Hardware Comparison on Tiny Shakespeare

Character-Level Language Model Training Across Heterogeneous Compute Platforms

Experiment Report 5

John Lee | Maximo Sanchez | Youssif Goda

March 3, 2026

Abstract

This study investigates whether hardware specification CPU versus GPU, and across GPU tiers produces measurable differences in training behavior when scaling a character-level language model across three complexity levels. A small-scale GPT model was trained on the Tiny Shakespeare dataset using the nanoGPT framework. Three block sizes (32, 64, 128) were crossed with three model complexities (Low, Mid, Large) across five hardware platforms: NVIDIA RTX 4060, NVIDIA RTX 3070, Apple M2 Silicon, AMD Ryzen 7 8845HS, and AMD Ryzen 7 5800H. Training and validation loss were hardware-invariant within each block-size configuration, confirming experimental integrity. Performance divergence appeared exclusively in runtime, token throughput, and energy-efficiency metrics. The RTX 4060 dominated throughput and runtime at medium-to-large complexity, while the RTX 3070 showed surprising energy-efficiency advantages at specific workload profiles which was most dramatically at Large complexity / block size 64, where it consumed 58% fewer joules than the 4060. The Apple M2 led at low complexity due to its unified memory architecture, but degraded severely at scale. These results demonstrate that hardware selection for language model training should be guided by the anticipated complexity tier and context window length rather than by raw hardware generation alone.

1. Introduction

Modern machine learning workloads span a wide range of hardware environments, from consumer-grade CPUs and integrated SoC architectures to discrete gaming GPUs. As small-scale model training becomes more accessible, practitioners increasingly face the question of whether the hardware available to them is adequate or whether specific workload profiles demand specific silicon.

This experiment uses the nanoGPT framework trained on the Tiny Shakespeare character-level dataset as a controlled benchmark. By systematically varying two independent variables model complexity and context window length (block size) across five hardware configurations, we isolate how the interaction between sequence length and model capacity affects training speed, throughput, and energy efficiency.

Loss curves serve as a control variable rather than a dependent variable: because all runs share identical hyperparameters, architecture, and data splits, convergence behavior is expected to be hardware-invariant. The meaningful signal lives in the performance metrics runtime, tokens per second, and joules consumed which differ substantially across platforms once model complexity exceeds a certain threshold.

2. Methodology

2.1 Hardware Configurations

Rank	Hardware	Type	Notes
1	NVIDIA RTX 4060	Discrete GPU	Highest theoretical compute capability
2	NVIDIA RTX 3070	Discrete GPU	Previous-gen discrete GPU
3	Apple M2 Silicon	Unified Memory SoC	Unified memory architecture
4	AMD Ryzen 7 8845HS	CPU	Modern x86 CPU
5	AMD Ryzen 7 5800H	CPU	Older x86 CPU

2.2 Model Complexity Levels

Three model configurations were defined, each increasing representational capacity:

Level	Attention Heads	Layers	Embedding Dimension
Low	3	3	126
Mid	6	6	384
Large	12	12	768

2.3 Block Sizes Under Investigation

Each complexity level was trained at three block sizes, making block size a first-class experimental variable alongside model complexity:

Block Size	Purpose
32	Minimal memory and compute demand per step
64	Baseline context, matches the repository default configuration
128	Extended context, stresses memory bandwidth and parallel capacity

2.4 Controlled Variables

All models were trained under identical conditions to ensure that any observed differences are attributable solely to hardware and block size:

- Identical iteration count per complexity level (2,000 iterations)
- Fixed hyperparameters: learning rate 1×10^{-3}, minimum LR 1×10^{-4}, beta2 = 0.99, dropout = 0.0
- Uniform batch size: 12
- Fixed random seeds and dataset splits (shakespeare_char)
- Gradient accumulation steps: 1
- Warmup iterations: 100

All runs were logged to a shared Weights & Biases project (GPUvsCPUResearch) and the codebase is available at: https://github.com/MaxiSanc37/GPU-vs-CPU-Comparative-Research

2.5 Metrics

Metric	Description	Availability
Training Loss	Model error on training data vs. iteration	All hardware
Validation Loss	Model error on held-out data vs. iteration	All hardware
Runtime (s)	Total wall-clock training duration vs. iteration	All hardware
Tokens Per Second	Instantaneous throughput — hardware-normalised proxy for speed	All hardware
Total Tokens / Total Joules	Cumulative energy efficiency trajectory over the full run	NVIDIA GPUs only
Token Throughput vs. Joules/Iteration	Instantaneous speed–efficiency scatter (Efficiency vs. Speed)	NVIDIA GPUs only

Note: Energy metrics (joules) are exclusively available for the RTX 4060 and RTX 3070 via the pynvml library, which interfaces with NVIDIA's proprietary GPU telemetry API. Apple M2 and AMD CPU platforms are evaluated through hardware-agnostic metrics only. This is an inherent instrumentation constraint, not an experimental oversight. Future work could address this via Intel RAPL or Apple Silicon profiling utilities.

3. Results

3.1 Loss Results (Control Variable)

Training and validation loss curves were nearly identical across all hardware configurations within each block-size group, as expected. Because all runs shared the same architecture, hyperparameters, dataset splits, and random seeds, the underlying optimization problem was unchanged. Loss convergence therefore serves as a control confirming experimental integrity rather than a differentiating metric.

Loss did, however, improve meaningfully with block size which is a finding attributable to model configuration rather than hardware:

Block Size	Complexity	Final Train Loss	Final Val Loss
32	Low	1.900	1.977
32	Mid	1.772	1.928
32	Large	1.920	1.992
64	Low	1.772	1.912
64	Mid	1.604	1.763
64	Large	1.695	1.839
128	Low	1.704	1.863
128	Mid	1.415	1.625
128	Large	1.566	1.743

This trend reflects the well-established relationship between context window length and model quality: larger block sizes expose the model to longer-range dependencies within the corpus, enabling more effective learning within the same iteration budget.

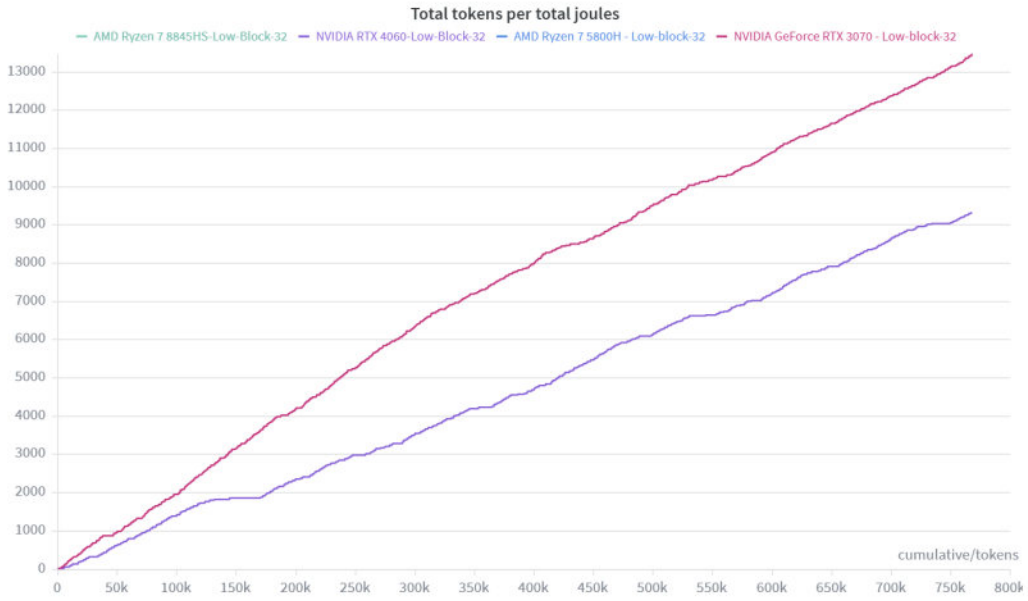
3.2 Block Size 32

Summary

Metric	Low Complexity Winner	Mid Complexity Winner	Large Complexity Winner
Total Tokens / Total Joules	RTX 4060 (+30%)	RTX 3070	RTX 4060 (+36%)
Efficiency vs. Speed	RTX 4060	RTX 4060	RTX 4060
Tokens Per Second	Apple M2	RTX 4060	RTX 4060
Runtime	Apple M2	RTX 4060 (−15 s)	RTX 4060 (−17 s)

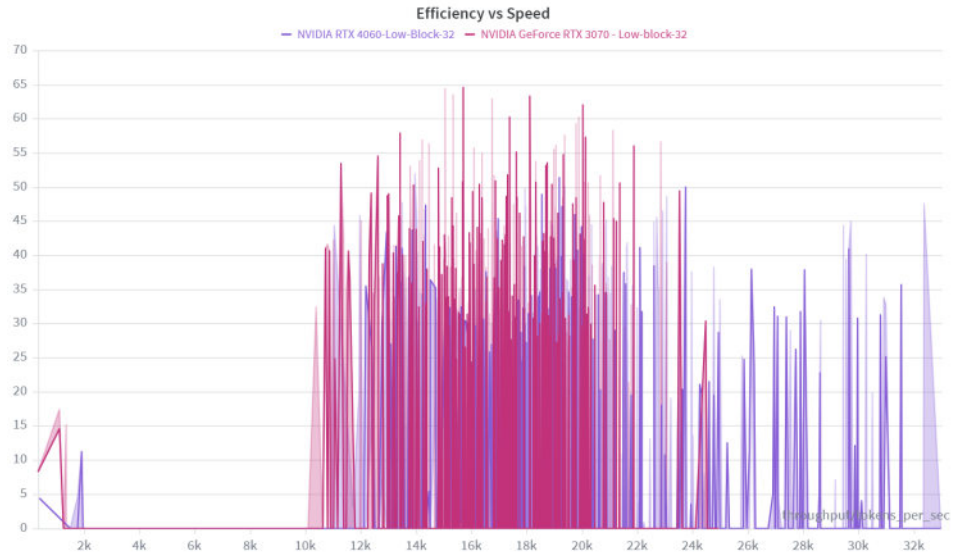
Low Complexity — Block Size 32

Energy efficiency (Total Tokens / Total Joules): The RTX 4060 consumed 9,318 joules at the 767k-token mark versus the RTX 3070's 13,466 joules — a 30% efficiency advantage.



▲ Low Complexity – 1

Efficiency vs. Speed: Both GPUs operated across a comparable throughput range of 10k–32k tokens/sec. The RTX 3070 produced frequent energy spikes up to ~65 J/iteration; the RTX 4060 remained stable between 40–50 J with far fewer outliers.

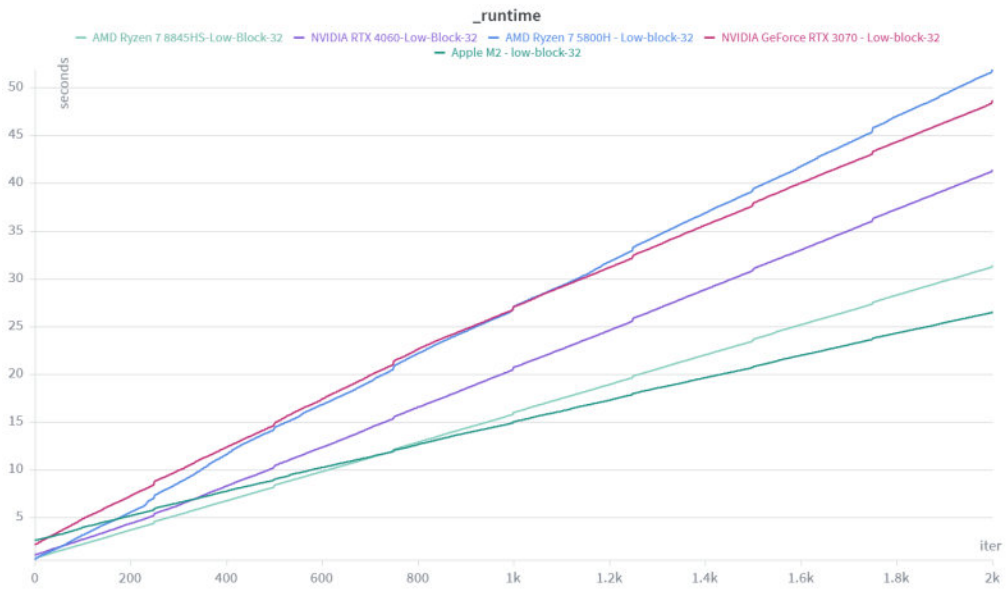


▲ Low Complexity - 2

Tokens Per Second: The Apple M2 averaged 25,000–35,000 tokens/sec, outpacing the RTX 4060 (~20,000–30,000) and the RTX 3070 (~10,000–20,000). The M2's unified memory architecture enables highly efficient data transfer at small model sizes where GPU parallelism is underutilised.

▲ Low Complexity - 3

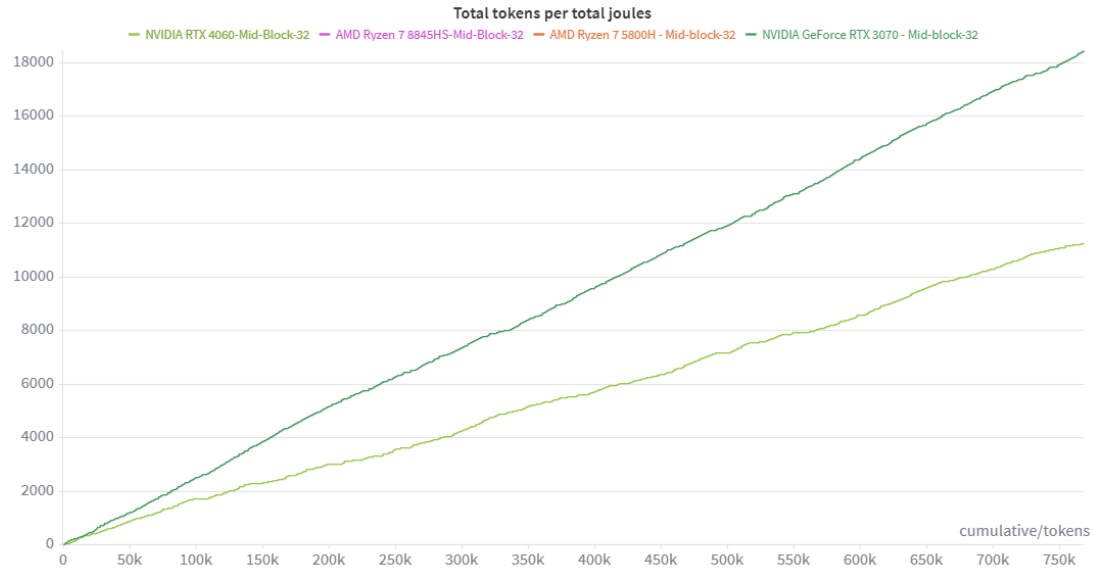
Runtime: The Apple M2 was the fastest hardware configuration, followed by the RTX 4060, RTX 3070, Ryzen 5800H, and Ryzen 8845HS.



▲ Low Complexity - 4

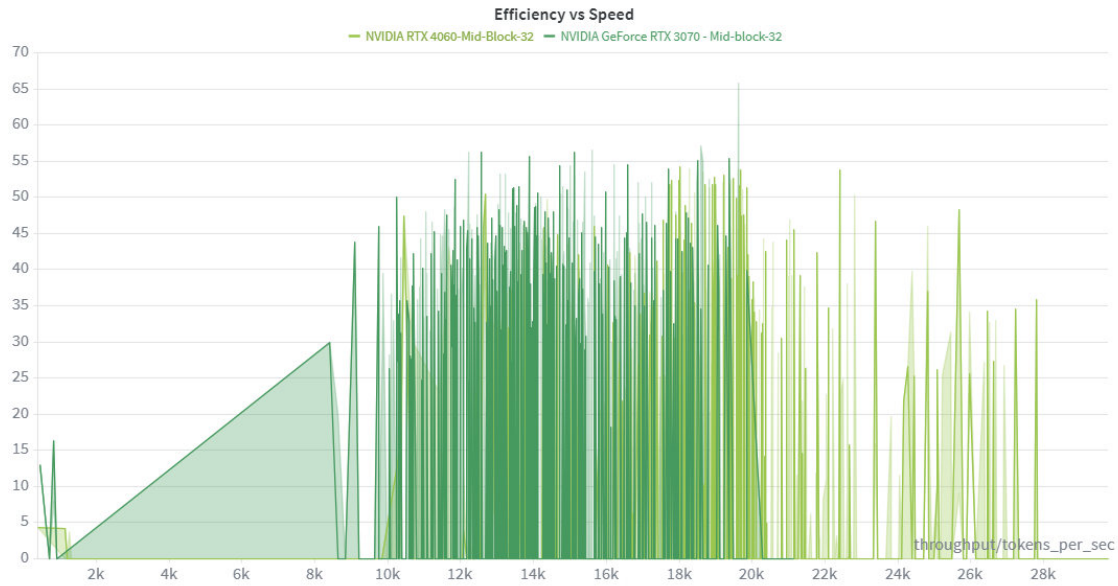
Mid Complexity — Block Size 32

Energy efficiency: The RTX 3070 reversed the low-complexity result decisively, reaching ~18,445 tokens/joule versus the RTX 4060's ~11,261 — a substantial 64% advantage to the older GPU.



▲ Mid Complexity – 1

Efficiency vs. Speed: The RTX 4060 extended past 28k tokens/sec while the RTX 3070 maxed out around 20–21k. The RTX 4060 held a clear throughput advantage with slightly lower peak joule spikes (~55–57 vs. ~65 for the RTX 3070).

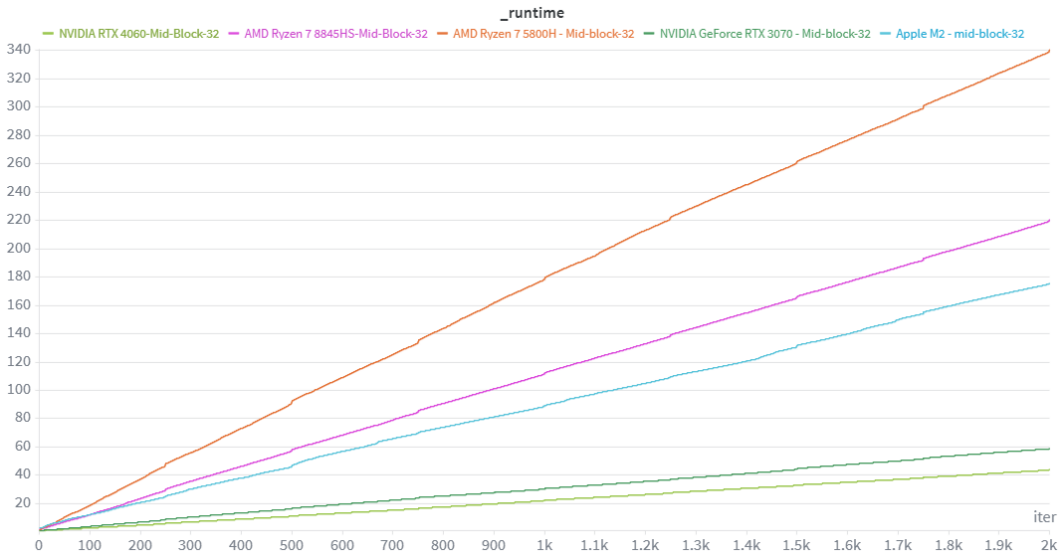


▲ Mid Complexity - 2

Tokens Per Second: Discrete GPUs dominated at this complexity tier. The RTX 4060 averaged 18,000–24,000 tokens/sec; the Apple M2 dropped to ~4,000–6,000; AMD CPUs fell to ~2,000–4,000.

▲ Mid Complexity – 3

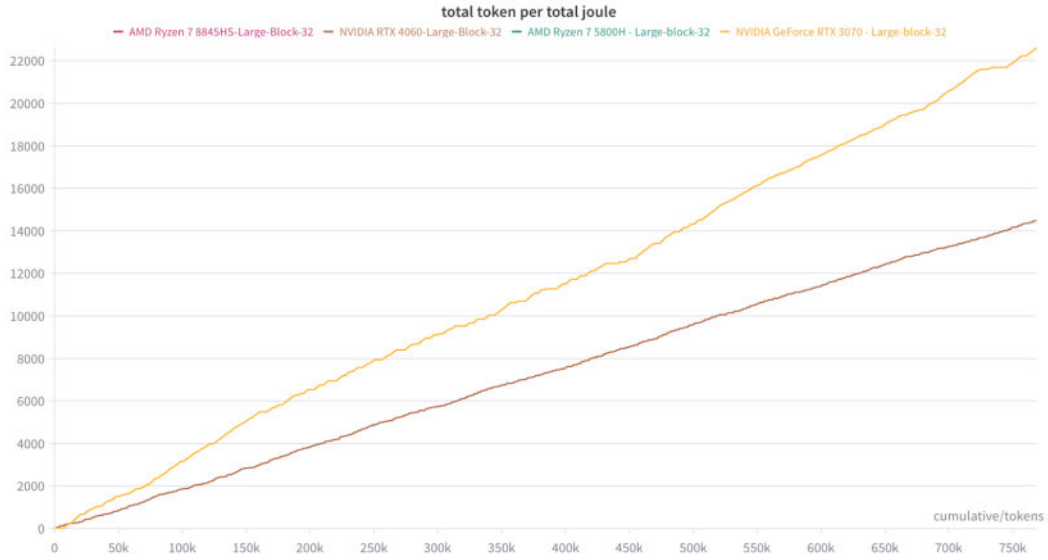
Runtime: The RTX 4060 completed 2,000 iterations in ~45 seconds versus the RTX 3070's ~60 seconds — a 15-second advantage.



▲ Mid Complexity – 4

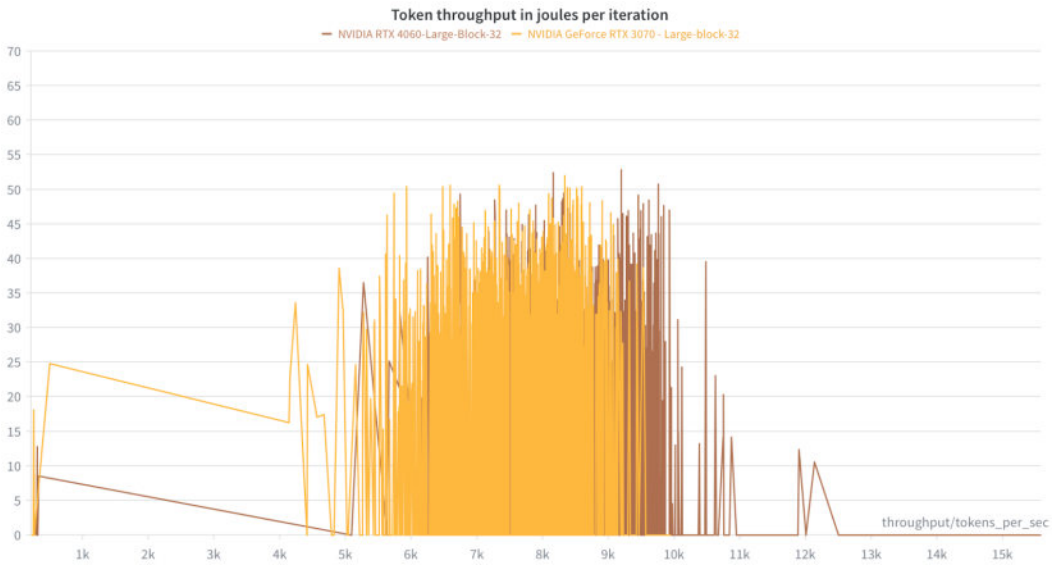
Large Complexity — Block Size 32

Energy efficiency: The RTX 4060 recovered its advantage, consuming only 14,000 joules at the 750k-token mark versus the RTX 3070's 22,000 — a 36% efficiency lead.



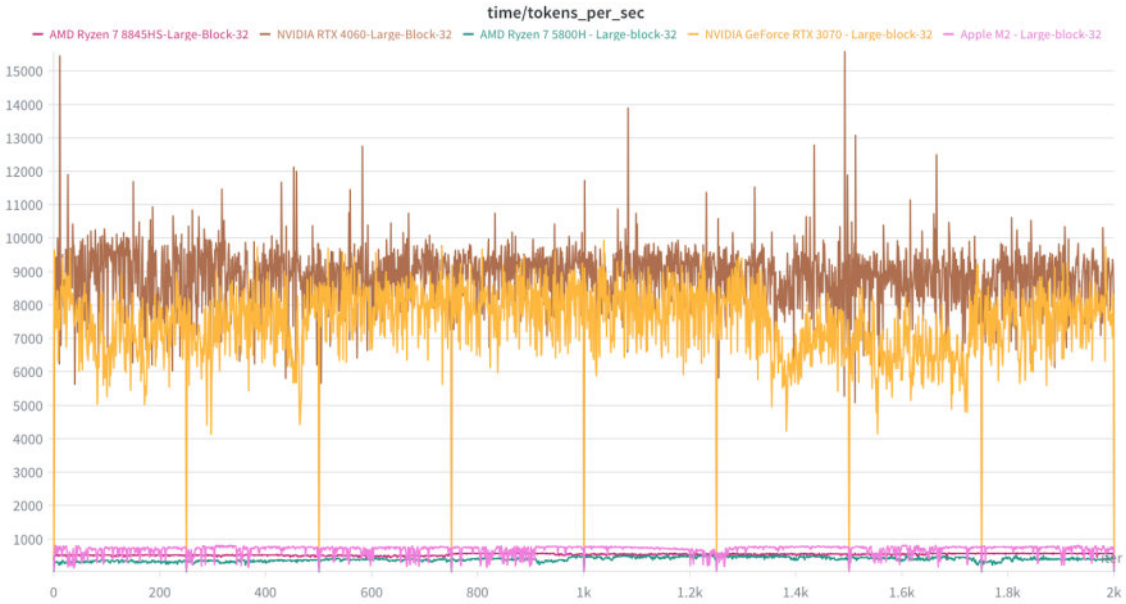
▲ Large Complexity – 1

Efficiency vs. Speed: Both GPUs operated in the 5k–12k tokens/sec range. The RTX 4060 reached 12k while the RTX 3070 was confined to 5k–9k, with nearly equal average joule consumption, making the 4060 faster at equivalent energy cost.



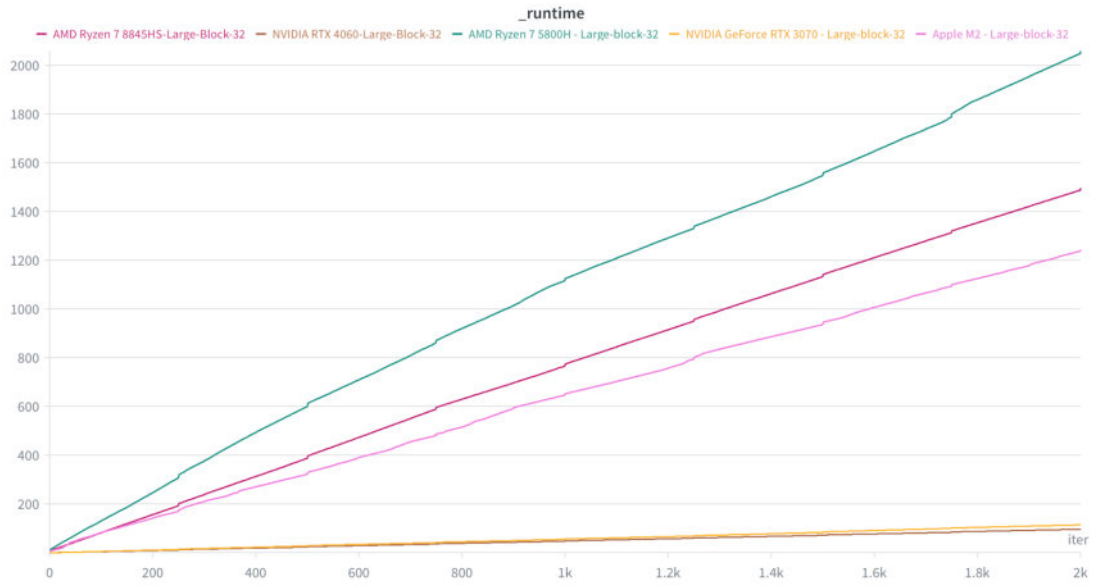
▲ Large Complexity - 2

Tokens Per Second: The RTX 4060 maintained 8,000–10,000 tokens/sec; the RTX 3070 managed 6,000–8,000; the Apple M2 dropped to ~1,000; AMD CPUs fell below 500.



▲ Large Complexity – 3

Runtime: The RTX 4060 outperformed the RTX 3070 by 17 seconds over 2,000 iterations.



▲ Large Complexity - 3

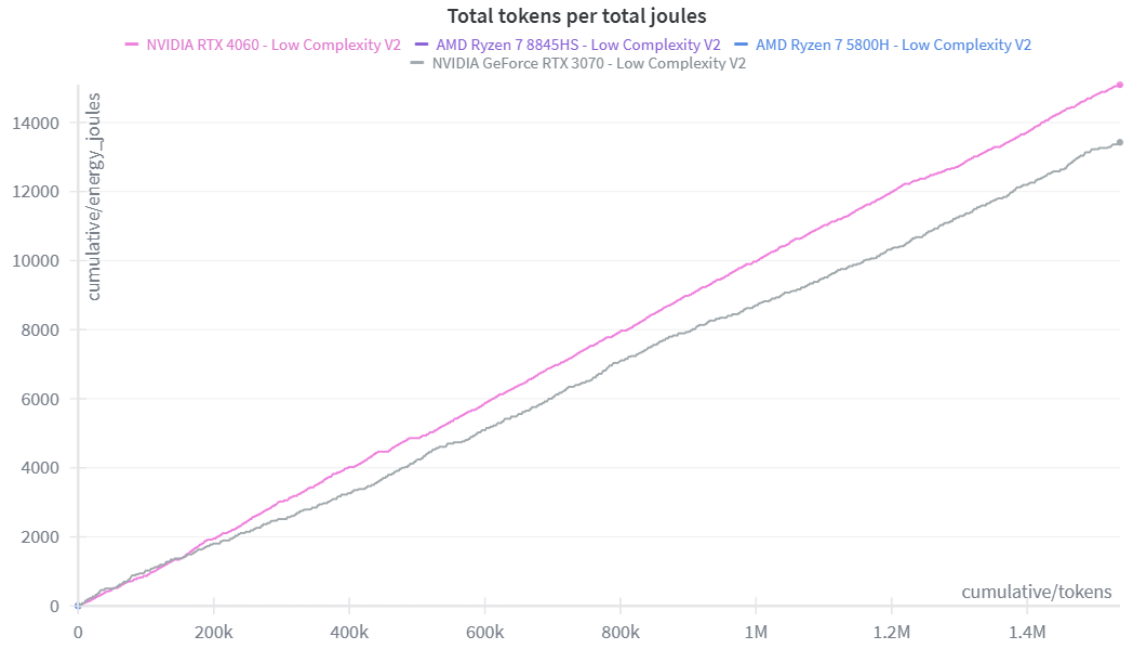
3.3 Block Size 64

Summary

Metric	Low Complexity Winner	Mid Complexity Winner	Large Complexity Winner
Total Tokens / Total Joules	RTX 3070 (+11.8%)	Tie (negligible)	RTX 3070 (+239%)
Efficiency vs. Speed	RTX 4060	RTX 4060 (marginal)	Split (4060 speed; 3070 efficiency)
Tokens Per Second	Apple M2	RTX 4060	RTX 4060
Runtime	Apple M2	RTX 4060 (−9 s)	RTX 4060 (−243 s)

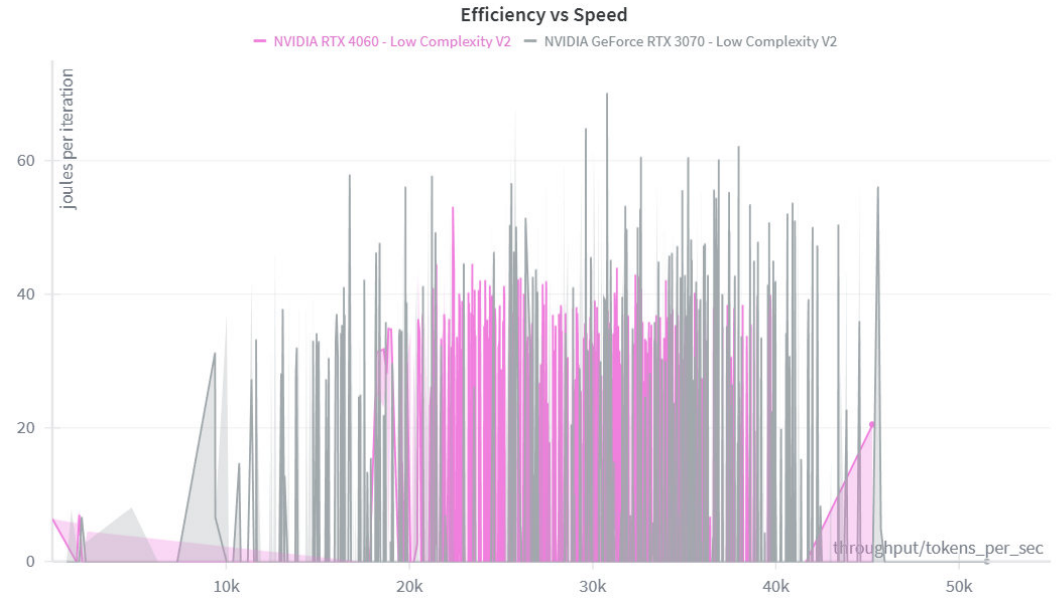
Low Complexity — Block Size 64

Energy efficiency: The RTX 3070 led at this tier — at the 1.5M-token mark it had consumed 13,227 joules versus the RTX 4060's 14,794 joules (an 11.8% advantage).



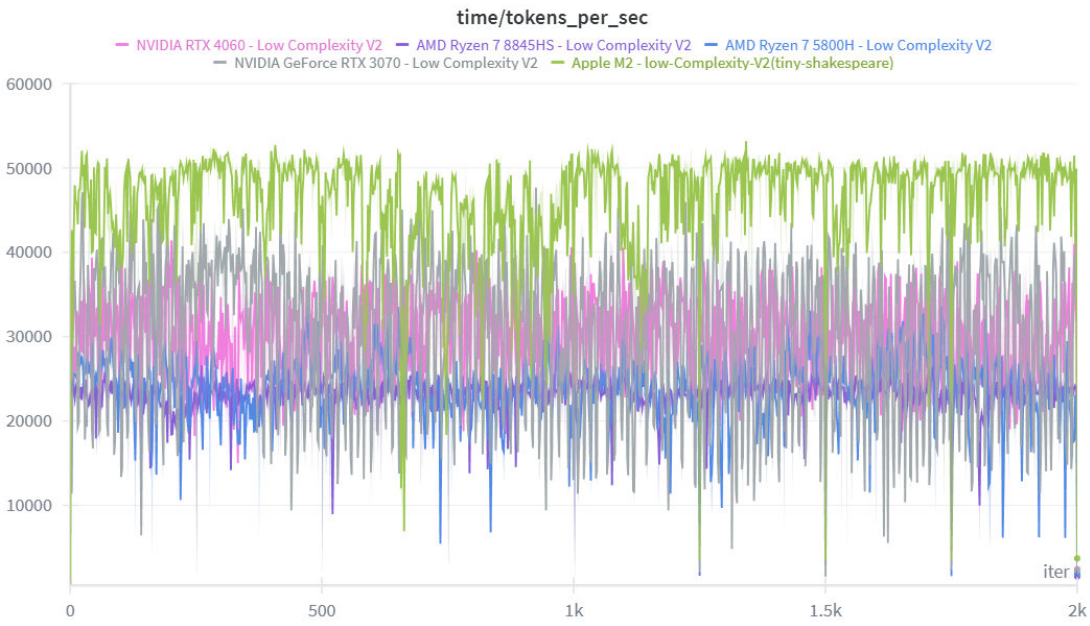
▲ Low Complexity - 1

Efficiency vs. Speed: Both GPUs operated in a comparable 10k–50k tokens/sec range. The RTX 4060 maintained lower and more stable joule expenditure (20–40 J) versus the RTX 3070's spikes to ~65 J.



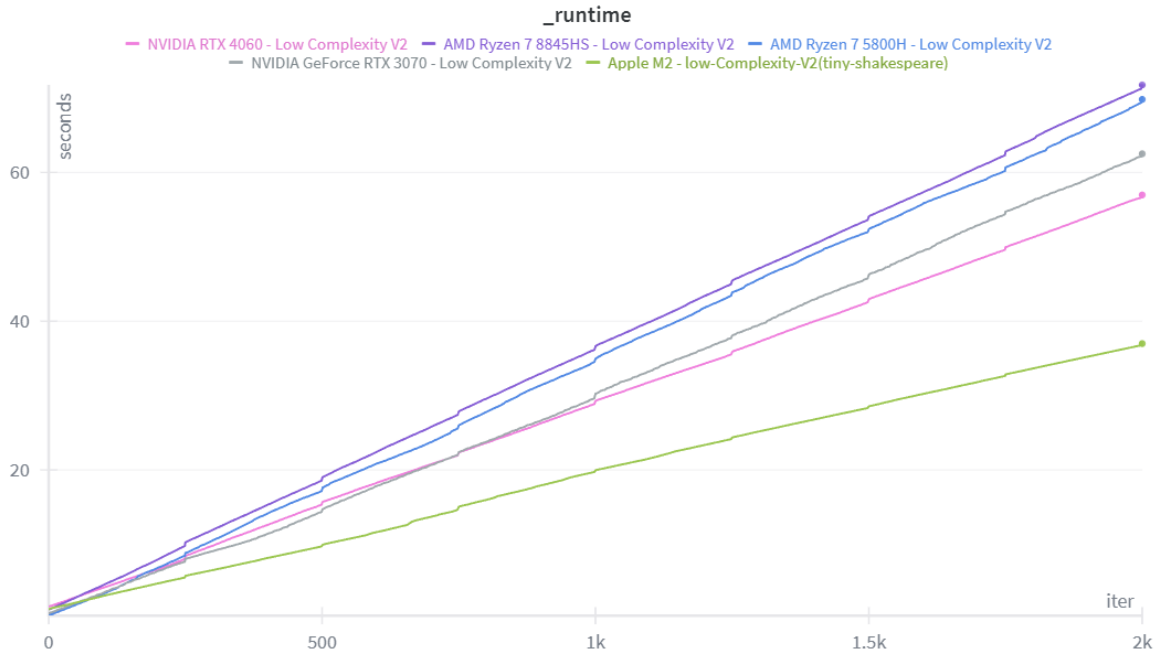
▲ Low Complexity - 2

Tokens Per Second: The Apple M2 averaged 40,000–50,000 tokens/sec, well ahead of both GPUs (~25,000–35,000 for the 4060; ~25,000–30,000 for the 3070).



▲ Low Complexity - 3

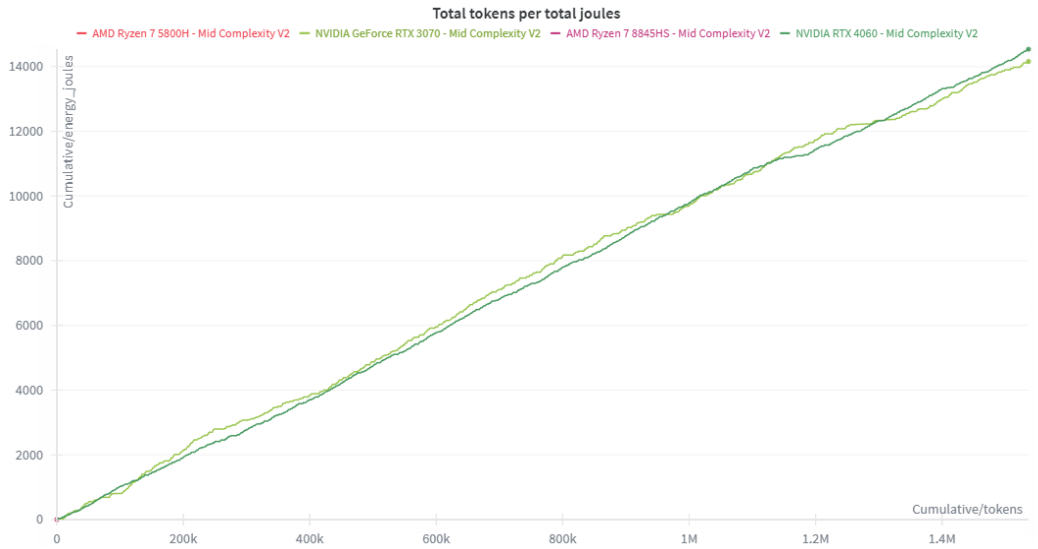
Runtime: Apple M2 was fastest, followed by RTX 4060, RTX 3070, Ryzen 5800H, and Ryzen 8845HS.



▲ Low Complexity - 4

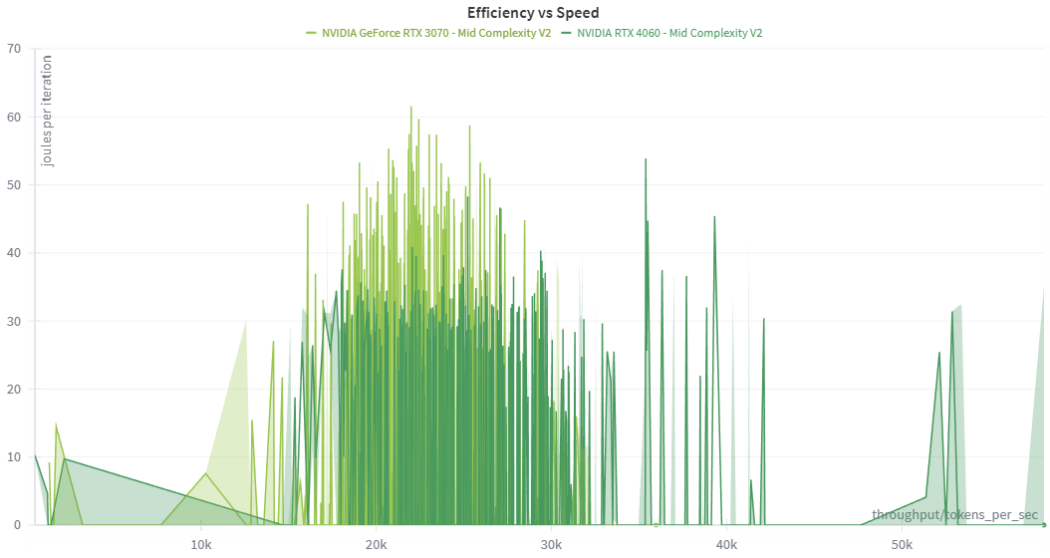
Mid Complexity — Block Size 64

Energy efficiency: RTX 4060 and RTX 3070 were effectively tied — at several token marks the 3070 briefly led, but differences were negligible. No clear winner.



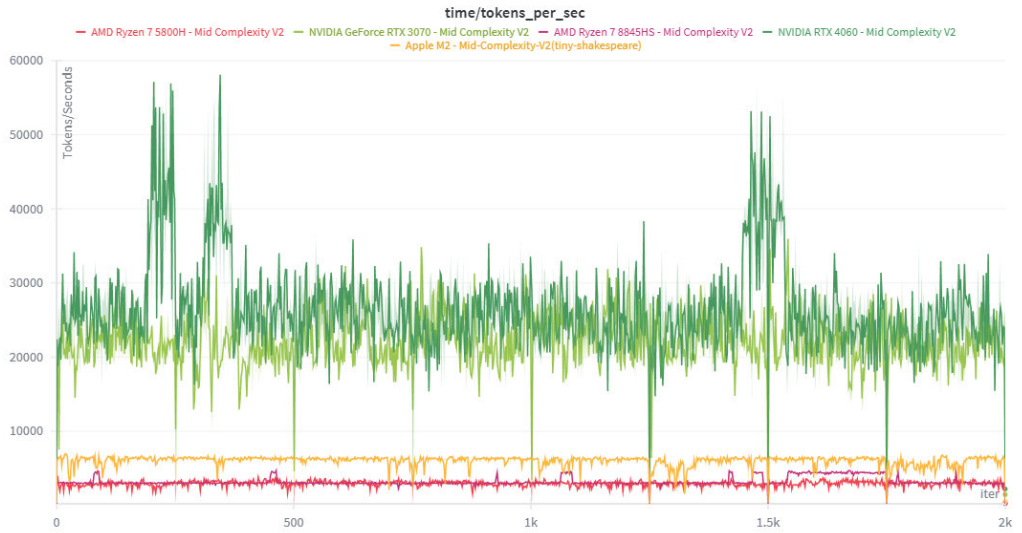
▲ Mid Complexity – 1

Efficiency vs. Speed: Both GPUs reached similar peak throughput (~55k tokens/sec) and similar peak joule values (~60–62 J). The RTX 4060 sustained its peak-energy iterations at marginally higher throughput.



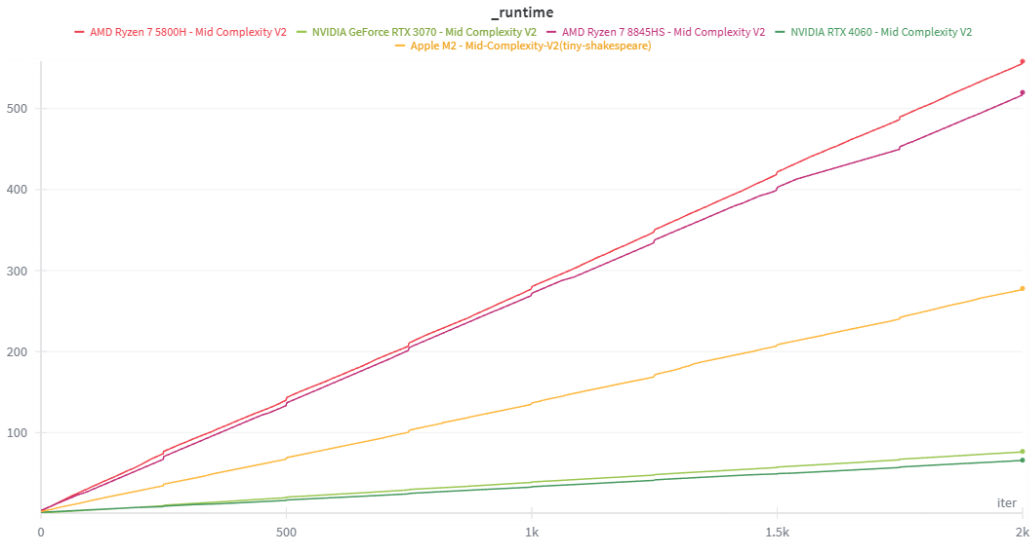
▲ Mid Complexity - 2

Tokens Per Second: RTX 4060 averaged 25,000–30,000 tokens/sec with spikes to 55,000+; RTX 3070 averaged 20,000–22,000. Apple M2 dropped to ~6,000; AMD CPUs to ~2,500–3,500.



▲ Mid Complexity – 3

Runtime: RTX 4060 outperformed by 9 seconds.

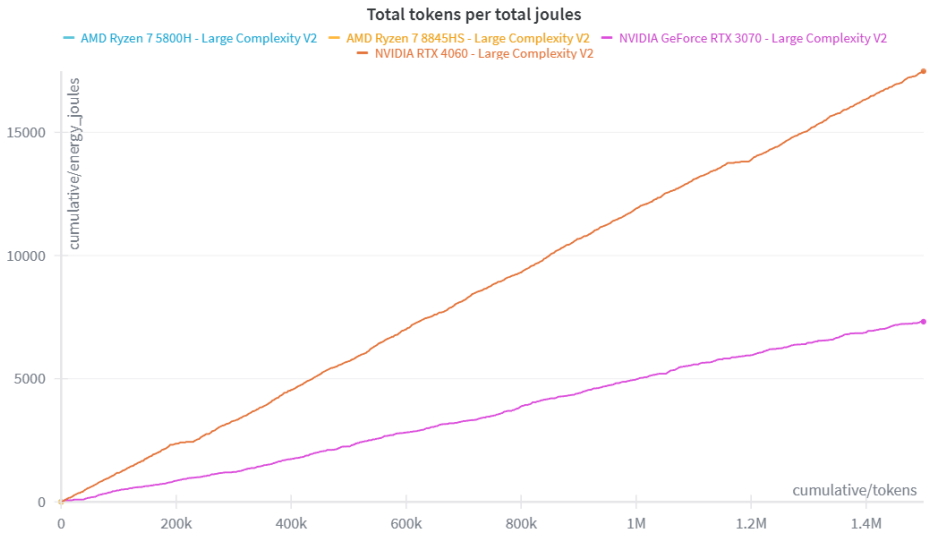


▲ Mid Complexity – 4

Large Complexity — Block Size 64

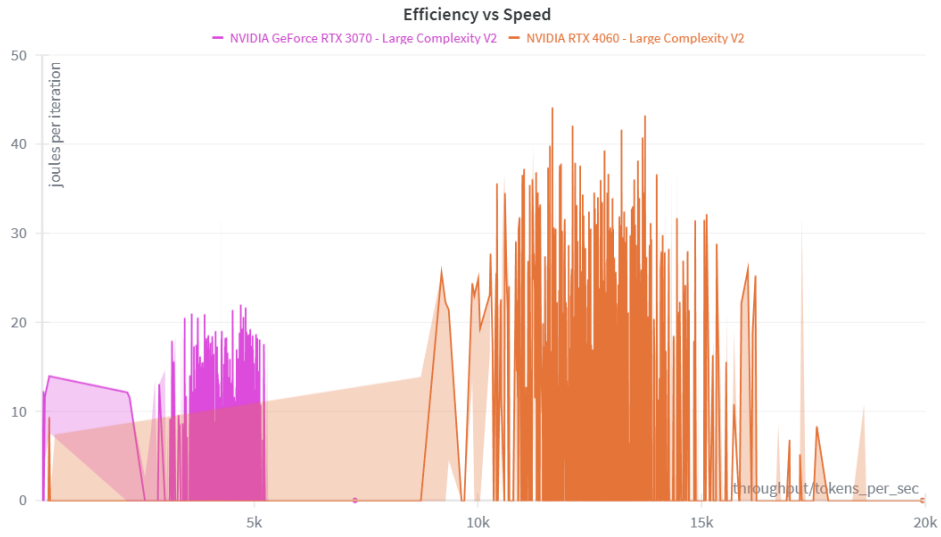
Notable finding: At large complexity with block size 64, the RTX 3070 consumed only 7,315 joules to process 1.5 million tokens, compared to the RTX 4060's 17,485 joules — a **239%**

energy-efficiency advantage favouring the older GPU. The authors attribute this to the RTX 3070's higher CUDA core count better matching this specific workload profile.



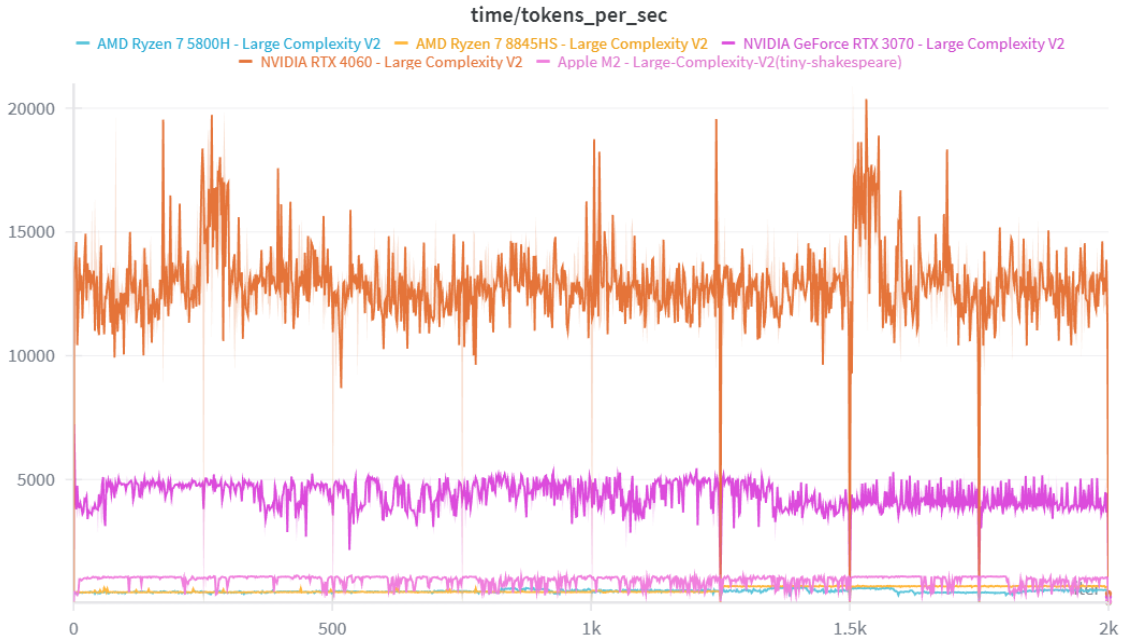
▲ Large Complexity – 1

Efficiency vs. Speed: The relationship inverted here. The RTX 3070 was confined to 1k–5k tokens/sec while the RTX 4060 operated at 8k–18k — approximately 3–4× faster. The 4060 paid for that speed with higher joule-per-iteration values (25–42 J vs. 10–20 J for the 3070). For training speed, the RTX 4060 wins decisively; for energy efficiency, the RTX 3070 wins at this scale.



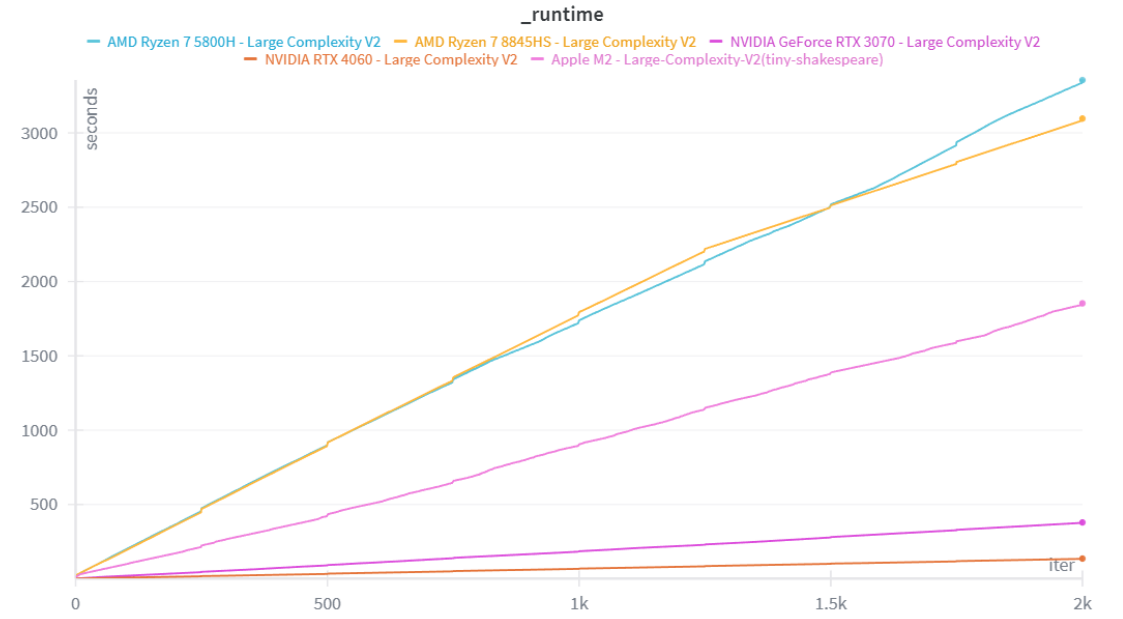
▲ Large Complexity - 2

Tokens Per Second: RTX 4060 averaged 12,000–13,000 tokens/sec; RTX 3070 averaged 4,000–5,000; Apple M2 ~1,000; AMD CPUs below 500.



▲ Large Complexity – 3

Runtime: RTX 4060 outperformed the RTX 3070 by 243 seconds — the largest absolute runtime gap in the study.



▲ Large Complexity - 4

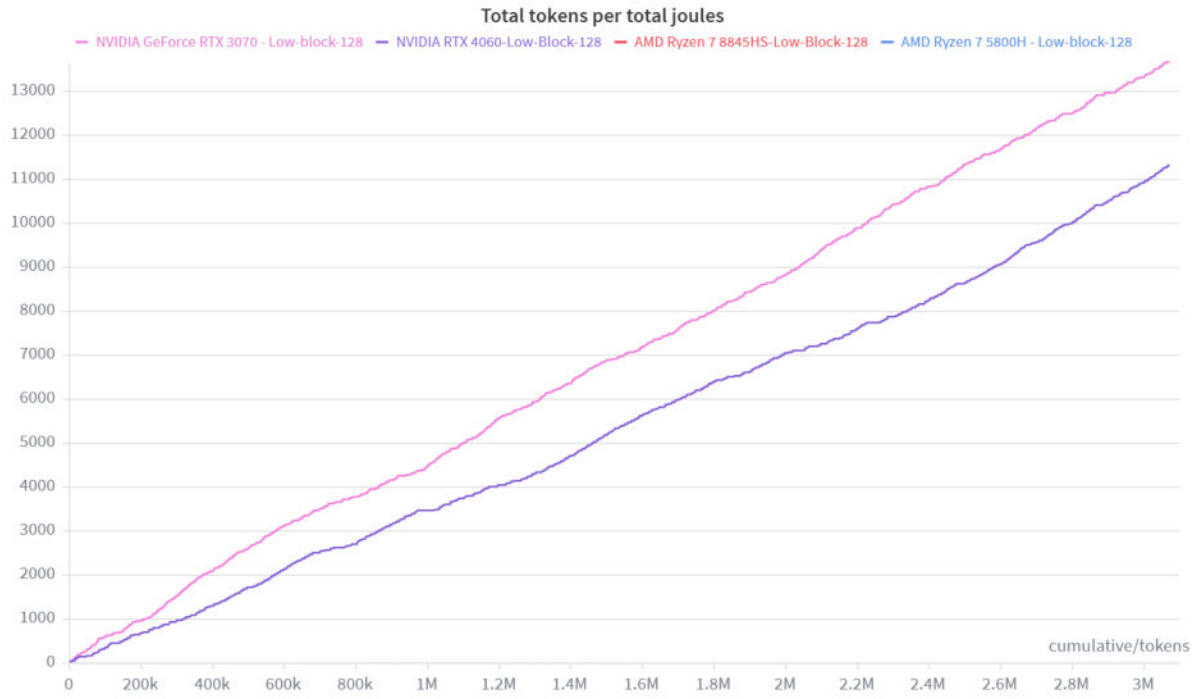
3.4 Block Size 128

Summary

Metric	Low Complexity Winner	Mid Complexity Winner	Large Complexity Winner
Total Tokens / Total Joules	RTX 4060 (+17%)	RTX 3070 (clear lead)	RTX 3070 (+15%)
Efficiency vs. Speed	RTX 4060	RTX 4060	RTX 4060
Tokens Per Second	RTX 4060	RTX 4060	RTX 3070 (marginal)
Runtime	RTX 4060	Tie (48 s each)	RTX 3070 (−10 s)

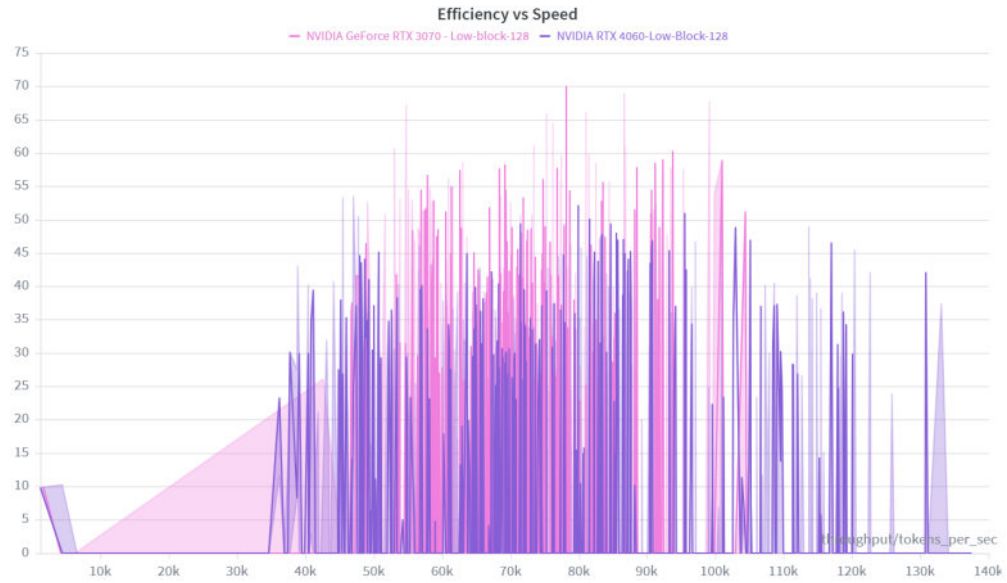
Low Complexity — Block Size 128

Energy efficiency: RTX 4060 led at the 3M-token mark with 11,290 joules versus the RTX 3070's 13,644 — a 17% advantage.



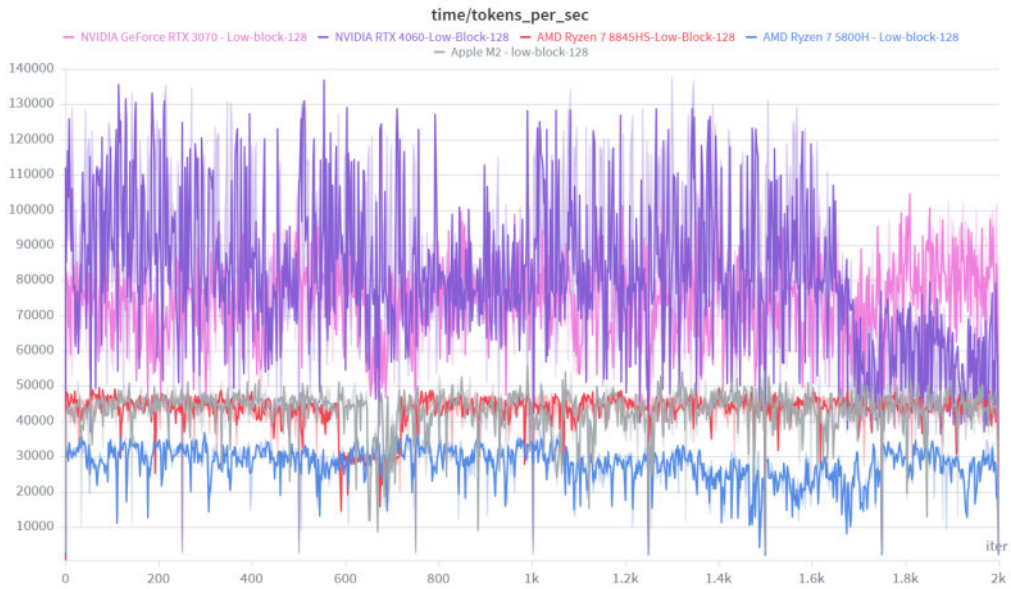
▲ Low Complexity - 1

Efficiency vs. Speed: Both GPUs operated across 30k–130k tokens/sec. RTX 3070 reached energy spikes up to ~70 J/iteration; RTX 4060 remained compressed between 40–50 J.



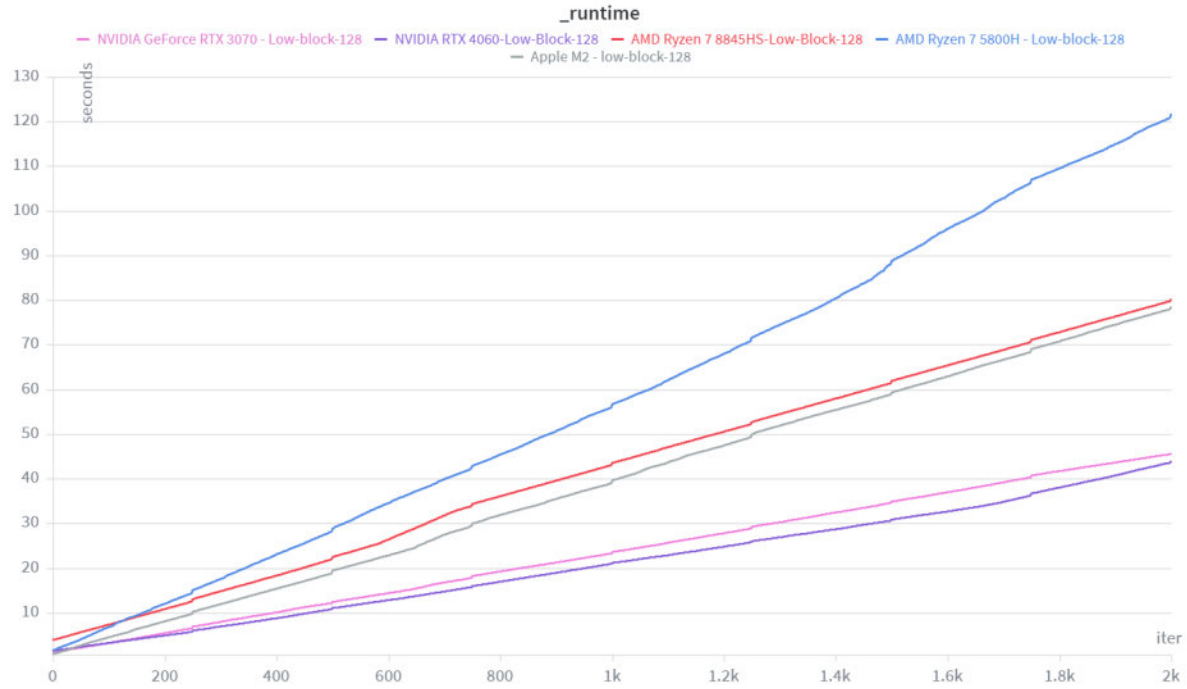
▲ Low Complexity – 2

Tokens Per Second: At block size 128, GPU architecture advantages emerged even at low complexity. RTX 4060 averaged 50,000–130,000 tokens/sec; RTX 3070 averaged 60,000–100,000; Apple M2 averaged 30,000–50,000. The larger context window allows the GPU's parallelism to be more fully utilised.



▲ Low Complexity - 3

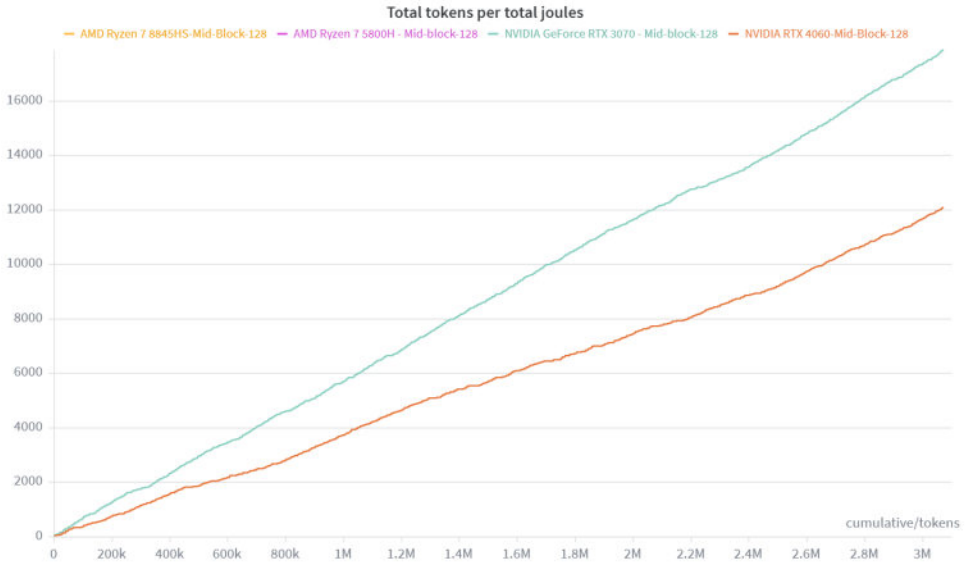
Runtime: RTX 4060 was fastest, followed by RTX 3070, Apple M2, Ryzen 8845HS, and Ryzen 5800H.



▲ Low Complexity - 4

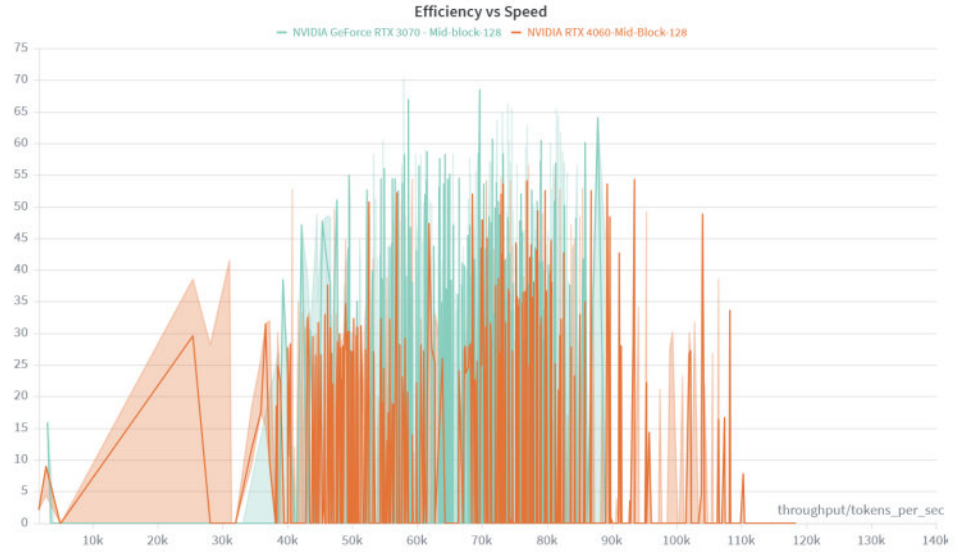
Mid Complexity — Block Size 128

Energy efficiency: The RTX 3070 dominated, reaching ~18,000 tokens/joule versus the RTX 4060's ~12,000 — with no crossover point. The RTX 3070's energy efficiency advantage was absolute at this configuration.



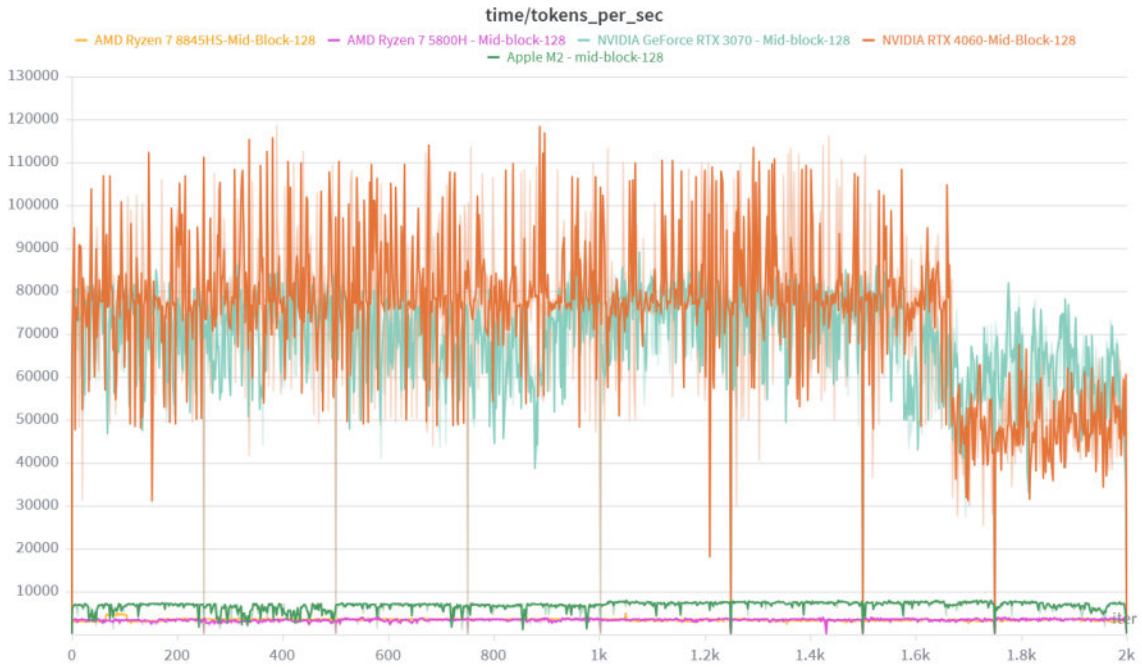
▲ Mid Complexity – 1

Efficiency vs. Speed: RTX 4060 extended to ~110k tokens/sec; RTX 3070 maxed out around 90k. RTX 4060 also displayed lower peak joule expenditure (~55–58 J vs. ~70 J for the RTX 3070).



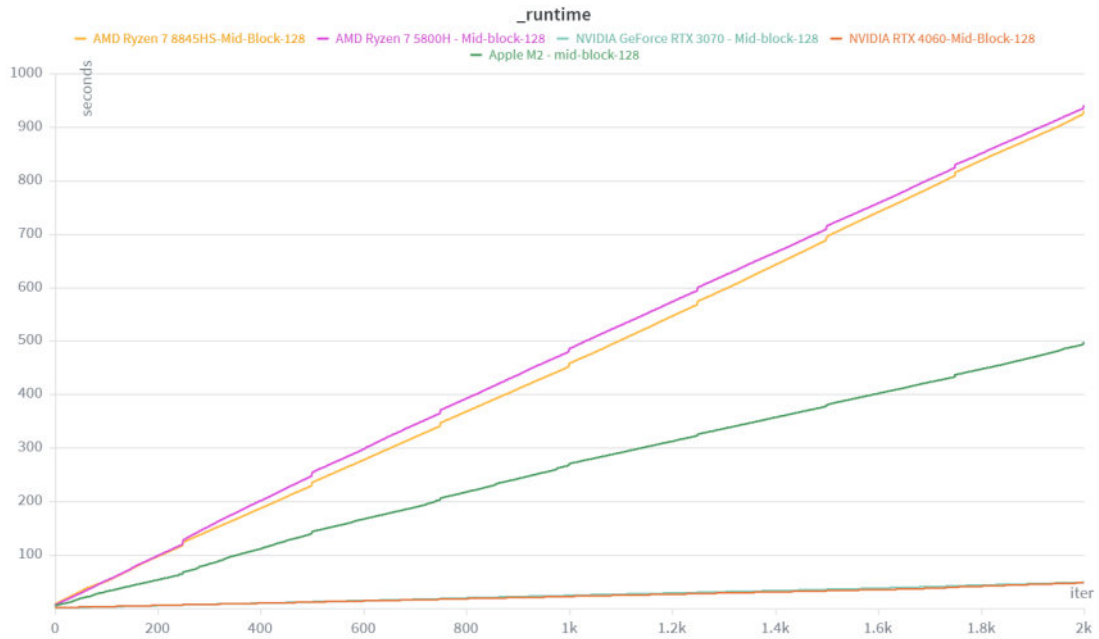
▲ Mid Complexity - 2

Tokens Per Second: RTX 4060 averaged 50,000–120,000 tokens/sec; RTX 3070 averaged 60,000–90,000; Apple M2 ~7,000; AMD CPUs ~3,000–4,000.



▲ Mid Complexity – 3

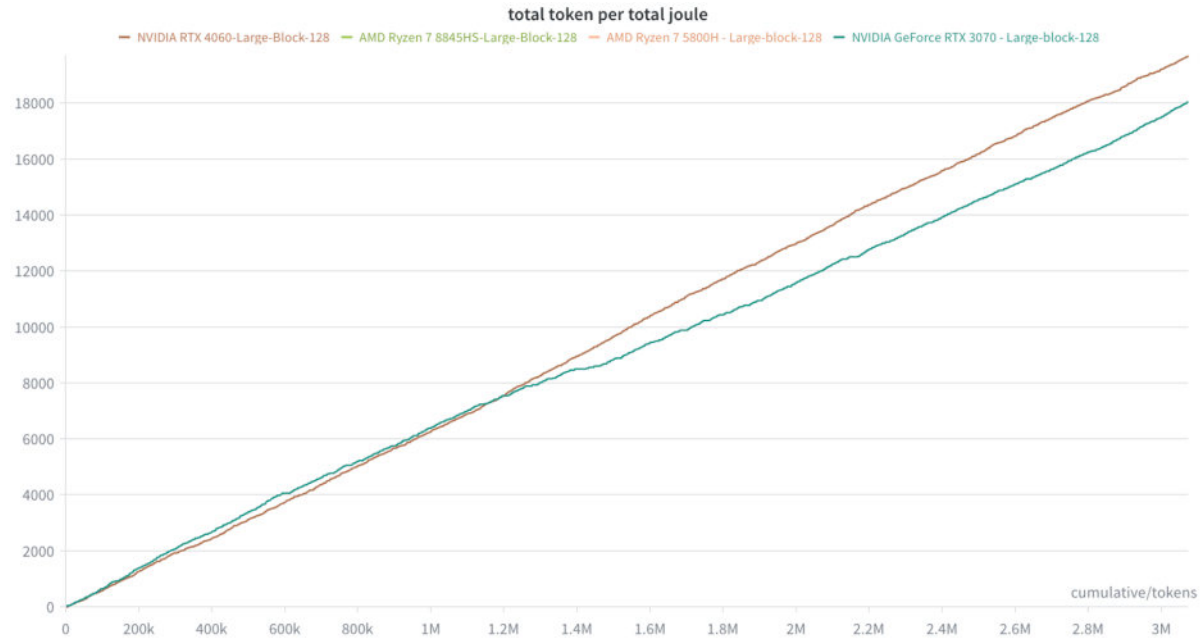
Runtime: RTX 4060 and RTX 3070 tied exactly at 48 seconds for 2,000 iterations.



▲ Mid Complexity - 4

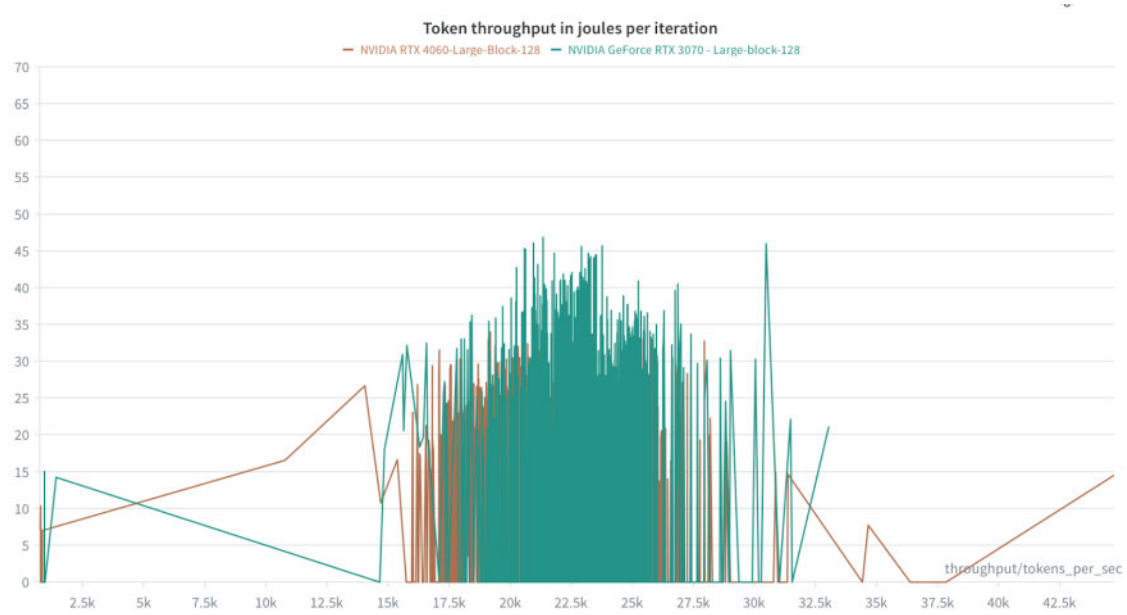
Large Complexity — Block Size 128

Energy efficiency: RTX 3070 led — at the 3M-token mark it had consumed 16,000 joules versus the RTX 4060's 19,000 (a 15% advantage).



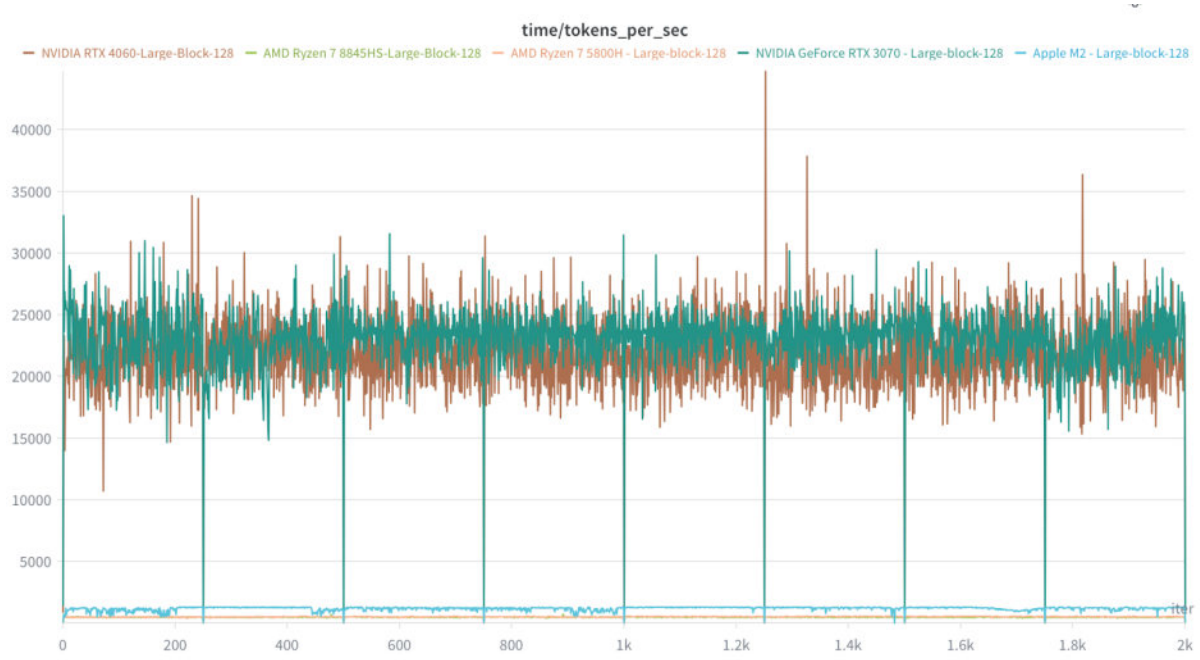
▲ Large Complexity - 1

Efficiency vs. Speed: Both GPUs operated in a comparable 10k–32k tokens/sec range. RTX 3070 produced energy spikes up to ~65 J/iteration; RTX 4060 remained compressed between 40–50 J.



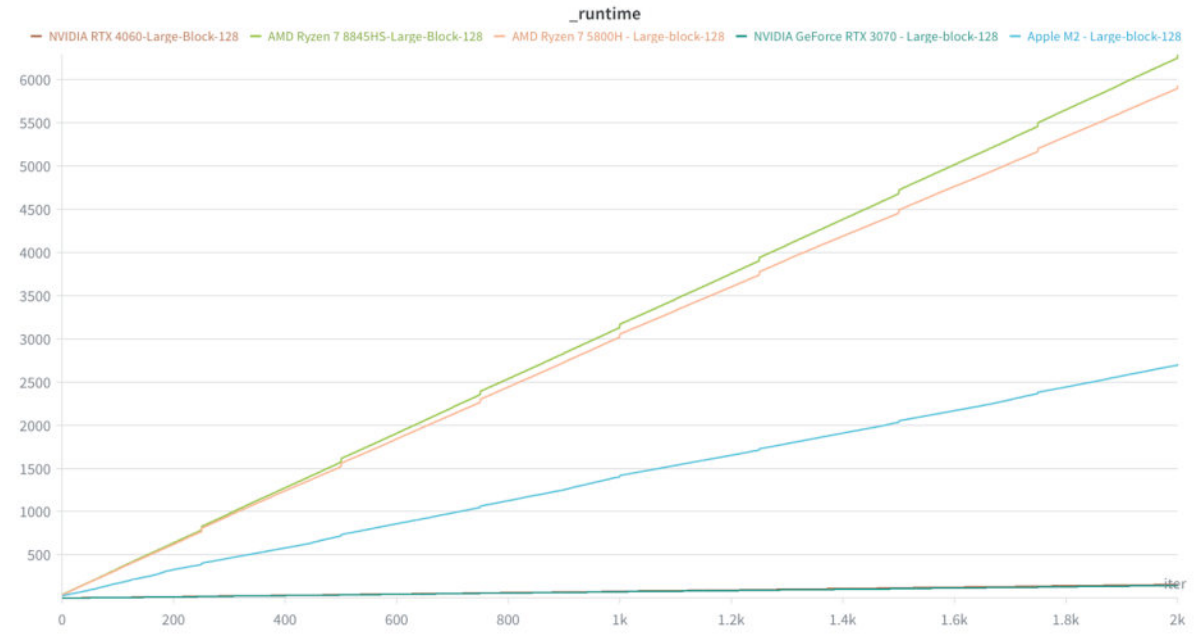
▲ Large Complexity - 2

Tokens Per Second: RTX 3070 averaged 20,000–25,000 tokens/sec — marginally higher and more consistent than the RTX 4060's 18,000–25,000.



▲ Large Complexity - 3

Runtime: RTX 3070 outperformed the RTX 4060 by 10 seconds — one of the few runtime wins for the older GPU in the study.



▲ Large Complexity - 4

4. Discussion

4.1 The RTX 4060 vs. RTX 3070 Dynamic

The most striking pattern across all 27 configurations (3 block sizes × 3 complexities × 3 metrics) is that neither GPU is universally superior. The RTX 4060 generally dominated runtime and instantaneous throughput, reflecting its newer architecture and faster memory bandwidth. However, the RTX 3070's higher CUDA core count translated into decisive energy-efficiency advantages at specific workload profiles, particularly mid-to-large complexity tasks where parallelism utilisation saturates.

The most dramatic illustration is the Large / Block-64 configuration, where the RTX 3070 consumed 58% fewer joules than the RTX 4060 to process the same number of tokens despite running at only 25–35% of the 4060's token throughput. For workloads where time-to-result is not the bottleneck but energy cost or thermal constraints are, the older GPU may be the more appropriate choice.

4.2 Apple M2 and the Unified Memory Effect

The Apple M2 was the fastest hardware at low complexity for block sizes 32 and 64, outperforming both discrete GPUs on tokens-per-second and runtime. This result is surprising given the M2's lower peak floating-point throughput and maybe it’s attributed to its unified memory architecture, which eliminates the PCIe transfer overhead that penalises discrete GPUs when data movement dominates computation.

This advantage evaporated rapidly as complexity increased. By mid complexity, the M2 had fallen to ~4,000–6,000 tokens/sec, and by large complexity it reached ~1,000 tokens/sec which is a 30–50× deficit versus the RTX 4060. The Apple M2 appears well-suited for inference or lightweight training at small context windows, but is not viable for production-scale model training.

4.3 CPU Degradation

Both AMD Ryzen CPUs followed a similar trajectory to the Apple M2 but with less resistance. At large complexity with block sizes 64 or 128, both CPUs fell below 500 tokens/sec — making training times measured in hours versus the seconds-per-iteration achieved by the discrete GPUs. CPUs should be considered last-resort hardware for any task above low complexity.

4.4 Loss Improvement with Block Size

The consistently lower loss values at larger block sizes are a useful secondary finding. Training loss improved from ~1.77 (block-32, mid) to ~1.60 (block-64, mid) to ~1.41 (block-128, mid), with validation loss following the same trend. This confirms that providing the model with more contextual information per step meaningfully improves learning quality — though further experiments would be needed to determine whether this holds as iterations and learning rates are also adjusted.

4.5 Limitations

- Energy metrics are limited to NVIDIA GPUs via pynvml. Cross-platform energy comparisons (M2, AMD) are not possible with current instrumentation.

- Tiny Shakespeare is a small, low-diversity corpus. Results may not generalise to larger datasets, longer training runs, or different domains.
- The fixed batch size of 12 is conservative and may underutilise GPU VRAM, particularly at low complexity. Larger batch sizes could alter the relative throughput and efficiency rankings.
- Only 2,000 training iterations were run. Sustained-performance differences (thermal throttling, memory pressure over time) are not captured.
- A single model run per hardware/configuration pair was conducted; no statistical averaging across seeds was performed for the performance metrics.

5. Conclusion

This experiment demonstrated that hardware performance in character-level language model training is deeply contingent on the interaction between model complexity and context window length. Training and validation loss remained hardware-invariant within each block-size configuration, confirming sound experimental controls. Performance divergence manifested exclusively through runtime, throughput, and energy-efficiency metrics.

The RTX 4060 dominated runtime and token throughput at medium and large complexity, making it the superior choice for time-constrained training. The RTX 3070 demonstrated surprising energy-efficiency advantages at specific workload profiles which most dramatically at large complexity with block size 64, where it was 239% more energy-efficient. The Apple M2 was competitive and sometimes superior at low complexity, but degraded severely at scale. AMD CPUs were viable only at the lowest workload tier.

These findings carry practical implications: hardware selection for language model training should be guided by the anticipated model complexity and sequence length rather than by GPU generation alone. For practitioners operating under energy or thermal constraints, the RTX 3070 may outperform newer hardware at specific configurations. For practitioners optimising for speed, the RTX 4060 is generally preferred except at very low model complexity.

The results also demonstrate that increasing block size consistently improves model quality within a fixed iteration budget, suggesting that longer context windows are an accessible lever for improving character-level language model performance.

6. Future Work

- Larger dataset: Tiny Shakespeare is a small, low-diversity corpus. Introducing a larger dataset such as OpenWebText would increase the number of required training iterations and provide a more realistic benchmark for sustained performance across longer runs.
- Cross-platform energy monitoring: Integrating Intel RAPL for x86 CPUs and Apple-specific profiling utilities would enable apples-to-apples energy comparisons across all five hardware configurations, removing the current NVIDIA-only constraint.
- Statistical replication: Running multiple seeds per hardware/configuration pair and reporting mean ± standard deviation for performance metrics would strengthen the statistical validity of the comparisons.
- Larger model scales: Extending beyond the Large complexity tier (12 heads / 12 layers / 768-dim) would determine whether the RTX 4060's throughput advantages over the RTX 3070 continue to grow, or whether VRAM becomes a bottleneck that inverts rankings at very large scales.