# Effort, Not Bits

How GLM-5.2, a 744B Open Model on a Desktop, Comes Within a Few Points of Opus 4.8 and GPT-5.5, and Why Reasoning Effort, Not Quantization, Closes the Gap

Dan Siroker

June 2026

**Abstract**

We study whether a 744-billion-parameter open-weight Mixture-of-Experts (MoE) model, GLM-5.2, (released under an MIT license) can be run usefully on a single workstation, and how it compares to two proprietary frontier systems (Claude Opus 4.8 and GPT-5.5) when reasoning effort is varied as an explicit condition rather than left at each system's default. We run three quantizations of GLM-5.2 (2-bit/217 GB, 4-bit/340 GB, and 4-bit-XL/434 GB) locally on a Mac Studio M3 Ultra with 512 GB of unified memory via `llama.cpp`, and evaluate all five configurations at both their lowest and highest reasoning effort, ten conditions in total, on four automatically gradable benchmarks (GPQA-Diamond, MATH-500, MMLU-Pro, HumanEval+), 100 items each, for $n$=400 items per condition with Wilson 95% confidence intervals. We report five findings. (1) At maximum effort the best local configuration reaches 91% versus 94 to 95% for the frontier, a real but narrow gap of a few points. (2) Almost all of that gap comes from one benchmark, GPQA-Diamond; on math, knowledge, and code the local model is statistically even with the frontier. (3) Reasoning effort dominates quantization: raising effort buys +5 to +10 points, whereas moving from a 217 GB model to a 434 GB model buys about 1. (4) Consequently 2-bit is the efficiency winner on these short-answer benchmarks. (5) The cost of local frontier-adjacent quality is latency, about 100 seconds per question versus about 10, not capability. An option-permutation probe finds the frontier's GPQA advantage is position-invariant, ruling out shallow answer-key contamination as its cause. We also document two harness artifacts that nearly produced false conclusions, and we are explicit about where the data is noisier than a single reported digit suggests. All code and raw generations are open-source at `github.com/dsiroker/local-llm-benchmarks`.

## 1 Introduction

The release of large open-weight language models under permissive licenses raises a practical question for builders: can a frontier-class model be self-hosted on hardware one owns, and if so, what is lost relative to proprietary APIs? We examine this concretely for GLM-5.2, a 744B-parameter MoE, by running it locally and comparing it to Opus 4.8 and GPT-5.5.

A fair comparison must confront a confound that informal benchmarks usually ignore: modern systems expose a reasoning-effort dial that trades latency for accuracy, and they ship with different defaults. Comparing one model's default against another's measures the vendors' product decisions, not the models. We therefore run every model at both the lowest and highest setting of its own dial. We are careful not to overstate this as "equal compute," since the dials are not commensurable across vendors; it is each model at its own endpoints. This framing also turns two deployment questions into measurable quantities: how much does quantization cost, and how much does effort

buy. Effort turns out to be the larger lever, the residual frontier advantage is small and localized to the hardest science questions, and careless methodology in grading or serving can produce confidently wrong conclusions.

# 2 Experimental Setup

## 2.1 Hardware and serving

All local inference ran on a Mac Studio M3 Ultra (32-core CPU, 80-core GPU) with 512 GB unified memory and a 16 TB SSD, purchased 14 February 2026 for $14,099. As of June 2026 this configuration is no longer available from Apple: the 512 GB option was removed in early March 2026 during the global DRAM shortage, as memory vendors shifted capacity toward high-bandwidth memory for AI accelerators, and lower-memory Mac Studio configurations have since gone intermittently unavailable with multi-month lead times. The study therefore runs on roughly the ceiling of what was briefly purchasable. We used `llama.cpp`'s OpenAI-compatible server (`llama-server`, flash attention enabled, all layers offloaded to the Metal backend). Because GLM-5.2 is an MoE with only about 30 to 40B parameters active per token, generation is bounded by memory bandwidth over the active experts rather than the full parameter count, which yields usable throughput despite the model's size. Loading the larger quantizations requires raising the macOS GPU wired-memory limit (`iogpu.wired_limit_mb`) above its default; we set 480 GB. We serve through the chat-completions endpoint so generation uses the model's chat template and stop tokens. An earlier version using the raw completion endpoint produced degenerate, repetitive output that mimicked a quantization cliff (Section 6).

## 2.2 Models and quantizations

We evaluate three Unsloth dynamic GGUF quantizations of GLM-5.2: `UD-Q2_K_XL` (2-bit, about 217 GB), `UD-IQ4_XS` (4-bit importance-matrix, about 340 GB), and `UD-Q4_K_M` (4-bit-XL, about 434 GB). Opus 4.8 and GPT-5.5 were accessed through their command-line interfaces with structured JSON output so token usage, including reasoning tokens, is recorded. The frontier numbers are point-in-time against hosted models and may drift; we did not pin exact model-snapshot identifiers, and benchmark runs were intended to be single-shot with tool use disabled (Section 7).

## 2.3 Benchmarks and grading

We use four hard, automatically gradable benchmarks, 100 items each (the first 100 of each test split): GPQA-Diamond (graduate-level multiple-choice science), MATH-500 (competition mathematics, free-form), MMLU-Pro (broad 10-way multiple-choice knowledge), and HumanEval+ (code generation, executed against the EvalPlus expanded test suite). Multiple-choice items are graded by extracting the selected option; MATH answers are graded by a normalizer that strips formatting (degree marks, units, variable prefixes, text after a final "="), with a symbolic-equivalence fallback via `sympy`; code is graded by execution. The first-100 slice is not a random sample and is not comparable to published full-set scores.

## 2.4 Reasoning configurations

Each model is evaluated at both its lowest and highest reasoning setting, giving ten conditions. For the frontier APIs we set the effort parameter directly (Claude: `low`/`max`; GPT-5.5: `low`/`xhigh`). For GLM we use `enable_thinking=false` as the low bookend and maximum reasoning as the

high bookend. Note that the GLM low bookend disables chain of thought entirely, so the GLM low-to-max delta partly reflects turning reasoning on at all and is not directly comparable to the frontier low setting, which still reasons. GLM's maximum effort does not reliably terminate on hard open-ended items, so the high bookend uses budget forcing (Section 4). Sampling used temperature 0.6, top-$p$ 0.95, and a fixed seed for the natural pass; forced commits used greedy decoding.

### 2.5 Statistics

Each item is scored binary correct or incorrect. We reach $n$=400 per condition by stacking four 25-item batches (offsets 0/25/50/75) and merging by item index; the harness is fully resumable, so reboots and a mid-run power outage cost no completed work. We report Wilson 95% confidence intervals on the 400-item overall accuracy. These intervals reflect item sampling only; they do not include generation-level variance (one sample per item) or the clustering induced by pooling four different benchmarks, both of which widen the true uncertainty. In total the sweep plus the contamination probe comprise 4,437 graded model responses and about 141 hours of cumulative compute (roughly 113 GPU-hours of local generation on the Mac, plus the API baselines and the probe).

## 3 Results

### 3.1 Overall accuracy

Table 1 and Figure 1 summarize accuracy. At maximum effort, GPT-5.5 (95%) and Opus 4.8 (94%) lead; the best local configuration, GLM 4-bit-XL at max effort, reaches 91%, with 2-bit and 4-bit at max effort both at 90%. The frontier advantage is real, the intervals separate at $n$=400, but narrow, a few points for a model running entirely on a workstation. We caution that the per-benchmark cells carry several points of run-to-run sampling variance (Section 5 quantifies this on GPQA), so the gap is best read as a few points rather than a precise value. Notably, the frontier models at low effort (92%) score about the same as the best local model at max effort.

Table 1: Accuracy per benchmark (out of 100) and overall ($n$=400) with Wilson 95% CIs. The low and max rows are each model at its own lowest and highest effort, not equal compute across vendors. "s/q" is median wall-clock per question; "tok/s" is median local generation throughput (local only; API throughput is provider-reported and not directly comparable). Running the released `analyze.py` reproduces this table.

| Condition | GPQA | MATH | MMLU-Pro | HE+ | **Overall** | s/q | tok/s |
|---|---|---|---|---|---|---|---|
| GPT-5.5, max | 99 | 95 | 90 | 96 | **95% [-3/+2]** | 11s | 60 |
| Opus 4.8, max | 95 | 95 | 91 | 94 | **94% [-3/+2]** | 12s | 75 |
| GPT-5.5, low | 88 | 94 | 89 | 95 | **92%** | 7s | 33 |
| Opus 4.8, low | 89 | 97 | 87 | 93 | **92%** | 6s | 44 |
| GLM 4-bit-XL, max | 85 | 95 | 92 | 93 | **91% [-3/+2]** | 103s | 14 |
| GLM 4-bit, max | 84 | 94 | 90 | 94 | **90%** | 106s | 14 |
| GLM 2-bit, max | 84 | 94 | 88 | 95 | **90%** | 103s | 17 |
| GLM 4-bit, low | 73 | 89 | 81 | 96 | **85%** | 22s | 14 |
| GLM 4-bit-XL, low | 74 | 86 | 82 | 91 | **83%** | 24s | 14 |
| GLM 2-bit, low | 63 | 88 | 83 | 90 | **81%** | 20s | 16 |

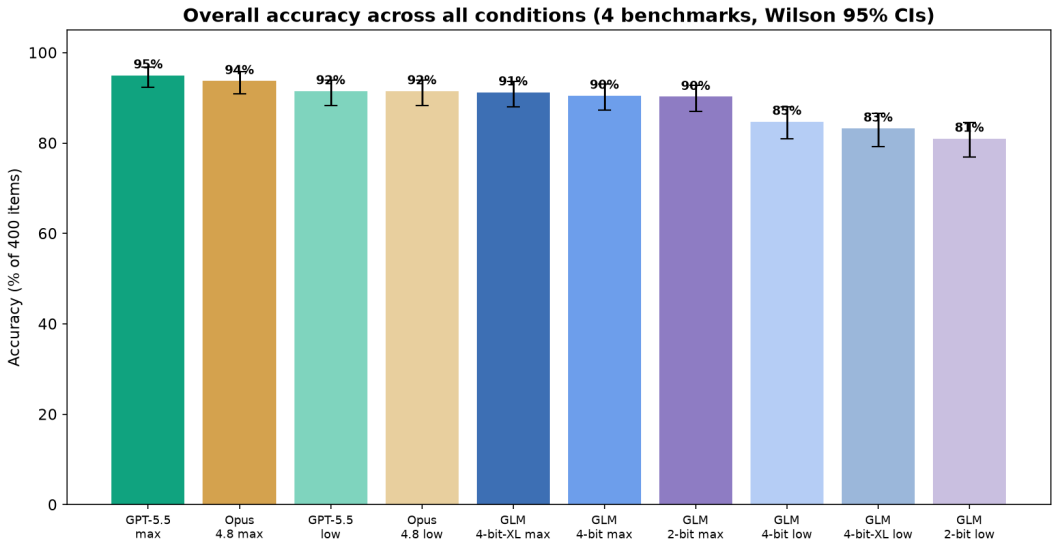


Figure 1: Overall accuracy with 95% Wilson confidence intervals across all ten conditions.

### 3.2 The gap is almost entirely GPQA

Decomposing the overall scores by benchmark localizes the frontier advantage to a single column (Table 2). On math, broad knowledge, and code the best local condition is within a point of the best frontier condition at the same effort, and the MMLU-Pro difference is within noise (a tie, not a local win). Only on GPQA-Diamond does a substantial gap remain. Two caveats attach to that gap. The GPQA absolutes here exceed published full-set numbers, which suggests the first-100 slice and the ungated mirror's answer key make this subset easier than the canonical set; and the exact $-14$ is the noisiest cell in the study, since the re-run (Section 5) reproduced GPQA several points lower. The direction is robust; the magnitude is soft.

Table 2: Best frontier vs. best local condition at max effort, per benchmark (out of 100).

| Benchmark | Best frontier (max) | Best local (max) | Gap |
|---|---|---|---|
| GPQA-Diamond (hard science) | 99 (GPT) | 85 | $-14$ |
| MATH-500 | 95 (GPT) | 95 | 0 |
| MMLU-Pro | 91 (Opus) | 92 | +1 (tied) |
| HumanEval+ (code) | 96 (GPT) | 95 | $-1$ |

### 3.3 Reasoning effort dominates quantization

The two deployment dials move quality by very different amounts. Raising reasoning effort buys +5 to +10 points (2-bit 81% to 90%; 4-bit 85% to 90%). Moving from the 217 GB 2-bit model to the 434 GB 4-bit-XL model, at matched max effort, buys about 1 point (90% to 91%). At max effort the three quantizations' confidence intervals overlap almost entirely (Figure 3); the largest file does not meaningfully outperform the smallest. We note that the GLM effort delta is inflated by the low bookend being reasoning-off rather than low-but-reasoning, so the magnitude overstates a true low-to-high comparison; the ordering, effort over quantization, is unaffected.

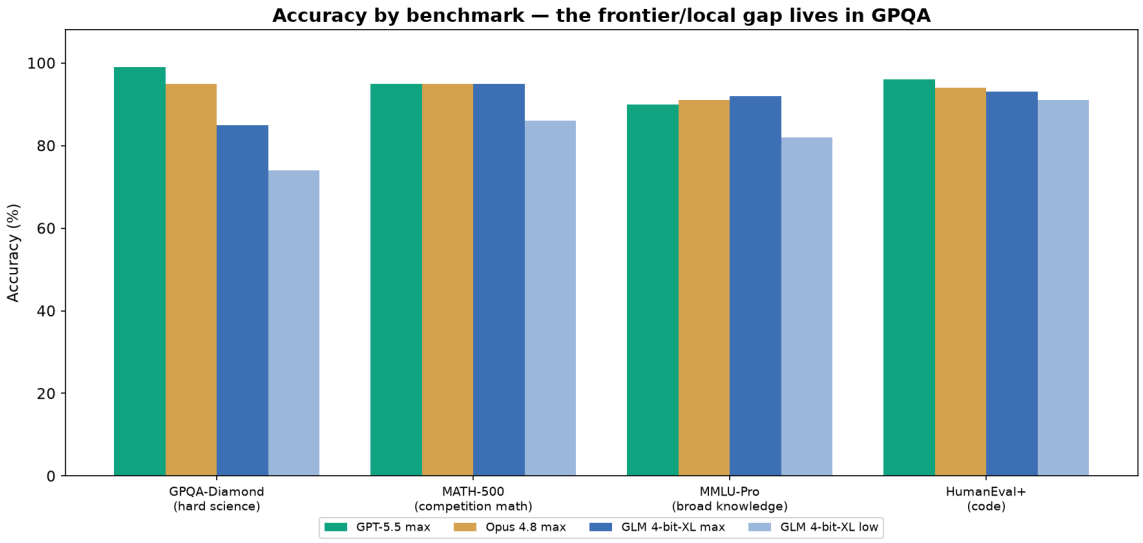


Figure 2: Per-benchmark accuracy. The frontier-local gap is confined to GPQA-Diamond.

### 3.4 2-bit is the efficiency winner on these benchmarks

GLM 2-bit at max effort (90%, 217 GB) ties 4-bit (90%, 340 GB) and trails 4-bit-XL by a single point (91%, 434 GB). At roughly half the memory of 4-bit-XL it gives up almost nothing on these four short-answer reasoning and coding benchmarks, and it leaves the most headroom to hold a large context window. This recommendation is scoped: long-context coherence, instruction following, and long-form or agentic generation are exactly where low-bit quantization tends to degrade, and none are tested here. The higher forced-commit rate at 2-bit (Table 3) is a mild signal of less stable convergence.

### 3.5 Throughput, latency, and cost

Local generation ran at 14 to 17 tok/s. The cost of the best local quality is latency, not capability: max-effort answers take a median of about 100 to 106 s per question, versus about 10 s for the frontier APIs, roughly an 8 to 10 times wall-clock penalty for the final few points of quality (Figure 4). Low-effort local is far more usable at about 20 s per question and still lands at 81 to 85%. There is also a cost asymmetry: the workstation was a $14,099 capital outlay, and at API prices these 400 questions cost cents. Local wins on privacy, ownership, offline use, and the absence of rate limits, not on dollars per token or speed. We report API tokens/s for context only; it reflects datacenter serving and is not directly comparable to local throughput.

## 4 Budget Forcing: Making Maximum Reasoning Terminate

GLM-5.2 at maximum reasoning effort will, on genuinely hard items, reason without converging. We observed single GPQA items consuming more than 48,000 tokens (over 85 minutes) without ever emitting a final answer. Left unbounded, a handful of items would never terminate and the condition could not be scored. Following the budget-forcing technique of Muennighoff et al. (s1), we let the model reason up to a 12,000-token budget and, if it has not answered, feed its own analysis back with thinking disabled and a strict format demand under greedy decoding, retrying once on a format miss. This guarantees termination while preserving the model's own reasoning.

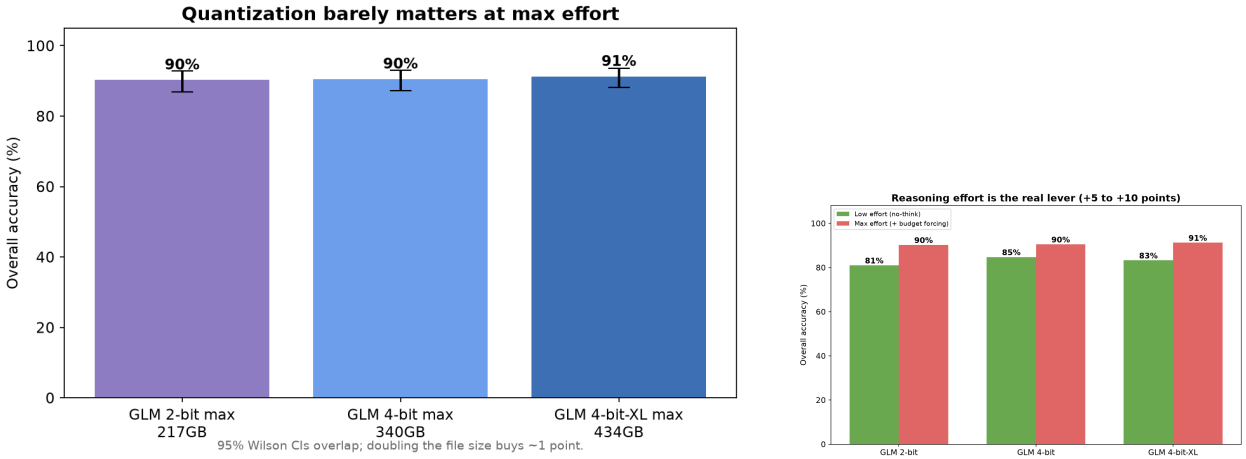


Figure 3: Left: the three GLM quantizations at max effort are statistically indistinguishable. Right: low vs. max effort within each quantization, the dominant lever.

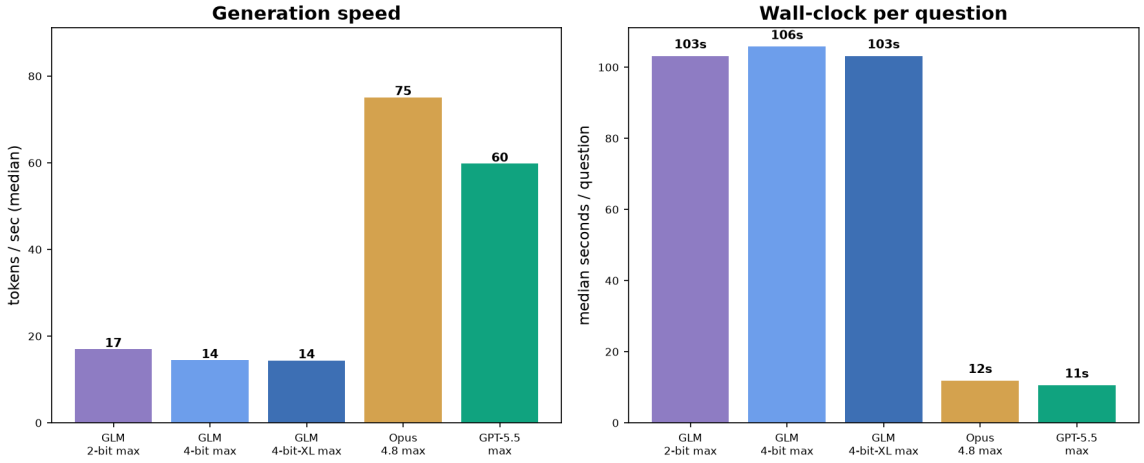


Figure 4: Quality versus speed. The local model matches the frontier on what it can answer, but not on how fast.

The forced-commit telemetry is informative (Table 3). Across the GLM max-effort conditions, 15 to 19% of items required forcing. The model is 93 to 94% accurate when it commits on its own and about 70% when forced, so the items it cannot converge on are the hard ones it would likely miss anyway. Two honest limits: the 12,000-token budget is a fixed choice we did not sweep, and the frontier models were not budget-capped, so the handling is asymmetric. A budget ablation (8k, 12k, 24k, 48k) would show whether 12k under-reports GLM; we have not run it. This reverses a conclusion from an earlier, smaller pilot, which, lacking a forcing mechanism and using a generous fixed cap, found that maximum reasoning "backfired" because it truncated. With termination guaranteed, maximum effort exceeds the low bookend by +5 to +10 points across every GLM quantization.

Table 3: Budget-forcing behavior for the GLM max-effort conditions ($n$=400 each).

| Condition | Forced-commit rate | Acc. (natural) | Acc. (forced) |
|---|---|---|---|
| GLM 2-bit, max | 19% | 94% | 72% |
| GLM 4-bit, max | 16% | 93% | 68% |
| GLM 4-bit-XL, max | 15% | 94% | 74% |

## 5 Contamination Robustness

Because the frontier-local gap is carried by GPQA-Diamond, a public benchmark that could appear in any model's pretraining data, we stress-tested that gap for memorization with an option-permutation probe. For each item we cyclically rotate the four answer choices by a per-item offset (1 to 3, never 0) so the correct content always lands on a different letter; the question, distractors, prompt, and grader are otherwise byte-identical. A model relying on a memorized question-to-letter mapping collapses under rotation; a model that reasons tracks the content to its new letter. We ran the original and permuted versions at maximum effort and compare paired by item index (Table 4).

Table 4: GPQA accuracy under option permutation (paired by item, max effort). $\Delta$ is permuted minus original.

| Condition | Original | Permuted | $\Delta$ | $n$ |
|---|---|---|---|---|
| GPT-5.5, max | 93% | 92% | $-1$ | 99 |
| Opus 4.8, max | 92% | 89% | $-3$ | 99 |
| GLM 4-bit-XL, max | 87% | 89% | $+2$ | 39 |

No condition moves by more than 3 points, within sampling noise, whereas a memorized answer key would drop accuracy by 20 to 40 points. The frontier's GPQA advantage is therefore position-invariant: it reflects reasoning rather than a scraped answer key, and the effect is symmetric (the local model behaves the same), so contamination of the answer mapping does not explain the gap. Three caveats temper this. First, the re-run scored GPT-5.5 and Opus at 93% and 92% on GPQA, versus 99% and 95% in the main run, several points of run-to-run variance at temperature $> 0$. Second, permutation rules out shallow letter-memorization but not deep solution-memorization; only a benchmark released after every model's training cutoff would settle that, and our high GPQA absolutes are themselves a soft contamination flag. Third, the GLM permutation cell is $n$=39 on one quant, so the symmetry claim is preliminary.

## 6 Two Methodological Cautionary Tales

**Grading.** A naive MATH grader that required near-exact string matches reported Opus 4.8 at 72% on MATH-500, penalizing correct answers formatted as `90` versus `90^\circ` or `5` versus `x=5`. A uniform re-grade with a normalizer raised Opus by 20 points and overturned the qualitative conclusion (an earlier draft claimed the local model beat Opus at math; it does not). Because the grader was loosened after an initial pass, it is a researcher degree of freedom; it is applied uniformly to all models, but its false-positive rate is not yet audited.

**Serving.** An early harness queried the raw completion endpoint, which, bypassing the chat template and stop tokens, caused the 2-bit model to ramble and repeat, producing arithmetic errors and a low math score that looked like a catastrophic quantization cliff. Serving the identical weights through the chat-completions endpoint eliminated the effect. We nearly published a false "2-bit is broken" claim. The lesson: validate the serving path before attributing quality differences to the model.

## 7 Limitations

(1) Per-benchmark resolution: although the overall $n$=400 CIs are tight, each per-benchmark cell is $n$=100 (about $\pm$5 points), and a re-run reproduced GPQA several points lower, so the $-14$ GPQA gap is robust in direction but soft in magnitude. (2) Single sample: one generation per item at temperature $> 0$; the permutation re-run gives a partial estimate (several points on GPQA), but generation variance is otherwise unmeasured. (3) Frontier access path: GLM via `llama.cpp`, Opus and GPT-5.5 via their CLIs, which impose their own system prompts and could in principle use tools; runs were intended to be single-shot with tool use disabled, and reproducers should confirm this, especially for HumanEval+ and GPQA. (4) Contamination and sampling: we used the first 100 items of each split rather than a random sample, and GPQA-Diamond came from an ungated mirror whose key we did not fully verify; the permutation probe rules out shallow answer-key memorization but not deep solution-memorization. (5) Throughput comparability: API tokens/s is provider-reported. The robust conclusions are the large effects, the effort-over-quantization ordering, the GPQA-localized gap, the budget-forcing behavior, and the latency penalty, rather than the exact ordering of the near-tied 90 to 92% conditions.

## 8 Conclusion

A 744B open-weight model runs usefully on a single 512 GB workstation and, at maximum reasoning effort, reaches 91% against a 94 to 95% frontier on a hard four-benchmark suite, a gap that is almost entirely attributable to graduate-level science and vanishes on math, knowledge, and code. For practitioners the actionable findings are clear: reasoning effort is a far larger quality lever than quantization; aggressive 2-bit quantization is the efficiency sweet spot on short-answer tasks; maximum reasoning is worth enabling if paired with a termination mechanism such as budget forcing; and the true cost of local frontier-adjacent quality is latency and capital, not capability. Our experience also underscores that local-model evaluation is dominated by harness, serving, and grading artifacts, and that controlled effort, uniform re-grading, and confidence intervals are not optional. All code and raw generations are at `github.com/dsiroker/local-llm-benchmarks`.

## Disclosure: Use of AI Tools

The author used Claude Opus 4.8 (Anthropic), one of the systems benchmarked in this paper, to assist with writing the benchmark harness, running and analyzing the experiments, and drafting and editing this manuscript. The author designed the study, reviewed and verified all results, code, and text, and takes full responsibility for the final paper. Per current publishing norms (ICMJE, COPE), AI tools are not credited as authors because they cannot take accountability for the work.

## References


[1] Z.ai. *GLM-5.2: An open-weight 744B Mixture-of-Experts model.* 2026.

[2] Muennighoff et al. *s1: Simple test-time scaling.* 2025.

[3] Rein et al. *GPQA: A Graduate-Level Google-Proof Q&A Benchmark.* 2023.

[4] Hendrycks et al. *Measuring Mathematical Problem Solving with the MATH Dataset.* 2021.

[5] Wang et al. *MMLU-Pro: A More Robust and Challenging Multi-Task Language Understanding Benchmark.* 2024.

[6] Liu et al. *Is Your Code Generated by ChatGPT Really Correct? (EvalPlus).* 2023.

[7] Gerganov et al. *llama.cpp: LLM inference in C/C++.* 2023–2026.