The Trunk Tax

What Four-Bit-Expert Models Spend Their Decode Bandwidth On, the Measured Refund, and the Quality Metric That Fails in Both Directions

Mohammad Alsufi* **Connor Boone***

and the Brainsless Research Lab AI Systems Research Group

*Equal lead contribution `research@brainsless.com`

July 2026 | Technical Report BRL-2026-13

Brainsless Research Lab | Technical Report **BRL-2026-13** | July 2026

DOI 10.5281/zenodo.21711187 | github.com/brainsless/the-trunk-tax

Abstract

The frontier mixture-of-experts releases from Moonshot and OpenAI ship 4-bit experts behind a 16-bit trunk, and Qwen's official Int4 release follows the same pattern. We show, from safetensors headers alone, that this makes the trunk the dominant term in batch-1 decode traffic: Kimi K3 reads **137.0 GB per token**, of which 111.2 GB is trunk. The figure circulating since the K3 release, 25 GB per token, is correct expert-only arithmetic; as the per-token planning number it is used for, it is low by 5.3×. Quantizing only the trunk converts the traffic back into speed: 1.36× **on OLMoE-1B-7B and** 1.32× **on gpt-oss-20b**, measured single-stream on lean stacks (MLX and llama.cpp/CUDA), with perplexity held; on today's datacenter engines the software floor we measured in BRL-2026-11 bounds the trunk-FP8 refund to roughly 1.11× until the floor is removed. The naive version of this refund fails on gpt-oss with a +53% chat-perplexity regression. Controlled attribution shows the regression is dominated by round-to-nearest quantization of the *embedding table* at group size 64 — a tensor whose decode-bandwidth cost is one row per token. The most-downloaded MLX build of gpt-oss-20b ships exactly this configuration; we measure the regression in the public artifact, show it is irreducible by temperature rescaling and repairable at zero bandwidth cost — and show it is invisible on wikitext, on a 1,172-item ARC eval, in greedy-generation divergence, and to a blind frontier-model judge over 300 sampled generation pairs (healed-versus-damaged win rate 51%, CI 42–60%). The regression is real and mechanistically attributed; by every user-facing probe we could construct it is practically latent, which indicts teacher-forced perplexity, the ecosystem's default quality metric for quantized builds, from both directions. A config-level audit of 103 quantized MLX MoE builds finds the same risky default across that ecosystem's most popular artifacts; family-by-family measurement finds the damage concentrated in exactly one of the six checkpoints tested across four lineages. Census tool, harness, and all raw cells are released.

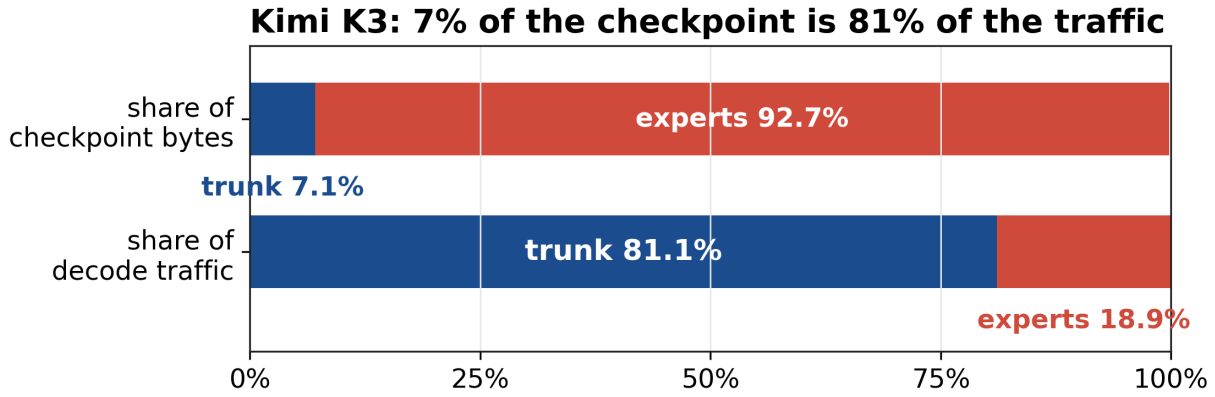

Figure 1. Kimi K3. Within each bar, left to right: trunk (BF16, blue) and routed experts (MXFP4, coral). Top: share of checkpoint bytes. Bottom: share of per-token decode traffic. Embeddings (one row per token) and the vision encoder are excluded from traffic.

model	experts	trunk	GB/token	trunk share
Qwen3.5-122B-A10B	BF16	BF16	18.02	59.8%
gpt-oss-120b	MXFP4	BF16	5.00	61.9%
Kimi-K2.6	INT4	BF16	32.99	64.0%
Kimi-K3	MXFP4	BF16	**136.99**	**81.1%**
Qwen3.5-122B-GPTQ-Int4	Int4	BF16	12.66	85.0%

Table 1. Dtype-verified census of five shipped checkpoints (MTP and vision modules excluded). Two controlled pairs sit inside the table: the Qwen pair isolates *precision* (same architecture, 59.8% → 85.0%); the K2.6→K3 pair isolates *architecture* at a fixed precision regime (64.0% → 81.1%).

1 The tax

A safetensors file begins with a JSON header listing every tensor, its dtype, and its shape. Reading the headers of all 96 shards of `moonshotai/Kimi-K3` costs about 60 MB of HTTP range requests against a 1.56 TB checkpoint. From the 497,220 tensors: routed experts 1,446.5 GB in MXFP4; trunk 111.2 GB in BF16 (attention 72.4, shared experts 24.3, latent projections 9.5, head, dense layer, and gates 5.0). Per token, decode touches 16 of 896 experts (25.83 GB) and the whole trunk (111.16 GB): **136.99 GB in total** (Figure 1).

The launch-week figure in circulation, "approximately 25 GB of data read per token," appears on the model's own Hugging Face discussion board (`moonshotai/Kimi-K3`, discussion #59). As expert-only arithmetic the post is correct; the problem is its downstream use, anchoring hardware plans and a 3–6 tok/s decode estimate as if it were the whole per-token cost. The trunk quadruples it. The model here counts weight reads only, at batch 1: KV reads add a context-growing term, and at server batch sizes the trunk amortizes across the batch while activated-expert coverage grows, so the bandwidth argument is a batch-1 statement; the capacity argument below survives batching.

Capacity is where the tax bites first at frontier scale. Shipped K3 weights (1.561 TB) exceed an 8×B200 node (1.536 TB) before any working memory. A trunk at FP4 brings weights to ∼1.48 TB, and the fit survives only because K3's KDA attention prices KV at 27.0 KB per token (a 128K-token stream costs 3.5 GB). Single-node 8×B200 K3 exists only behind trunk quantization, at batch-1 class concurrency, against a working-memory budget of roughly 7 GB per GPU whose non-KV

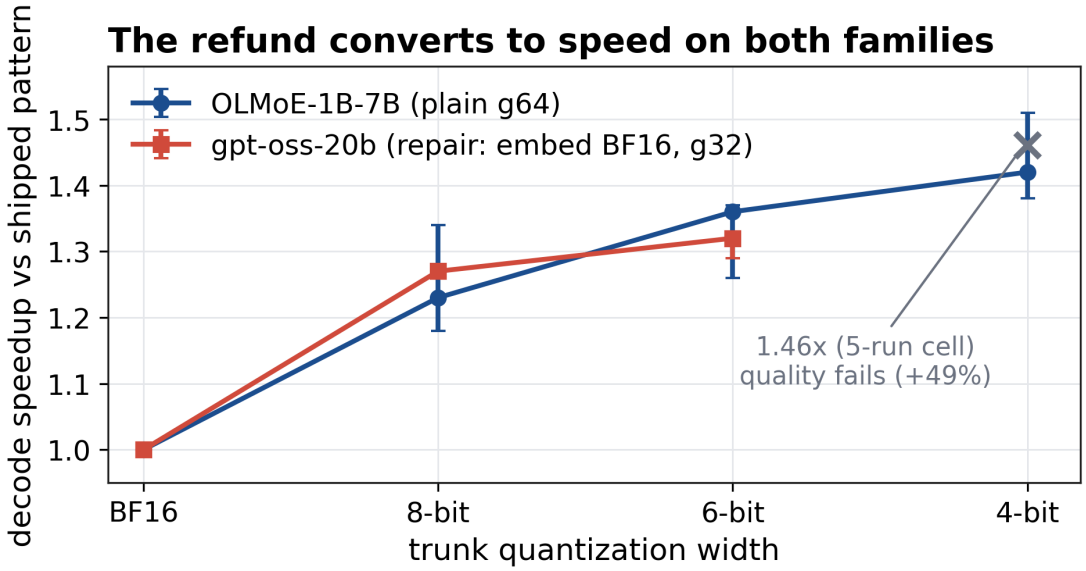

Figure 2. Decode speedup over each model's shipped pattern versus trunk width. OLMoE: plain g64 mixes. gpt-oss-20b: the repair (embeddings BF16, group size 32); its 4-bit point is excluded from the curve and shown only as an annotated marker because quality fails there (+49%). Whiskers are bootstrap 95% CIs over 12 hygiene runs.

terms are untested at 2.8T scale; the fit is stated as conditional on them. On today's engines the speed side is bounded by the software floor we measured in BRL-2026-11: roughly 1.11× for trunk-FP8 on vLLM's day-0 stack, 1.68× only as the floor approaches zero.

2 The refund, measured

To convert the byte accounting into measured speed we ran the full expert-precision × trunk-precision matrix on OLMoE-1B-7B (Apple M4 Pro, MLX, batch-1 greedy). Every headline number is the median of 12 runs across 3 shuffled passes with cooldowns — single-pass numbers proved untrustworthy by up to 37% and are labeled wherever they appear.

cell	tok/s	wikitext PPL
BF16 everything	68.3	11.887
4-bit experts, BF16 trunk (*the shipped pattern*)	118.2	12.191
+ trunk 8-bit	145.9 (1.23×)	12.193 (+0.02%)
+ trunk 6-bit	160.4 (1.36×)	12.210 (+0.16%)
+ trunk 4-bit	167.5 (1.42×)	12.454 (+2.2%)
inverse control: 4-bit trunk, BF16 experts	80.8 (1.18×)	12.134

Table 2. OLMoE-1B-7B matrix. The inverse control fixes the causal order: trunk quantization pays little while the experts stay wide. The certified 6-bit point holds on a chat corpus through OLMoE's own template as well: 5.0875 vs ship 5.0914 (−0.08%); 4-bit reads +0.86% there (artifact olmoe_chat_cells.json).

The 4-bit point costs +2.2% perplexity, and the cost decomposes near-additively across tensor classes (head 41%, attention 41%, embeddings 21%; the 103% sum exceeds what integer rounding alone can produce, so a small interaction term of order one point is present). On gpt-oss-20b — whose MXFP4 expert tensors pass through every cell bit-identical (216 tensors MD5-verified) —

the refund with the repair applied measures 37.9 → 49.9 tok/s, **1.32**×, with chat perplexity 13.43 against the ship pattern's 13.76 and wikitext 136.1 against 143.6. Paired per-chunk ΔNLL across 39 chunks: mean −0.025, bootstrap 95% CI [−0.070, +0.020] — the interval includes zero, so we claim quality no worse than ship at the resolution the instrument demonstrated.

The quality side of the refund holds at frontier scale, with one honest gradient: on official Qwen3.5-122B-A10B weights, measured by quantize–dequantize in floating point, baseline 3.639, whole trunk 8-bit 3.6381, recipe configuration 3.6334 — nothing measurable. The K3 trunk additionally contains tensor classes absent from our end-to-end models — shared experts and latent projections — and the capacity arithmetic of Section 1 assumes they quantize; on Qwen1.5-MoE-A2.7B (a shared-expert architecture), quantize–dequantize of the shared-expert class costs +0.03% at 8-bit g64 and +0.03% at 6-bit g32 (4.2902 / 4.2901 vs baseline 4.2888) — one architecture's evidence, stated as such, that the class is not a second fragile tensor. On the two mid-size Qwen checkpoints the recipe configuration reads +0.43% (Qwen3-30B, 5.1301 vs 5.1082) and +1.69% (Qwen3.6-35B, 10.098 vs 9.9301) — the latter near this instrument's ±2% resolution and the largest recipe delta we measured anywhere.

The speed side replicates on CUDA. On an NVIDIA L4 (llama.cpp, `llama-bench`, 5 repetitions), the public gpt-oss-20b GGUF variants — which share identical MXFP4 expert tensors and differ only in trunk precision — decode at 62.1 tok/s with the F16 trunk, **84.2 tok/s (1.36×) at Q8_0**, and 92.5 tok/s (1.49×) at Q4_K_M (run-to-run s.d. ≤ 0.3). The shared-expert identity across these files follows from llama.cpp's pass-through of pre-quantized MXFP4 tensors and is not hash-verified here (the MD5 check of Section 2 covers the MLX pipeline). The 4-bit trunk is usable in that ecosystem precisely because k-quants protect the embedding and output tensors by default — an independent stack arriving at the same repair this paper isolates by measurement; we did not re-verify Q4_K_M quality ourselves. Quantization level also changes kernel dispatch in llama.cpp, so these deltas are trunk-bytes-plus-kernel effects, not bytes alone.

3 The fragile tensor

The naive refund fails on gpt-oss. Whole-trunk 8-bit at group size 64 moves chat perplexity 13.76 → 21.06 (+53%; distinct from the +49% residual of the *repaired* 4-bit trunk in Figure 2 — similar magnitudes, different cells); 6-bit reads 31.67 (+130%); full 4-bit destroys the model (PPL 1235). Holding one class at BF16 at a time isolates the culprit (Figure 3).

A kernel-free control fixes the mechanism's location: applying the same quantize–dequantize to the embedding weights in floating point, with no quantized layer in the graph, reproduces the damage (22.31; the fact that this slightly exceeds the whole-trunk-8-bit cell's 21.06 indicates measurement noise or a mild non-additive interaction, and we do not interpret the difference). The arithmetic is destructive on this tensor, independent of implementation. The scope matters: this is a statement about RTN affine g64 on this model family. OLMoE's embeddings tolerate the same treatment at +0.02%; official Qwen3.5-122B weights tolerate it with no measurable change; the same operation on gpt-oss-120b produced no catastrophe in a floating-point pipeline whose out-of-distribution absolute levels limit its sensitivity — and Section 3.1's structural screen argues that null reflects the instrument, not the model.

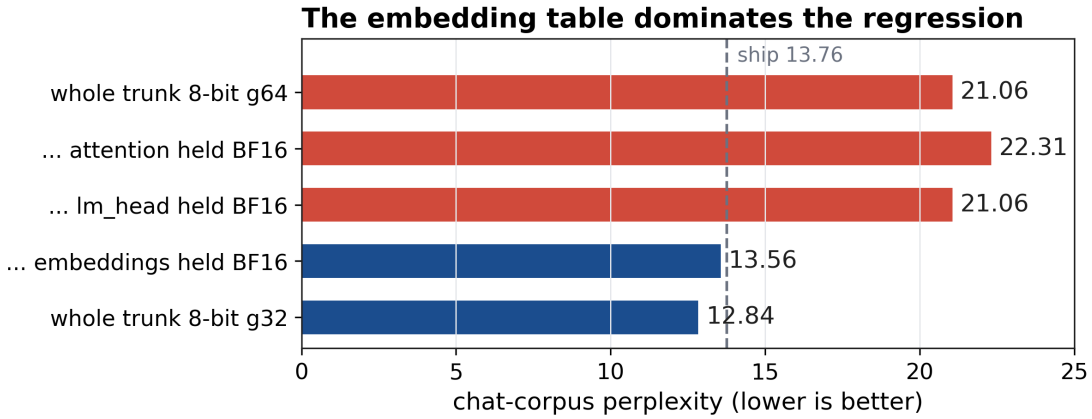

Figure 3. gpt-oss-20b chat perplexity by cell. Holding attention or lm_head at BF16 recovers nothing; holding embeddings at BF16 recovers everything, as does group size 32 alone. Dashed line: shipped baseline (13.76).

3.1 Why this tensor

Distributional analysis of the embedding tables separates the fragile family from the immune ones with a single structural statistic (Figure 4). gpt-oss-20b's embedding rows are extreme outlier carriers: the median per-row ratio of absolute maximum to RMS is **14.1** (99th percentile 20.8), against 3.9 for OLMoE and 3.8 for Qwen3-30B. Under min–max affine quantization, the outlier dimension sets its group's scale, so every other weight in that group is quantized with steps roughly an order of magnitude coarser relative to its own magnitude; halving the group size confines each outlier's blast radius to half as many neighbors, which is why g32 alone recovers quality. The measured per-row reconstruction error at g64 is 1.8× the immune families' (mean relative RMS 0.96% vs 0.54%) — yet still small in absolute terms, with no row exceeding 50% error. The group-local account is measurable directly: defining an outlier group as one whose absolute maximum exceeds 8× its median magnitude, **99.97%** of gpt-oss-20b's g64 embedding groups are outlier groups (OLMoE: 1.1%), and within them the quantization error relative to the group's *typical* weight is 10.5%, against 0.8–1.8% elsewhere (artifact group_stat.json). The damage is therefore invisible not only to standard evaluation corpora (Section 5) but to the reconstruction-error metrics quantization pipelines typically optimize; the norm-dominant outlier coordinates absorb the error budget while the low-magnitude coordinates that carry the row's semantic content are coarsened. The worst-hit rows are a mix of common and rare token fragments; none are special tokens. The absmax/RMS statistic is computable from a checkpoint in seconds. Computed for every checkpoint in this paper plus K3, it separates the gpt-oss lineage from everything else with no overlap: gpt-oss-20b 14.1 and gpt-oss-120b 13.4, against 3.6–4.0 for OLMoE, Qwen1.5-MoE, Qwen3-30B, Qwen3.6-35B, Qwen3.5-122B, and Kimi K3 (artifacts embed_mechanism.json, embed_screen_extra.json, k3_embed_screen.json). The 120b reading revises an earlier inference: its outlier structure matches its fragile sibling, so the null result of our out-of-distribution 120b QDQ pipeline is best read as instrument insensitivity, not robustness — we withdraw the scale-dependent-fragility suggestion in favor of "the structure is a family trait; the 120b functional test needs a better instrument." We propose the statistic as a screen, not a proof, and the functional pathway from coarsened low-magnitude coordinates to next-token damage remains to be traced at the activation level. For K3 specifically (screen 4.0, immune band, simulated QDQ error 0.53%), the capacity-unlocking trunk quantization of Section 1 carries no embedding risk by this screen.

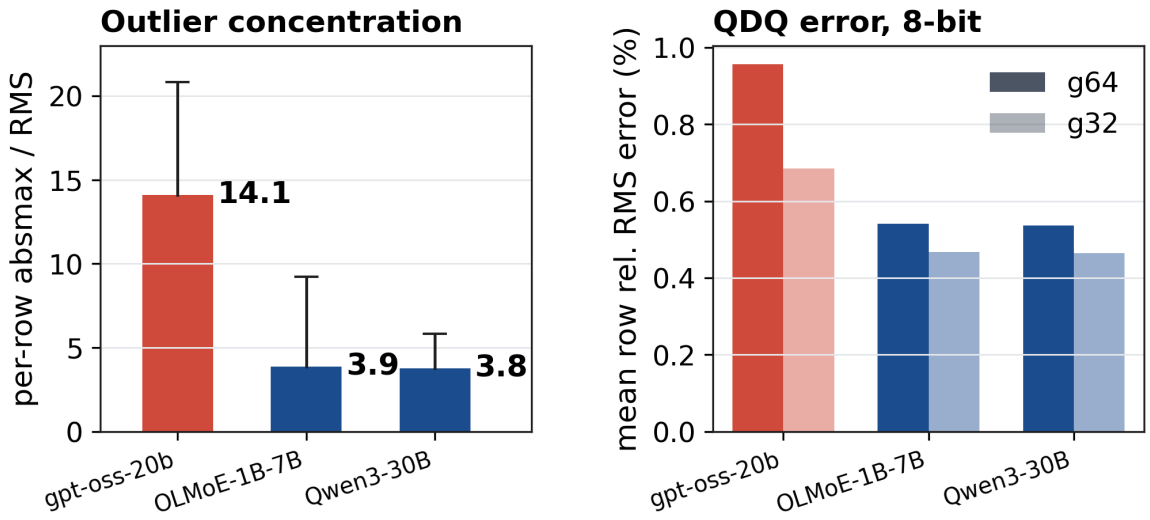

Figure 4. Mechanism. Left: per-row outlier concentration (absmax/RMS; bar = median, whisker to 99th percentile) separates the fragile family (coral) from the immune ones (blue), 14.1 vs ∼3.9. Right: mean per-row relative RMS error under 8-bit RTN at g64 and g32 — small everywhere, which is exactly why reconstruction metrics miss the damage.

Embeddings cost one row per token to read: quantizing them buys nearly nothing and carries all the measured risk, while the tensors that carry the traffic tolerate 8-bit cleanly on both models.

4 The repair

Two zero-cost fixes, applied per family by the attribution measurement: keep embeddings in BF16 (no decode-bandwidth cost), or quantize them at group size 32 (+3% bytes). The repaired 6-bit trunk on gpt-oss is the 1.32× cell of Section 2. On OLMoE, whose embeddings are robust, the optimal mix skips the protection: plain 6-bit g64 (1.36×) beats the protected g32 mix (1.20×) because finer groups carry kernel overhead. The protected-embedding mix itself is standard practice in llama.cpp k-quants and GPTQ/AWQ conventions; the contribution is the attribution instrument that says which protection a given family needs, and the measured curve with experts held fixed.

5 In the wild

`mlx-community/gpt-oss-20b-MXFP4-Q8`, the most-downloaded MLX build of gpt-oss-20b (348,084 downloads), quantizes embeddings at 8-bit affine g64 — its own configuration says so — and measures chat perplexity **21.01**, reproducing our attribution cell (21.06) on the public artifact. Healing the artifact in memory — restoring only the original BF16 embedding rows, fetched by byte range, everything else untouched — recovers it to **13.43**, within 0.004 of our independently converted repair build (13.4306). One tensor separates the damaged public build from clean. The damage is:

- *invisible on wikitext* (140.9 vs 143.6), the corpus class the ecosystem evals against;
- *irreducible by temperature* (best-fit $\tau = 1.0$ on a 0.7–2.0 grid; a global temperature story is excluded, richer calibration stories are untested);
- *not visible on the full 1,172-item ARC-Challenge eval* (ship 49.4%, community 49.1%, repair 48.6%; SE ≈ 1.5 points; a 300-item pilot agreed) — noting the plain-completion probe scores this chat model near 49%, so the null also reflects the probe's weak coupling to the chat distribution where the regression lives;
- *not visible in greedy generation space*: with each build generating 48 greedy 160-token continuations and both builds scoring both generation sets, each scorer rates the two sets within 0.002 nats/token of each other (a constant scorer-level offset of 0.02–0.03 separates the scorers themselves; artifact gen_divergence.json);
- *not visible to a blind frontier judge on sampled generations*: 300 prompt pairs, the shipped build versus the same build with only the embedding healed, identical sampling seeds (temperature 1.0, top-p 0.95), judged blind in both orderings by Kimi K3 served via a third-party inference API. Among order-consistent decisive pairs the healed build wins 51.3% (CI 42–60%). The judge exhibits a 58.4% first-position preference among decisive single judgments; the order-consistent design controls for this by counting only pairs judged identically under both orderings (75 order-split pairs excluded). The study was run twice: an initial protocol whose truncated outputs failed validity checks was replaced before analysis, and both runs' logs are released (artifact judged_verdicts.json).

The severity picture is therefore final and two-sided: the damage is large and mechanistically attributed at the teacher-forced distribution level, and no user-facing probe we constructed — powered multiple-choice, greedy divergence, or blind judged sampling — can detect it, with judged harm bounded above by the CI. Teacher-forced perplexity thus misreports quantization quality in both directions on this artifact: the ecosystem's standard corpora missed a +53% regression, and the regression itself overstates what users experience. Which metric to trust for quantized-build quality is, on this evidence, an open problem the field should treat as such.

A config-level audit of 103 quantized MLX MoE builds finds the same default (quantized embeddings, affine g64) in **70 of 103**, including the most-downloaded artifacts in that ecosystem: `lmstudio-community/Qwen3.6-35B-A3B-MLX-4bit` (566K downloads, embeddings at 4-bit), `Qwen3-Coder-30B` (158K), and `Qwen3-30B` (85.7K). The GGUF ecosystem's k-quant defaults protect embeddings and are not audited here. Family-by-family measurement answers what config inspection cannot: on official Qwen3-30B weights, embedding quantize–dequantize costs nothing at 8-bit or 4-bit (5.113 / 5.103 vs baseline 5.108); the shipped 24K-download artifact confirms by paired healing (6.019 as-shipped vs 6.027 healed — no bleed); and the 566K build's family clears at both widths on official Qwen3.6-35B weights (9.930 baseline; 9.783 / 9.652).

Six embedding-tested checkpoints across four lineages: one bleeds, and it is the one whose most popular build ships damaged. The risky default is widespread, the damage is concentrated, and nothing but the per-build test — minutes with the released tools — says which side a checkpoint falls on.

6 Related work

Protected embedding and output tensors are long-standing practice in deployed quantization: llama.cpp's k-quant mixes keep `token_embd` and `output` at higher precision (with explicit `-token-embedding-type` / `-output-tensor-type` overrides), and GPTQ (Frantar et al., arXiv:2210.17323) and AWQ (Lin et al., arXiv:2306.00978) conventionally exclude embeddings from weight quantization; selective per-tensor precision is also the basis of community "dynamic" GGUF mixes. Our contribution on that axis is not the mix but the controlled attribution instrument, the family-level measurement of when protection is actually needed, and the structural statistic that predicts it. The outlier phenomenon we measure in embedding rows is a relative of the activation-outlier channels documented by LLM.int8 (Dettmers et al., arXiv:2208.07339) and addressed by rotation methods such as QuaRot (Ashkboos et al., arXiv:2404.00456) and SpinQuant (Liu et al., arXiv:2405.16406); those works target activations and linear layers, and we are not aware of a published per-family fragility census of *embedding* tables in shipped MoE artifacts. On the traffic side, batch-1 decode memory floors are measured across GPUs by arXiv:2605.30571, and arXiv:2507.15465 argues MLA and MoE moved the serving bottleneck away from attention — a pre-quantization-era framing that our census inverts for the 4-bit-expert generation: once experts are 4-bit, the dense trunk again dominates the bytes. We are not aware of prior checkpoint-level, dtype-verified accounting of trunk versus expert decode traffic across shipped MoE releases.

7 Methods

Census. HTTP range requests against shard headers; tensor set verified against the index weight map (497,220 exact), byte totals against the index (exact), predicted shard sizes against served Content-Length (exact on shards 1/13/96); classification audited per template with architectural counts ($82{,}432 = 92 \times 896$ expert tensors; 69 KDA +24 MLA attention layers). The tool reproduces our published K2.6 numbers.

Speed. Batch-1 greedy, one 200-token generation per run, short context; medians of 12 across 3 shuffled passes, 20 s cooldowns, warmup discarded. Bootstrap 95% CIs on every quoted tok/s median are released; the headline comparisons are CI-separated (gpt-oss ship $[37.1, 38.2]$ vs repair $[49.0, 50.2]$; OLMoE ship $[117.8, 121.5]$ vs 6-bit $[149.1, 161.5]$), adjacent trunk widths sometimes are not, and all tok/s cells use one fixed prompt on one machine. Ratio CIs divide by the ship-median point estimate without propagating denominator uncertainty, and 12-run percentile bootstraps on a median are coarse instruments; the headline separations survive both caveats, and we do not lean on adjacent-width distinctions.

Quality. Fixed deterministic corpora; the chat corpus is ultrachat `test_sft[:120]` through each model's own chat template (∼40K tokens scored); wikitext-2 test is the second corpus; gpt-oss is out-of-distribution on raw wikitext (∼140 for every variant) and that corpus is used only for within-model deltas. OLMoE is `allenai/OLMoE-1B-7B-0924-Instruct` via a locally patched config (the cached snapshot predates the `rms_norm_eps` key; value 10^{-5}). One empty-class cell reproduced a 4-decimal-identical perplexity across independent reconversions, and the repair cell reproduced

13.4306 across two independent converts. Perplexity cells are deterministic point measurements on fixed corpora (∼40K scored tokens gives roughly ±2% resolution); the ship-versus-repair comparison additionally carries the paired per-chunk CI, and the fragility deltas of interest (+53%, +130%, PPL 1235) exceed this resolution by an order of magnitude or more. Two paired cells deviate from the 40K default and say so here: the Qwen3-30B shipped-artifact healing pair ran at 12K tokens (a 40K attempt exhausted machine memory; absolute PPL shifts with corpus length, pairing does not), and the frontier QDQ cells ran at 20K, where observed favorable-direction movements of up to 2.8% (Qwen3.6-35B embed cells) indicate an effective resolution nearer ±3% for that instrument. A sixth checkpoint (DeepSeek-V2-Lite) was downloaded and dropped for disk space before any measurement; no measured result was omitted from this report except where the text says so.

8 Limitations

Two model families measured end-to-end for speed on Apple/MLX plus one CUDA replication (L4, llama.cpp, gpt-oss-20b only); the 122B, 30B, and 35B quality cells are quantize–dequantize in floating point (no speed measurement at those scales). The 4-bit trunk on gpt-oss remains broken (+49%) under the training-free levers used here; the residual damage sits in attention weights, and rotation- or scaling-based repairs are untested. Speedups under-realize byte ratios as bytes shrink (kernel floor), and tok/s numbers are short-context by construction. Perplexity-below-baseline readings for repaired cells are reported as observed and claimed only as "no measurable loss." The audit is MLX-scoped.

9 Artifacts

`census_any.py` and `census_dtype.py` (byte census of any Hugging Face checkpoint from headers; no download), the K3 tensor census and classification audit, the five-model control cells, the full OLMoE and gpt-oss matrices with hygiene distributions and bootstrap CIs, the community-build measurements, the ARC and temperature tests, the 103-build audit, the healer and QDQ family cells (Qwen3-30B, Qwen3.6-35B, Qwen3.5-122B, gpt-oss-120b), the expert-tensor hash check, and every run log. Artifacts release with the paper at `github.com/brainsless/the-trunk-tax`.