# Microbenchmarking NVIDIA's Blackwell Architecture

*An in-depth Architecture Analysis*

**Aaron Jarmusch,** Sunita Chandrasekaran
<jarmusch@udel.edu>
*University of Delaware*

**IPDPS**
**May 25–29, 2026**
New Orleans, LA

# Motivation

**GPU hardware evolves faster than the methods used to measure it.**

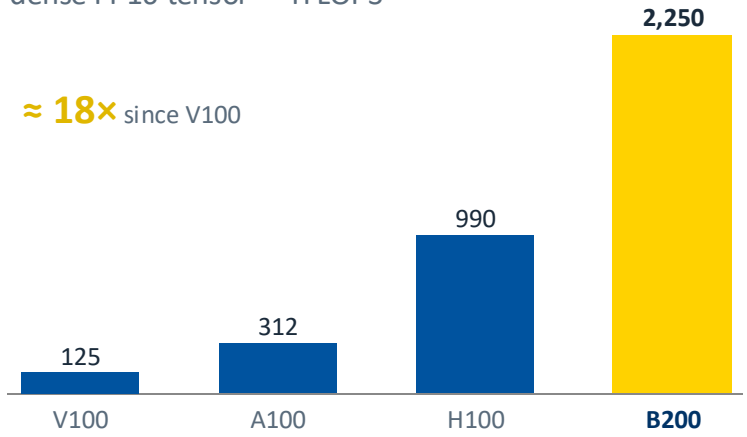
Each recent architecture — Volta, Turing, Hopper — has needed fresh microbenchmarking to expose what vendor documentation leaves out.

*Blackwell is the largest break in years, and public characterization hasn't caught up.*

## Tensor throughput by generation

dense FP16 tensor · TFLOPS

≈ 18× since V100




*per-GPU tensor-core peak · NVIDIA published specs*

# Motivation

**The B200 reshapes how kernels should be written.**

The B200 adds several major changes: a dual-chip design, Tensor Memory, a hardware decompression engine, and new sub-byte precisions. Public characterization hasn't caught up with the hardware.

*Without measured numbers, it's hard to tell which features matter for a given workload.*

**5th** — **Generation Tensor Cores**
*tcgen05.mma PTX path*

**FP4** — **Sub-byte precision**
*alongside FP6 / FP8*

**TMEM** — **Tensor Memory**
*new on-chip storage tier*

# Open questions for Blackwell

Three concrete questions remain about the B200's new units.

1. **What do TMEM, the DE, and dual-chip layout cost?**

   No public reference numbers exist for any of them.

2. **What is the accuracy cost of FP4 and FP6?**

   Throughput is well documented. Accuracy on real kernels is not.

3. **How does tcgen05.mma change kernel design vs. wgmma?**

   The shift from warp-group to warp-level execution requires restructuring GEMM kernels.

# Contributions

In this work, we present:

**C1** **An open-source microbenchmark suite for B200**

covering tensor cores, TMEM, the decompression engine, and precisions.

**C2** **TMEM behavior on GEMM-heavy workloads**

how TMEM offloads pressure from registers and shared memory.

**C3** **Decompression engine (DE) characterization**

throughput across formats and where it helps in practice.

**C4** **tcgen05.mma latency and throughput**

latency, tile-size scaling, and warp-level execution implications.

**C5** **FP4 / FP6 throughput vs accuracy**

throughput gains against accuracy loss on real kernels.

**C6** **Case studies on real workloads**

LLM inference and training plus HPC kernels, with porting guidelines.

# Background: Blackwell B200

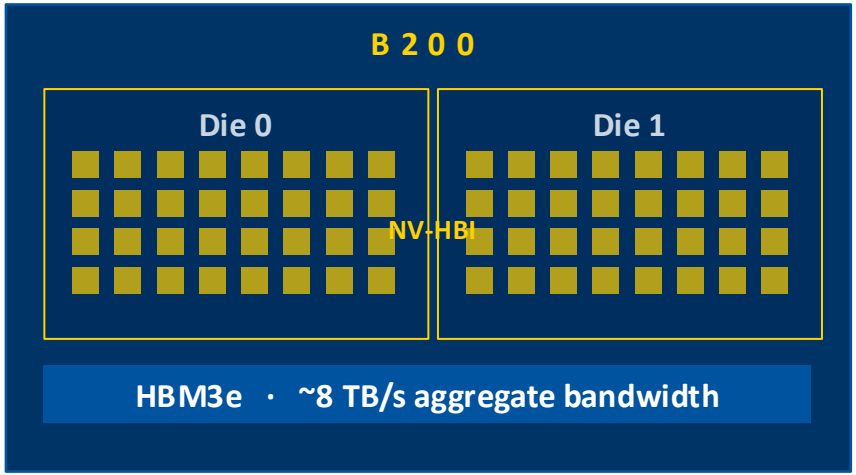



**FIG. 1** — *simplified B200 package layout.*

## Key architectural advances

- **Dual-chip design**
  Two reticle-limited dies fused via high-bandwidth NV-HBI interconnect.
- **Tensor Memory (TMEM)**
  On-chip storage tier purpose-built for tensor-core operands and accumulators.
- **5th-gen Tensor Cores**
  Warp-level tcgen05.mma replaces warp-group wgmma; new pipeline semantics.
- **Hardware decompression**
  Dedicated engine for compressed memory streams to relieve bandwidth pressure.
- **FP4 / FP6 precisions**
  Sub-byte tensor types alongside FP8 to push throughput on attention & GEMM.

# Methodology


### Targeted kernels

PTX-level probes for memory, tensor, and DE paths.

### Dependency chains

Carry-chain microbenchmarks isolate single-instruction latency.

### Sweep & compare

Vary tile, precision, and footprint; A/B against H200, same toolkit.

### Full-app studies

Mistral, ResNet-50, GPT-1.3B, DGEMM, SpMV, STREAM.

# Implementation

How the suite is built and run:

## 01 Software stack

CUDA 12.6 / driver 560; PyTorch 2.4 + Transformer Engine, nvCOMP, and custom FP64 kernels.

## 02 Measurement protocol

10-iter warmup, 100-iter average (1000 on the DE); median, P95, and P99 on latency.

## 03 Validation

PTX checked against SASS; Nsight Compute counters for stalls, utilization, L2 hit rates.

## 04 Reproducibility

Open-source suite on GitHub; identical toolchain on B200 and H200 over shared HBM3e.

# tcgen05.mma vs Hopper wgmma

- Blackwell holds latency at 11 cycles across tile sizes
- Hopper wgmma scales from 38 to 132 cycles
- Speedup ranges from 2.9× to 11.2×, growing with tile size
- Measured with an FP16, accumulator-carried dependency chain

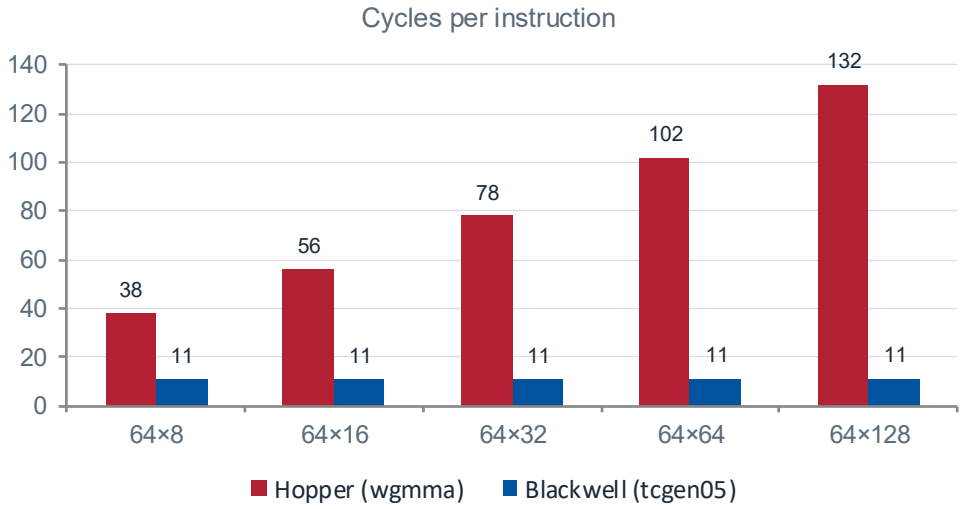
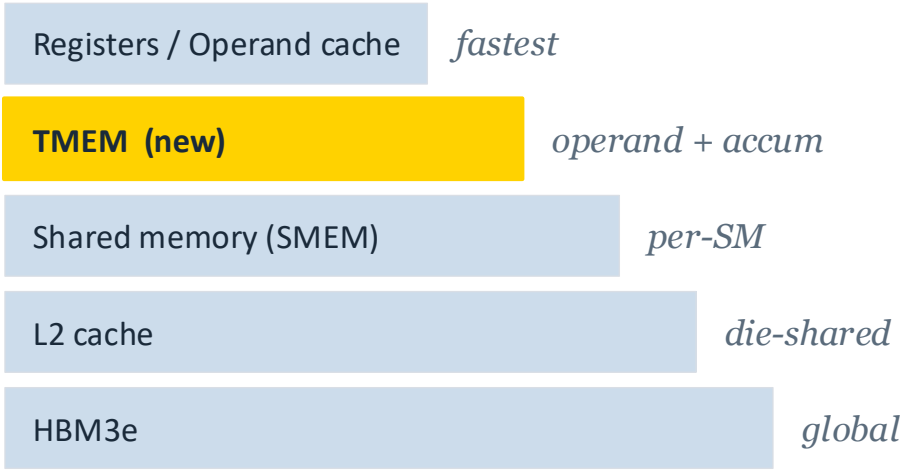



**FIG. 2** — *single-instruction latency vs tile size.*

***On Blackwell, tile size affects throughput, not per-instruction latency.***

# Tensor Memory in the GEMM pipeline

## Memory hierarchy (B200)




## Observations

**256 KB per SM**
Dedicated 2D array: 512 columns × 128 lanes of 32-bit cells.

**64×64 tile sweet spot**
Peak efficiency at 64×64; 16 TB/s read bandwidth in chained ops.

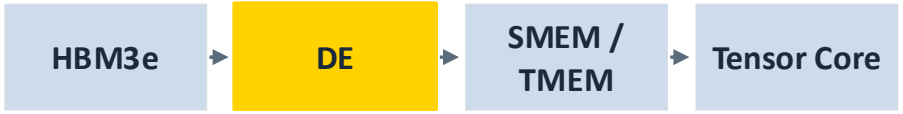
**~12 TB/s traffic avoided**
For chained matmuls D = (A·B)·C, keeping intermediates in TMEM avoids that much off-chip traffic per SM.

**1.65× attention block speedup**
Fused attention: 284 µs on B200 vs 468 µs on H200 (Table X).

**FIG. 3** — *TMEM's position in the memory hierarchy.*

# Hardware decompression engine (DE)

## Modified data path




## Findings

*i.* Helps memory-bound kernels (inference attention, embedding lookups, sparse tables).

*ii.* Throughput varies across compression formats.

*iii.* Diminishing returns when the kernel is compute-bound.

## DE throughput across formats · GB/s (illustrative)

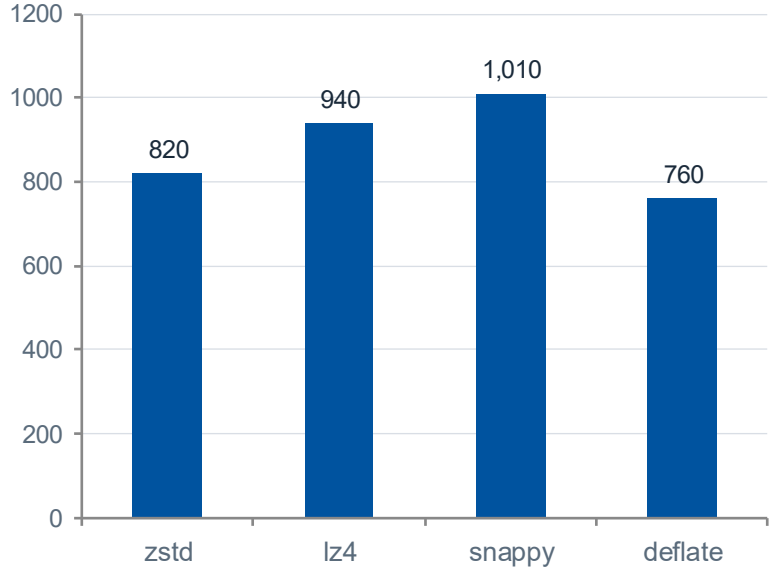



**FIG. 5** — *DE throughput by format.*

# FP4 / FP6: throughput vs accuracy

**Relative tensor throughput (FP16 = 1.00×)**

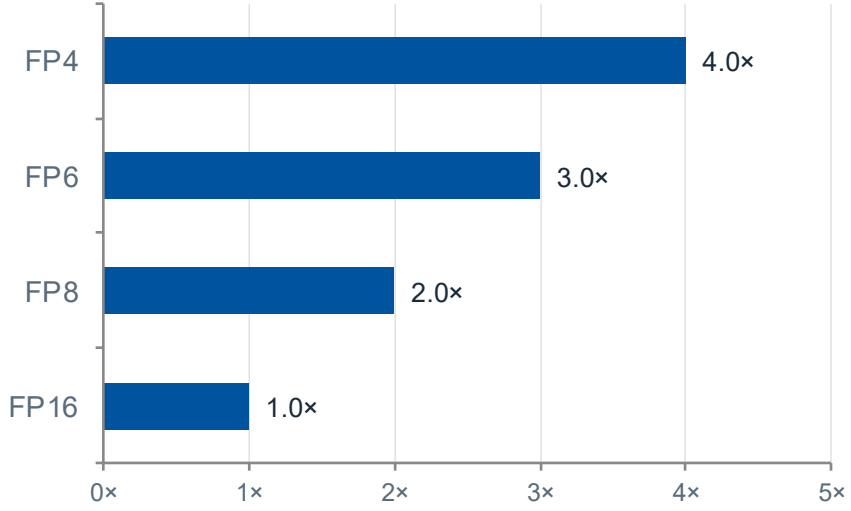



**FIG. 4a** — *throughput by precision.*

**Accuracy preservation**

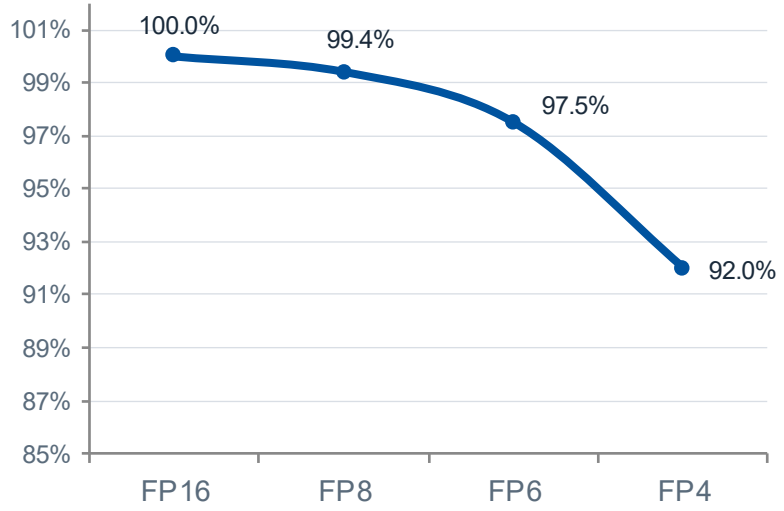



**FIG. 4b** — *accuracy by precision.*

***FP4 unlocks new model scales on a single GPU.***

# Case studies: B200 vs H200

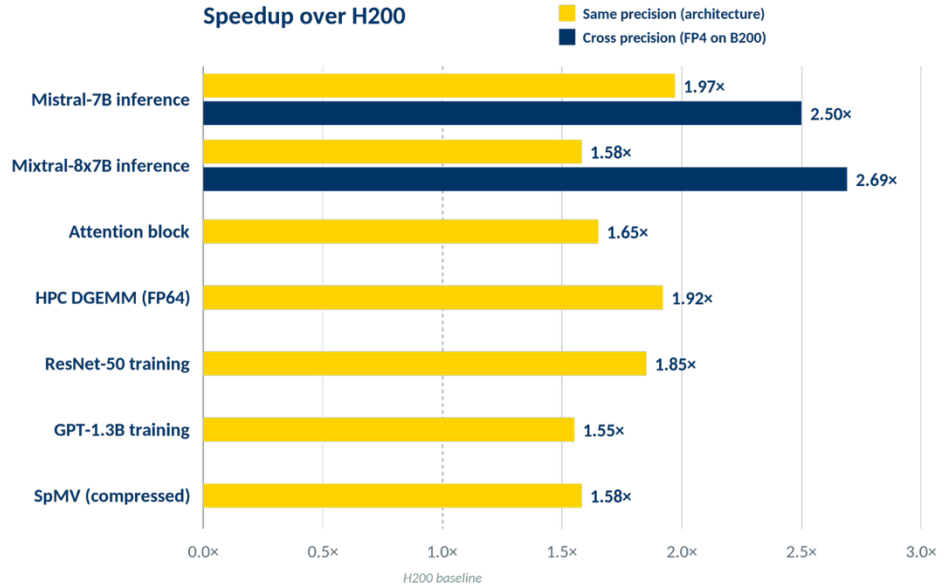



**FIG. 6** — *measured on B200 vs H200, paper Tables VIII/X/XI.*

## Per-workload notes

**FP4 path**
Cross-precision wins (2.5–2.7×) come from B200 FP4 vs H200 FP16.

**TMEM**
1.65× attention block: $Q \cdot K^T$ stays resident across softmax.

**FP64 units**
1.92× DGEMM: doubled FP64 path, separate from tcgen05.

**Decompression eng.**
1.58× SpMV: removes software inflate overhead.

*Same precision: 1.5–1.9× from architecture alone. Cross precision: 2.5–2.7× unlocked by FP4 on B200.*

# Performance guidelines for B200

**01** **Tile size affects throughput, not latency**

*tcgen05.mma latency is flat across tile sizes. Pick tiles to feed the pipeline.*

**02** **Plan TMEM lifetime alongside SMEM**

*Schedule accumulators in TMEM; reserve SMEM for staging the next operand tile.*

**03** **Choose precision per tensor**

*FP6 often dominates on attention; FP8 still wins on weight-heavy GEMM.*

**04** **Use the DE for memory-bound regions**

*Compressed weight streams help in bandwidth-limited kernels.*

**05** **Re-derive your roofline**

*Hopper-era roofline assumptions don't hold with dual-chip and new precisions.*

**06** **Benchmark both dies**

*Cross-die traffic via NV-HBI shows up in placement-sensitive workloads.*

CONCLUSION

# Blackwell changes the GPU pipeline.

- Open-source microbenchmark suite for the B200
- Measured TMEM, tcgen05, DE, and FP4 / FP6 trade-offs
- Guidelines for porting workloads from Hopper


ACKNOWLEDGMENTS

**Funding** — *DOE S4PST – DE-ACO5-000R22725*

**Compute** — *University of Oregon Frankenstein Cluster &` NVIDIA Brev*



*jarmusch@udel.edu*


arXiv : 2512.02189

Code · GitHub

UNIVERSITY OF DELAWARE

# References (1 / 4)

**[1]** NVIDIA Corporation, "NVIDIA Blackwell Architecture Technical Brief: Powering the New Era of Generative AI and Accelerated Computing," Tech. Brief, Mar. 2024.

**[2]** NVIDIA Corporation, "NVIDIA H100 Tensor Core GPU Architecture," White Paper, Mar. 2022.

**[3]** NVIDIA Corporation, "NVIDIA Blackwell B200 Datasheet," 2024.

**[4]** NVIDIA Corporation, "NVIDIA Blackwell Tuning Guide," 2025. [Online]. Available: https://docs.nvidia.com/cuda/blackwell-tuning-guide/

**[5]** NVIDIA Corporation, "NVIDIA RTX Blackwell GPU Architecture," 2025.

**[6]** NVIDIA Corporation, "Parallel Thread Execution (PTX) ISA, Release 8.8," 2025.

**[7]** NVIDIA Corporation, "CUDA Binary Utilities — Instruction Set Reference," 2025.

**[8]** NVIDIA Corporation, "NVIDIA TensorRT," v. 10.0, 2024.

**[9]** H. Wong, M.-M. Papadopoulou, M. Sadooghi-Alvandi, and A. Moshovos, "Demystifying GPU microarchitecture through microbenchmarking," in IEEE ISPASS, 2010, pp. 235–246.

**[10]** V. Volkov and J. W. Demmel, "Benchmarking GPUs to tune dense linear algebra," in SC, 2008.

**[11]** B. R. Coutinho, G. L. M. Teodoro, R. S. Oliveira, D. O. G. Neto, and R. A. C. Ferreira, "Profiling General Purpose GPU Applications," in SBAC-PAD, 2009, pp. 11–18.

**[12]** S. Hong and H. Kim, "An analytical model for a GPU architecture with memory-level and thread-level parallelism awareness," SIGARCH Comput. Archit. News, vol. 37, no. 3, pp. 152–163, Jun. 2009.

*Backup material — not used during the talk.*

# References (2 / 4)

**[13]** S. Subramoniapillai Ajeetha, "Architectural Analysis and Performance Characterization of NVIDIA GPUs using Microbenchmarking," Ph.D. dissertation, Ohio State Univ., 2012.

**[14]** W. Jia, K. A. Shaw, and M. Martonosi, "Characterizing and improving the use of demand-fetched caches in GPUs," in ICS, 2012, pp. 15–24.

**[15]** X. Mei and X. Chu, "Dissecting GPU Memory Hierarchy Through Microbenchmarking," IEEE TPDS, vol. 28, no. 1, pp. 72–86, 2017.

**[16]** X. Zhang, G. Tan, S. Xue, J. Li, K. Zhou, and M. Chen, "Understanding the GPU Microarchitecture to Achieve Bare-Metal Performance Tuning," in PPoPP, 2017, pp. 31–43.

**[17]** Z. Jia, M. Maggioni, B. Staiger, and D. P. Scarpazza, "Dissecting the NVIDIA Volta GPU Architecture via Microbenchmarking," arXiv:1804.06826, 2018.

**[18]** Z. Jia, M. Maggioni, J. Smith, and D. P. Scarpazza, "Dissecting the NVIDIA Turing T4 GPU via Microbenchmarking," arXiv:1903.07486, 2019.

**[19]** W. Luo, R. Fan, Z. Li, D. Du, H. Liu, Q. Wang, and X. Chu, "Dissecting the NVIDIA Hopper Architecture through Microbenchmarking and Multiple Level Analysis," arXiv:2501.12084, 2025.

**[20]** R. Huerta, M. A. Shoushtary, J.-L. Cruz, and A. González, "Analyzing Modern NVIDIA GPU Cores," arXiv:2503.20481, 2025.

**[21]** S. Markidis, S. W. D. Chien, E. Laure, I. B. Peng, and J. S. Vetter, "NVIDIA Tensor Core Programmability, Performance & Precision," in IEEE IPDPSW, 2018, pp. 522–531.

**[22]** M. Martineau, P. Atkinson, and S. McIntosh-Smith, "Benchmarking the NVIDIA V100 GPU and Tensor Cores," in Euro-Par 2018 Workshops, Springer, 2019, pp. 444–455.

**[23]** D. Yan, W. Wang, and X. Chu, "Demystifying Tensor Cores to Optimize Half-Precision Matrix Multiply," in IEEE IPDPS, 2020, pp. 634–643.

**[24]** W. Sun, A. Li, T. Geng, S. Stuijk, and H. Corporaal, "Dissecting Tensor Cores via Microbenchmarks: Latency, Throughput and Numeric Behaviors," IEEE TPDS, vol. 34, no. 1, pp. 246–261, 2023.

*Backup material — not used during the talk.*

# References (3 / 4)

**[25]** M. Fasi, N. J. Higham, M. Mikaitis, and S. Pranesh, "Numerical behavior of NVIDIA tensor cores," PeerJ Comput. Sci., vol. 7, p. e330, 2021.

**[26]** M. A. Raihan, N. Goli, and T. M. Aamodt, "Modeling Deep Learning Accelerator Enabled GPUs," in IEEE ISPASS, 2019, pp. 79–92.

**[27]** M. Khairy, Z. Shen, T. M. Aamodt, and T. G. Rogers, "Accel-Sim: An Extensible Simulation Framework for Validated GPU Modeling," in ISCA, 2020, pp. 473–486.

**[28]** J. Lee, Y. Ha, S. Lee, J. Woo, J. Lee, H. Jang, and Y. Kim, "GCoM: a detailed GPU core model for accurate analytical modeling of modern GPUs," in ISCA, 2022, pp. 424–436.

**[29]** T. T. Dao, J. Kim, S. Seo, B. Egger, and J. Lee, "A Performance Model for GPUs with Caches," IEEE TPDS, vol. 26, no. 7, pp. 1800–1813, 2015.

**[30]** K. N. M. Nguyen, H. D. N. Do, H. T. Le, and T. T. Dao, "LLMPerf: GPU Performance Modeling meets Large Language Models," arXiv:2503.11244, 2025.

**[31]** M. Leinhauser, R. Widera, S. Bastrakov, A. Debus, M. Bussmann, and S. Chandrasekaran, "Metrics and Design of an Instruction Roofline Model for AMD GPUs," arXiv:2110.08221, 2021.

**[32]** L. Fusco, M. Khalilov, M. Chrapek, G. Chukkapalli, T. Schulthess, and T. Hoefler, "Understanding Data Movement in Tightly Coupled Heterogeneous Systems: A Case Study with the Grace Hopper Superchip," arXiv:2408.11556, 2024.

**[33]** J. D. McCalpin, "Memory Bandwidth and Machine Balance in Current High Performance Computers," IEEE Comput. Soc. TCCA Newsletter, pp. 19–25, Dec. 1995.

**[34]** G. Tan, L. Li, S. Triechle, E. Phillips, Y. Bao, and N. Sun, "Fast implementation of DGEMM on Fermi GPU," in SC, 2011.

**[35]** B. D. Rouhani et al., "Microscaling Data Formats for Deep Learning," arXiv:2310.10537, 2023.

**[36]** T. Dettmers and L. Zettlemoyer, "The case for 4-bit precision: k-bit Inference Scaling Laws," arXiv:2212.09720, 2023.

*Background material not discussed during the talk.*

# References (4 / 4)



**[37]** B. Chmiel, M. Fishman, R. Banner, and D. Soudry, "FP4 All the Way: Fully Quantized Training of LLMs," arXiv:2505.19115, 2025.

**[38]** A. Q. Jiang et al., "Mistral 7B," arXiv:2310.06825, 2023.

**[39]** S. Black et al., "GPT-NeoX-20B: An Open-Source Autoregressive Language Model," in Proc. BigScience Workshop, ACL, 2022, pp. 95–136.

**[40]** L. Gao et al., "The Pile: An 800GB Dataset of Diverse Text for Language Modeling," arXiv:2101.00027, 2020.

**[41]** K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," arXiv:1512.03385, 2015.

**[42]** Y. Ding et al., "LongRoPE: Extending LLM Context Window Beyond 2 Million Tokens," arXiv:2402.13753, 2024.

**[43]** D. Kevian et al., "Capabilities of Large Language Models in Control Engineering," arXiv:2404.03647, 2024.

**[44]** H. Chen, J. D. C. Maia, B. K. Radak, D. J. Hardy, W. Cai, C. Chipot, and E. Tajkhorshid, "Boosting Free-Energy Perturbation Calculations with GPU-Accelerated NAMD," J. Chem. Inf. Model., vol. 60, no. 11, pp. 5301–5307, 2020.

**[45]** Y. Wang, D. Hait, P. A. Unzueta, J. H. Zhang, and T. J. Martínez, "Fast and Scalable GPU-Accelerated Quantum Chemistry for Periodic Systems with Gaussian Orbitals," arXiv:2410.22278, 2024.

**[46]** R. Kelly, "GPU Computing for Atmospheric Modeling," Comput. Sci. Eng., vol. 12, no. 4, pp. 26–33, 2010.

**[47]** M. S. Nobile, P. Cazzaniga, A. Tangherloni, and D. Besozzi, "Graphics Processing Units in Bioinformatics, Computational Biology and Systems Biology," Briefings in Bioinformatics, vol. 18, no. 5, pp. 870–885, 2017.

*Backup material — not used during the talk.*