



DeepStack

Facilitating Co-Design Exploration of 3D DRAM-Stacked Accelerators for Distributed LLM Inference

Zhiwen Mo¹, Guoyu Li¹, Hao (Mark) Chen¹, Yu Cheng², Zhengju Tang², Qianzhou Wang¹, Lei Wang², Shuang Liang¹, Lingxiao Ma³, Yuxiao Guo³, Wayne Luk¹, Jilong Xue³, Hongxiang Fan¹

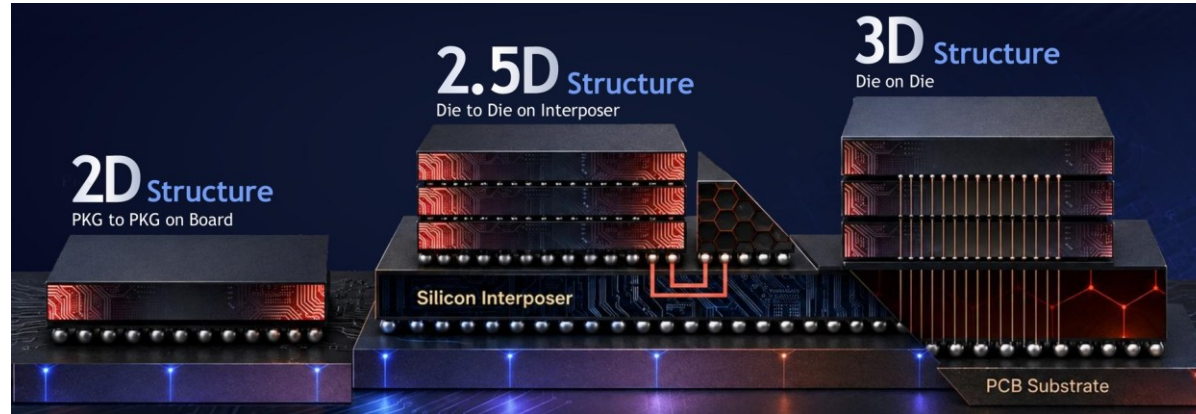
¹ Imperial College London ² Peking University ³ Tile-AI

Imperial College
London

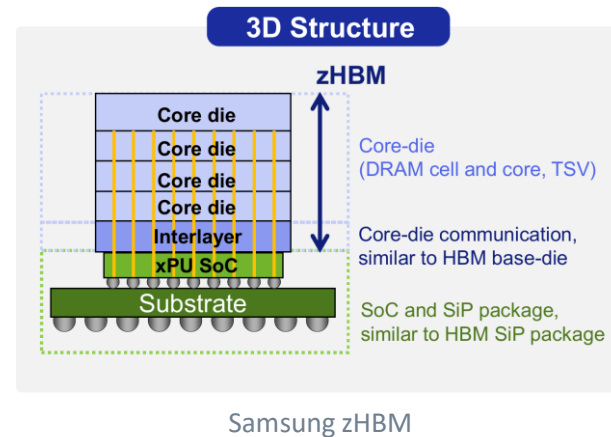
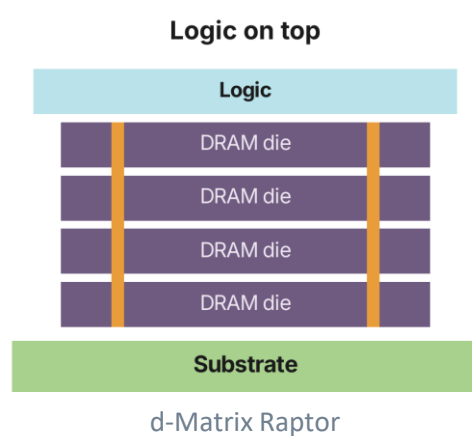


Increasing industry attention to 3D DRAM stacking

Motivation · Hot Chips 2026 and beyond



SK hynix: memory packaging moves 2D → 2.5D → 3D (die on die)



d-Matrix · Raptor

3D DRAM bonded under N4 logic; 32 GB, ~100 TB/s per card

Qualcomm · AI250

compute die under stacked DRAM; planned for 2027

Cerebras · CS-6

DRAM wafer on the compute wafer (roadmap)

Samsung · zHBM

DRAM directly on the xPU; ~2.3× BW, ~100 W less vs. HBM4E

SK hynix × NVIDIA

HBM stacked directly on GPU logic (reported R&D)

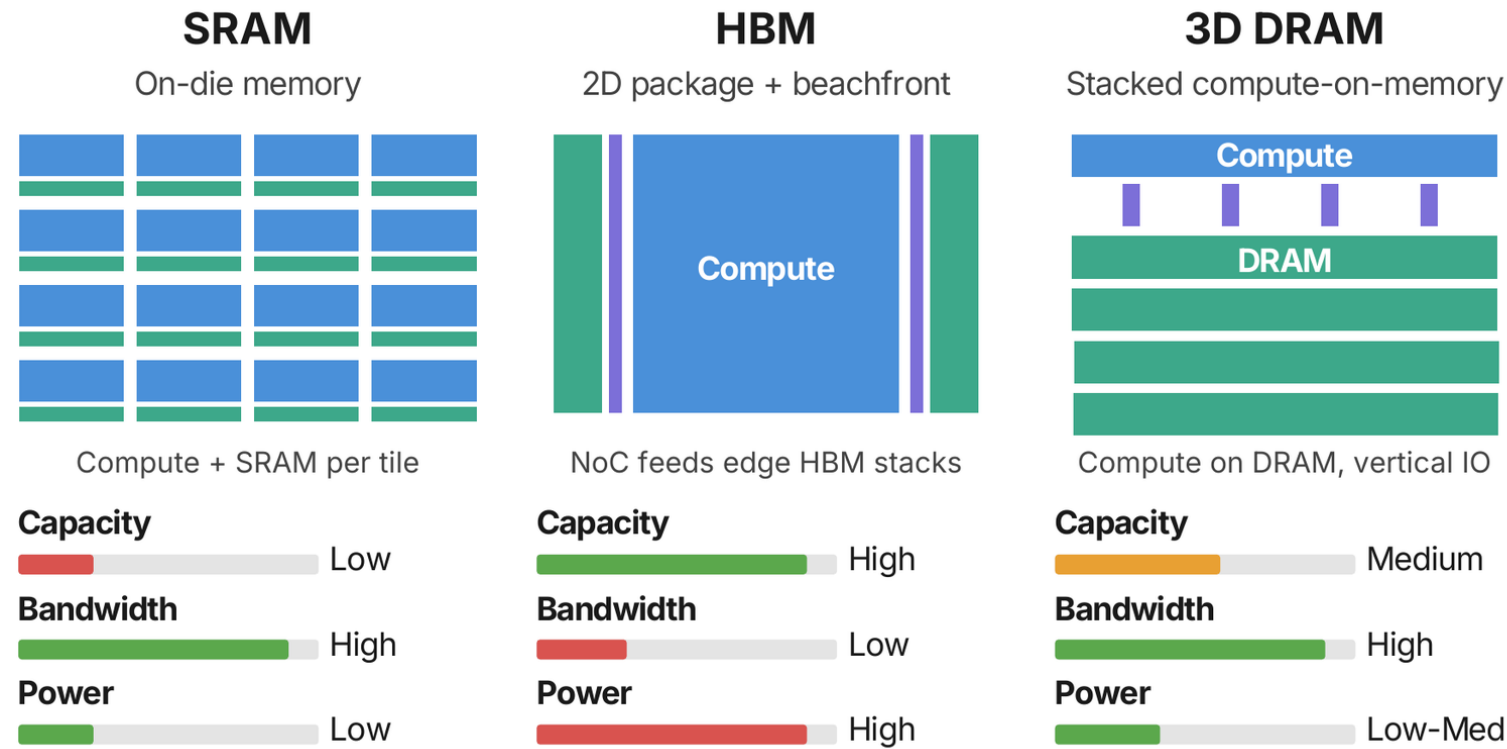
Academia

Stratum, H²-LLM, STCO, ... mostly single-chip / single-node

3D DRAM-on-logic is moving from research to products, mostly for LLM decode

3D DRAM: high bandwidth close to compute

Motivation · bandwidth and energy per bit



The energy ladder

Memories & interconnect	Energy / bit
SRAM (on-die)	~50 fJ
On-chip wire	~35 fJ/mm
3D vertical IO	0.3–0.4 pJ
Interposer trace	~500 fJ/mm
2.5D HBM4 (system)	2.5pJ + 3pJ (on chip)

3D IO lands **~10× below HBM** — PHY-less, mm-scale vertical path instead of cm-scale interposer + PHY

Figure: d-Matrix, Hot Chips 2026 (vendor figures, not DeepStack results)

- Decode is memory-bound: SRAM-class bandwidth at ~1/10 of HBM's energy per bit → faster, cheaper decode
- But heat limits stacking depth (Raptor: one DRAM layer, ≤ 0.5 W/mm²)

Model footprints exceed single-chip capacity

Motivation · how many devices just to hold the weights?

- DeepSeek-V3: **671 GB** of FP8 weights + a KV cache that grows with context and batch

Model weights

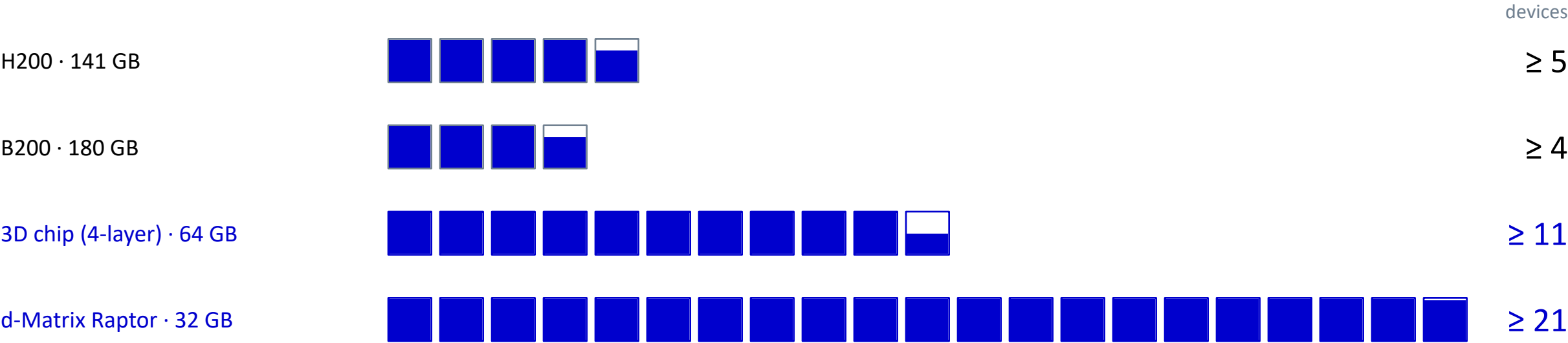


671 GB, one copy

Model footprints exceed single-chip capacity

Motivation · how many devices just to hold the weights?

■ DeepSeek-V3: **671 GB** of FP8 weights + a KV cache that grows with context and batch



Each square = one device, filled by weights. 3D chip: 4-layer configuration from our evaluation (Table I); Raptor: per card, Hot Chips 2026. KV cache not included.

Even with stacked DRAM, distributed inference is not merely beneficial but essential
(d-Matrix: 72 Raptor cards, 2.3 TB, for one frontier model at 1M context)

Compound parallelism strategies

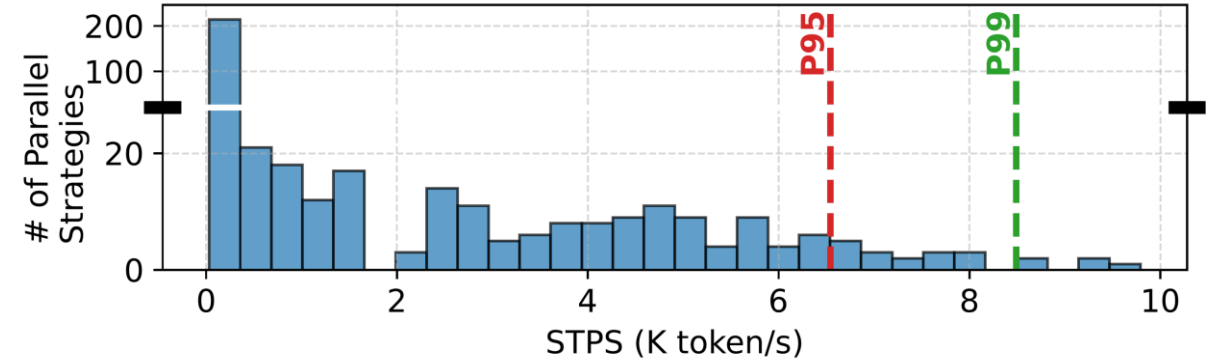
Motivation · distributed inference at rack scale



d-Matrix Raptor: chiplet $\rightarrow$ card $\rightarrow$ tray $\rightarrow$ 72-card rack (Hot Chips 2026)

Cerebras CS-4 links 3 wafers; Qualcomm AI250 and SambaNova SN50 pair decode chips with GPUs for prefill

One model, one chip design: STPS across parallel strategies



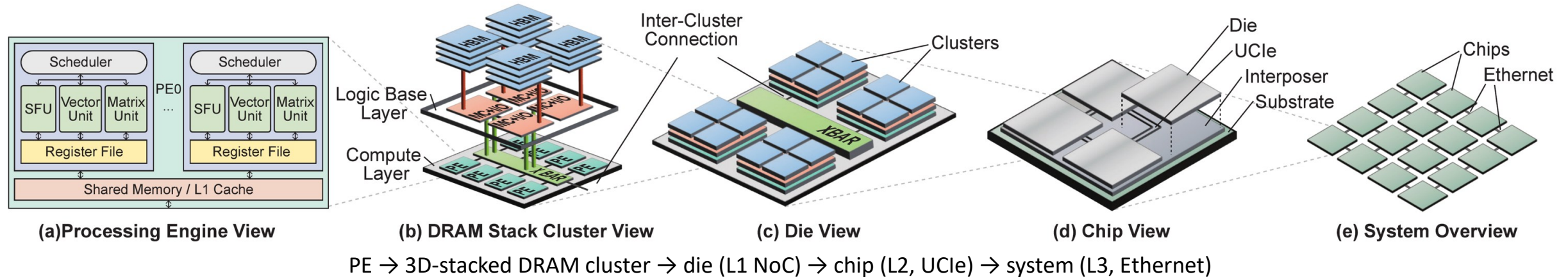
wide spread; high-performance strategies are rare

Existing frameworks: simplified cost models,
or parallelism restricted to TP / PP / DP

An incomplete search space can mislead DSE into
chip designs irrecoverable after fabrication

Vast system–hardware co-design space

Hardware hierarchy and design dimensions



Hardware

DRAM stacking: stacked vs. connected layers
Per-chip architecture: compute, buffering
On-/off-chip network topology and bandwidth

System

Parallelism: TP / PP / DP / EP / SP / CP / FSDP
Scheduling: tiling, fusion, pipelining
Collectives and compute–communication overlap

Tightly coupled, under area and thermal constraints: $\sim 2.5 \times 10^{14}$ design points

Challenge: fine-grained 3D hardware modeling

Little's Law and bank-level access semantics

Little's Law

$$\text{Effective BW} \leq \frac{\text{buffer size}}{\text{latency}}$$

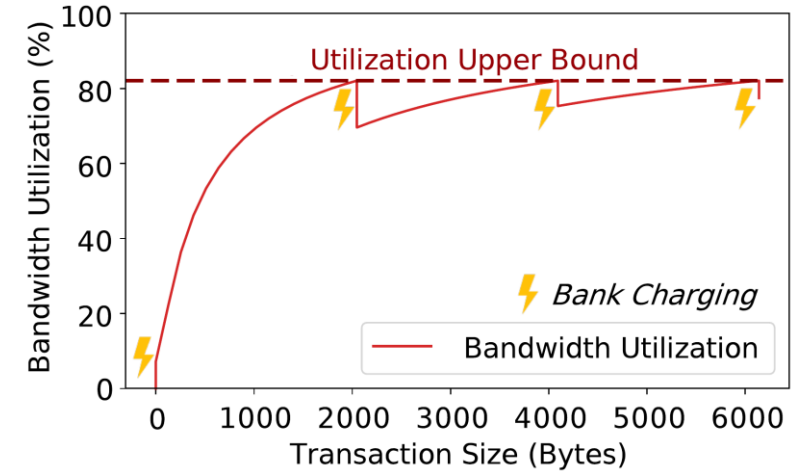
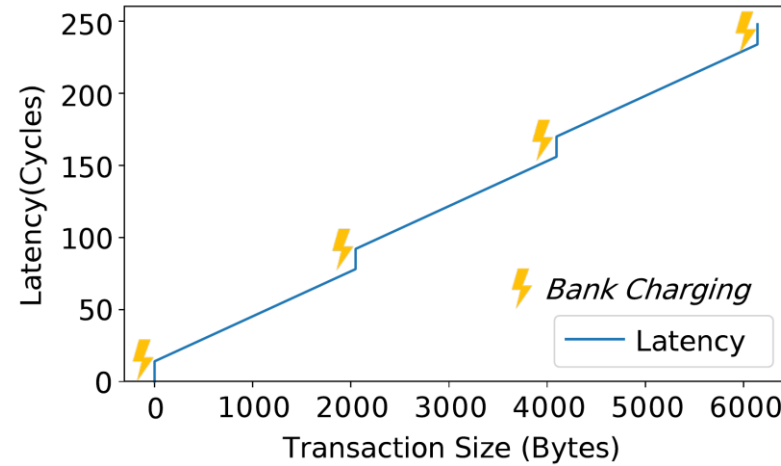
3D bandwidth demands larger on-chip buffers

Bank-level semantics

Activate + precharge per new row

Partial rows can't amortize it

Peak BW needs full-row streaming across many banks

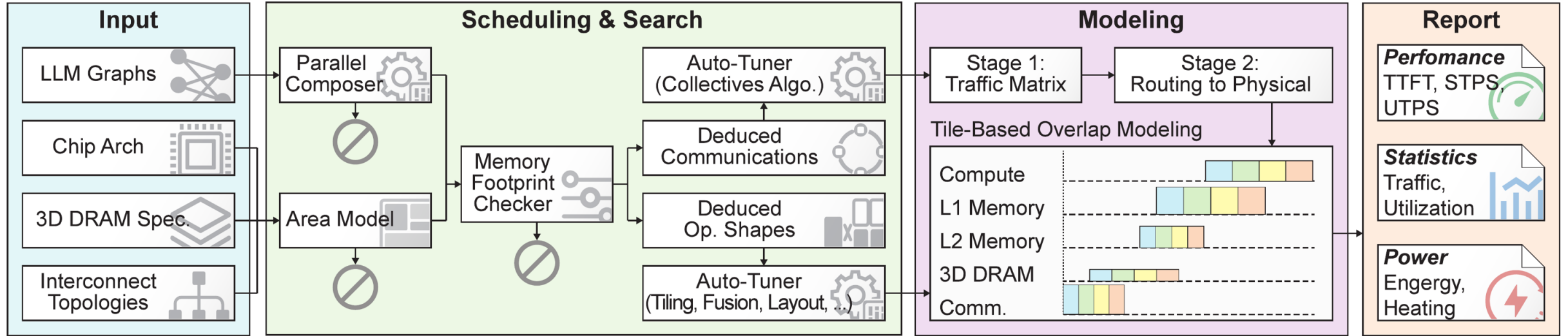


3D-stacked DRAM latency and bandwidth utilization (TSV, no interleaving)

Prior models assume idealized interleaving and overestimate achievable bandwidth

DeepStack overview

Full-stack modeling and DSE framework



1 Input

LLM graph, workload,
chip and NoC candidates

2 Search

area → parallelism →
footprint → auto-tuners

3 Modeling

3D memory, network,
tile-level overlap

4 Report

TTFT, UTPS, STPS,
power, temperature

Bank-level 3D memory modeling

Bank-level effects on a tile-level abstraction

(i) Transaction-size-dependent bandwidth

vendor C-model curve, per tile

(ii) Little's-Law buffering

$$BW_{\text{eff}} = \min\left(BW, \frac{\text{buffer}}{\text{latency}}\right)$$

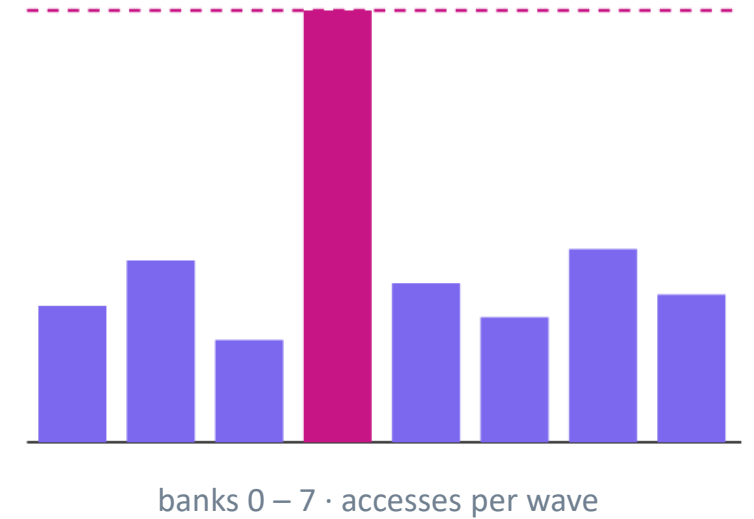
(iii) Bank parallelism limited by operator size

small ops leave banks idle

(iv) Bank conflicts from data layout

many tiles hit the same bank

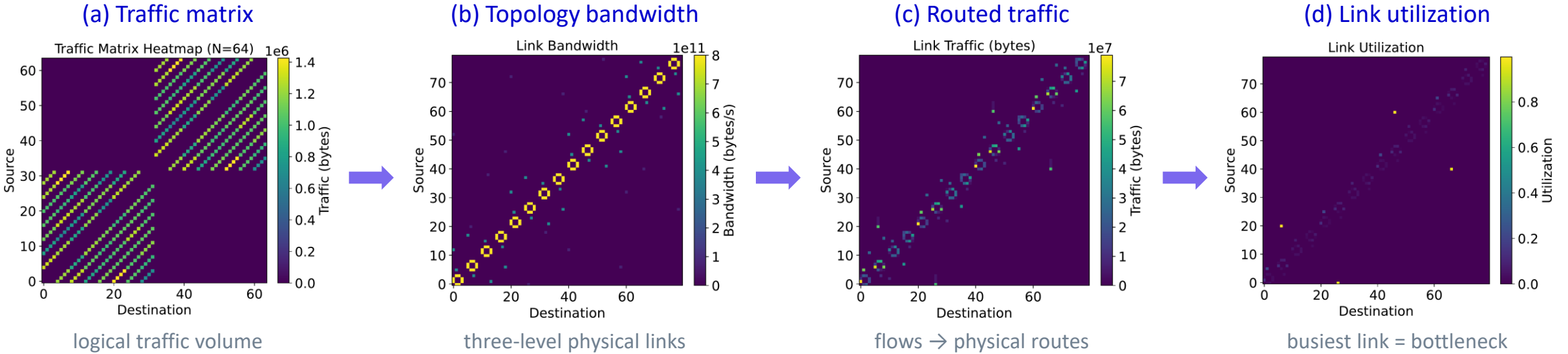
(iii)+(iv): per-wave bank histogram



The busiest bank sets the wave's service time

Dual-stage network abstraction

64-node EP all-to-all on a torus-mesh-mesh topology



Stage 1 · Traffic-matrix construction

Entry (s, d): bytes from s to d

Topology-independent:
reused across hardware configs

Stage 2 · Physical mapping and routing

$$T_{\text{net}} = \underbrace{\max_{p \in \mathcal{P}} \sum_{\ell \in p} \delta_{\ell}}_{L_{\text{net}}: \text{hop latency}} + \underbrace{\max_{\ell \in \mathcal{L}} \frac{V_{\ell}}{BW_{\ell}}}_{T_{\text{cong}}: \text{bottleneck link}}$$

Parallel composer and hierarchical search

Making $\sim 2.5 \times 10^{14}$ design points tractable

$$TP \times EP \times SP \times CP \times DP \times PP = N$$

2,574 valid strategies for 256 devices ($\pm$ FSDP)

Per-module: e.g. EP for MoE, TP for attention

Collective tuner: ring, tree, Rabenseifner, ...

Operator tuner: tiling, fusion, pipelining

Raw space $\sim 2.5 \times 10^{14}$

① Pareto-dominated HW (>10 layers)

② Invalid parallel strategies ($\sim 80\%$)

③ Memory-footprint check ($\sim 50\%$)

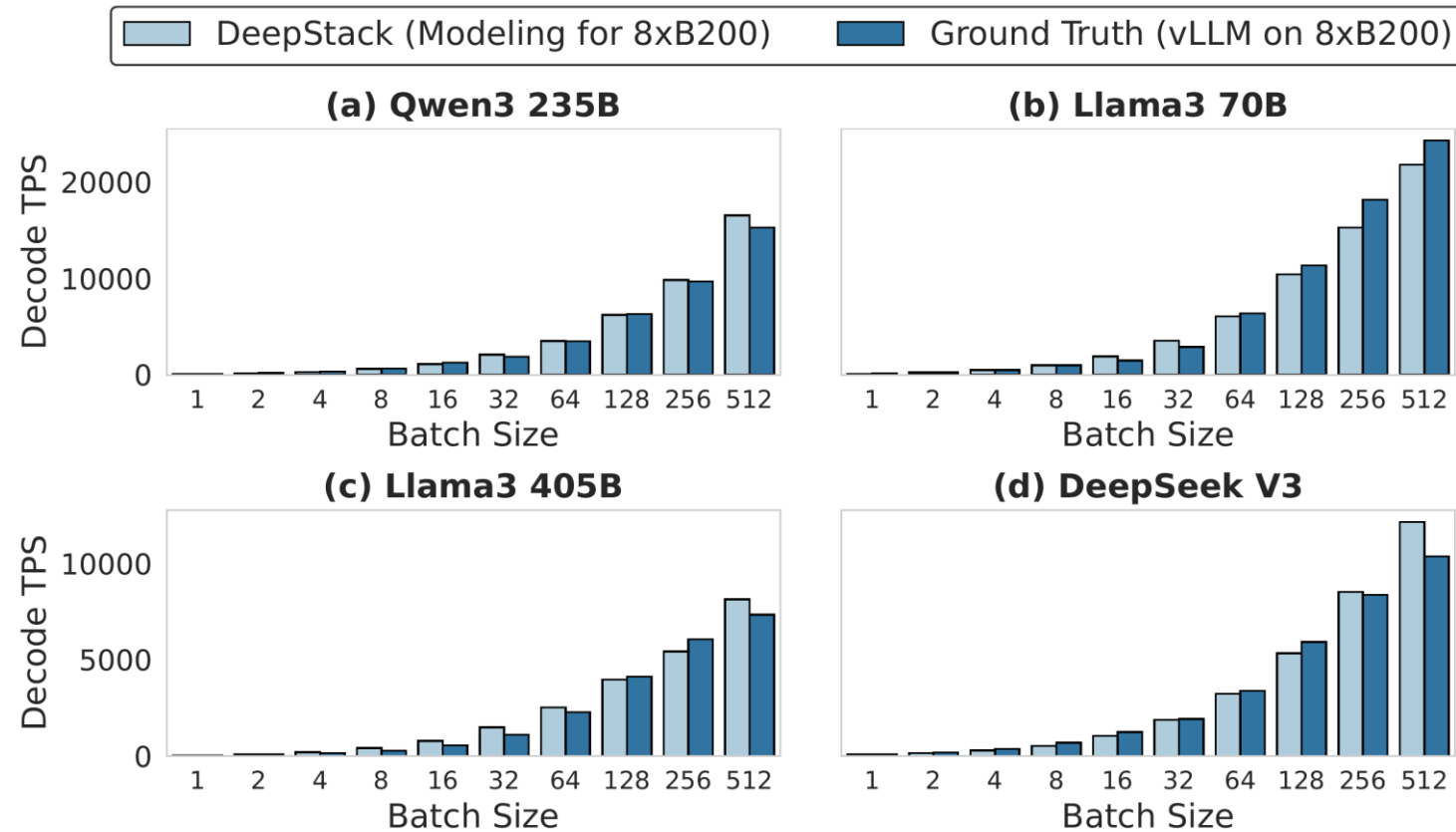
④ Hierarchical NoC search (top 5%)

$\sim 10^8$ evaluated $\cdot$ ~ 2 days

on a 512-core CPU

Modeling accuracy

Cross-validation against real systems and state-of-the-art simulators



DeepStack (light) vs. vLLM measured (dark), decode on 8xB200, 4 models $\times$ 10 batch sizes

End to end

12.9% MAPE vs. vLLM

Network

$\sim 2\%$ vs. ASTRA-sim NS-3

up to $10^5\times$ faster (0.1 s vs. 3 h)

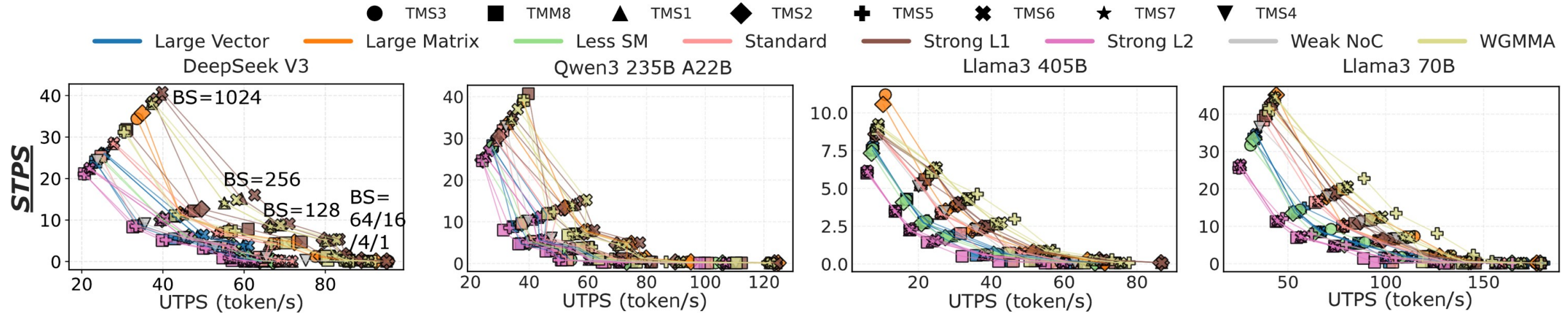
3D hardware

$< 5\%$ vs. Cadence Palladium

(in-house 3D design)

System–hardware co-design exploration

Decode Pareto frontiers per architecture and NoC; 3D vs. 2.5D at equal area



Curve: one architecture + NoC · point: one batch size (1–1024), parallelism re-searched · up and right is better

Small-batch decode is NoC-sensitive

Llama → TMS5 (TP-heavy)
DeepSeek → TMS6 (EP-heavy)

Best parallelism shifts with batch

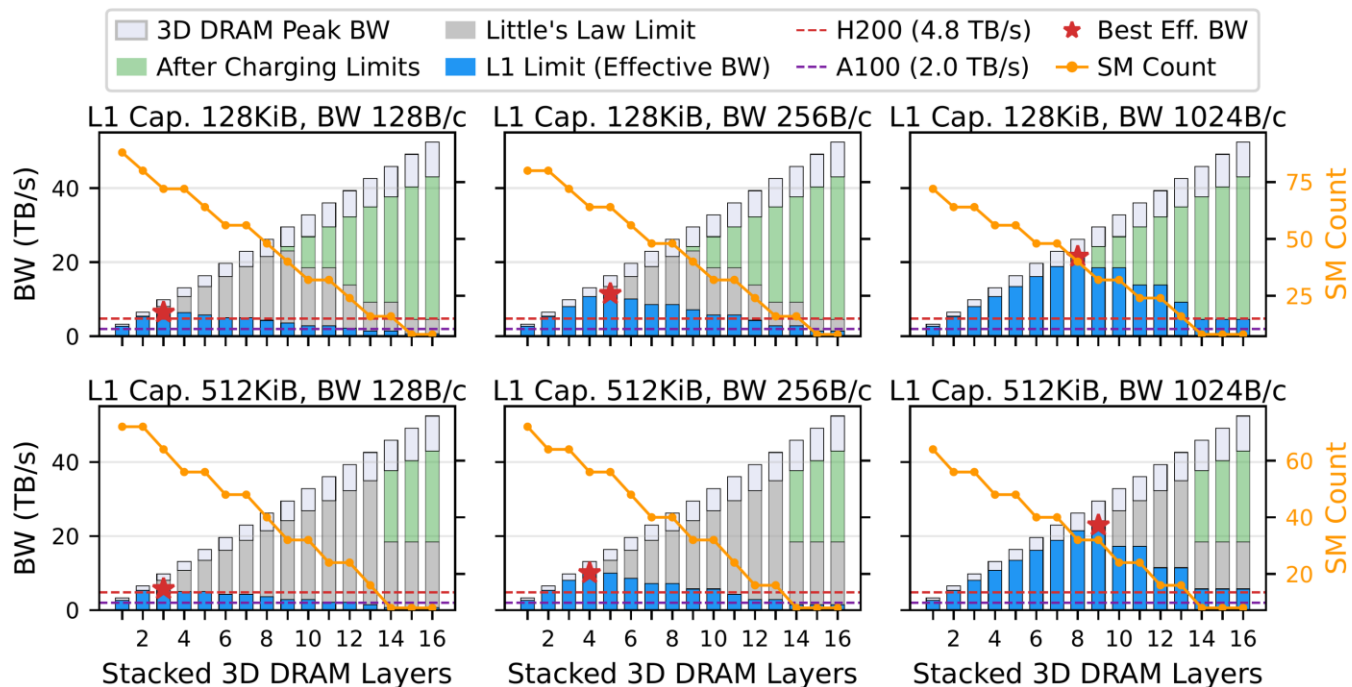
TP at small batch sizes
PP and EP as batch grows

3D vs. 2.5D, equal area

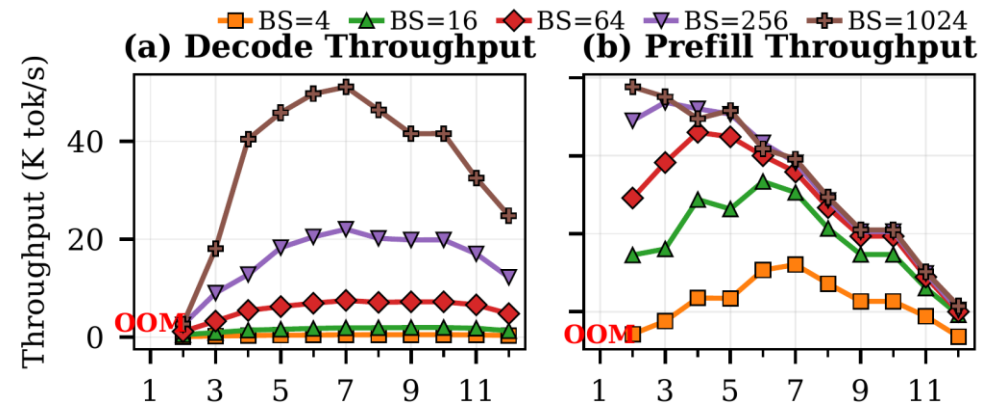
decode: 1.3–2.8× faster
prefill: on par with H200*

Case study: how many DRAM layers to stack?

Layer–bandwidth trade-off under a fixed area budget



Takeaway: an inverted-U curve — beyond ~9 layers, effective bandwidth decreases (12 layers: -39.6%)



DeepSeek-V3 optimal depth

small-batch decode: ~9

large-batch decode: 6–7

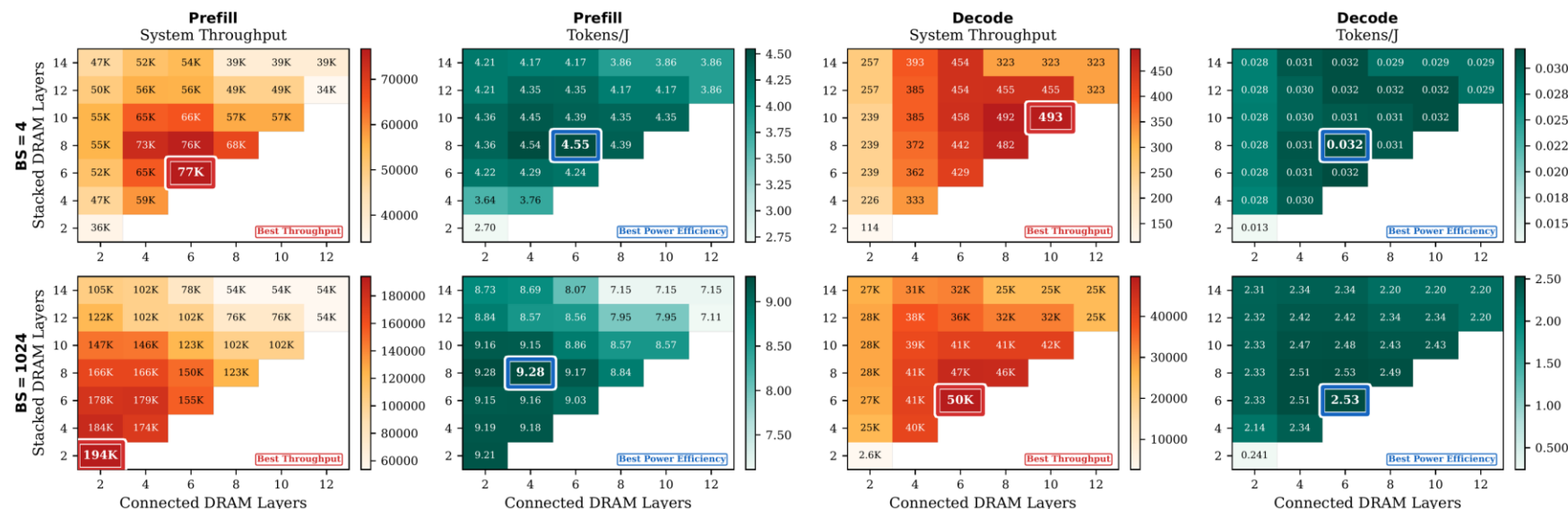
small-batch prefill: ~7

large-batch prefill: 2

Takeaway: batch size defines a more fundamental architectural divide than prefill/decode

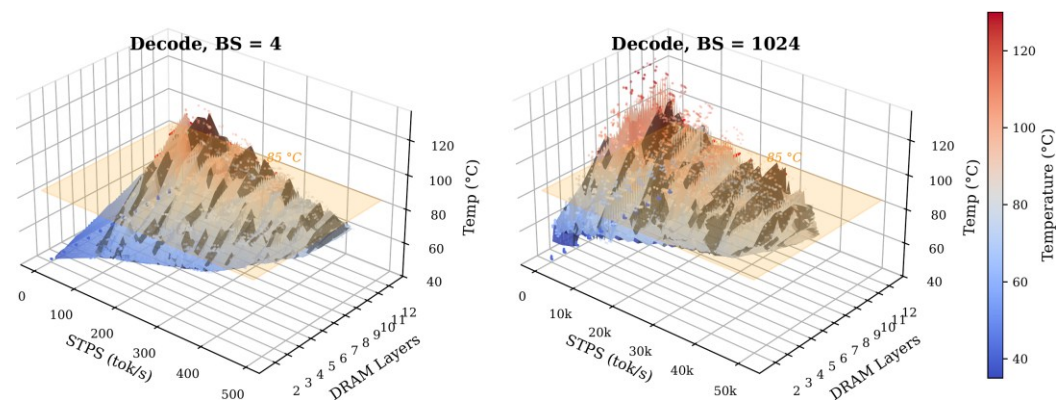
Connected vs. stacked layers and thermal feasibility

Decoupled design: throughput-optimal vs. energy-optimal



m stacked: capacity
 $n \leq m$ connected: bandwidth

Takeaway: energy-efficient designs favor more stacked but fewer connected layers (3–24% higher tokens/J)



Large-batch decode: most thermally challenging

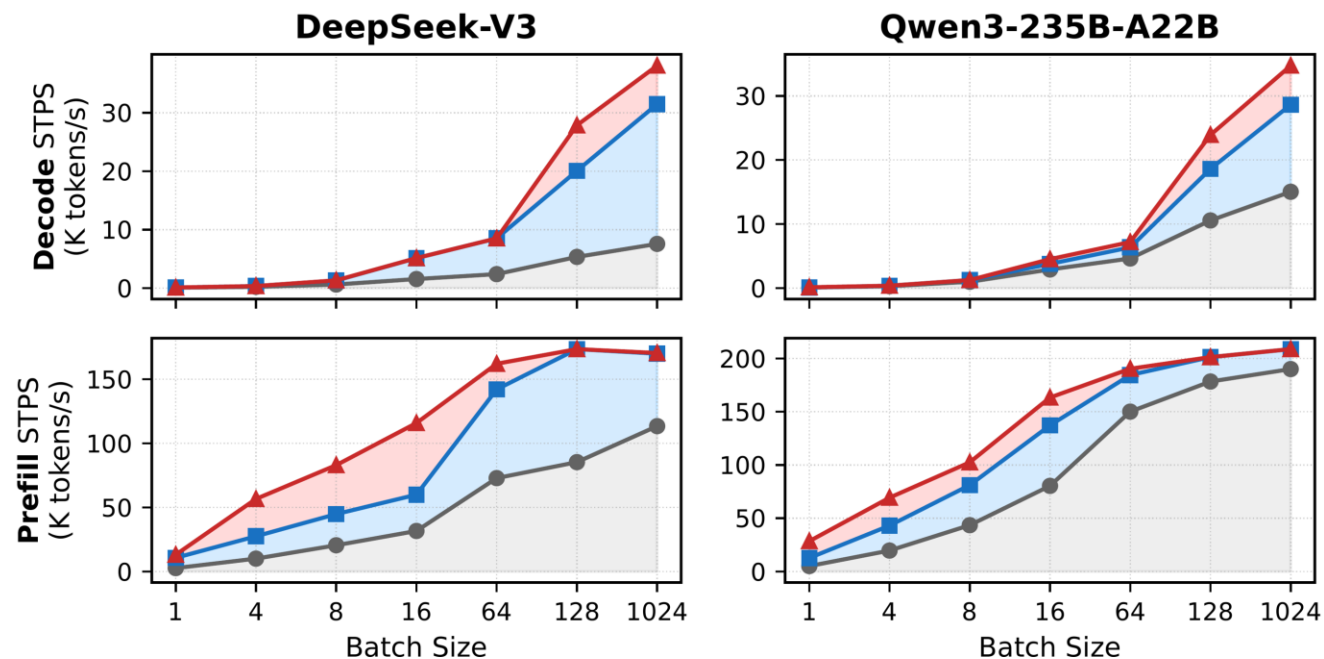
All feasible designs $< \sim 0.8 \text{ W/mm}^2$

Takeaway: thermal limits constrain the viable space; thermal-aware DSE is necessary

Ablation: incomplete schedule search distorts chip design

Expanded parallelism search space (decode)

—●— ASTRA-sim* ■ + More Parallel Dims ▲ + Flexible Module Parallelism (DeepStack)



■ ASTRA-sim*: TP / PP / DP only

■ + EP / SP / CP

■ + per-module parallelism (DeepStack)

Decode, BS = 1024: 5.03× DeepSeek-V3,
2.31× Qwen3-235B

Without EP: 8 layers, 5 SMs → power wall
With EP: 7 layers, 6 SMs → within limits

Restricted DSE misses up to 9.5×
modeled throughput (ablation)

Takeaway: incomplete schedule search permanently distorts architecture design;
the silicon mismatch is irrecoverable by software tuning

Conclusion

DeepStack and its design implications

DeepStack: accurate performance modeling and efficient DSE for distributed 3D-stacked LLM inference, up to $10^5\times$ faster at comparable accuracy

Optimal DRAM stacking depth is governed by a non-trivial bandwidth–area trade-off

Batch size creates a more fundamental divide than prefill/decode

Parallelism strategy tightly couples with hardware architecture: optimizing one without the other leads to suboptimal designs irrecoverable by software



DeepStack

github.com/tile-ai/DeepStack



Thank you

Zhiwen Mo

MICRO 2026 · Athens, Greece

Imperial College
London



DeepStack

github.com/tile-ai/DeepStack

