Thermally-Aware System-Technology Co-Optimization for AI Systems

Dedeepyo Ray*, George Karfakis*, Alexander Graening*, David Ratchkov†, Puneet Gupta*
*University of California, Los Angeles
†Anemoi Software

***Abstract*—Rising computational demands in modern AI workloads are pushing systems toward larger designs. Single-die approaches run into practical limits, which makes chiplet-based integration an increasingly useful option for avoiding those drawbacks. Chiplet-based integration can achieve near-monolithic performance, but results in platforms that are often thermally constrained. Packaging choices, such as thermal interface materials (TIMs), underfill/infill composition, and 2.5D versus 3D integration, influence the worst-case temperature and therefore the application-level system *performance*. Most prior models treat performance and thermal considerations separately, incorrectly estimating the impact of packaging on system performance. To address this gap, we present a thermally-aware system-technology co-optimization (STCO) framework that closes the loop between workload behavior, power, packaging, and thermal throttling to predict end-to-end runtime for large language model (LLM) training and inference workloads. In this work, we show that for our example 2.5D and 3D systems, adjusting the TIM can result in a workload performance improvement up to 6.8% while increasing the overall heat transfer coefficient (HTC) can result in an improvement up to 12.5%.**

I. Introduction

As Moore's Law scaling slows, chiplet-based systems are becoming increasingly important. For instance, 2.5D architectures are now commonly employed in commercial products like the NVIDIA A100, H100, and B200 GPUs [1]–[3]. 3D stacking, which is widely used for high bandwidth memories (HBMs) in commercial products, is also being explored for stacking memory with compute to increase the bandwidth between memory and compute. The AMD MI300 series [4], [5] stacks memory on compute and the Intel Ponte Vecchio [6] stacks compute on memory. This trend towards greater density and vertical stacking has driven a significant rise in Thermal Design Power (TDP).

Temperature directly impacts performance in high-performance systems. In high-performance server-class GPU nodes, high temperatures force HBMs to refresh more frequently while forcing GPUs to down-clock under dynamic voltage frequency scaling (DVFS), significantly degrading performance.

The temperature profile of 3D stacked systems is more strongly dependent on the composition of fill and interface materials than 2.5D systems due to the larger number of these layers on the thermal path. In addition, heat flux is higher due to the increase in density of active silicon [7].

The goal of using 2.5D and 3D integration is typically to increase performance by increasing the amount of compute or memory available in a single package. While packing more resources densely into a package can enhance performance, it also increases the power density for the system. This can make cooling a serious issue and require using larger heatsinks, bigger fans, liquid cooling, two-phase cooling, or other advanced cooling solutions. If the cooling system is not sufficient to cool the system at peak performance, it is necessary to use some form of thermal throttling. Chiplet-based systems often reduce their clock frequency or temporarily disable components [8] when operating above safe temperature limits. This thermal throttling reduces throughput and increases workload execution time.

Recent work has begun to close the loop between thermal simulation and performance by quantifying temperature-induced GPU and HBM throttling in 2.5D GPU+HBM packages using closed-loop workflows on small synthetic benchmarks [9]. In contrast, we apply this loop to full end-to-end LLM training and inference and broaden the design space to both 2.5D and 3D packaging, while using a more detailed throttling model.

To the best of our knowledge, no previous study has developed a unified flow that analyzes the thermal-aware performance of a chiplet system with the ability to capture the performance impact of materials used for packaging and chiplet stacking architectures for real LLM workloads. In this work, we built a system-technology co-optimization (STCO) framework to allow early design stage thermal-aware performance comparisons. We can assess the impact of changing the thermal properties of materials such as infill, underfill, or TIMs. We can assess the impact of package architecture changes: HBM stack height, 2.5D vs. 3D, and memory on top of compute vs. compute on top of memory in a 3D stack. Finally, we can model the performance of real workloads on these systems and show how our results change for training vs. inference across different machine learning models.

We make the following contributions:

- We develop a flow that allows us to evaluate the thermal-aware performance impacts of changing thermal properties of materials, packaging architecture, and workloads for chiplet systems.
- We evaluate the impacts of changing various materials in the stackup. Specifically, we run simulations for different underfill, infill, and TIM materials. We show the best improvement comes from improving the TIM material, giving up to a 6.8% performance improvement. Improving the underfill and infill materials had negligible impact.
- We compare the impacts of improving the underfill, infill, and TIM materials to the results of improving the overall HTC. We show that improving the HTC provides up to a 12.5% improvement on our workloads.

- We provide a comparison of the relative performance of 2.5D and 3D systems after considering the thermal consequences of stacking. 3D with compute on top showed better performance by up to 3.7% compared to 3D with memory on top. Meanwhile, 2.5D stacking showed less performance degradation due to thermal throttling, but overall has lower performance than the 3D configurations.
- We make our framework available on GitHub at https://github.com/nanocad-lab/thermal_stco for others to use.

In Section II, we describe the flow we use to evaluate the systems described in Section III. We show and discuss our results in Section IV before concluding with Section V.

II. Thermal-Aware STCO Framework

We evaluate thermal-aware performance using a closed-loop workflow that couples runtime prediction with steady-state thermal simulation, as shown in Figure 1. The goal is to predict end-to-end runtime under realistic temperature-driven throttling by explicitly closing the loop between workload behavior, power, packaging materials, cooling strength, and the GPU and HBM temperature limits.

A. Thermal-Aware Performance Evaluation Loop

Given a workload, chiplet stackup, and system micro-architecture configuration, we first run the performance simulator to estimate the power for each chiplet in our design. These power numbers are passed to our thermal simulator to determine whether or not the HBM and/or GPU would be forced to thermal throttle. The throttling information is then passed back to the performance simulator to estimate updated power numbers for the chiplets. We iterate until the throttling state and peak temperatures converge. We stop when the peak GPU temperature is below safe temperature limit and the peak HBM temperature changes by less than 0.1°C between successive iterations.

B. Performance Simulator

Rapid-LLM [10] is a unified performance modeling framework for LLM training and inference on GPU clusters. It generates hardware-aware, operator-level traces from an abstract hardware and model specification, and executes those traces to estimate runtime.

We use this tool to predict end-to-end training and inference runtime and the associated workload-driven utilization for a given model, parallelism strategy, and hardware configuration, as shown in Figure 2. We translate utilization into per-chiplet power for the GPU and HBM, accounting for cases where stalls change the activity and resulting average power of the chiplets. In our closed-loop flow, we modify the effective device parameters such as peak GPU throughput, GPU-to-HBM bandwidth, and GPU-to-HBM latency based on thermal simulation results and temperature-driven GPU DVFS and HBM refresh throttling. Rapid-LLM returns updated runtime and GPU utilization/idle behavior, which we use to update power inputs for the next thermal iteration.

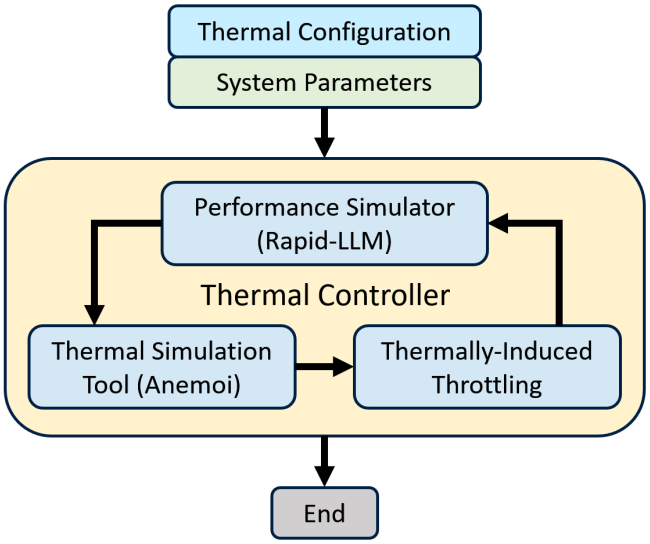


Fig. 1: Thermal-Aware Performance Evaluation Flow

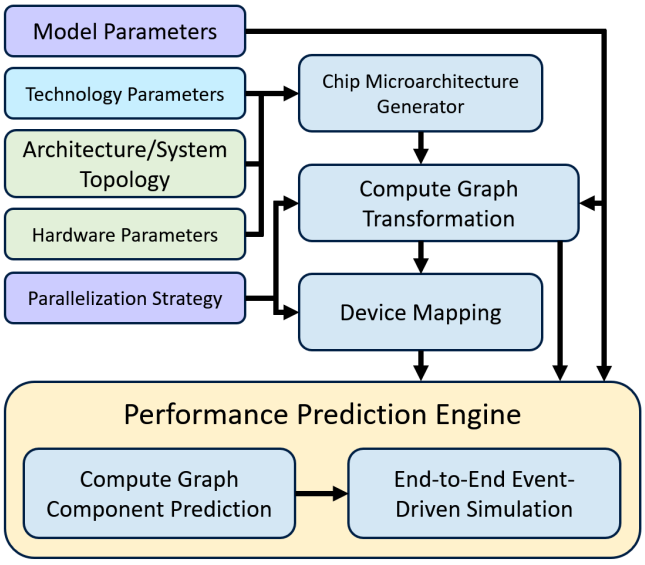


Fig. 2: Performance Modeling Framework (Rapid-LLM)

C. Thermal Simulator

General-purpose analytical thermal solvers such as Ansys Icepak [11] are highly accurate but are often too slow for the large number of iterations and design points required in early-stage design-space exploration. We therefore use Anemoi [12], a faster commercial thermal simulator. Using this tool, we estimate steady-state temperatures for a given chiplet layout, stackup materials, and power map under a specified cooling boundary condition modeled with an effective HTC.

Although chiplet power varies during ML workload execution, package temperatures evolve on much slower time scales than workload behavior under typical conditions. The package and cooling path have thermal time constants typically on the

Temperature (°C)	Refresh Rate	Relative Refresh Energy	Relative Latency	Relative Bandwidth
$T < 85$	1×	1.0	1.00	1.00
$85 \leq T < 95$	2×	2.0	1.24	0.91
$T \geq 95$	4×	4.0	1.71	0.73

TABLE I: Normalized HBM Performance for Different Refresh Rates (from [9]).

order of seconds, while the dominant phases in LLM execution occur on microsecond-to-millisecond scales and repeat many times within an iteration. As a result, temperature tracks a cycle-averaged power level, and the throttling behavior that determines end-to-end runtime is governed by sustained operating points. For this reason, we use steady-state simulations for this work.

Our tool supports both 2.5D and 3D assemblies by generating thermal models directly from a chiplet system description adapted from [13] that includes placement location, layer stackups, and thermal properties of materials. The simulator outputs per-chiplet peak temperatures which we use to apply DVFS and refresh-rate throttling models in our closed-loop iterative flow.

D. Throttling Model

1) HBM: HBM consists of stacks of Dynamic Random Access Memory (DRAM). DRAM capacitor leakage increases with temperature, reducing memory cell retention time [14] and requiring more frequent refreshes at higher temperatures [15], [16]. According to the JEDEC specifications, the HBM must double its refresh rate beyond two manufacturer-dependent trip point temperatures. Previous works have reported that these trip temperatures lie between 75°C and 95°C [8], [17]–[19]. For the purposes of this analysis, it is assumed that the two trip points are at 85°C and 95°C. As shown in Table I from [9], these trip points can have significant performance effects. We account for elevated HBM refresh rates in the performance simulation as reductions in effective bandwidth and increases in latency.

The refresh energy of the HBM is assumed to be 12% of its total energy consumption when unthrottled [20]. The peak HBM temperature and Table I determine the power consumed by HBM in our workflow introduced in Section II-A.

2) GPU: GPU throttling is modeled by adjusting its peak clock frequency, and hence, performance, based on its temperature. The maximum safe temperature for the GPU is assumed to be 95°C [9]. For higher temperatures, we model GPU DVFS following quadratic frequency-power scaling. We capture GPU DVFS behavior as a reduction in effective peak throughput.

E. Power-Temperature Calibration

Running a full thermal simulation at every loop iteration can dominate exploration time, especially when sweeping many package variants. To reduce this time, we precompute a power-to-temperature map for each system by sweeping GPU power from idle to peak and HBM power across the relevant range, then recording the resulting peak GPU and HBM temperatures into a calibration table. During closed-loop iterations, we replace repeated thermal simulator runs with table lookup and interpolation of $(T_{\text{GPU}}, T_{\text{HBM}})$ given $(P_{\text{GPU}}, P_{\text{HBM}})$. This calibration step is tool-agnostic and can be generated using any thermal solver.

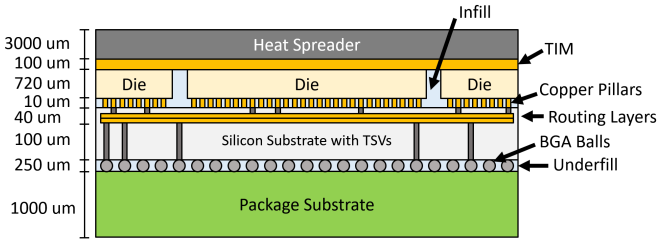


Fig. 3: Vertical Cross-Section of Stack. Each stacking configuration constrains the stacked dies to fit in the 720 μm height shown here.

Note that a single peak GPU temperature and a single peak HBM temperature are sufficient since we are evaluating single-GPU configurations and consider uniform throttling across all HBMs. For a larger or less uniform system, it may be necessary to calibrate a larger number of peak temperatures.

III. Case Studies

In this work, we consider both 2.5D (GPU adjacent to HBM) and 3D (stacked GPU and HBM) packaging since their divergent thermal paths can result in different throttling behavior even for the same workload and total power. As case studies, we compare 2.5D and 3D stacked GPU configurations modeled after the NVIDIA A100 under identical cooling assumptions and report the end-to-end impacts on LLM training and inference across multiple models.

A. Thermal Simulation Parameters

Our system is inspired by the NVIDIA A100 GPU which consists of one large GPU die surrounded by 6 HBM stacks. In Figure 3, we show the layers in the stack and their corresponding thicknesses. The stacked layers from bottom to top are as follows:

1) Package substrate: 1 mm thick.
2) BGA bonding layer with an underfill of epoxy: 250 μm thick.
3) Silicon substrate: 100 μm thick.
4) Silicon substrate routing layers: 40 μm thick.
5) Bonding layer composed of cylindrical Cu pillars: 10 μm thick.
6) Chiplets (GPU or HBM): 720 μm thick.
7) Thermal interface material: 100 μm thick.
8) Heat spreader: 3 mm thick, connected to a heat sink modeled as a single effective HTC.

The infill between chiplets is assumed to be epoxy or an aluminum nitride matrix. TSVs supply power vertically to the chiplet-based systems. The substrate is modeled as silicon with

Chiplet	GPU	HBM
Width (mm)	32.4	7.56
Length (mm)	25.5	11.49
Power (W)	370	5
# Chiplets	1	6

TABLE II: Specifications of Chiplets.

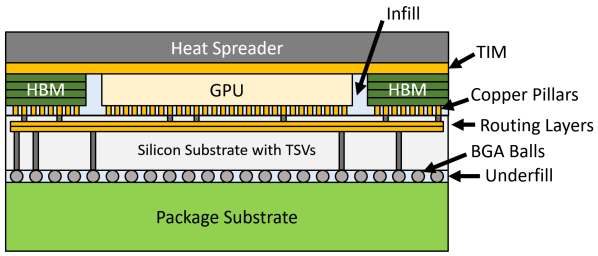
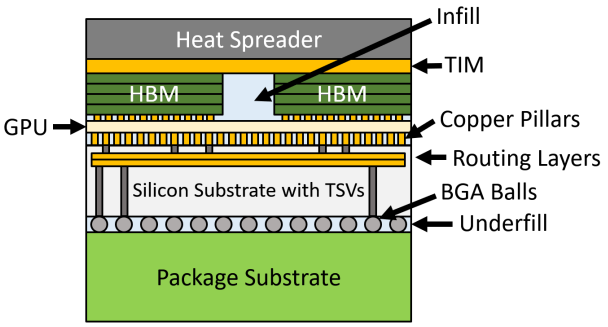


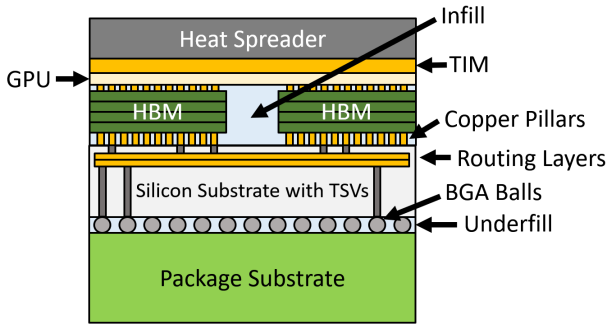
Fig. 4: 2.5D Stackup

five metal routing layers. They are assumed to be 50% Cu and 50% Si.

Each of our GPUs and HBMs was modeled on the publicly available specifications of the NVIDIA A100 chip. We assumed a TDP of 400 W for each set of 1 GPU and 6 HBM chiplets, modeling it on a commercially available NVIDIA A100. The specifications for our system are shown in Table II.



(a) 3D Stackup with Memory on Compute.



(b) 3D Stackup with Compute on Memory.

Fig. 5: Comparison of 3D Stackup Configurations.

B. 2.5D and 3D packaging

1) 2.5D GPU and HBM: We assumed the 2.5D system consists of HBMs stacked side-by-side with GPU. In addition, we considered HBM stacks with 8 DRAM dies (the stack height of HBM3E [21]) and with 16 DRAM dies (the maximum supported stack height of HBM4 [22]). Each 8-tall HBM stack was assumed to consist of one 410 μm tall DRAM die at the top, one 60 μm tall memory controller die at the bottom, and seven 30 μm tall DRAM dies between these. These are all bonded by 5 μm tall interstitial layers. The GPU was assumed to consist of silicon with 20 metal routing layers. The structure is shown in Figure 4.

2) 3D GPU and HBM stacking: The individual die thicknesses in the 3D stackup match those in the 2.5D stackup, except that the top DRAM die is 30 μm thick and the GPU die is thinned, in order to limit the total stack height to 720 μm. We evaluate two 3D integration organizations (Figure 5). In memory-on-compute, six HBM stacks are placed above the GPU die. In compute-on-memory, six HBM stacks are first arranged as a 2×3 array in the x–y plane (all stacks having equal height), and a single GPU die is stacked above and spans the entire 2×3 HBM footprint so that all six HBM stacks share one GPU. UCIe-3D [23] predicts a 25× increase in bandwidth from 2.5D to 3D. We pessimistically assumed the GPU-to-HBM bandwidth for 3D is 4× that of the 2.5D configuration.

C. Effective HTC

We model the cooling solution as a convective boundary condition applied at the top surface of the package, parameterized by an effective heat transfer coefficient, h_{eff}. The boundary condition follows Newton's law of cooling, so heat removal scales with surface area and the temperature difference between the package surface and some reference coolant temperature. At the package level, $P \approx h_{\text{eff}} A (T_{\text{surf}} - T_{\text{ref}})$, where P is dissipated power, A is the cooled surface area, T_{surf} is the package surface temperature, and T_{ref} is the coolant ambient reference used by the solver.

h_{eff} is a lumped parameter that captures the net strength of the external cooling path, including cold plate or heatsink design, flow rate, internal convection effectiveness, contact geometry and thermal convection quality, and any additional spreading resistance between the stack and the cooler. This abstraction lets us compare packaging and materials choices under the same assumed cooling capability without tying results to one specific cooling solution.

To choose a representative baseline, we back-calculate h_{eff} using an NVIDIA A100 GPU as a reference point. The coolant is assumed to be at an ambient temperature of 45°C. Using the parameters from Table III and an assumed 38°C package to coolant temperature drop gives $h_{\text{eff}} \approx 7$ kW/(m^{2}K), which we use as our baseline. We also test higher values, for example 10 kW/(m^{2}K), to represent stronger liquid cooling and quantify how improved heat extraction changes throttling and end to end runtime.

D. Workloads

We consider two machine learning models as workloads for our training and inference runs: Llama2-7B [24] [25] and

Parameter	Value
Power (W)	400
Package area (mm^2)	1500
Heat flux (kW/m^2)	267
Temperature drop (K)	38
HTC ($kW/(m^2K)$)	7

TABLE III: Thermal Parameters of Our NVIDIA A100 Inspired GPU System.

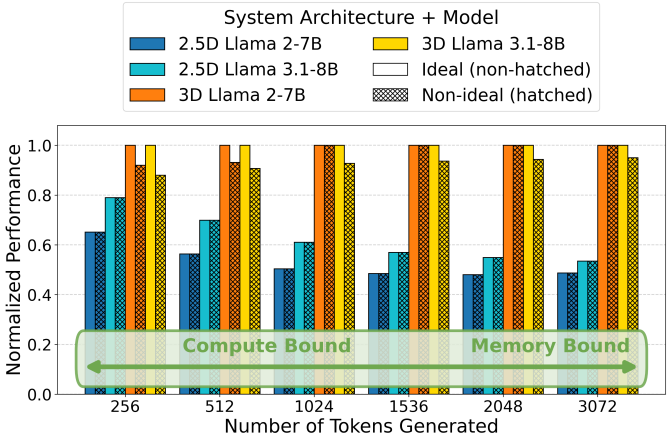


Fig. 6: Inference Performance vs. Generated/Decode Tokens for 2.5D and 3D Systems. Performance is normalized to the ideal (non-throttled) 3D case for the corresponding LLM workload for each number of tokens. The 3D case here is memory stacked on top of compute.

Llama3.1-8B [26] [27]. We used sequence length of 1024 and global batch size of 1 for training runs, and sequence length of 4096 and batch size of 32 for inference runs. We look at both training and inference to showcase thermal effects on more and less memory bound workloads. Llama3.1-8B is less memory bound during inference than Llama2-7B.

IV. Results

In this section, we present the performance impact of changing thermal conductivities, chiplet stack configurations, and workloads.

A. Performance Scaling Across Workloads

We compare 2.5D and 3D under our baseline thermal configuration across a range of inference workloads, shown in Figure 6. Each point is normalized to the ideal (non-throttled) 3D performance for the corresponding workload and generated-token count, so the plot directly shows the relative gap between 2.5D and 3D.

Compute vs. Memory Bound Behavior: Compute-bound points benefit less from 3D's higher GPU-to-HBM bandwidth than memory-bound points. Decode becomes increasingly memory bound as the number of generated tokens grows because KV-cache traffic increases. As a result, 3D benefits most at long decode lengths, while 2.5D saturates at lower normalized performance.

Model Dependence: Llama 3.1-8B uses grouped-query attention (GQA), which reduces KV-cache bandwidth pressure and increases arithmetic intensity relative to Llama 2-7B. This narrows the relative benefit of 3D bandwidth compared to Llama 2-7B, as memory bandwidth sensitivity is lower, and memory throttling is less impactful.

2.5D vs. 3D: For Llama 3.1-8B, higher GPU activity (and higher temperature) means none of the 3D points reach ideal performance in the baseline cooling setup. In Section IV-B, we show that 2.5D experiences very little GPU thermal throttling compared to 3D. Therefore, the gap between 2.5D and 3D in Figure 6 is driven primarily by memory-bandwidth differences rather than throttling in 2.5D.

B. Effects of Improving Material Thermal Properties

A sensitivity analysis of packaging materials reveals a clear hierarchy in thermal bottlenecks. Among the internal packaging materials evaluated, the TIM conductivity has the largest impact on performance. Our "Baseline" thermal setups use a TIM thermal conductivity of $5\,W/mK$ [28], infill material thermal conductivity of $1.6\,W/mK$, underfill thermal conductivity of $1.6\,W/mK$, and cooling solution with an HTC of $7\,kW/(m^2K)$. Improved TIM has a thermal conductivity of $50\,W/mK$, Improved infill and underfill materials use thermal conductivities of $19\,W/mK$. The results are shown in Figure 7 and Figure 8.

Dominance of TIM: Increasing TIM thermal conductivity from 5 W/mK to 50 W/mK yields a significant performance recovery of up to 6.8% for 3D-stacked configurations. This gain is due to the relative thickness of the TIM and lack of solder or copper connections through the layer.

Negligible Impact of Infill and Underfill: In contrast, enhancing the thermal conductivity of underfill and infill materials (from 1.6 W/mK to 19 W/mK) produces no measurable performance improvement. This is due to both the thinness of these layers and the presence of highly conductive vertical interconnects. In our 3D stack model, die-to-die connections are considered to be Cu pads/pillars (thermal conductivity of about 400 W/mK [29]), reducing the importance of a high-conductivity underfill. Note that copper interconnects provide substantially higher thermal conductivity than solder microbumps (thermal conductivity of about 50 W/mK [29]). An additional reason that underfill and infill thermal conductivity may have less impact than the TIM thermal conductivity is that all of the heat generated below the heat spreader is dissipated through the TIM, while lower underfill layers will have a lower heat flux than higher underfill layers due to distance from the heat spreader. Infill on the other hand is likely not on the primary heat dissipation path.

System-Level Cooling vs. Packaging: While TIM optimization is the most effective packaging material change, system level cooling enhancements can have a larger impact. Increasing the overall heat transfer coefficient (HTC) provides a performance boost of up to 12.5%, nearly eliminating thermal throttling in the 3D systems we studied.

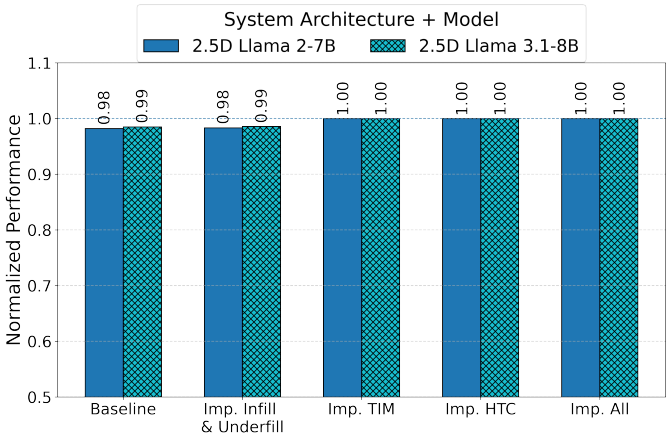


Fig. 7: Training Performance Impact of Thermal Properties for 2.5D System. Results are normalized to the ideal (non-throttled) result for the corresponding model at each point.

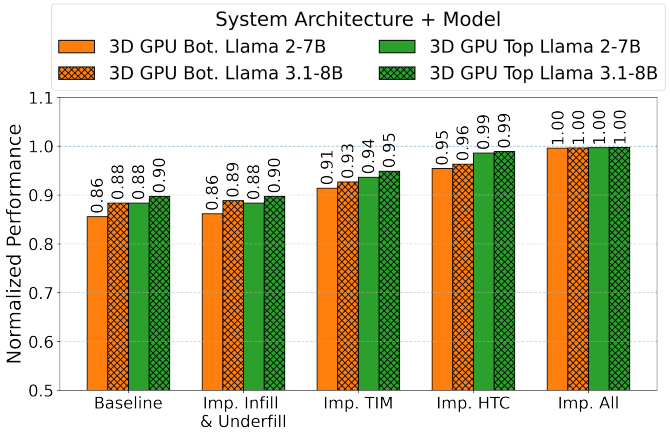


Fig. 8: Training Performance Impact of Thermal Properties for 3D System. Results are normalized to the ideal (non-throttled) result for the corresponding model. We show both 3D with memory on top and 3D with GPU on top.

These results demonstrate that STCO efforts should prioritize TIM selection and external cooling solutions over secondary material optimizations, which offer negligible returns in the current stackup architecture.

C. 3D Stacking Order

In Figure 8, we see better performance for compute stacked on top of memory than we do for memory stacked on top of compute. This is because the GPU can run at a higher clock frequency while staying in a safe operating range when it is closely coupled to the heatsink.

D. HBM Stack Height

We compared the impact of increasing the HBM stack height from 8 DRAM dies as used in the HBM3E specification to 16 DRAM dies as supported in the HBM4 specification. Our results (shown as fraction of non-throttled performance) in Figure 9 indicate that there is more thermal throttling in

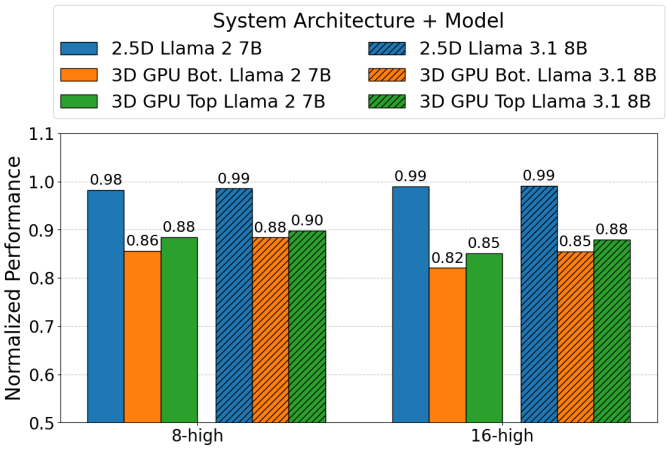


Fig. 9: Training Performance Impact of HBM Stack Height. Results are shown as the fraction of ideal performance achieved for a given model and stack height. Results are normalized to the ideal (non-throttled) result for the corresponding model and configuration. Note that while 2.5D shows less performance degradation than 3D, 3D has better ideal performance.

the case with the larger HBM stack. This is in line with our expectation that thermal considerations will take more and more prominence as stack height increases.

V. Conclusions

Advanced packaging such as 3D stacking promises increased memory bandwidth and increased performance, but thermal throttling can erase a significant portion of that benefit unless packaging, architecture, and cooling are co-designed. A closed-loop model is required to predict end-to-end performance since workloads impact average power, power impacts temperature, temperature impacts throttling, and throttling in turn impacts performance. In this work, we developed an end-to-end STCO tool to enable this analysis.

We showed that in our example systems, 2.5D suffers a small thermal tax (about 1% to 2%), while 3D suffers a large one (about 10% to 18%) under the same cooling solution. When we improve the thermal properties of materials, an improved TIM shows the largest impact while underfill and infill improvements show negligible benefit in this stackup. Using a better cooling solution with a higher HTC is the most effective change we tested and can nearly eliminate thermal throttling for a 3D stacked system.

Compute bound workloads suffer more from thermal throttling than memory bound workloads and stacking compute on top of memory provided better performance than stacking memory on top of compute in our examples.

In future work, we plan to add support for transient analysis and evaluate larger systems with multiple GPUs or accelerators in a single package.

VI. Acknowledgements

The authors would like to thank David Ratchkov from Anemoi software for his valuable assistance in modeling the thermal systems presented in this paper.

This work was supported in part by CHIMES, one of the seven centers in JUMP 2.0, a Semiconductor Research Corporation (SRC) program sponsored by DARPA, NSF, and ASML.

References

[1] NVIDIA Corporation, "NVIDIA A100 Tensor Core GPU," https://www.nvidia.com/en-us/data-center/a100/, 2022, accessed: 2026-02-19.

[2] ——, "NVIDIA H100 Tensor Core GPU," https://www.nvidia.com/en-us/data-center/h100/, 2024, accessed: 2026-02-19.

[3] ——, "NVIDIA DGX B200: The Foundation for Your AI Factory," https://www.nvidia.com/en-us/data-center/dgx-b200/, 2025, accessed: 2026-02-19.

[4] H. Wen and W. Zhang, "Exploiting gpu with 3d stacked memory to boost performance for data-intensive applications," in *2018 IEEE High Performance extreme Computing Conference (HPEC)*, 2018, pp. 1–6.

[5] A. Smith, E. Chapman, C. Patel *et al.*, "11.1 amd instincttm mi300 series modular chiplet package – hpc and ai accelerator for exa-class systems," in *2024 IEEE International Solid-State Circuits Conference (ISSCC)*, vol. 67, 2024, pp. 490–492.

[6] H. Jiang, "Intel's ponte vecchio gpu : Architecture, systems software," in *2022 IEEE Hot Chips 34 Symposium (HCS)*, 2022, pp. 1–29.

[7] P. Shukla, A. K. Coskun, V. F. Pavlidis *et al.*, "An overview of thermal challenges and opportunities for monolithic 3d ics," in *Proceedings of the 2019 Great Lakes Symposium on VLSI*, ser. GLSVLSI '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 439–444. [Online]. Available: https://doi.org/10.1145/3299874.3319485

[8] S. Pandey and P. R. Panda, "Neuromap: Efficient task mapping of deep neural networks for dynamic thermal management in high-bandwidth memory," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 11, pp. 3602–3613, 2022.

[9] G. Karfakis, M. Bouzidi, Y. Im *et al.*, "Optimizing thermal performance in 2.5d systems using embedded isolators," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 15, no. 3, pp. 458–468, 2025.

[10] G. Karfakis, F. Tahmasebi, B. Chen *et al.*, "RAPID-LLM: resilience-aware performance analysis of infrastructure for distributed llm training and inference," 2025. [Online]. Available: https://arxiv.org/abs/2512.19606

[11] *Ansys Icepak*, Ansys, Inc., accessed: 2026-02-19. [Online]. Available: https://www.ansys.com/products/electronics/ansys-icepak

[12] Anemoi Software Inc., "DankaThermal." [Online]. Available: https://anemoisoftware.com

[13] A. Graening, J. Talukdar, S. Pal *et al.*, "Catch: A cost analysis tool for co-optimization of chiplet-based heterogeneous systems," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 1–1, 2025.

[14] JEDEC Solid State Technology Association, "Ddr3 sdram specification," JEDEC Solid State Technology Association, Standard, 2012.

[15] ——, "High bandwidth memory dram (hbm3)," JEDEC Solid State Technology Association, Standard JESD238A, January 2023.

[16] ——, "High bandwidth memory dram (hbm1, hbm2)," JEDEC Solid State Technology Association, Standard JESD235D, February 2021.

[17] *High Bandwidth Memory (HBM2) Interface Intel® FPGA IP User Guide*, Intel Corporation, March 2020, accessed: 2026-02-19. [Online]. Available: https://docs.altera.com/r/docs/683189/21-2-19-6-1/high-bandwidth-memory-hbm2-interface-intel-fpga-ip-user-guide/download-document

[18] L. Siddhu, R. Kedia, and P. R. Panda, "Corememdtm: Integrated processor core and 3d memory dynamic thermal management for improved performance," in *2022 Design, Automation Test in Europe Conference Exhibition (DATE)*, 2022, pp. 1377–1382.

[19] S. Pandey, L. Siddhu, and P. R. Panda, "Neurocool: Dynamic thermal management of 3d dram for deep neural networks through customized prefetching," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 29, no. 1, Dec. 2023. [Online]. Available: https://doi.org/10.1145/3630012

[20] J. Liu, B. Jaiyen, R. Veras *et al.*, "Raidr: Retention-aware intelligent dram refresh," in *2012 39th Annual International Symposium on Computer Architecture (ISCA)*, 2012, pp. 1–12.

[21] Micron Technology, Inc., "HBM3E — Micron Technology Inc." https://www.micron.com/products/memory/hbm/hbm3e, 2026, accessed: 2026-02-21.

[22] N. Kamath, "Hbm4 elevates ai training performance to new heights," https://semiengineering.com/hbm4-elevates-ai-training-performance-to-new-heights/, Semiconductor Engineering, May 2025, accessed: 2026-02-21.

[23] D. Das Sharma, G. Pasdast, S. Tiagaraj *et al.*, "High-performance, power-efficient three-dimensional system-in-package designs with universal chiplet interconnect express," *Nature Electronics*, vol. 7, pp. 244–254, Mar. 2024. [Online]. Available: https://doi.org/10.1038/s41928-024-01126-y

[24] H. Touvron, L. Martin, K. Stone *et al.*, "Llama 2: Open foundation and fine-tuned chat models," 2023. [Online]. Available: https://arxiv.org/abs/2307.09288

[25] Meta, "Llama 2 7B (meta-llama/llama-2-7b) model card," https://huggingface.co/meta-llama/Llama-2-7b, 2023, hugging Face model repository. Model card notes training between January 2023 and July 2023.

[26] A. Grattafiori, A. Dubey, A. Jauhri *et al.*, "The llama 3 herd of models," 2024. [Online]. Available: https://arxiv.org/abs/2407.21783

[27] Meta, "Llama 3.1 8B (meta-llama/llama-3.1-8b) model card," https://huggingface.co/meta-llama/Llama-3.1-8B, 2024, hugging Face model repository. Model release date: July 23, 2024.

[28] *DOWSIL™ TC-5550 Thermal Conductive Compound Technical Data Sheet*, Dow Chemical Company, 2022, accessed: 2026-02-19. [Online]. Available: https://www.ulbrich.cz/chemical-technical-products/TDS_DOWSIL_TC_5550_Thermal_Conductive_Compound_eng.pdf

[29] The Engineering ToolBox, "Thermal conductivity of metals, metallic elements and alloys," https://www.engineeringtoolbox.com/thermal-conductivity-metals-d_858.html, 2005, accessed: 2026-02-26.