Training-Free Mixture of Agents for CUDA Kernel Translation and Optimization

Oak Ridge National Laboratory · Iowa State University · SC26 Chicago Nov 15–20 · Genesis Mission · U.S. Department of Energy

Motivation

GPU kernel generation and optimization require more than one LLM

- **Single-model CUDA/HIP generation is brittle:** correctness, launch configuration, indexing, and performance interact tightly
- **Optimization is not one decision;** kernels need memory coalescing, tiling, atomics/reductions, loop restructuring, and profiling feedback
- High-performance computing (HPC) deployment also requires visibility into **token cost, latency, GPU power, and which agent influenced the final code**

Research Question

Can small, open, local agents collaborate to produce correct and optimized GPU kernels while exposing cost and attribution?

System Overview

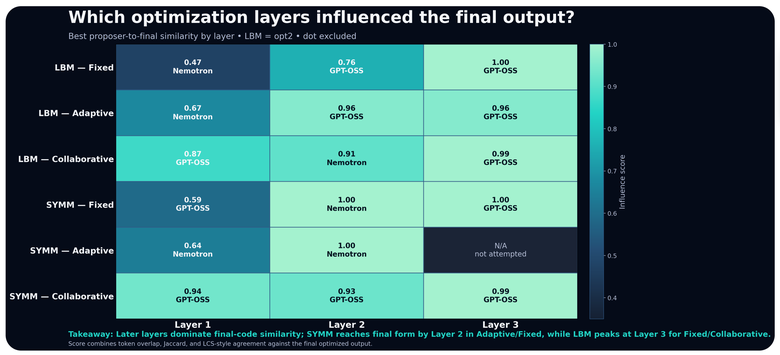
Accelerating production of AI assistants for HPC

- Nemotron 3 Nano 30B-A3B: reasoning model with the ability to switch reasoning on/off
- Llama 3.1 7B: dense model with no reasoning
- GPT-OSS 20B: powerful open-source reasoning model
- Translation layer: serial or kernel code → executable GPU kernel candidates
- Optimization layer: repeated mixture-of-agents edits under fixed, adaptive, or collaborative control
- Instrumentation: token ledger, inference latency, NVIDIA Management Library power sampling, CUDA-event timing, and Nsight Compute hooks

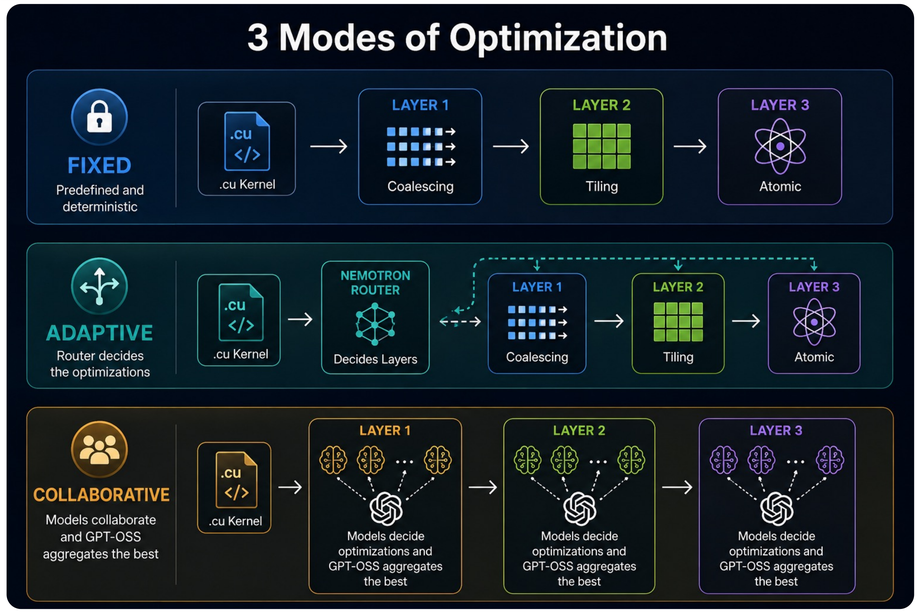
Model Layer Influence

Superior results faster, at lower cost, and with modest resources

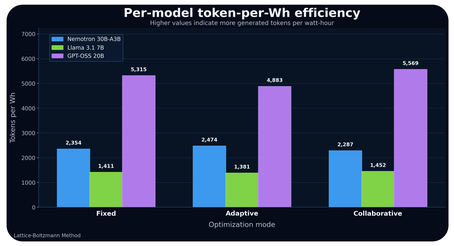
Higher influence scores indicate stronger structural similarity between an intermediate layer and the final optimized kernel. The results suggest that the lattice-Boltzmann method (LBM) benefits from deeper staged refinement, whereas SYMM converges earlier, with Layer 2 producing the dominant final transformation.

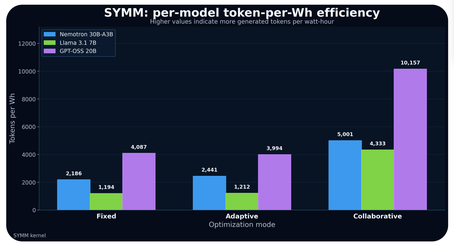


Three Optimization Methods: What Each Does



Token-per-Wh Efficiency





Token-per-Wh efficiency indicates how much optimization reasoning is produced per unit energy. GPT-OSS achieves the highest efficiency across both LBM and SYMM, while SYMM shows the clearest gain under collaborative optimization, indicating that multi-agent aggregation can improve energy-normalized reasoning throughput.

Contact Us!

Arijit Bhattacharjee: arbhatt9@iastate.edu
Pedro Valero-Lara: valerolarap@ornl.gov
Ali Jannesari: jannesar@iastate.edu
William F. Godoy: godoywf@ornl.gov
Keita Teranishi: teranishik@ornl.gov
Jeffrey S. Vetter: vetter@ornl.gov

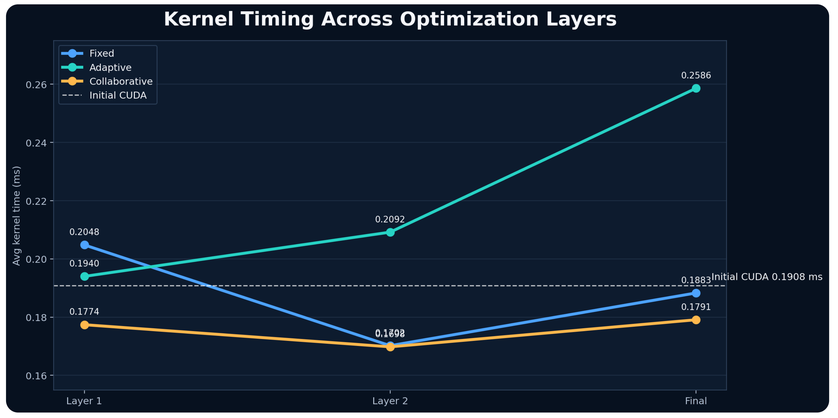
Acknowledgments: *This research used resources of the Oak Ridge Leadership Computing Facility and the Experimental Computing Laboratory at Oak Ridge National Laboratory, which is supported by the US Department of Energy's (DOE's) Office of Science under contract DE-AC05-00OR22725. This material is based upon work supported by the DOE Office of Science's Advanced Scientific Computing Research program as part of the Advancements in Artificial Intelligence for Science program, Ellora and Durban projects.*

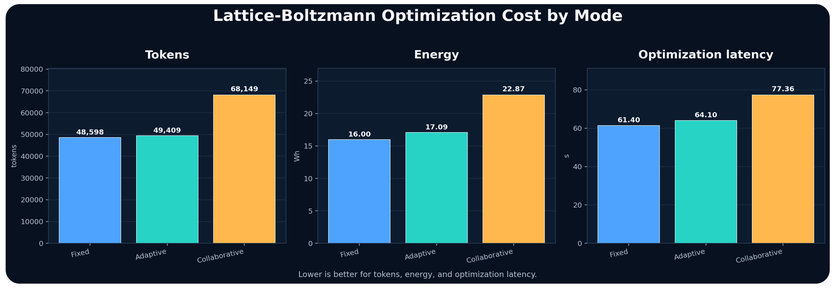
Conclusions

More than a code optimization framework

Kernel-aware adaptive optimization provides the best balance between performance and cost. LBM benefits from deeper layer refinement, whereas SYMM converges earlier after coalescing and tiling, showing that effective GPU optimization should stop once additional layers no longer improve the final kernel.

Lattice-Boltzmann Kernel





SYMM Kernel

