MICRO 2026 · Athens, Greece · 04/11/2026

DeepStack

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

Zhiwen Mo[1], Guoyu Li[1], Hao (Mark) Chen[1], Yu Cheng[2], Zhengju Tang[2], Qianzhou Wang[1], Lei Wang[2], Shuang Liang[1], Lingxiao Ma[3], Yuxiao Guo[3], Wayne Luk[1], Jilong Xue[3], Hongxiang Fan[1]

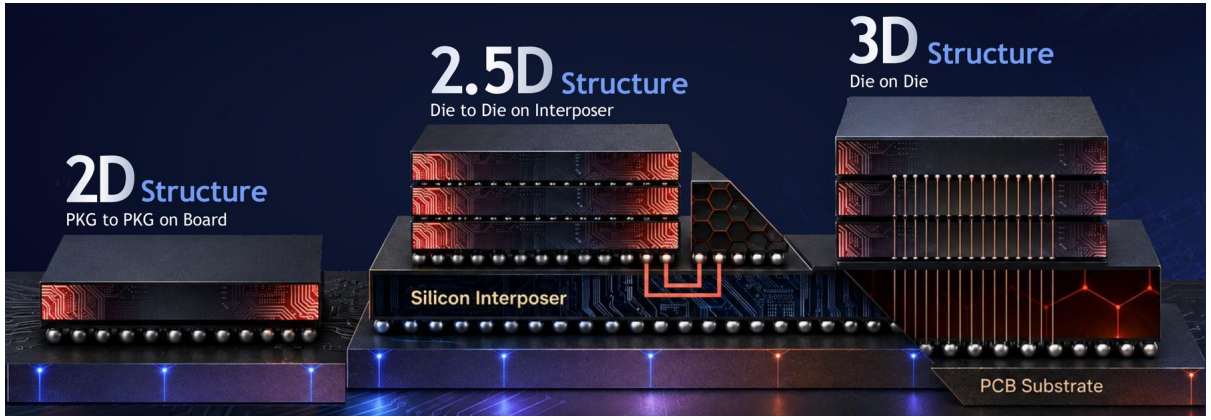
[1] Imperial College London [2] Peking University [3] Tile-AI

Imperial College London

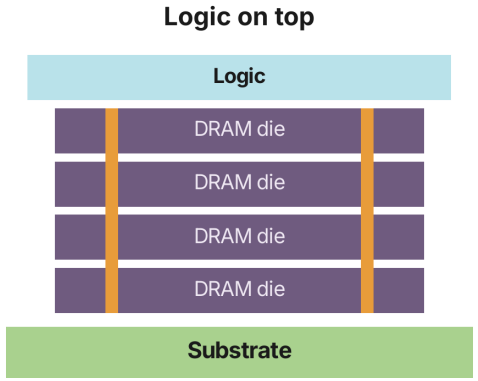
Artifacts Available V1.1 · Artifacts Evaluated Functional V1.1 · Results Reproduced V1.1

3D DRAM shifting from academia to industry

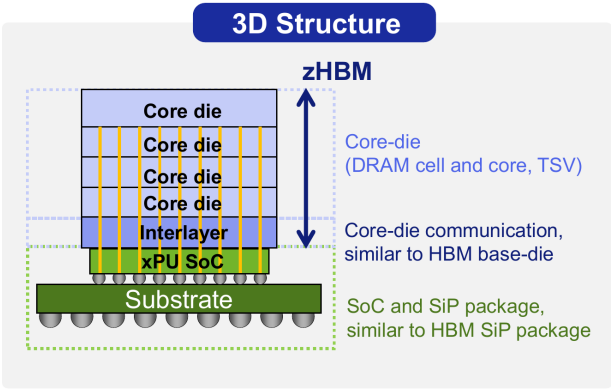
Increasing industry attention to 3D DRAM stacking



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



d-Matrix Raptor



Samsung zHBM

Figures: Hot Chips 2026 talks

d-Matrix · Raptor
3D DRAM bonded under N4 logic, 32 GB, ~100 TB/s

Qualcomm · AI250
compute die under stacked DRAM, planned for 2027

Cerebras · CS-6
DRAM wafer on the compute wafer (roadmap)

Samsung · zHBM
DRAM on the xPU, ~2.3× BW, ~100 W less than HBM4E

SK hynix × NVIDIA
HBM stacked directly on GPU logic (reported R&D)

Academia
mostly focus on single chip / node

> **3D DRAM: higher bandwidth, lower energy, and moderate capacity compared to DRAM**

Key contributions

DeepStack: modeling and design space exploration (DSE) for distributed 3D DRAM-stacked LLM inference

C1 · Fine-grained 3D hardware modeling

– Bank-level DRAM accesses, transaction-size curve

– Little's-Law buffering limits

– Area, power and thermal feasibility in a full-stack framework

C2 · Fast, accurate distributed modeling

– Dual-stage network abstraction

– Tile-level compute–communication overlap

– Up to 10^5× faster than NS-3 backend, 12.9% / 8.4% error vs. vLLM on 8×B200 / H100

C3 · Vast System–hardware co-design DSE

– ~2.5×10^{14} hardware and software design points

– Hierarchical search, ~2 days on one CPU server

– Up to 9.5× higher modeled throughput than restricted DSE

C4 · Design insights for 3D systems

– Optimal stacking depth: a bandwidth–area trade-off

– Batch size divides designs more than prefill/decode

– Parallelism and hardware must be co-designed

Model footprints exceed single-chip capacity

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

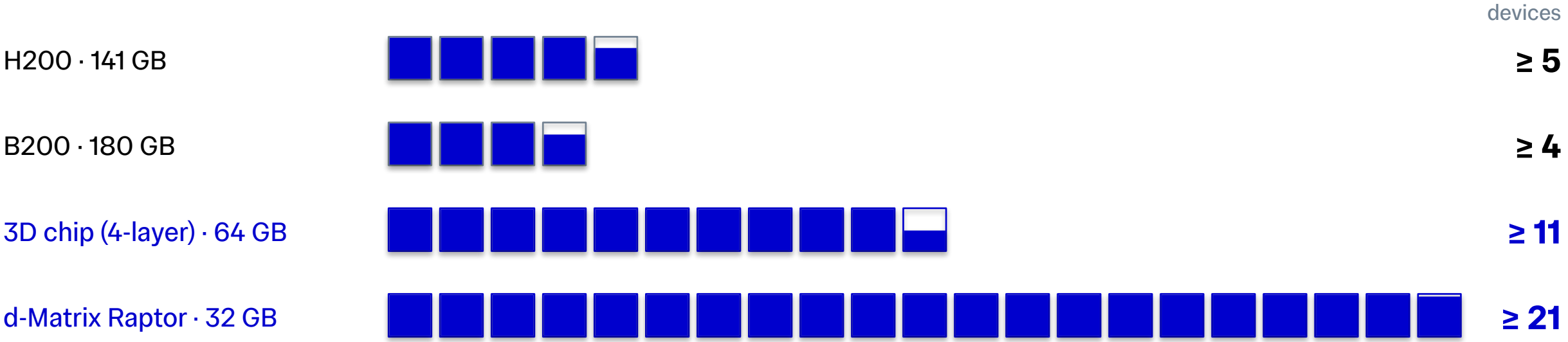


671 GB, one copy

Model footprints exceed single-chip capacity

Motivation · how many devices just to hold the weights?

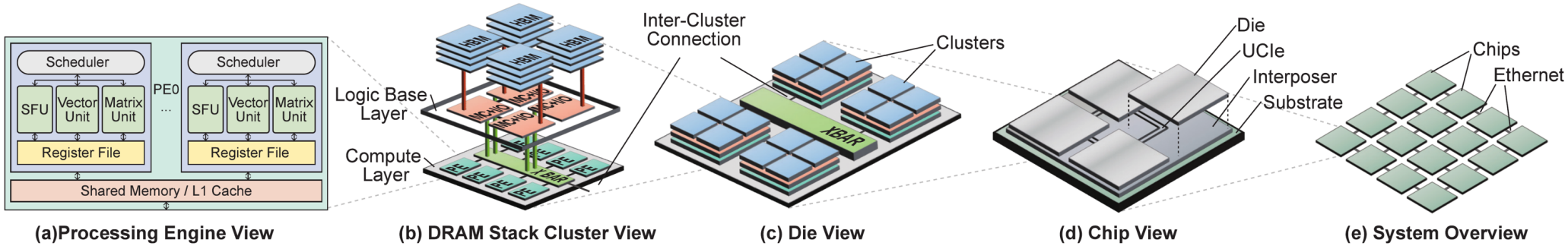
■ DeepSeek-V3: **671 GB** of FP8 weights + a KV cache that grows with context and batch



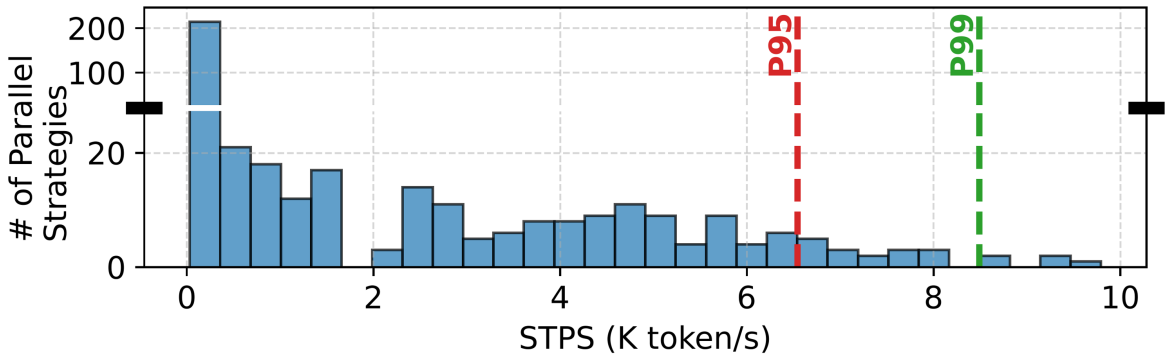
> Distributed is a must, and the performance is tightly coupled with parallelism strategy
> We cannot design the architecture itself individually

Vast system–hardware co-design space

Tightly coupled under area and thermal limits: ~2.5×10^{14} design points



System token per second (STPS) across 2,574 strategies



Best parallelism strategies are rare.

Prior tools: TP / PP / DP only

Hardware

DRAM stacking: stacked, connected layers

Per-chip: Vec / Tensor / SFU, cache capacity & BW

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

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

The vast co-search space

Hardware: area allocation



Compute matrix / vector / SFU / ALU

Buffers register, TMEM, L1 SMEM / L2 cache

MC + I/O memory controllers + I/O for 3D DRAM

Network L1 / L2 / L3 topology, bandwidth and latency

Overhead control and routing

Software: map the model onto it

- **Parallelism**
- **Tiling**
- **Layout**
- **Fusion**
- **Collective**

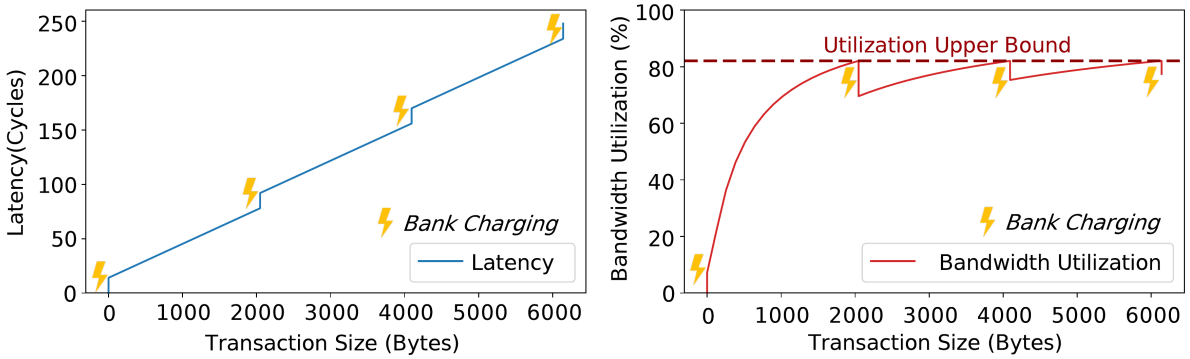
Constraints

Area | Footprint ≤ capacity | Power | Thermal

Fine-grained 3D memory modeling

Challenge: idealized interleaving overestimates bandwidth → C1: bank-level model

Challenge



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

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

3D bandwidth needs larger on-chip buffers

In fully connected DRAM bank setting, charging cannot be interleaved.

Prior models assume idealized interleaving and overestimate achievable bandwidth

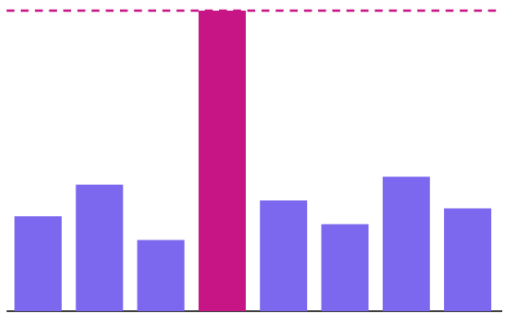
C1 · DeepStack's bank-level model

(i) Transaction-size bandwidth curve

(ii) Little's-Law buffering

(iii) Small operators leave banks idle

(iv) Layouts cause bank conflicts

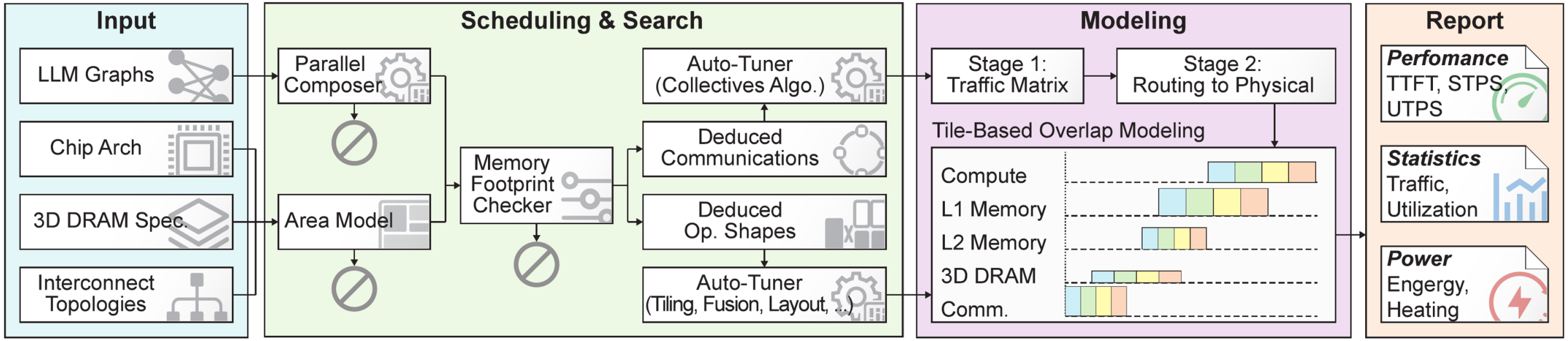


per-wave accesses, banks 0–7

The busiest bank sets the wave's time

DeepStack overview

Full-stack modeling and DSE framework



1 Input

LLM graph, workload, chip and NoC candidates

2 Search

2,574 parallel strategies, pruned to ~10^8 points

3 Modeling

3D memory, network, tile-level overlap

4 Report

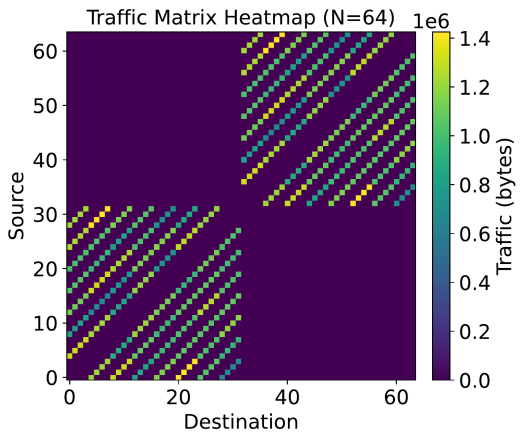
TTFT, UTPS, STPS, power, temperature

A restricted search misses up to 9.5×:
4.27× from software, 2.22× from hardware co-search

Dual-stage network abstraction

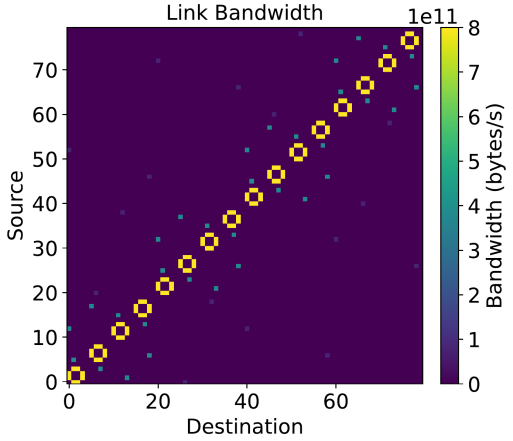
(Single acclerator performance modeling derived from TileSight)

(a) Traffic matrix



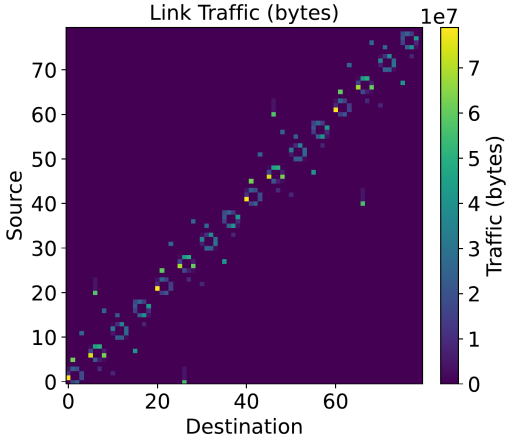
logical traffic volume

(b) Topology bandwidth



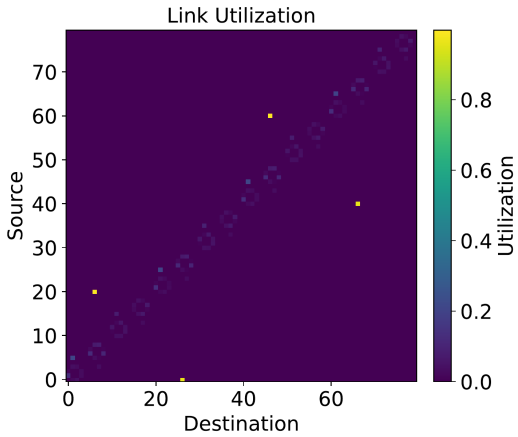
three-level physical links

(c) Routed traffic



flows → physical routes

(d) Link utilization



busiest link = bottleneck

Stage 1 · Traffic-matrix construction

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

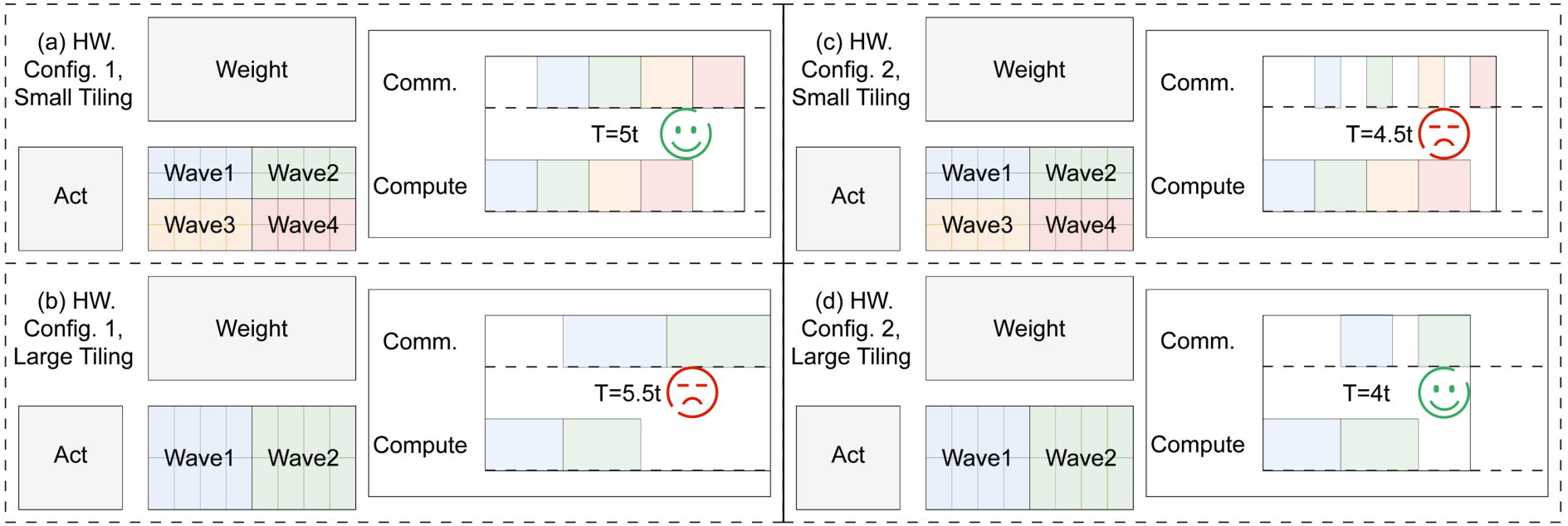
Topology-independent:
reused across hardware configs

Stage 2 · Physical mapping and routing

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

Tile-level compute–communication overlap

Wavefront pipeline model



$$T_{\text{e2e}} = \underbrace{\tau_{\text{comp}}}_{\text{prologue}} + \underbrace{(\tau_{\text{comm}} + L_{\text{net}})}_{\text{epilogue}} + \underbrace{\max\left((W-1)\,\tau_{\text{comp}},\ (W-1)\,\tau_{\text{comm}} + L_{\text{net}}\right)}_{\text{steady state}}$$

Wave i communicates while wave i+1 computes

Low NoC BW: small tiles, more overlap

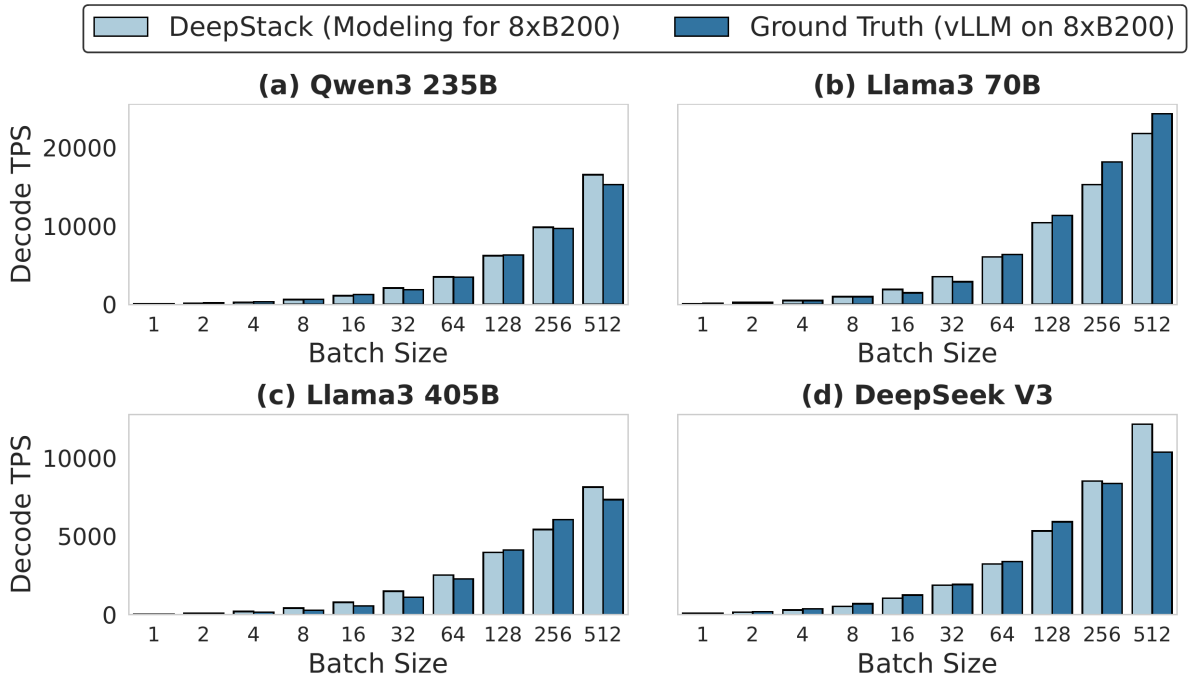
High NoC BW: large tiles, higher compute efficiency

Modeling accuracy

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

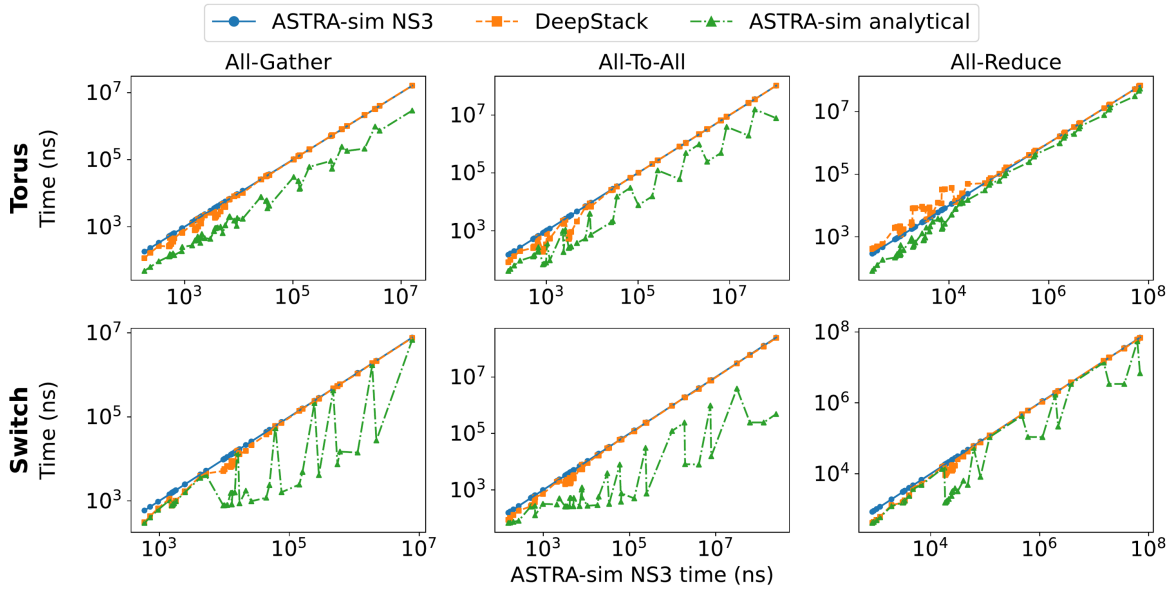
End to end: vLLM decode on 8×B200

12.9% MAPE over 40 model–batch points



Network: vs. ASTRA-sim NS-3

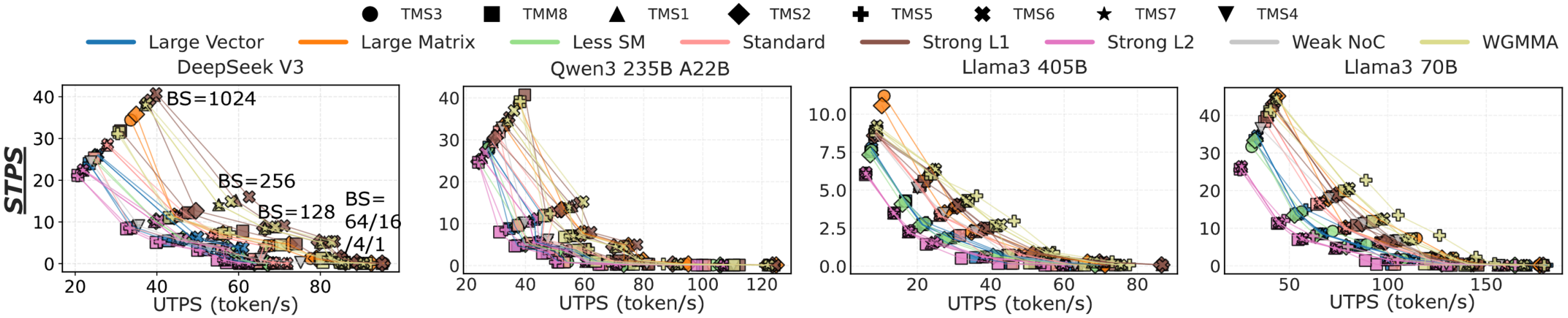
2.12% (Switch) · 1.62% (Torus) · up to 10^5× faster



Also, Triton-distributed compute-communication overlaping kernels on 8×H100 8.4%

System–hardware co-design exploration

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



Curve from pareto optimal point

Small batches are NoC-sensitive

Llama → strong local network (L1)
DeepSeek (large EP) → L3 (global)

Best parallelism shifts with batch

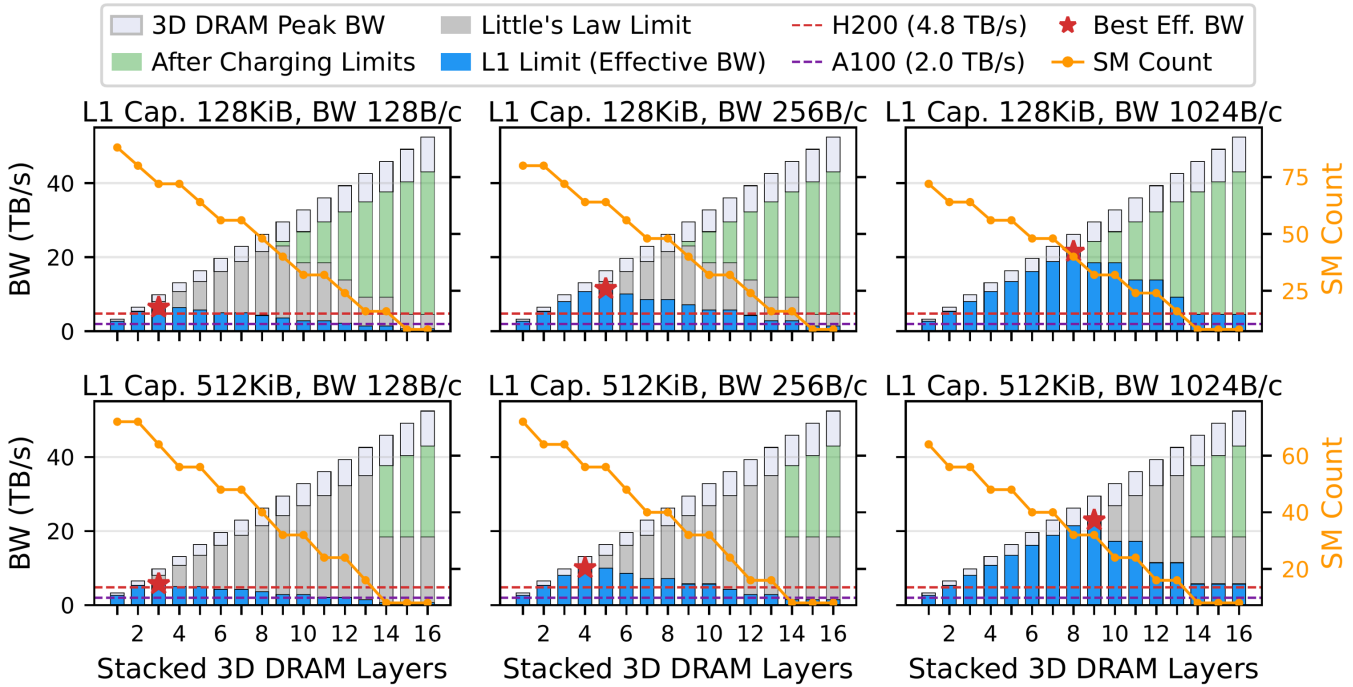
TP at small batch sizes
DP and EP as batch grows

3D vs. 2.5D, equal area

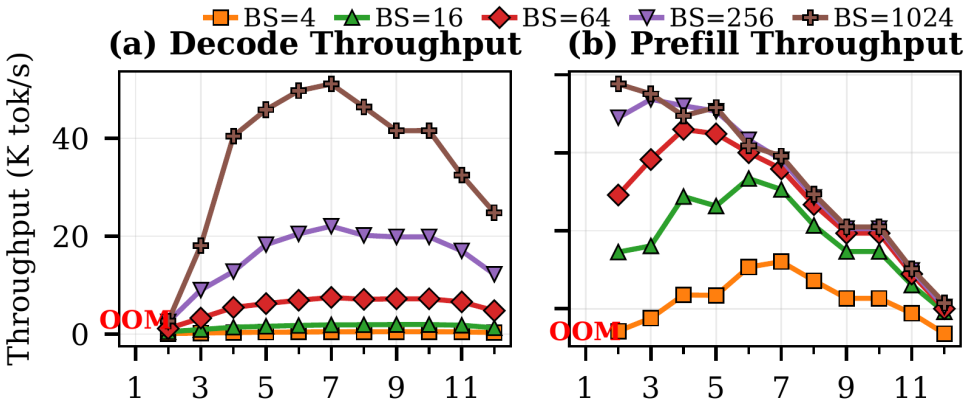
decode: 1.3–2.8× faster
prefill: on par with H200*

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

lmedium-batch decode: 6–7

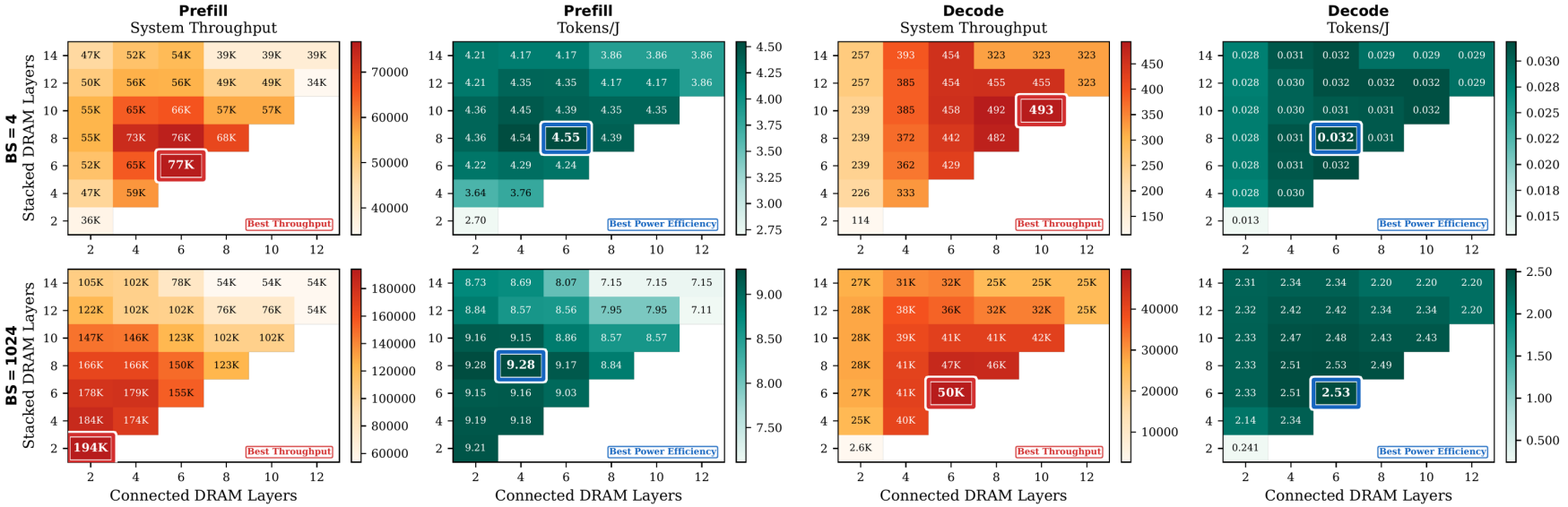
small-batch prefill: ~7

large-batch prefill: 2

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

Connected vs. stacked layers and thermal feasibility

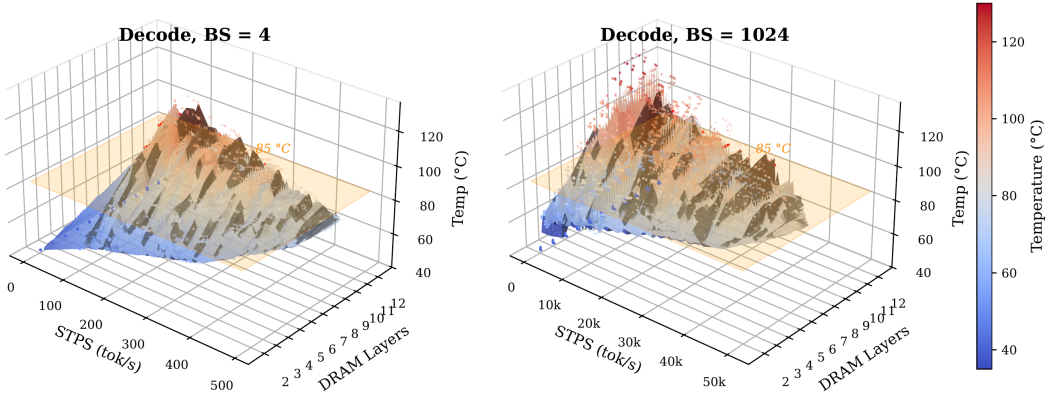
Decoupled design: throughput-optimal vs. energy-optimal



m stacked: capacity

n ≤ m connected: bandwidth

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



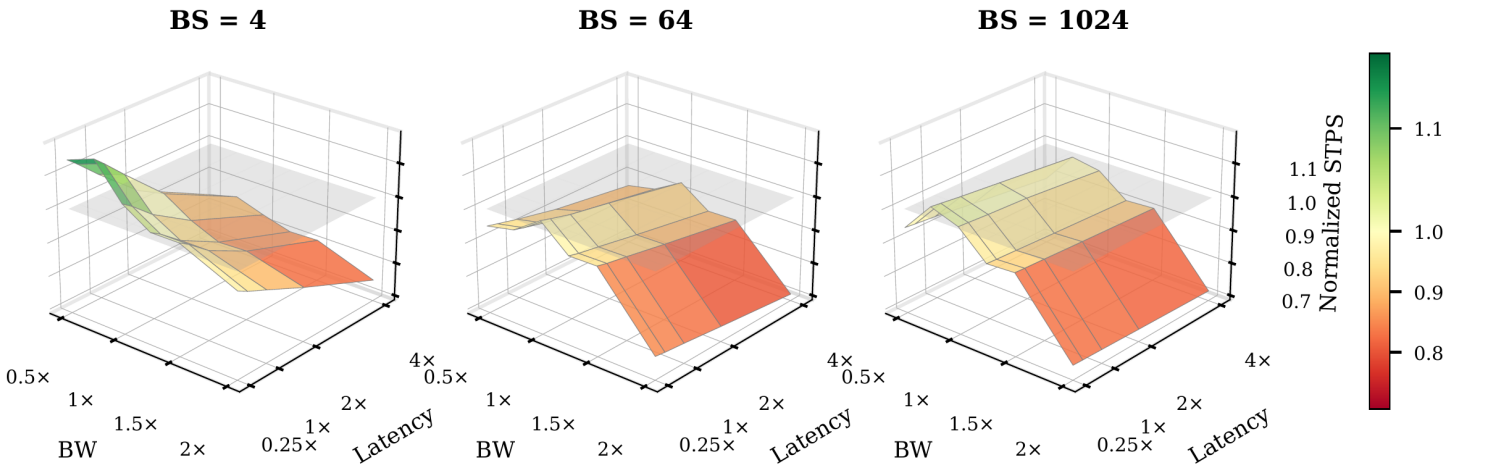
Large-batch decode: most thermally challenging

All feasible designs < ~0.8 W/mm²

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

NoC provisioning should be batch-aware

NoC bandwidth and hop latency co-optimized with compute area, parallelism re-searched

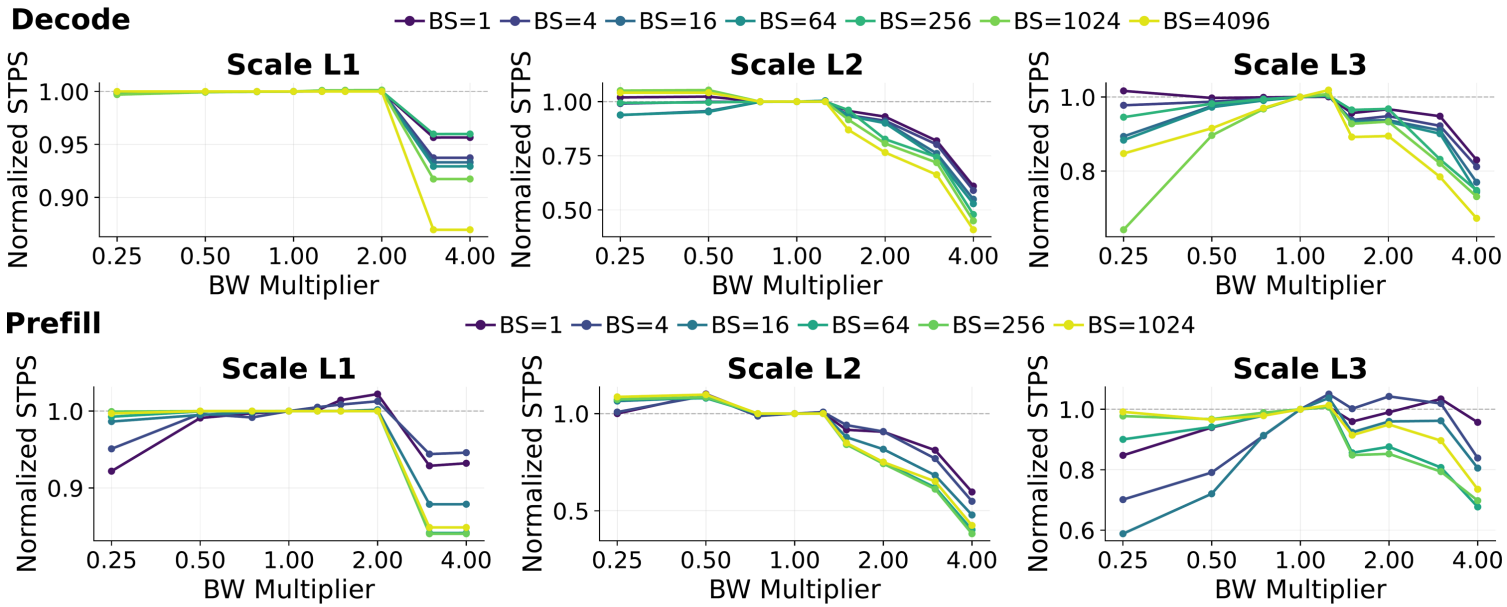


All levels together:

BS = 4: hop latency dominates (up to 31%)

BS = 1024: bandwidth only matters via area (2× BW: −27%, fewer SMs)

Small batches need low hop latency



Each level alone:

L2 (inter-tile) can be modestly reduced

L3 (off-chip) gains from a slight increase

Large batches need high L3 bandwidth for large-scale EP

Conclusion

Design takeaways for future 3D-stacked LLM accelerators

DeepStack An accurate performance model and efficient DSE framework for distributed 3D-stacked LLM inference

1. **Match TP x EP degree to the strong-network scale-up domain (∝ model size)**
2. **At 7 nm, stacking beyond ~9 DRAM layers stops paying off**
3. **Batch size divides architectures more than prefill / decode**
4. **NoC provisioning should be batch-aware**
5. **Energy-efficient and most performant designs differ**

Proposal: three 3D chips for three scenarios

1. **Small-batch decode**
   "Deep" stack (~9 layers)
   Low-latency interconnect
2. **Small-batch prefill + medium-batch decode**
   Moderate stack (6–7 layers)
   Network scale-up domain should match large-scale EP x TP degree
3. **Large-batch prefill**
   No need for stack at all

Co-design the chip with its parallel schedule: a mismatch is irrecoverable by software

Thank you

Scan the QR code and try DeepStack!



Watch 0 | Fork 5 | Starred 67

DeepStack

github.com/tile-ai/DeepStack

Zhiwen Mo[1], Guoyu Li[1], Hao (Mark) Chen[1], Yu Cheng[2], Zhengju Tang[2], Qianzhou Wang[1], Lei Wang[2], Shuang Liang[1], Lingxiao Ma[3], Yuxiao Guo[3], Wayne Luk[1], Jilong Xue[3], Hongxiang Fan[1]

[1] Imperial College London [2] Peking University [3] Tile-AI

Imperial College London

Acknowledgements: This work was supported by ARIA through its Scaling Compute programme.