEX)	↑

Case-Study: Response-Time Performance



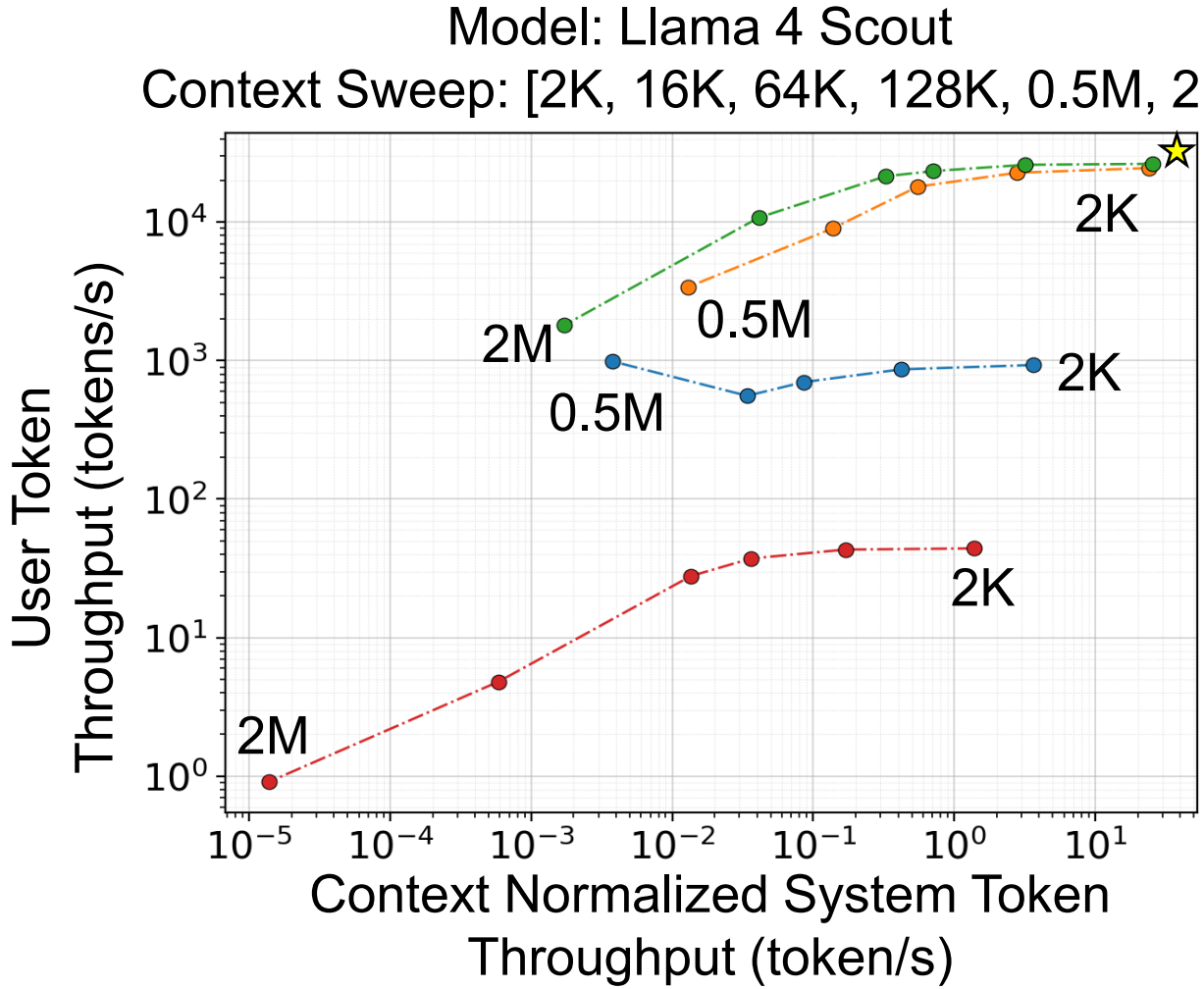
Batch Sweep: B = 1 to 128

- Static (optimal B = 8)
- OS 1T1C (optimal B = 2)
- FeNOR (optimal B = 2)
- HBM4 (optimal B = 64)

★ Preferred Corner

Model-residency shifts latency limit to longer context

Case-Study: Throughput Performance



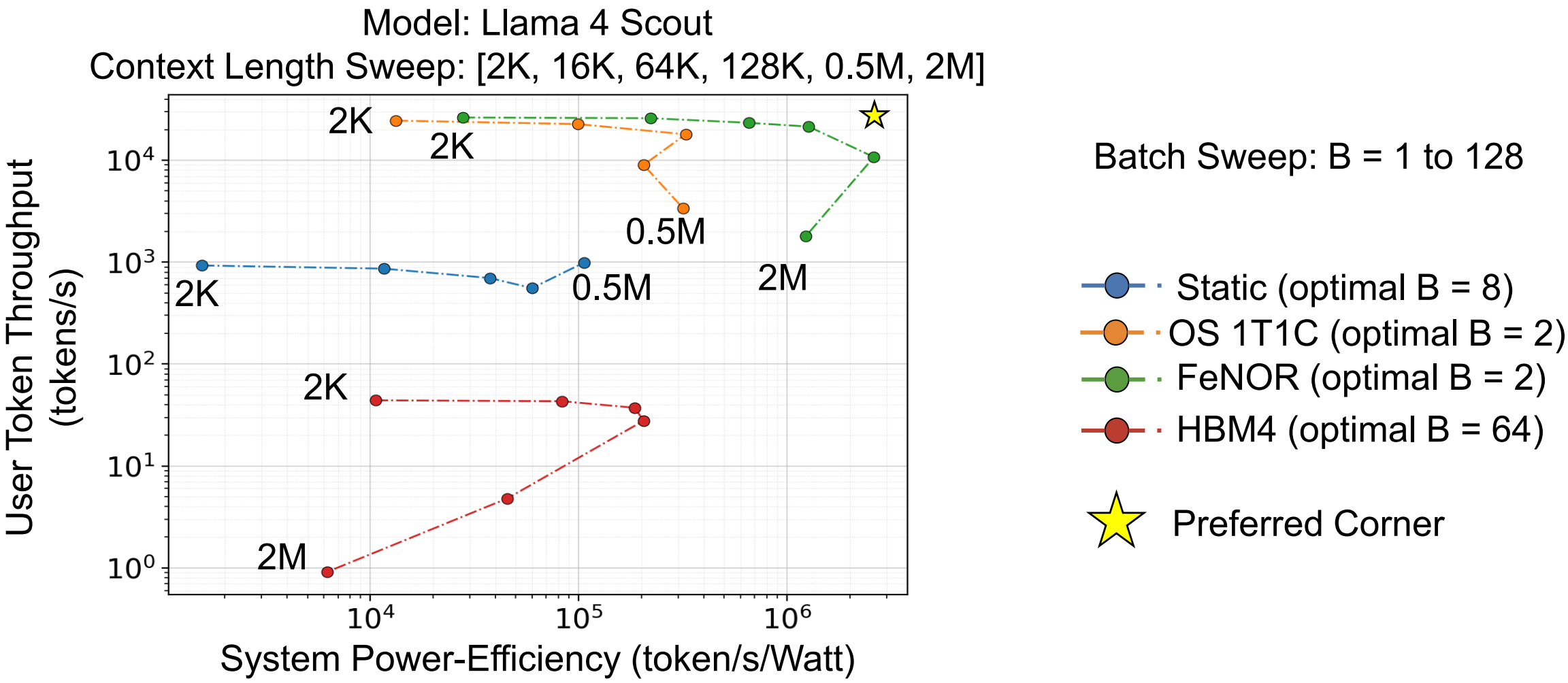
Batch Sweep: B = 1 to 128

- Static (optimal B = 8)
- OS 1T1C (optimal B = 2)
- FeNOR (optimal B = 2)
- HBM4 (optimal B = 64)

★ Preferred Corner

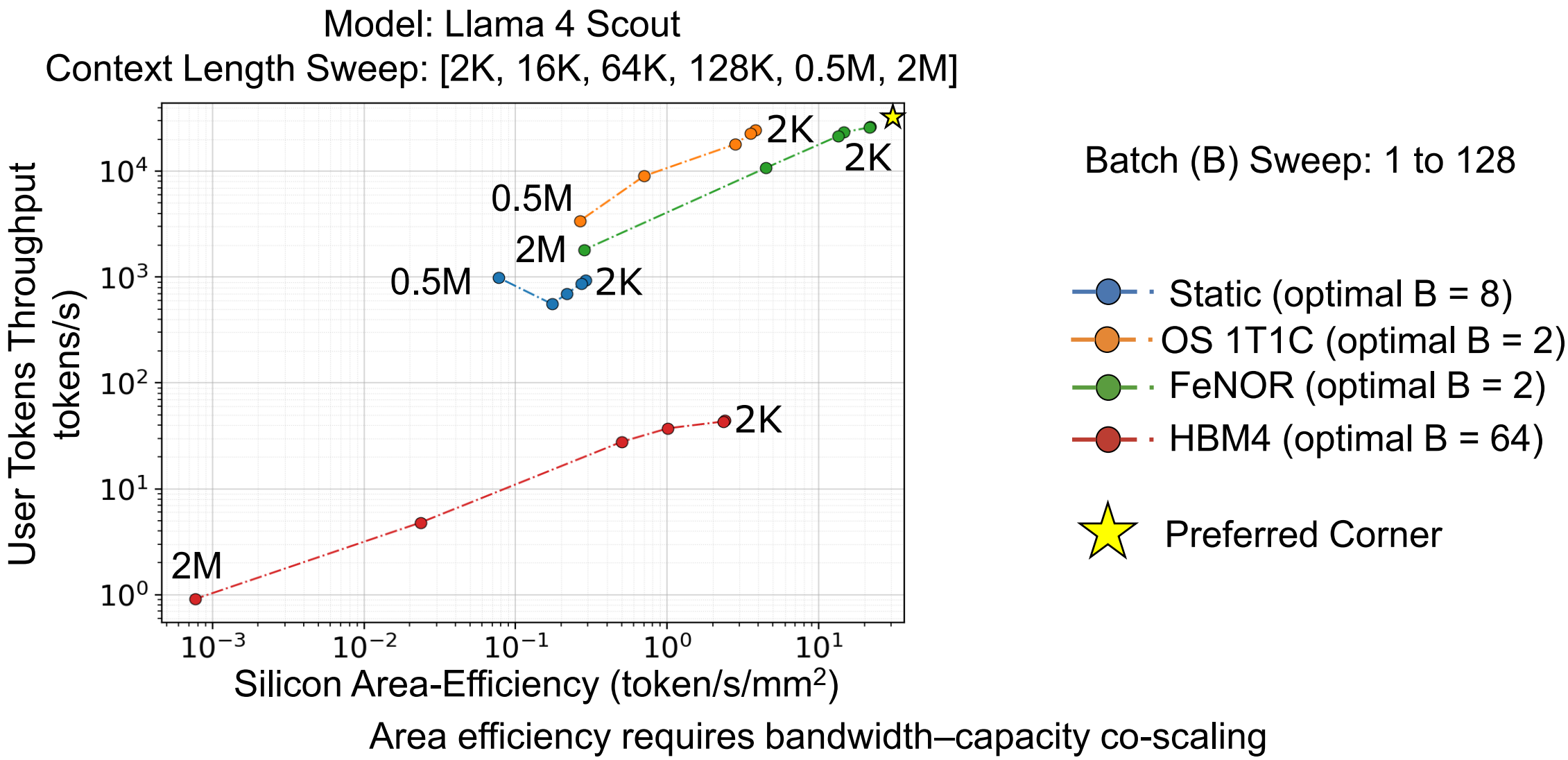
Bandwidth and capacity density determine usable throughput

Case-Study: Power Performance



Must co-optimize energy-efficiency with bandwidth density

Case-Study: Silicon-Area Performance



Area efficiency requires bandwidth–capacity co-scaling

Conclusion

Take Away:

- ❑ Model-resident architecture key for AI inference
- ❑ Requires high-density and high read bandwidth

Future Work:

- ❑ Multi-die-stacking
- ❑ Workload modeling (parallelism, routing skew)
- ❑ Compute & NoC micro-architecture detail
- ❑ Publish open-source

Call for Action:

- ➢ Memory makers to provide high-density, high read bandwidth LtRAMs