OpenEvolve-Driven Selection and Tuning of Ozaki-Scheme GPU Matrix Multiplication

Connor Vitz
University of Delaware
Newark, US

Aaron Jarmusch
University of Delaware
Newark, US

Sunita Chandrasekaran
University of Delaware
Newark, US

***Abstract*—Ozaki-Scheme is an established numerical method for emulating high-precision GEMM on low-precision GPU hardware. Using it means navigating a costly accuracy/performance tradeoff across algorithm variants and their tuning parameters. This work automates that process with a multi-stage pipeline combining rule-based variant selection with an LLM-driven evolutionary tuner (OpenEvolve). New features include a monotonicity-constrained XGBoost classifier predicting whether an accuracy target is reachable before spending GPU time and a throughput-priority mode trading bounded accuracy for up to 106.6% higher GFLOPS. A per-island-search fix stabilizes OpenEvolve convergence, widening the best-valid range from about 45-50K to 61-108K GFLOPS. In the Quantum ESPRESSO AUSURF112 workload, native cuBLAS remains $1.21\times$ faster than the tuned complex-GEMM configuration, and the pipeline reaches 20.6$\times$ native cuBLAS on GB10 versus 1.42$\times$ on GH200. Together, these features and corresponding results further validate the pipeline while narrowing but not yet closing the gap to native cuBLAS on a real application workload.**

***Index Terms*—GPU computing, GEMM, Ozaki-I, Ozaki-II, OpenEvolve, numerical accuracy, automated optimization, feasibility classification**

I. Introduction and Contributions

GPU vendors continue to diverge on double-precision throughput: some emulate FP64 accuracy on low-precision tensor cores, while others prioritize native FP64 hardware. Ozaki-Scheme is an established numerical method [1], [2] that reconstructs FP64-accurate GEMM results from a series of low-precision GEMM operations, letting emulation-oriented GPUs stand in for scarce native FP64 throughput. Using it well requires navigating a wide accuracy/performance tradeoff space: several algorithm variants, each with tunable parameters, with no single configuration dominating across matrix shape, hardware, and accuracy target. Evaluating a candidate configuration also requires a real GPU run, so exhaustive search is expensive. Prior work in this line [3] built a multi-stage pipeline, combining rule-based variant selection with an OpenEvolve-driven [4] evolutionary tuner (Gemini 2.5 Flash [5]), that automates this selection and tuning process, showing untuned Ozaki variants reaching up to 42% higher DGEMM throughput than native cuBLAS FP64 on GH200, though gains did not consistently carry over to the Quantum ESPRESSO [6] AUSURF112 application workload.

Supported by NSF ACCESS allocation CIS250803 (NCSA DeltaAI) and DOE ASCR contract DE-AC05-00OR22725 (S4PST). Text partly drafted alongside Claude Sonnet 5 (Anthropic). The authors reviewed, edited, and take full responsibility for all content.

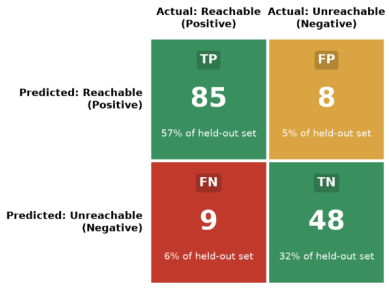


Fig. 1. Confusion matrix (true/false positives and negatives) for the feasibility classifier on held-out configurations.



Fig. 2. Normal mode versus `--throughput-priority` mode on an identical 4096^3 DGEMM, same code, one CLI flag flipped.

This work extends that pipeline with four contributions:

- **Feasibility classifier:** predicts whether a requested accuracy target is reachable before OpenEvolve spends GPU time searching for it.
- **Throughput-priority mode:** a search objective that trades bounded accuracy loss for substantially higher throughput.
- **Search-stability fix:** corrects OpenEvolve's per-island search dynamics, which stalled on one candidate.
- **Cross-platform validation:** the unmodified pipeline evaluated on a second GPU architecture, NVIDIA's GB10 [7], alongside an updated AUSURF112 benchmark.

II. Results and Observations

A monotonicity-constrained XGBoost classifier was trained to predict, from workload and accuracy-target features, whether OpenEvolve can reach a requested error bound without running the search. Evaluated on held-out configurations deliberately biased toward the decision boundary, the classifier reaches 91.4% precision (95% CI [83.9%, 95.6%]) and 90.4% recall (95% CI [82.8%, 94.9%]). Gating the search on this prediction avoids the GPU time typically spent chasing infeasible targets (Figure 1).

The `--throughput-priority` flag changes which variant the selector routes to for accuracy targets it can still satisfy, biasing toward higher throughput at bounded accuracy

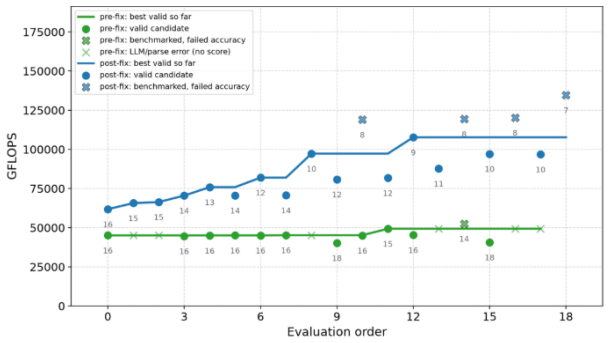


Fig. 3. OpenEvolve's `num_moduli` exploration before and after the per-island dilution fix.

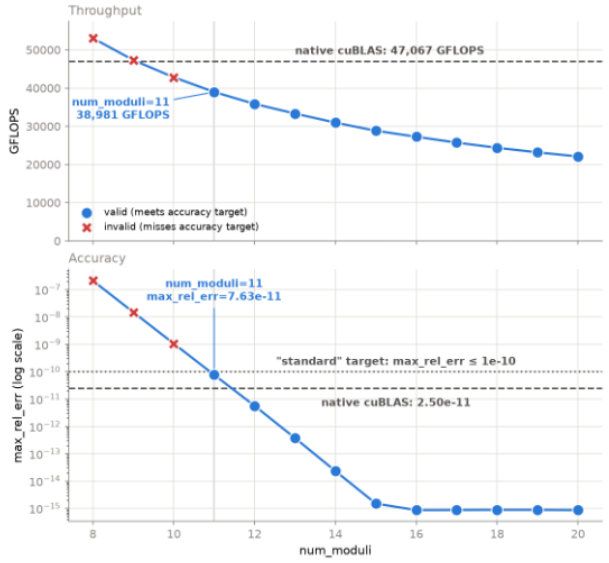


Fig. 4. Best tuned complex-GEMM candidate versus native cuBLAS on the 50-shape AUSURF112 workload profile.

cost. On an identical 4096^3 DGEMM, throughput-priority mode reaches 182,299 GFLOPS versus 88,247 GFLOPS in normal mode, a 106.6% increase, while relative error grows from 2.25e-11 to 3.02e-04 (Figure 2).

OpenEvolve's search itself required a stability fix. Before correcting a per-island dilution issue, repeated search runs got stuck proposing the same candidate (`num_moduli=16`), and the best-valid throughput found across runs stayed in a narrow band of roughly 45,000–50,000 GFLOPS. After the fix, the search explores the `num_moduli` space more broadly and steadily narrows toward better candidates, with best-valid throughput spanning roughly 61,000–108,000 GFLOPS across runs (Figure 3).

Applying the pipeline to the same complex-valued AUSURF112 workload used previously, across the top 50 GEMM shapes extracted via cuBLAS interception, the best candidate reaches 38,980.55 GFLOPS (7.63e-11 relative error) versus 47,067.14 GFLOPS for matched native cuBLAS, narrowing the gap from roughly 2.4× to 1.21× (Figure 4). This improvement reflects corrections to how the benchmark harness constructs and measures wide, complex-valued workloads.

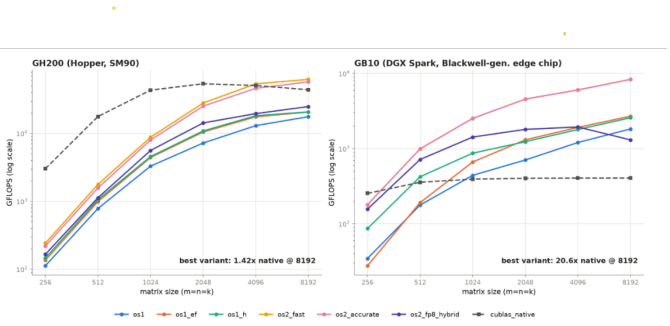


Fig. 5. `os2_fast` throughput relative to native cuBLAS on GH200 versus GB10.

The same unmodified pipeline was also swept on NVIDIA's GB10, a compact edge chip distinct from full datacenter Blackwell parts, using matched narrow- and wide-distribution sweeps (7 trials × 20 repeats, median reported). The best variant reaches 1.42× native cuBLAS throughput on GH200 but 20.6× on GB10 (Figure 5). GB10's native FP64 throughput stays nearly flat across the sweep, reflecting how FP64-starved the chip is relative to its low-precision throughput, exactly the gap Ozaki-Scheme exploits.

III. Future Work and Conclusion

Each new feature measurably improves the pipeline, and the pipeline transfers unmodified to GB10, where its advantage over native cuBLAS far exceeds that on GH200. The AUSURF112 workload remains the clearest sign of unfinished work: even after benchmark-harness corrections, native cuBLAS remains 1.21× faster than the tuned complex-GEMM configuration on this real application workload, and the evaluator still fits one `num_moduli` value across all 50 shapes rather than per shape, likely making that gap conservative.

References

[1] H. Ootomo, K. Ozaki, and R. Yokota, "DGEMM on integer matrix multiplication unit," 2023, *arXiv:2306.11975*.

[2] K. Ozaki, Y. Uchino, and T. Imamura, "Ozaki Scheme II: A GEMM-oriented emulation of floating-point matrix multiplication using an integer modular technique," 2025, *arXiv:2504.08009*.

[3] C. Vitz, A. Jarmusch, and S. Chandrasekaran, "Preliminary evaluation of a multi-agent optimization system for Ozaki Scheme GPU matrix multiplication," poster, *IEEE Int. Conf. Cluster Comput. (CLUSTER)*, 2026. [Online]. Available: https://github.com/WhiZarD31415/IEEE26-Evaluating-Multi-Agent-for-OzakiScheme-Poster

[4] Algorithmic Superintelligence, "OpenEvolve: An open-source implementation of AlphaEvolve," 2026. [Online]. Available: https://github.com/algorithmicsuperintelligence/openevolve

[5] Google DeepMind, "Gemini 2.5 Flash model card," 2025. [Online]. Available: https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-2-5-Flash-Model-Card.pdf

[6] P. Giannozzi *et al.*, "QUANTUM ESPRESSO: A modular and open-source software project for quantum simulations of materials," *J. Phys.: Condens. Matter*, vol. 21, no. 39, Art. no. 395502, 2009, doi: 10.1088/0953-8984/21/39/395502.

[7] NVIDIA Corporation, "NVIDIA DGX Spark: Personal AI supercomputer powered by Blackwell," 2025. [Online]. Available: https://www.nvidia.com/en-us/products/workstations/dgx-spark/