BYTE-CONSTRAINED LAYER-WISE QUANTIZATION WITH COST-NORMALIZED SENSITIVITY

Anonymous authors
Paper under double-blind review

ABSTRACT

Mixed-precision post-training quantization allocates storage across tensors of different sizes and sensitivities, but nominal bit width omits format metadata. We instantiate a byte-constrained multi-choice knapsack formulation with measured sensitivity and runtime-supported formats, then allocate precision using quality gain per byte. Across five models from three families, the resulting GGUF artifacts lower perplexity relative to six production `llama.cpp` presets at 25 of 30 matched-size points. Paired block-bootstrap intervals resolve 23 improvements and one regression. Controlled ablations separate storage accounting from ranking quality. Uncorrected nominal budgets overshoot file-size targets by 10.6% to 17.9%, but nominal and byte pricing produce similar perplexity after size matching. In contrast, raw-sensitivity ranking is worse at 29 of 30 matched-size points, by up to 6.36 perplexity. Across six downstream tasks, Benjamini-Hochberg correction at 5% identifies 21 gains and three regressions among 180 comparisons, with 19 gains on models of at most 3B parameters. A same-model untying control attributes most shared-embedding sensitivity to the language-model head. These results distinguish accurate artifact sizing from effective precision allocation, while exposing the limits of a separable sensitivity objective.

1 INTRODUCTION

Deploying a language model under a storage limit requires choosing which tensors retain precision and which tolerate stronger quantization. Qwen2.5-7B occupies 14.2 GiB in 16-bit precision, compared with 2.64-4.15 GiB for the production presets evaluated here. These weight files represent only part of deployment memory: runtime buffers, application state and the operating system also consume RAM. Post-training quantization (PTQ) reduces the weight footprint, making storage allocation a practical constraint for local inference.

Layer-wise mixed-precision PTQ assigns a quantization format to each tensor. Prior methods use Hessian-derived sensitivity (Dong et al., 2019; 2020; Shen et al., 2020; Guan et al., 2024), geometric proxies or activation information (Xiao et al., 2025) to guide allocation. Allocation strategies include joint constrained optimizers, which account for tensor size in the budget (Dong et al., 2020; Li et al., 2023; Huang et al., 2025; Lee et al., 2025), and scalar rules, which rank tensors for precision protection. Nominal average bit width summarizes precision but omits format metadata. Two questions then remain distinct in an executable implementation: how accurately the constraint predicts storage, and how effectively the allocation uses that storage.

Sensitivity must be evaluated relative to storage cost. Sensitivity measures quality degradation, whereas the budget constrains storage. Ranking tensors without accounting for their size can therefore spend disproportionate amounts of memory protecting large tensors; this failure mode is specific to the size-blind scalar rules, since joint optimizers already account for n_i in the constraint. The disparity is large in grouped-query attention (GQA): on Qwen2.5-7B, `attn_k` and `attn_v` each hold 0.7% of the parameters while `ffn_gate` holds 25%. The relevant quantity is quality gain per byte, and the two rankings are only loosely related: their rank correlation ranges from -0.04 to +0.44 across the five models. Protecting attention can be valuable because it is inexpensive, without implying intrinsic fragility.

Format metadata changes storage costs. The evaluated block-quantized formats store scales and other metadata, so their realized widths exceed nominal widths by format-dependent factors. Nominal

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

width therefore does not directly specify storage cost. The two constraints need not induce the same feasible assignments (Proposition 1).

The precision regime also affects the observed benefit of allocation. On one model, the measured-versus-random allocation gap widens by 11.0× on the log-perplexity scale when moving to a lower precision band. This pattern persists with the importance matrix disabled. Most evaluated preset budgets lie below the high-band floor, motivating our focus on lower-precision allocations (Section 4). This experiment characterizes the tested allocations without bounding other methods' gains.

Contributions. We instantiate a byte-constrained multi-choice knapsack problem over executable formats and provide a cost-normalized allocator (Sections 2 and 3). We establish sufficient conditions for equivalence between nominal and realized storage budgets, with a counterexample when they fail. Matched-size controls then distinguish the sizing effect of byte accounting from the quality effect of cost normalization (Section 7). Evaluation spans five models, production presets, off-domain corpora and downstream tasks. Finally, solver, surrogate and weight-tying audits identify where the formulation works and where its assumptions fail (Sections 3 and 8).

2 Problem formulation

We use the block-wise quantization formats implemented in `llama.cpp`. For example, `q4_0` stores 32 weights using 4-bit codes and one FP16 scale, requiring 4.5 bits per weight. The K-quant formats store additional sub-block metadata, giving 2.5625, 3.4375 and 5.5 bits per weight for `q2_K`, `q3_K` and `q5_K`, respectively (Section A). An activation-derived *importance matrix* (imatrix) calibrates quantization error. On Qwen2.5-1.5B, uniform `q2_K` reaches a perplexity of 395 without an imatrix and 31.9 with one, illustrating its importance at low precision. The production presets are themselves hand-tuned mixed-precision assignments, providing a practical baseline for our executable allocator.

Let $\mathcal{T}$ index the quantizable tensors and $\mathcal{L}$ the available levels. Writing n_i for the parameter count of tensor i, $\mathrm{bpw}(L)$ for the *realized* width of level L, and $s(i, L)$ for the quality cost of assigning L to i, the allocation problem is

$$\min_{L_1,\ldots,L_{|\mathcal{T}|}} \sum_{i\in\mathcal{T}} s(i, L_i) \qquad s.t. \qquad \sum_{i\in\mathcal{T}} (n_i \cdot \mathrm{bpw}(L_i))/8 \;\le\; B, \tag{1}$$

a multi-choice knapsack problem (MCKP) whose budget B is in *bytes*. Here B is the budget for the quantizable tensor payload. The full GGUF also contains metadata, alignment padding, and tensors outside $\mathcal{T}$; the build–measure–retarget loop accounts for these costs before accepting an artifact. File size constrains persistent storage but does not fully predict resident memory. Our measured resident-memory-to-file-size ratios are 1.07 to 1.62 (Section 6); runtime buffers and the evaluation workload also consume memory.

Nominal and realized storage. A mean nominal-width constraint, $\bar{b} = \sum_i n_i b_i / \sum_i n_i$, already weights tensors by parameter count. The distinction is therefore the format cost, rather than whether tensor size appears in the constraint.

Proposition 1 (sufficient conditions for equivalent storage budgets)**.** *Let $C_{\mathrm{nom}} = \sum_i n_i b(L_i)$ and $C_{\mathrm{real}} = \sum_i n_i \mathrm{bpw}(L_i)$, with fixed $N = \sum_i n_i$. If $\mathrm{bpw}(L) = a\, b(L) + c$ for all levels, where $a > 0$, then $C_{\mathrm{real}} = aC_{\mathrm{nom}} + cN$. Thus a nominal budget K and the realized-bit budget $aK + cN$ define identical feasible assignments and optimal objective values. Constant realized-to-nominal width is the special case $c = 0$. Without this affine relation, equivalence is not guaranteed.*

Proof and counterexample. Substitution gives the identity and the equivalence of the two inequalities. For non-equivalence, consider two equal tensors and nominal widths 2, 3 and 4, with realized widths 2.5625, 3.4375 and 4.5. Assignments $(2, 4)$ and $(3, 3)$ have equal nominal cost but realized costs 7.0625 and 6.875 per tensor parameter. If each tensor's sensitivity costs at those levels are $(4, 3, 0)$, a nominal budget of 6 selects $(2, 4)$ or $(4, 2)$, whereas a realized budget of 6.875 selects $(3, 3)$. Hence nominally equal assignments can differ in both feasibility and optimality under a specified realized budget. These costs form a mathematical counterexample, not experimental measurements. □

For our palette, the realized-to-nominal ratios are 1.100 for `q5_K`, 1.125 for `q4_0`, 1.146 for `q3_K`, and 1.281 for `q2_K`. The realized widths are also non-affine in nominal width: successive increments from 2 to 3, 3 to 4, and 4 to 5 bits are 0.875, 1.0625, and 1.0 bits per weight. The sufficient condition therefore does not apply to this palette. Relative to nominal payload, metadata adds 28% at the

lowest level and 10% at the highest. Nominal accounting consequently distorts the storage trade-offs between levels. Section 7 tests this effect with built, size-matched artifacts.

The level set $\mathcal{L}$ contains concrete formats supported by the runtime. This restricts precision choices but ensures that each assignment can be built and executed directly. Every allocator result is evaluated as a GGUF model file loaded by `llama.cpp`.

Equation (1) approximates joint degradation by summing isolated tensor costs. It omits interactions and uses KL divergence as a proxy for log-perplexity degradation. Sections C and 6 assess the combined approximation error and identify the model with a within-budget ranking inversion.

3 Sensitivity estimation and allocation

Figure 1 summarizes the workflow. Four-format probes against FP16 produce the sensitivity table $s(i, L)$; allocation combines it with tensor sizes. Measured GGUF size then guides retargeting until the artifact meets tolerance.

The cost table. We measure sensitivity through one-at-a-time (OAT) perturbations against a common FP16 reference. For each tensor i and level L, we replace only that tensor and measure the output Kullback–Leibler (KL) divergence on calibration text. Writing P_t^{ref} and $P_t^{i,L}$ for the next-token distributions at position t under the FP16 reference and under the model with only tensor i set to level L,

$$s(i, L) \;=\; \frac{1}{T} \sum\nolimits_{t=1}^{T} D_{\mathrm{KL}}\Big(P_t^{\text{ref}} \,\Big\|\, P_t^{i,L} \Big)\,, \tag{2}$$

the forward divergence from the reference, averaged over the T calibration positions, so s is a per-token quantity. Weights are read from GGUF files through `gguf.quants.dequantize`, preserving the deployed quantizer's values and imatrix calibration.

We use paired, token-level KL because perplexity (PPL) changes from individual tensor perturbations can be comparable to measurement noise. A fixed FP16 reference also makes sensitivity differences comparable across levels. Measuring KL against a quantized reference instead would assign a non-negative divergence even to a beneficial promotion. Across the profiling runs, 98% of tensors have strictly monotone costs (1106 of 1129 tensors, including the untying control). Exceptions account for 1.9% of the palette-wide effect; greedy filling accepts only upgrades with positive measured gain.

Class-level per-parameter sensitivities span a 118-fold range (Table 1). Less sensitive feed-forward classes contain most parameters, allowing precision to be reassigned to smaller tensors.

All five models are profiled per tensor; Qwen2.5-7B requires 198×4 probes. Profiling costs 22 to 113 GPU-minutes per model on one V100, or 3.0 to 8.6 seconds per probe, incurred once per model and amortized across budgets (Section I). Section G evaluates a class-level approximation.

The solver. We solve Equation (1) approximately, by placing a shadow price λ on payload storage and moving the constraint into the objective: $L_i(\lambda) = \arg\min_L [\, s(i, L) + \lambda\, n_i\, \mathrm{bpw}(L)/8 \,]$. For fixed λ, tensors choose independently, and total size is non-increasing in λ. Bisection therefore replaces a search over $|\mathcal{L}|^{|\mathcal{T}|}$ assignments with $|\mathcal{T}|$ four-way comparisons per step. The relaxation selects lower-convex-hull levels in the (bytes, s) plane. Levels strictly above that hull cannot be selected by bisection but may enter through greedy filling. With four discrete levels, the relaxation leaves 0.0% to 4.4% of the budget unspent. Greedy filling then selects feasible one-level upgrades with the greatest positive gain per added byte until none fits.

We audit the heuristic using a dynamic-programming lower bound on the optimal objective value. Integer byte costs permit exact discretization when the state space is tractable. Otherwise, rounding costs down relaxes the constraint and yields a valid lower bound. At payload budgets set to the 30 preset file sizes, the upper bounds span 0.3% to 3.7% of the objective across the 30 instances (Section B). This solver diagnostic precedes file-size retargeting; it does not certify the final artifacts or their measured perplexity.

Surrogate calibration and ordering. Regressing measured Δ logPPL on predicted cost gives model-dependent slopes: 1.09 on Llama-3.2-1B, 0.99 on Mistral-7B-v0.3, 0.87 on Llama-3.2-3B, 0.73 on Qwen2.5-1.5B, 0.45 on Qwen2.5-7B. This discrepancy can reflect both tensor interactions and the difference between KL divergence and log-perplexity. Within-budget ranking is more stable.

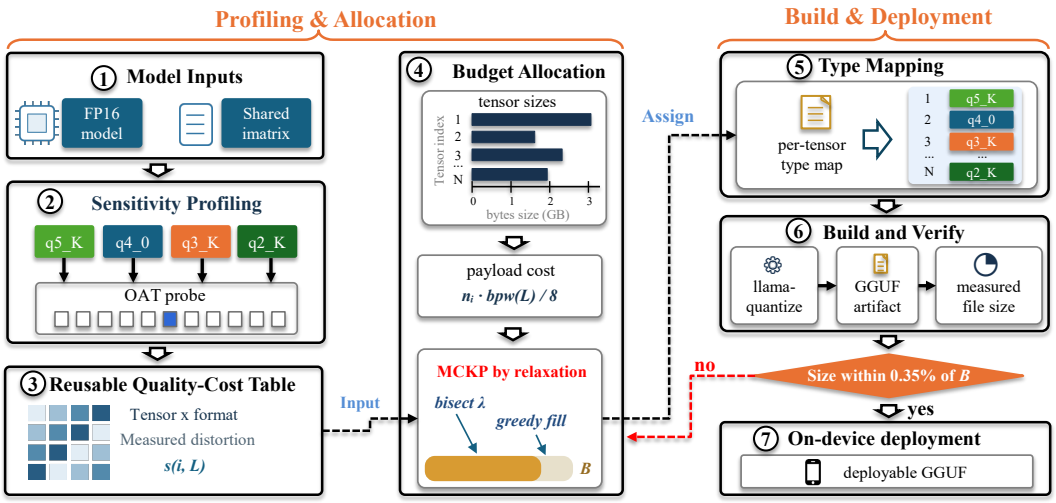


Figure 1: Workflow. Sensitivity is profiled once per model; allocation and GGUF size verification repeat per budget. The red feedback adjusts the payload budget using measured file size. The size check uses the full-file target; Equation 1 constrains only the quantizable payload.

Table 1: Per-parameter sensitivity on Qwen2.5-7B relative to the unquantized model ($\times 10^{-9}$).

Class	Role	Params	%	q3_K	q2_K
attn_v	attention value proj.	51 M	0.7	0.271	0.700
output	LM head	545 M	7.2	0.113	0.448
attn_k	attention key proj.	51 M	0.7	0.080	0.196
attn_output	attention output proj.	360 M	4.7	0.038	0.117
ffn_down	feed-forward down	1901 M	25.0	0.053	0.058
ffn_up	feed-forward up	1901 M	25.0	0.019	0.056
ffn_gate	feed-forward gate	1901 M	25.0	0.015	0.046
attn_q	attention query proj.	360 M	4.7	0.011	0.029
token_embd	input embedding lookup	545 M	7.2	0.001	0.006

Excluding predicted-cost differences below 5% as ties, the two rankings disagree in 1 of 60 within-budget comparisons; the inversion occurs on Qwen2.5-7B. The threshold sweep gives 15 of 90 at 0%; 1 of 60 at 2%; 1 of 60 at 5%; 1 of 60 at 10%. Thus the inversion count is stable from 2% upward, although near-ties introduce additional disagreements. A small surrogate optimization gap does not guarantee minimum measured perplexity.

4 Precision Regime and Observed Allocation Gains

We compare three allocations at a budget corresponding to 49% of the achievable storage saving within each precision band. SMART ranks by measured KL density, SENS ranks by total KL, and RANDOM uses three seeds. The high band demotes q8_0 to q4_0; the low band demotes q4_0 to q2_K. The gap between SMART and RANDOM characterizes the observed allocation range on Qwen2.5-1.5B, not an upper bound for other allocators.

On the additive log-PPL scale, this gap increases from 0.054 to 0.594, an 11.0× widening. The corresponding uniform-endpoint gap increases from 0.095 to 1.097. Disabling the importance matrix changes the low-band uniform gap to 3.582, but leaves the allocation gap near 0.586. The widening remains 10.3× without the imatrix. Calibration nevertheless affects the ranking comparison: the density-versus-raw gap increases from 0.082 to 0.474 when the imatrix is removed. All primary matched-size comparisons therefore retain calibration on both sides.

The evaluated storage targets also motivate the lower band. On Qwen2.5-1.5B, uniform q8_0→q4_0 returns 736 of 1570 MiB and costs 0.96 PPL, a 10% relative increase. Scaling the high-band floor by parameter count places only seven of the 30 preset budgets in that band. All seven use the two

loosest presets, where measured gains are smallest. This calculation contextualizes the tested budgets without establishing a universal precision threshold (Section D).

5 EXPERIMENTAL PROTOCOL

We evaluate Qwen2.5-1.5B and 7B, Llama-3.2-1B and 3B, and Mistral-7B-v0.3 (Qwen Team, 2024; Grattafiori et al., 2024; Jiang et al., 2023). These models include tied and untied embeddings, 32K–152K vocabularies, and grouped-query attention ratios of 4:1–7:1. Disjoint character spans of the WikiText-2 test split (Merity et al., 2017) supply calibration and evaluation: `calib` $[0, 120\text{K})$ builds the importance matrix and sensitivity map; `eval` $[600\text{K}, 760\text{K})$ supplies the held-out perplexity score. Table 13 reports an additional held-out `val` span. Neither held-out span selects an assignment. One fixed allocator produces a final artifact for each budget after size-only retargeting.

Each artifact targets the preset's measured file size within a two-sided 0.35% tolerance. The allocator and presets share calibration text, the importance matrix, and evaluation settings. The four-bit slot differs: ours uses `q4_0`, while presets use `q4_K`; Section 7 tests that asymmetry. We also evaluate The Pile and C4 (Gao et al., 2020; Raffel et al., 2020) and six downstream tasks (Zellers et al., 2019; Sakaguchi et al., 2020; Clark et al., 2018; Mihaylov et al., 2018; Hendrycks et al., 2021) on the deployed GGUFs, with identical item order for paired tests. Exact slices, task sizes and commands appear in Section J.

Reprofiling on an independent span gives density-rank correlations of 0.969 to 0.998 and pre-retargeting assignment agreement on 85–100% of tensors; 26 of 30 matched-size points remain improvements (Table 6). This control tests sensitivity to the profiling span while keeping the importance matrix fixed.

6 RESULTS

Perplexity at matched size. Our allocator improves held-out perplexity at 25 of 30 points across five models and six production presets, by up to 7.50 PPL (Table 2). The presets are `Q2_K_S`, `Q2_K`, `Q3_K_S`, `Q3_K_M`, `Q3_K_L` and `Q4_K_S`, abbreviated 2S, 2K, 3S, 3M, 3L and 4S. To distinguish small point differences from resolved effects, we report paired 95% moving-block bootstrap intervals over 70–78 consecutive evaluation windows, with block length $L = 4$. The mean absolute lag-one autocorrelation is 0.12, so permutation tests also flip signs by block. The block-length sweep and full adjusted tests are in Section T.2 and Table 12.

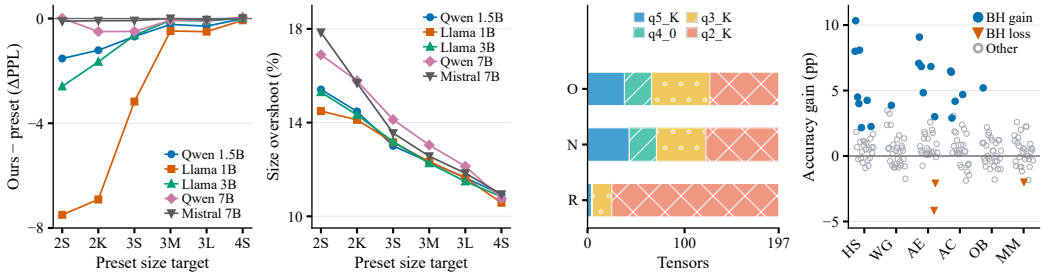
The intervals resolve 23 improvements and one regression. The regression occurs on Qwen2.5-7B at `Q4_K_S`, by +0.047 perplexity (95% CI $[+0.016, +0.079]$); the remaining 6 are not resolved either way, with margins of at most 0.067 perplexity. Of the unresolved cells, 5 of the 6 occur at the three loosest budgets. Across all 30 tests, 23 of the 30 survive a Benjamini–Hochberg control at the five per cent level, 22 of them favouring the allocation; 21 survive Benjamini–Yekutieli, whose guarantee holds under arbitrary dependence. These are the Benjamini–Hochberg (BH) and Benjamini–Yekutieli (BY) procedures. Adjusted tests retain most gains and complement the interval estimates.

All files are within 0.27% of their targets. Restricting attention to files no larger than their presets leaves 9 of the 30 matched-size points, and the allocation wins all 9 of them. The local size–quality sensitivity calculation in Section E assesses the possible effect of positive mismatch; it does not adjust any score. The largest improvement occurs where the preset degrades most (Llama-3.2-1B at `Q2_K_S`, where the preset reaches 18.78 and ours 11.28), and gains narrow near four bits. At equal perplexity, interpolating the measured curves gives files smaller than the preset that reaches it at 17 of 19 such points, by up to 14%. These are bracketed log–log estimates, not additional builds.

Off-domain evaluation. We rescore the same artifacts on The Pile and C4, neither used for profiling or allocation. Our allocation lowers PPL in 53 of 60 comparisons, with improvements up to 13.07 PPL and regressions no greater than 0.14 PPL (Table 3). Several WikiText-2 losses reverse on both corpora: Qwen2.5-7B at `Q2_K_S`, Mistral-7B-v0.3 at `Q3_K_M` and Mistral-7B-v0.3 at `Q4_K_S`. Relative allocation quality therefore depends on the corpus; the full controls retain both favourable and unfavourable cells.

Table 2: Held-out perplexity at matched GGUF size with identical importance-matrix calibration. Size is our file size; Mis. is its relative mismatch against the preset. Δ is ours minus preset, with a paired 95% moving-block bootstrap confidence interval (CI). Blue favours ours; orange favours the preset. Colour indicates direction, not significance. Grey intervals include zero.

Model	Preset	Size (MiB)	Mis.	`llama.cpp`	Ours	Δ (95% CI)
Qwen2.5-1.5B	Q2_K_S	611.0	+0.08%	13.9119	12.3852	-1.527 [-1.785, -1.234]
	Q2_K	645.4	+0.07%	12.7869	11.5760	-1.211 [-1.441, -1.002]
	Q3_K_S	725.8	+0.02%	11.3565	10.6702	-0.686 [-0.787, -0.603]
	Q3_K_M	786.0	-0.00%	10.5153	10.2886	-0.227 [-0.308, -0.130]
	Q3_K_L	839.3	-0.01%	10.4261	10.1356	-0.290 [-0.354, -0.212]
	Q4_K_S	896.6	-0.02%	10.0173	10.0093	-0.008 [-0.071, +0.069]
Qwen2.5-7B	Q2_K_S	2707.8	+0.19%	8.5039	8.5134	+0.010 [-0.082, +0.110]
	Q2_K	2884.1	+0.27%	8.3998	7.9018	-0.498 [-0.582, -0.404]
	Q3_K_S	3336.4	+0.17%	7.9853	7.4905	-0.495 [-0.540, -0.439]
	Q3_K_M	3637.8	+0.16%	7.4453	7.3804	-0.065 [-0.104, -0.018]
	Q3_K_L	3905.1	+0.16%	7.4245	7.3594	-0.065 [-0.118, -0.008]
	Q4_K_S	4256.9	+0.13%	7.2737	7.3206	+0.047 [+0.016, +0.079]
Llama-3.2-1B	Q2_K_S	529.3	+0.07%	18.7799	11.2837	-7.496 [-7.859, -6.988]
	Q2_K	553.9	-0.01%	17.8561	10.9452	-6.911 [-7.396, -6.326]
	Q3_K_S	612.0	+0.00%	13.4558	10.2840	-3.172 [-3.485, -2.869]
	Q3_K_M	658.5	-0.05%	10.3549	9.8795	-0.475 [-0.544, -0.386]
	Q3_K_L	698.5	-0.02%	10.1457	9.6429	-0.503 [-0.574, -0.418]
	Q4_K_S	739.8	+0.02%	9.5401	9.4727	-0.067 [-0.131, +0.015]
Llama-3.2-3B	Q2_K_S	1215.9	+0.05%	11.4935	8.9002	-2.593 [-2.823, -2.380]
	Q2_K	1300.9	+0.01%	10.0247	8.3714	-1.653 [-1.895, -1.415]
	Q3_K_S	1471.3	-0.00%	8.4516	7.8097	-0.642 [-0.732, -0.549]
	Q3_K_M	1609.0	-0.00%	7.7006	7.6324	-0.068 [-0.104, -0.028]
	Q3_K_L	1731.2	-0.00%	7.6090	7.5135	-0.096 [-0.134, -0.052]
	Q4_K_S	1839.2	+0.02%	7.4206	7.4342	+0.014 [-0.025, +0.061]
Mistral-7B-v0.3	Q2_K_S	2416.5	+0.05%	6.4791	6.3761	-0.103 [-0.134, -0.069]
	Q2_K	2600.0	+0.12%	6.1799	6.0971	-0.083 [-0.116, -0.054]
	Q3_K_S	3022.6	+0.03%	5.8812	5.8013	-0.080 [-0.101, -0.055]
	Q3_K_M	3362.6	+0.09%	5.7367	5.7402	+0.003 [-0.014, +0.022]
	Q3_K_L	3650.4	+0.04%	5.7241	5.6760	-0.048 [-0.067, -0.027]
	Q4_K_S	3954.5	+0.04%	5.6399	5.6494	+0.010 [-0.004, +0.027]



(a) PPL at equal size (b) Nominal size error (c) Format allocation (d) Downstream tasks

Figure 2: (a) Matched-size PPL differences, ours minus preset (paired intervals in Table 2). (b) Uncorrected nominal-budget size overshoot. (c) Format counts for Qwen2.5-1.5B at 2S; O/N/R denote ours, nominal-matched and raw. (d) Task accuracy differences: filled markers indicate Benjamini–Hochberg (BH) $q < 0.05$ across all 180 cells. HS/WG/AE/AC/OB/MM denote HellaSwag, WinoGrande, ARC-Easy, ARC-Challenge, OpenBookQA, and MMLU.

Downstream accuracy. Of 180 model–budget–task cells, 121 favour ours. Exact paired McNemar tests and paired item-bootstrap intervals quantify uncertainty (Tables 10 and 11). After correction, 24 of the 180 cells survive a Benjamini–Hochberg false-discovery-rate control at the five per cent level, 21 of them favouring the allocation and 3 against; 17 survive Benjamini–Yekutieli, which assumes no dependence structure at all. Most discoveries occur at tight budgets: 21 of those 24 sit at the three tightest budgets, consistent with the precision-regime pattern. BH discoveries concentrate at

Table 3: Held-out perplexity on two corpora outside the WikiText family, at the same matched file sizes. Neither corpus is used for calibration, profiling or allocation. Blue and orange indicate negative and positive differences (ours minus preset), respectively, not statistical significance.

		The Pile			C4		
Model	Preset	`llama.cpp`	Ours	Δ	`llama.cpp`	Ours	Δ
Qwen2.5-1.5B	Q2_K_S	16.214	14.327	-1.888	19.643	17.634	-2.008
	Q2_K	14.828	13.664	-1.164	18.020	16.729	-1.291
	Q3_K_S	13.140	12.412	-0.727	16.110	15.123	-0.987
	Q3_K_M	12.126	12.037	-0.090	14.841	14.565	-0.276
	Q3_K_L	12.002	11.840	-0.162	14.679	14.351	-0.329
	Q4_K_S	11.651	11.685	+0.034	14.070	14.207	+0.137
Qwen2.5-7B	Q2_K_S	10.787	10.388	-0.399	12.811	12.272	-0.539
	Q2_K	10.551	10.001	-0.550	12.443	11.821	-0.622
	Q3_K_S	10.129	9.538	-0.591	11.894	11.215	-0.679
	Q3_K_M	9.449	9.425	-0.024	11.225	11.137	-0.088
	Q3_K_L	9.394	9.363	-0.031	11.178	11.090	-0.089
	Q4_K_S	9.253	9.284	+0.030	11.030	11.013	-0.018
Llama-3.2-1B	Q2_K_S	24.276	15.007	-9.269	29.151	17.116	-12.036
	Q2_K	23.284	14.502	-8.782	29.602	16.531	-13.070
	Q3_K_S	17.542	13.747	-3.795	20.537	15.284	-5.253
	Q3_K_M	13.815	13.258	-0.557	15.538	14.654	-0.884
	Q3_K_L	13.582	13.018	-0.564	15.175	14.189	-0.986
	Q4_K_S	12.877	12.844	-0.033	14.026	13.836	-0.190
Llama-3.2-3B	Q2_K_S	15.707	12.876	-2.830	18.212	14.474	-3.737
	Q2_K	13.748	12.245	-1.503	15.862	13.470	-2.393
	Q3_K_S	11.914	11.234	-0.681	13.324	12.013	-1.311
	Q3_K_M	10.948	10.984	+0.036	11.751	11.621	-0.130
	Q3_K_L	10.873	10.808	-0.066	11.612	11.252	-0.359
	Q4_K_S	10.602	10.653	+0.051	11.075	11.139	+0.064
Mistral-7B-v0.3	Q2_K_S	7.562	7.491	-0.071	8.717	8.584	-0.132
	Q2_K	7.256	7.284	+0.028	8.363	8.274	-0.089
	Q3_K_S	6.968	6.922	-0.045	7.957	7.896	-0.062
	Q3_K_M	6.836	6.835	-0.001	7.788	7.785	-0.004
	Q3_K_L	6.817	6.779	-0.038	7.766	7.712	-0.054
	Q4_K_S	6.767	6.763	-0.004	7.688	7.674	-0.014

smaller scales: 19 of 21 gains occur on models with at most 3B parameters, with no BH-significant regressions in that subset. The two 7B models yield 2 BH-significant gains and 3 regressions across 72 cells; none of Mistral's 36 cells shows a BH-significant gain. Perplexity gains do not guarantee task gains: on Qwen2.5-7B at 2K, PPL improves by 0.498 while ARC-Easy worsens significantly. Predicting task accuracy from sensitivity remains an open problem.

Runtime cost. On the two benchmarked Qwen models, matched-size files differ in peak resident memory by at most 11 MiB. Throughput ratios span 0.84 to 1.40 and energy-per-token ratios 0.78 to 1.08, without a resolved systematic advantage. These shared-server measurements have substantial repeat variability; absolute rates and exclusion criteria appear in Tables 15 and 16. Cross-budget comparisons at attained preset quality save 35 to 447 MiB of peak resident memory. They do not establish mobile latency or energy gains.

Academic baselines. GPTQ and AWQ use published code and calibration protocols (Frantar et al., 2023; Lin et al., 2024); QuaRot's unresolved integration is documented but excluded (Ashkboos et al., 2024). Because Hugging Face and `llama.cpp` disagree even at FP16, comparison uses degradation relative to each harness's own reference. Our allocation has lower normalized degradation in 10 of 14 comparisons against GPTQ, AWQ and round-to-nearest (RTN). The four losses occur at 4.16 effective bits per weight. Effective width covers only quantized tensors; the baselines retain embeddings, norms and the LM head in FP16. Their estimated complete checkpoint footprints are 1.43 to 1.92 times our measured GGUF sizes at the same effective width. The Qwen2.5-1.5B untying control attributes 96.1% of decomposed sensitivity to the head (Section 8), underscoring the importance of this precision asymmetry. These comparisons match linear-layer width, not deployment storage (Section O and Table 18).

7 SEPARATING STORAGE ACCOUNTING FROM SENSITIVITY RANKING

Our formulation changes both the storage constraint and the ranking rule. Three controls examine these choices, sharing the sensitivity map, importance matrix and format palette (Table 4). NOMINAL uses nominal costs and is built without size correction; NOMINAL-MATCHED retargets that cost model to the preset's measured size; RAW greedily ranks demotions by their unnormalized sensitivity increase at the same measured size.

Accounting controls size. Uncorrected nominal files overshoot by 10.6% to 17.9%. After retargeting, the absolute PPL differences between nominal and realized pricing are at most 0.054 on four of the five models and 0.315 on Qwen2.5-7B, despite different tensor assignments. These cost tables affect quality less than their uncorrected sizes suggest. Explicit byte accounting chiefly ensures accurate artifact sizing. Qwen2.5-7B is the sole model on which NOMINAL-MATCHED improves all six budgets, by 0.315 PPL at 2S and at most 0.036 elsewhere. It also has the least calibrated surrogate (slope 0.45) and the only within-budget ranking inversion. This pattern is consistent with surrogate error: a more accurate storage model need not yield lower measured perplexity (Sections C and T.3).

Cost-normalized allocation improves on raw ranking. Raw sensitivity is worse at 29 of 30 matched-size points, by up to 6.36 PPL; raw ranking improves the remaining point by 0.10 PPL. Its mean penalty is 79% of our mean margin over the presets, a ratio of means rather than a per-cell statistic. At the tightest 1.5B budget, raw ranking leaves 172 of 197 tensors at `q2_K`, whereas cost normalization spreads protection across more tensors (Figure 2). This comparison evaluates allocation procedures at matched size; it changes both ranking and search, and does not reimplement a published allocator. Full assignments and the exponent sweep are in Sections P and T.3.

The four-bit palette asymmetry is conservative on the model we tested. Substituting `q4_K` for `q4_0` on Qwen2.5-1.5B is better at 6 of 6 budgets, by 0.01 to 0.26 perplexity; our allocation still wins at 6 of 6 budgets. This direction remains untested on the other four models (Section D). The presets use `q4_K`, whereas our allocator uses `q4_0` at the same realized width. Format choice therefore confounds the narrowing margins near four bits and the resolved regression on Qwen2.5-7B at `Q4_K_S`. Neither effect can be attributed solely to the allocation policy.

8 INTERPRETING SENSITIVITY UNDER WEIGHT TYING

A tied embedding combines input lookup and language-model-head (LM-head) roles. On Qwen2.5-1.5B, demoting the shared `token_embd` alone to `q2_K` costs 0.578 KL, 11 times the entire `attn_v` class. Yet per parameter it ranks 25th among 197 tensors. Large total sensitivity therefore does not establish exceptional density or identify which computational role causes the effect.

We separate the roles within the same Qwen2.5-1.5B model by copying the shared weights into an independently stored LM head. Before profiling, logits remain bit-identical. At `q2_K`, input and output sensitivity densities are 0.098 and 2.394 ($\times 10^{-9}$ per parameter), respectively; 96.1% of the combined cost belongs to the head. The decomposed sum closely matches the original tied density: 2.492 against 2.475 (Figure 5c). The head-to-input ratio is 24 to 31 across the four levels.

This control attributes the measured sensitivity to computational roles within one model. The head affects all vocabulary logits while lookup retrieves one row per token, but that explanation requires further testing. For allocation, protecting the shared head also raises input-embedding precision; an untied model can protect them separately. Cross-model comparisons and the full interpretation are retained in Section T.4.

9 RELATED WORK

PTQ methods improve rounding, reconstruction or outlier handling: AdaRound and BRECQ (Nagel et al., 2020; Li et al., 2021), second-order methods (Frantar et al., 2022; 2023), and outlier decomposition, scaling or rotation (Dettmers et al., 2022; Xiao et al., 2023; Lin et al., 2024; Ashkboos et al., 2024). Low-bit methods also use sparse corrections (Dettmers et al., 2023; Kim et al., 2024), incoherence processing (Chee et al., 2023), or quantization-aware training (Liu et al., 2023; Wang et al., 2023). These methods may complement allocation by improving fixed-width quantization.

Table 4: Ranges across six budgets per model. Overshoot is the uncorrected nominal file-size error. Both Δ columns report control PPL minus ours at matched size; positive values favour ours.

Model	NOMINAL overshoot	NOMINAL-MATCHED Δ	RAW Δ
Qwen2.5-1.5B	10.9–15.4%	-0.033 to +0.012	+0.41 to +1.44
Qwen2.5-7B	10.8–16.9%	-0.315 to -0.002	-0.10 to +0.24
Llama-3.2-1B	10.6–14.5%	-0.006 to +0.053	+0.20 to +6.36
Llama-3.2-3B	10.8–15.3%	-0.001 to +0.017	+0.05 to +1.47
Mistral-7B-v0.3	10.9–17.9%	-0.004 to +0.001	+0.04 to +0.18

Mixed-precision methods use Hessian, geometric or activation-derived sensitivity (Dong et al., 2019; 2020; Shen et al., 2020; Guan et al., 2024; Xiao et al., 2025). HAWQ-V2 already combines additive sensitivity with a Pareto frontier over model size; later automated allocators search parameter-weighted budgets (Li et al., 2023; Huang et al., 2025; Lee et al., 2025). Our contribution is the executable instantiation and controlled evaluation, rather than constrained allocation itself. Comparing these nominal-width, non-GGUF methods at matched file size requires runtime integration. Our internal RAW control does not represent them or establish an advantage over these methods.

Hardware-aware allocation (Wang et al., 2019) and integer inference (Jacob et al., 2018) likewise motivate runtime-specific costs. Our GGUF assignments execute directly in `llama.cpp`, supporting comparisons with production presets under shared calibration and matched file size.

10 CONCLUSION

We combine measured tensor sensitivity with executable quantization formats under a byte constraint. Matched-size controls distinguish storage accounting from allocation quality: byte accounting improves size accuracy, while the cost-normalized allocator generally outperforms raw ranking. Across five models from three families, the allocator lowers perplexity at 25 of 30 matched-size points, with 23 improvements resolved by paired block-bootstrap intervals. The Qwen2.5-1.5B untying control attributes 96.1% of decomposed sensitivity to the head.

Limitations. The separable objective can misorder allocations; more exact optimization cannot correct surrogate error. Downstream evaluation uses teacher-forced multiple-choice likelihoods. Multilingual text, long contexts, generative reasoning (Cobbe et al., 2021), code synthesis (Chen et al., 2021) and further commonsense suites (Bisk et al., 2020) remain untested. Profiling cost grows with tensor count, and KL sensitivity does not consistently predict task accuracy. Shared-server benchmarks do not establish mobile latency or energy gains. Academic baselines use dequantized weights and different evaluation harnesses (Sections O, Q and S). Future work should model interactions, broaden evaluation, and compare allocators at matched checkpoint sizes within one runtime.

AI USE STATEMENT

Generative AI tools assisted manuscript editing and organization, figure presentation, bibliographic formatting and verification, experimental-code organization and bug checking, and consistency checks between claims, code and recorded results. The authors reviewed the AI-assisted material against experimental artifacts and primary sources; no experimental measurements were generated by AI. The authors take responsibility for the final content.

REPRODUCIBILITY STATEMENT

Sections 2, 3 and 5 specify allocation and evaluation protocols. The appendix provides commands, implementation checks, full results, statistics and limitations. Measurement-derived tables and macros link reported values to the supplied data.

REFERENCES

Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L. Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. QuaRot: Outlier-free 4-bit inference in rotated LLMs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about physical commonsense in natural language. In *AAAI Conference on Artificial Intelligence*, 2020.

Jerry Chee, Yaohui Cai, Volodymyr Kuleshov, and Christopher De Sa. QuIP: 2-bit quantization of large language models with guarantees. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? Try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. LLM.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Tim Dettmers, Ruslan Svirschevski, Vage Egiazarian, Denis Kuznedelev, Elias Frantar, Saleh Ashkboos, Alexander Borzunov, Torsten Hoefler, et al. SpQR: A sparse-quantized representation for near-lossless LLM weight compression. *arXiv preprint arXiv:2306.03078*, 2023.

Zhen Dong, Zhewei Yao, Amir Gholami, Michael Mahoney, and Kurt Keutzer. HAWQ: Hessian aware quantization of neural networks with mixed-precision. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

Zhen Dong, Zhewei Yao, Daiyaan Arfeen, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer. HAWQ-V2: Hessian aware trace-weighted quantization of neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Elias Frantar, Sidak Pal Singh, and Dan Alistarh. Optimal brain compression: A framework for accurate post-training quantization and pruning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. GPTQ: Accurate post-training quantization for generative pre-trained transformers. In *International Conference on Learning Representations*, 2023.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The Pile: An 800GB dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Ziyi Guan, Hantao Huang, Yupeng Su, Hong Huang, Ngai Wong, and Hao Yu. APTQ: Attention-aware post-training mixed-precision quantization for large language models. *arXiv preprint arXiv:2402.14866*, 2024.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021.

Wei Huang, Haotong Qin, Yangdong Liu, Yawei Li, Qinshuo Liu, Xianglong Liu, Luca Benini, Michele Magno, Shiming Zhang, and Xiaojuan Qi. SliM-LLM: Salience-driven mixed-precision quantization for large language models. In *International Conference on Machine Learning (ICML)*, 2025.

Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023.

Sehoon Kim, Coleman Hooper, Amir Gholami, Zhen Dong, Xiuyu Li, Sheng Shen, Michael W. Mahoney, and Kurt Keutzer. SqueezeLLM: Dense-and-sparse quantization. In *International Conference on Machine Learning (ICML)*, 2024.

Sangjun Lee, Seung-taek Woo, Jun-gyu Jin, Changhun Lee, and Eunhyeok Park. AMQ: Enabling AutoML for mixed-precision weight-only quantization of large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025.

Shiyao Li, Xuefei Ning, Ke Hong, Tengxuan Liu, Luning Wang, Xiuhong Li, Kai Zhong, Guohao Dai, Huazhong Yang, and Yu Wang. LLM-MQ: Mixed-precision quantization for efficient LLM deployment. In *NeurIPS Workshop on Efficient Natural Language and Speech Processing*, 2023.

Yuhang Li, Ruihao Gong, Xu Tan, Yang Yang, Peng Hu, Qi Zhang, Fengwei Yu, Wei Wang, et al. BRECQ: Pushing the limit of post-training quantization by block reconstruction. In *International Conference on Learning Representations*, 2021.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. AWQ: Activation-aware weight quantization for on-device LLM compression and acceleration. In *Proceedings of Machine Learning and Systems (MLSys)*, 2024.

Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, et al. LLM-QAT: Data-free quantization aware training for large language models. *arXiv preprint arXiv:2305.17888*, 2023.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? A new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.

Markus Nagel, Rana Ali Amjad, Mart van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? adaptive rounding for post-training quantization. In *International Conference on Machine Learning (ICML)*, 2020.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An adversarial winograd schema challenge at scale. In *AAAI Conference on Artificial Intelligence*, 2020.

Sheng Shen, Zhen Dong, Jiayu Ye, Linjian Ma, Zhewei Yao, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer. Q-BERT: Hessian based ultra low precision quantization of BERT. In *AAAI Conference on Artificial Intelligence*, 2020.

Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Huaijie Wang, Lingxiao Ma, Fan Yang, Ruiping Wang, et al. BitNet: Scaling 1-bit transformers for large language models. *arXiv preprint arXiv:2310.11453*, 2023.

Kuan Wang, Zhijian Liu, Yujun Lin, Ji Lin, and Song Han. HAQ: Hardware-aware automated quantization with mixed precision. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. SmoothQuant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning (ICML)*, 2023.

He Xiao, Qingyao Yang, Dirui Xie, Wendong Xu, Zunhai Su, Runming Yang, Wenyong Zhou, Haobo Liu, Zhengwu Liu, and Ngai Wong. Exploring layer-wise information effectiveness for post-training quantization in small language models. *arXiv preprint arXiv:2508.03332*, 2025.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

A EXECUTABLE FORMATS AND METADATA OVERHEAD

Table colours. Pale blue bands distinguish numeric columns. For signed quality differences, blue favours our allocation and orange favours the preset; shading does not denote statistical significance. Boldface and reported p- and q-values retain their stated meanings.

Block quantization in `llama.cpp`. We use `llama.cpp` because it executes quantized GGUF models on commodity devices and exposes per-tensor format assignment through a text file of regular expressions. Assignments can therefore be built and evaluated directly.

The legacy `q4_0` format groups weights into blocks of 32. Within each block, the largest-magnitude entry determines a half-precision scale. Weights use biased 4-bit codes stored as unsigned nibbles and decoded as signed multiples of the scale. Amortizing the 16-bit scale over 32 weights gives an effective width of 4.5 bits per weight rather than 4. This metadata overhead is omitted by nominal-bit accounting. The K-quant family (`q2_K`, `q3_K`, `q5_K`) uses 256-weight super-blocks with quantized sub-block scales and, for asymmetric formats, minima. The sub-blocks contain sixteen groups of sixteen weights in `q2_K` and `q3_K`, or eight groups of thirty-two in `q5_K`. The resulting widths are 2.5625, 3.4375 and 5.5 bits per weight, respectively. For all formats in this palette, metadata raises realized width above nominal width.

All primary comparisons use the same activation-derived importance matrix for the allocator and production presets. Uniform `q2_K` on Qwen2.5-1.5B reaches a perplexity of 395 without an imatrix and 31.9 with one, demonstrating the importance of calibration in this setting. The presets themselves are mixed-precision assignments that protect selected tensors, including value projections and output heads. They therefore provide a stronger deployment comparison than uniform quantization.

B SOLVER PROCEDURE AND OPTIMALITY AUDIT

We solve Equation (1) approximately, by placing a shadow price λ on payload storage and moving the byte constraint into the objective:

$$L_i(\lambda) \;=\; \arg\min_{L\in\mathcal{L}} \left[\, s(i,L) \;+\; \lambda \cdot \tfrac{n_i \,\mathrm{bpw}(L)}{8} \,\right]. \tag{3}$$

For fixed λ, every tensor selects its level independently and the total size is non-increasing as λ grows. Bisection therefore replaces a search over $|\mathcal{L}|^{|\mathcal{T}|}$ assignments with $|\mathcal{T}|$ independent four-way comparisons per step: 197 tensors on Qwen2.5-1.5B and 198 on 7B. We expand the upper bound on λ, run 64 bisection steps, and then spend any residual budget with the greedy fill described next.

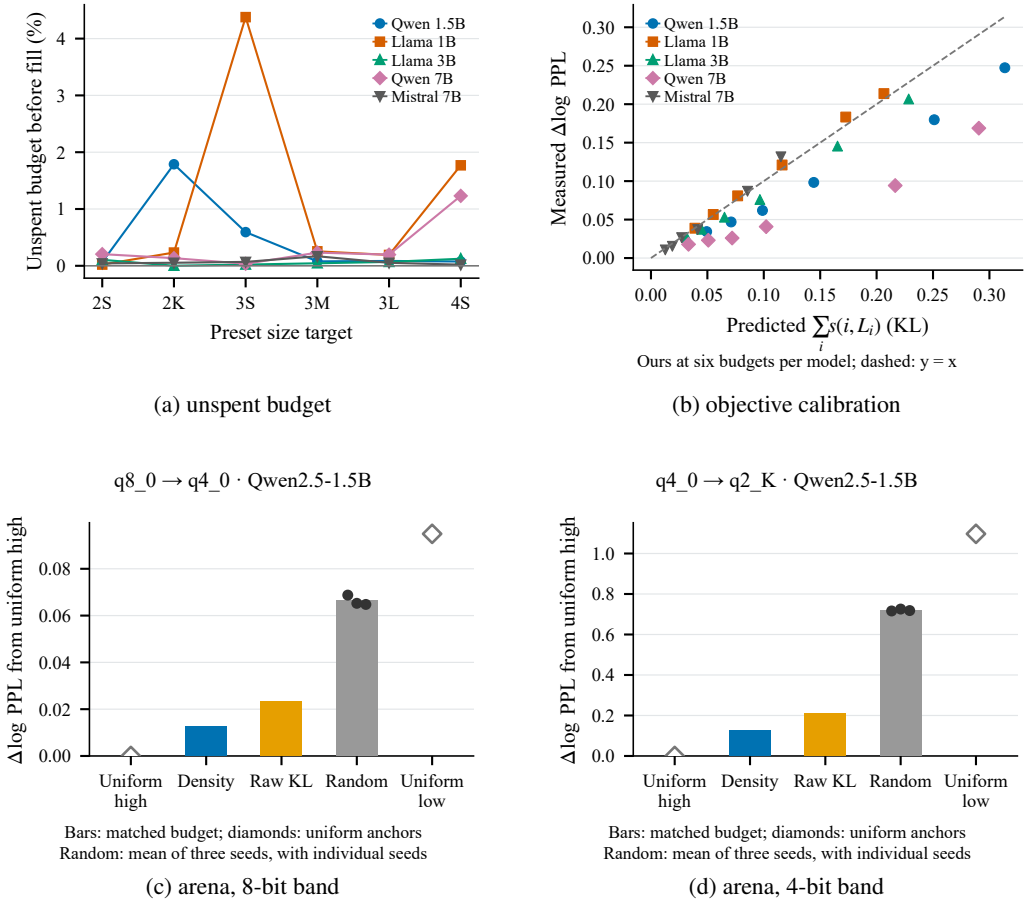
An exact pseudo-polynomial dynamic program is possible after discretizing the byte budget. We use the relaxation because it requires only $64\,|\mathcal{T}|\,|\mathcal{L}|$ comparisons. Exact optimization would remove solver error but would leave surrogate error unresolved. The relaxation selects lower-convex-hull levels; greedy filling then uses the remaining budget.

Bounding surrogate suboptimality. We assess this choice against a dynamic-programming lower bound. Integer byte costs permit exact discretization, provided the state space is tractable. Otherwise, rounding each cost *down* relaxes the constraint, making the dynamic-programming optimum a lower bound. For this solver diagnostic, we set the payload budget numerically equal to each preset's full file size, before the build–measure–retarget loop. Across these 30 instances, the relaxation-plus-fill objective exceeds the bound by 0.3% to 3.7% of the objective across the 30 instances. Relative surrogate suboptimality on these diagnostic instances is therefore bounded above by 3.7%; these gaps are upper bounds, not point estimates. Retargeting changes the payload budget and can change the assignment. The reported bounds do not certify the final matched-size artifacts. Table 5 gives every instance, its bound, and which discretization the bound was computed on. This bound does not establish optimality: single-level filling cannot reach assignments requiring simultaneous changes to two tensors. The bound concerns only the surrogate objective. The separate calibration audit below is needed to relate this objective to measured perplexity.

Allocation procedure. (1) Start with $\lambda_{\mathrm{lo}} = 0$ and enlarge λ_{hi} by factors of four until the induced assignment fits B; (2) run 64 bisection steps, moving the lower bound when the assignment is too large and the upper bound otherwise; (3) assign every tensor by Equation (3) at λ_{hi}; and (4) repeatedly apply the fitting one-level upgrade with the largest decrease in $s(i,L)$ per added byte. The result is feasible because both the bisection output and every accepted upgrade respect B.

Unused budget after relaxation. Equation (3) selects levels on the lower convex hull. Total size changes discretely as λ crosses a selection threshold. One or more tensors may change level, moving the assignment from above to below budget. The returned solution can therefore be substantially *under* budget, leaving bytes that could support further quality-improving upgrades.

At the same 30 diagnostic payload budgets, the relaxation leaves 0.0% to 4.4% of the budget unspent (Figure 3a). We close the gap with a greedy fill: repeatedly apply the feasible single-level upgrade with the greatest positive quality gain per added byte, until no such upgrade fits. The largest shortfall is 4.4%, precluding a matched-size comparison before filling.



(a) unspent budget

(b) objective calibration

(c) arena, 8-bit band

(d) arena, 4-bit band

Figure 3: Solver and precision-regime diagnostics. (a) Unspent payload budget before greedy filling at the diagnostic payload budgets. (b) KL objective versus measured log-perplexity degradation for our six assignments per model; the dashed line is equality, not a fitted calibration curve. (c,d) Qwen2.5-1.5B in the high and low precision bands. Bars share a budget; uniform endpoints are size-unmatched anchors. Random bars show three-seed means with the individual seed values, not confidence intervals.

Table 5: Surrogate optimality diagnostic at the 30 preset file-size targets, used here as payload budgets before file-size retargeting. *Heuristic* is the objective attained by relaxation plus greedy filling; *bound* is the dynamic-programming optimum on the specified discretization grid. All 30 instances use a `relaxed` grid because exact integer discretization exceeds the state-space budget. Rounding costs down relaxes the byte constraint, so the computed optimum is a lower bound. The reported gap therefore bounds heuristic suboptimality from above; it does not measure a perplexity gap. Section C separately compares the objective with observed quality loss.

Model	Preset	Heuristic	Bound	Gap	Quantum (KiB)	Bins	Grid
Llama-3.2-1B	Q2_K_S	0.19527	0.19477	+0.26%	15	35398	relaxed
Llama-3.2-1B	Q2_K	0.16351	0.16217	+0.83%	16	35399	relaxed
Llama-3.2-1B	Q3_K_S	0.11061	0.10751	+2.88%	18	35399	relaxed
Llama-3.2-1B	Q3_K_M	0.071631	0.07068	+1.35%	19	35398	relaxed
Llama-3.2-1B	Q3_K_L	0.05189	0.051432	+0.89%	20	35399	relaxed
Llama-3.2-1B	Q4_K_S	0.036559	0.035946	+1.71%	21	35398	relaxed
Llama-3.2-3B	Q2_K_S	0.21555	0.21221	+1.57%	61	20304	relaxed
Llama-3.2-3B	Q2_K	0.15869	0.1545	+2.71%	66	20304	relaxed
Llama-3.2-3B	Q3_K_S	0.094381	0.092689	+1.83%	74	20304	relaxed
Llama-3.2-3B	Q3_K_M	0.063637	0.062553	+1.73%	81	20304	relaxed
Llama-3.2-3B	Q3_K_L	0.043603	0.042615	+2.32%	87	20304	relaxed
Llama-3.2-3B	Q4_K_S	0.030899	0.030009	+2.97%	93	20304	relaxed
Mistral-7B-v0.3	Q2_K_S	0.10738	0.10521	+2.06%	140	17699	relaxed
Mistral-7B-v0.3	Q2_K	0.081586	0.079396	+2.76%	150	17699	relaxed
Mistral-7B-v0.3	Q3_K_S	0.041087	0.039874	+3.04%	175	17699	relaxed
Mistral-7B-v0.3	Q3_K_M	0.027135	0.026263	+3.32%	194	17699	relaxed
Mistral-7B-v0.3	Q3_K_L	0.018946	0.018278	+3.66%	211	17699	relaxed
Mistral-7B-v0.3	Q4_K_S	0.012691	0.012456	+1.89%	229	17699	relaxed
Qwen2.5-1.5B	Q2_K_S	0.29318	0.28711	+2.11%	31	20304	relaxed
Qwen2.5-1.5B	Q2_K	0.23893	0.23428	+1.99%	33	20304	relaxed
Qwen2.5-1.5B	Q3_K_S	0.13865	0.13585	+2.07%	37	20304	relaxed
Qwen2.5-1.5B	Q3_K_M	0.09504	0.091839	+3.49%	40	20304	relaxed
Qwen2.5-1.5B	Q3_K_L	0.068378	0.066902	+2.21%	42	20304	relaxed
Qwen2.5-1.5B	Q4_K_S	0.047326	0.046078	+2.71%	45	20304	relaxed
Qwen2.5-7B	Q2_K_S	0.27248	0.26758	+1.83%	137	20202	relaxed
Qwen2.5-7B	Q2_K	0.20526	0.19884	+3.23%	146	20202	relaxed
Qwen2.5-7B	Q3_K_S	0.10076	0.098362	+2.44%	169	20202	relaxed
Qwen2.5-7B	Q3_K_M	0.071074	0.069226	+2.67%	184	20202	relaxed
Qwen2.5-7B	Q3_K_L	0.05032	0.04907	+2.55%	198	20202	relaxed
Qwen2.5-7B	Q4_K_S	0.032945	0.032091	+2.66%	215	20202	relaxed

Figure 3 summarizes unused budget, surrogate calibration against artifact measurements, and allocation quality in both precision bands.

C ADDITIVITY OF THE SEPARABLE OBJECTIVE

Additivity. Equation (1) sums one-at-a-time (OAT) sensitivity costs, which assumes tensor perturbations do not interact. Every artifact has a predicted cost, $\sum_i s(i, L_i)$, and measured degradation, $\log \mathrm{PPL} - \log \mathrm{PPL}_{\mathrm{ref}}$. This audit requires no additional quantization; the ablations in Section 7 supply multiple allocations per budget. We distinguish calibration error from ordering error. A common positive rescaling preserves the ranking of allocations at a fixed budget. An *ordering* error can instead favor an assignment with poorer measured quality.

The fitted slope of measured degradation against predicted cost varies across models: 1.09 on Llama-3.2-1B, 0.99 on Mistral-7B-v0.3, 0.87 on Llama-3.2-3B, 0.73 on Qwen2.5-1.5B, 0.45 on Qwen2.5-7B (Figure 3b). This discrepancy can reflect both non-additive tensor interactions and the difference between KL sensitivity and log-perplexity. Calibration is not determined by model size: Llama-3.2-3B is better calibrated than the smaller Qwen2.5-1.5B. Within-budget ordering is more stable than

absolute calibration. We compare every pair of allocations at the same budget, treating predicted-cost differences below 5% as unresolved ties. The predicted and measured rankings disagree in 1 of 60 within-budget comparisons, on Qwen2.5-7B. To assess the 5% threshold, we report a threshold sweep: 15 of 90 at 0%; 1 of 60 at 2%; 1 of 60 at 5%; 1 of 60 at 10%. The count is stable from 2% upward. It increases at a zero threshold when near-ties are included. This result identifies a surrogate limitation on Qwen2.5-7B: the assignment preferred by the separable objective need not minimize measured perplexity. Exact optimization cannot remove this surrogate error (Section S).

D PRECISION-REGIME EXPERIMENT

We compare density-ranked and random assignments under matched relative storage budgets.

We fix a budget at 49% of the achievable storage saving within each band. SMART ranks by measured KL density, SENS ranks by total KL, and RANDOM uses three seeds. The high band H demotes `q8_0` to `q4_0`; the low band L demotes `q4_0` to `q2_K`. We call the measured-versus-random log-PPL gap the *allocation range*, labelled "arena" in Figures 3c and 3d. Three random seeds and one measured rule do not bound other allocators' performance.

The allocation range widens from 0.054 in H to 0.594 in L, a factor of 11.0×. The uniform-endpoint log-PPL gap widens from 0.095 to 1.097. Log-PPL is additive in token-level negative log-likelihood, avoiding the exponential scaling of raw PPL differences.

Calibration control. Disabling the importance matrix increases the low-band uniform gap from 1.097 to 3.582. The allocation range changes little, from 0.594 to 0.586; its cross-band widening remains 10.3×. Calibration does change the density-versus-raw ranking gap, which grows from 0.082 to 0.474 without the imatrix. Omitting the imatrix would therefore overstate the advantage of cost normalization on this model. Primary matched-size comparisons retain calibration on both sides. Academic quantizers use their own calibration procedures (Section O).

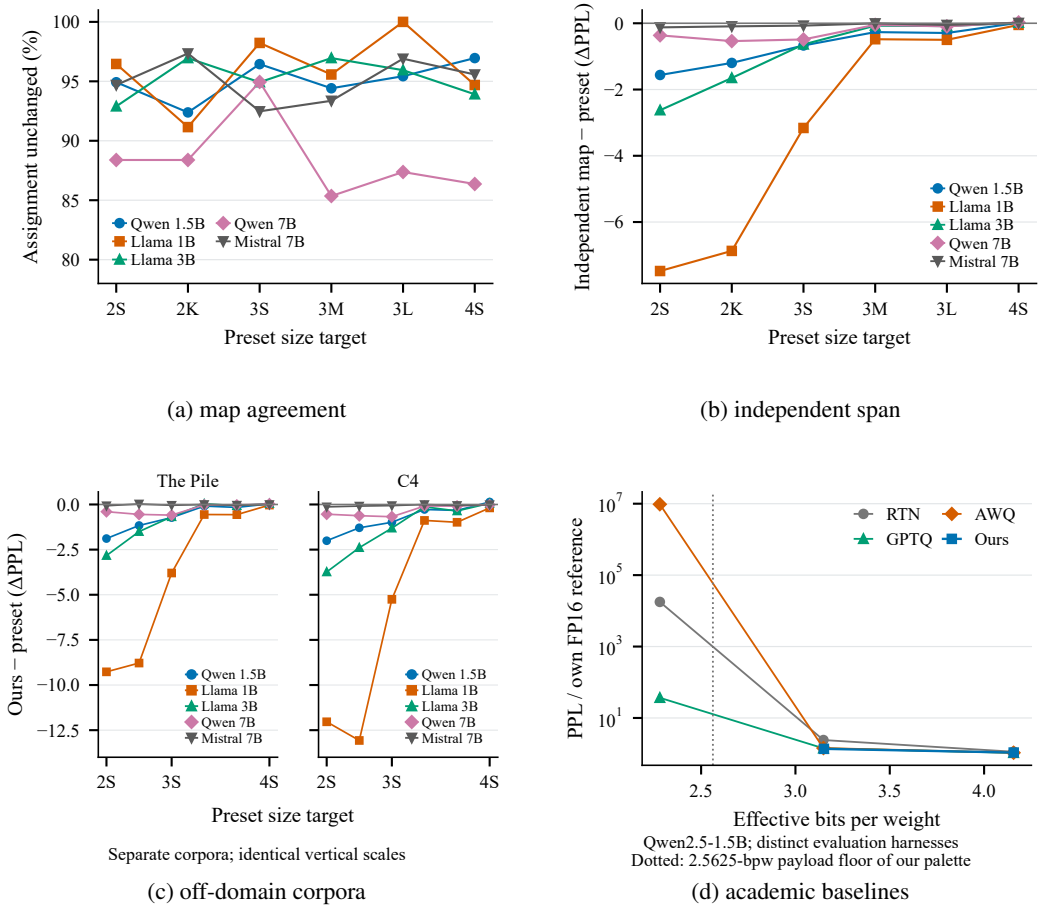
Relation to the evaluated budgets. On Qwen2.5-1.5B, moving from uniform `q8_0` to `q4_0` returns 736 of 1570 MiB and costs 0.96 PPL, a 10% relative increase. The next 345 MiB, from 834 to 489 MiB, costs 21.2 PPL. The high band spans 834–1570 MiB, whereas the evaluated presets span 610–897 MiB. The corresponding 7B anchors are 7.6 GiB and 4.0 GiB, versus presets of 2.6–4.2 GiB. Scaling the floor by parameter count places seven of the 30 model–budget points in the high band, all at the two loosest presets. These comparisons motivate the lower-precision experiments. They neither establish a universal limit on allocation gains nor determine whether a particular device has enough total resident memory.

Production controls. Our palette uses `q4_0`, whereas production presets use `q4_K` at the same realized width. On Qwen2.5-1.5B, we profile `q4_K` and replace the four-bit slot while keeping four available levels. Adding both formats would require handling zero-byte transitions in greedy filling. At matched size, the substitution is better at 6 of 6 budgets, by 0.01 to 0.26 perplexity; the allocator still wins at 6 of 6 budgets. The original asymmetry therefore reduces the observed advantage on this model. Its direction remains untested on the other four models.

E CONTROLS

The independent sensitivity span. Because `calib` supplies both the imatrix and the primary sensitivity map, we repeat every probe on `sens` $[400\text{K}, 560\text{K})$, separated from its neighbours by 40K characters, and rebuild and rescore every allocation from the resulting map. With the imatrix fixed, the per-byte density rank correlations are 0.969 to 0.998; pre-retargeting assignments agree on 85–100% of tensors, and 26 of 30 matched-size points remain improvements (Figure 4a and Table 6).

This control fixes the importance matrix computed on `calib`. Candidate tensors are quantized with that matrix before probing; only the text used to score KL changes, not the quantizer's activation statistics. It isolates variation across profiling spans without testing changes to the importance matrix. Given the importance matrix's effect at low precision (Section A), a priority is an end-to-end control using disjoint text for imatrix and sensitivity estimation. The control is run per tensor on every model, not on class means, so it tests the map the allocator actually consumes. Rebuilding and rescoring every matched-size point from the independent map, 26 of 30 matched-size points remain



(a) map agreement (b) independent span

(c) off-domain corpora (d) academic baselines

Figure 4: Calibration and baseline controls. (a) Pre-retargeting assignment agreement and (b) matched-size perplexity differences using an independent sensitivity span. (c) Pile and C4 differences shown separately on identical axes. (d) Qwen2.5-1.5B academic baselines, normalized by each harness's own FP16 reference; the dotted line marks our palette's payload floor. Panel (d) matches effective width, not memory use.

improvements (Figure 4b). The sensitivity signal remains useful when profiling moves to a disjoint span, conditional on the original importance matrix.

Table 6: Independent-span sensitivity control. Agreement is the fraction of the assignment left unchanged when the sensitivity map is measured on a disjoint calibration span, using preset file sizes as payload budgets before retargeting; Δ compares the final independent allocation with the preset at the same file size.

Model	Preset	Agreement	`llama.cpp`	Ours (`calib`)	Ours (indep.)	Δ
Qwen2.5-1.5B	`Q2_K_S`	95%	13.9119	12.3852	12.3506	-1.561
Qwen2.5-1.5B	`Q2_K`	92%	12.7869	11.5760	11.5901	-1.197
Qwen2.5-1.5B	`Q3_K_S`	96%	11.3565	10.6702	10.6859	-0.671
Qwen2.5-1.5B	`Q3_K_M`	94%	10.5153	10.2886	10.2488	-0.267
Qwen2.5-1.5B	`Q3_K_L`	95%	10.4261	10.1356	10.1325	-0.294
Qwen2.5-1.5B	`Q4_K_S`	97%	10.0173	10.0093	10.0326	+0.015
Qwen2.5-7B	`Q2_K_S`	88%	8.5039	8.5134	8.1392	-0.365
Qwen2.5-7B	`Q2_K`	88%	8.3998	7.9018	7.8614	-0.538
Qwen2.5-7B	`Q3_K_S`	95%	7.9853	7.4905	7.4966	-0.489

Table 6: (continued).

Model	Preset	Agreement	`llama.cpp`	Ours (`calib`)	Ours (indep.)	Δ
Qwen2.5-7B	`Q3_K_M`	85%	7.4453	7.3804	7.3976	-0.048
Qwen2.5-7B	`Q3_K_L`	87%	7.4245	7.3594	7.3312	-0.093
Qwen2.5-7B	`Q4_K_S`	86%	7.2737	7.3206	7.2994	+0.026
Llama-3.2-1B	`Q2_K_S`	96%	18.7799	11.2837	11.3010	-7.479
Llama-3.2-1B	`Q2_K`	91%	17.8561	10.9452	10.9846	-6.872
Llama-3.2-1B	`Q3_K_S`	98%	13.4558	10.2840	10.2950	-3.161
Llama-3.2-1B	`Q3_K_M`	96%	10.3549	9.8795	9.8731	-0.482
Llama-3.2-1B	`Q3_K_L`	100%	10.1457	9.6429	9.6456	-0.500
Llama-3.2-1B	`Q4_K_S`	95%	9.5401	9.4727	9.4908	-0.049
Llama-3.2-3B	`Q2_K_S`	93%	11.4935	8.9002	8.8685	-2.625
Llama-3.2-3B	`Q2_K`	97%	10.0247	8.3714	8.3727	-1.652
Llama-3.2-3B	`Q3_K_S`	95%	8.4516	7.8097	7.8123	-0.639
Llama-3.2-3B	`Q3_K_M`	97%	7.7006	7.6324	7.6286	-0.072
Llama-3.2-3B	`Q3_K_L`	96%	7.6090	7.5135	7.5233	-0.086
Llama-3.2-3B	`Q4_K_S`	94%	7.4206	7.4342	7.4375	+0.017
Mistral-7B-v0.3	`Q2_K_S`	95%	6.4791	6.3761	6.3553	-0.124
Mistral-7B-v0.3	`Q2_K`	97%	6.1799	6.0971	6.0869	-0.093
Mistral-7B-v0.3	`Q3_K_S`	92%	5.8812	5.8013	5.8106	-0.071
Mistral-7B-v0.3	`Q3_K_M`	93%	5.7367	5.7402	5.7340	-0.003
Mistral-7B-v0.3	`Q3_K_L`	97%	5.7241	5.6760	5.6738	-0.050
Mistral-7B-v0.3	`Q4_K_S`	96%	5.6399	5.6494	5.6507	+0.011

Evaluation outside the calibration domain. WikiText-2 supplies the calibration span, the sensitivity map and the evaluation span, so a win measured on it could be an artifact of that shared domain. We rescore every artifact on a slice of The Pile and one C4 validation shard, neither of which is used for calibration, profiling or allocation. Our allocation lowers PPL in 53 of 60 comparisons, with improvements up to 13.07 PPL and regressions no greater than 0.14 PPL (Table 3). The largest effect occurs at the same configuration as on WikiText-2: Llama-3.2-1B at `Q2_K` reaches a 13.1 PPL margin on C4.

Several differences change sign across corpora: Qwen2.5-7B at `Q2_K_S`, Mistral-7B-v0.3 at `Q3_K_M` and Mistral-7B-v0.3 at `Q4_K_S`. Each is a loss on WikiText-2 and a gain on both off-domain corpora. These reversals show that relative allocation quality depends on the evaluation corpus. Table 3 reports every cell, retaining both gains and losses.

Unresolved differences and size verification. Unresolved intervals accompany the smallest point differences, and 5 of the 6 occur at the three loosest budgets. The 25 of 30 improvement count refers to point estimates, whereas the 23 resolved improvements exclude intervals containing zero.

Every artifact is within 0.27% of its target. The search uses `llama-quantize --dry-run` to estimate payload size without writing a file. Final verification uses the built file, including GGUF metadata, alignment padding and tensors outside the level set. We record each configuration's overhead and rebuild mismatched artifacts without adjusting quality scores.

Sensitivity to positive size mismatch. The two-sided tolerance permits slightly larger allocator files. Restricting comparisons to files no larger than the preset leaves 9 of the 30 matched-size points, and the allocation wins all 9 of them. For the complementary subset, there are 21 points where ours is the larger file, by at most 0.27%; on the preset series' own size–quality curve those bytes are worth at most 0.017 perplexity, and the margin exceeds that figure at every one of the 16 we win. This calculation estimates the local size–quality slope from six preset points and assumes it applies to a different allocation. It is therefore a sensitivity analysis, not a rigorous bound. Neither figure is used to adjust a score.

Largest gains and equal-quality interpolation. The largest margin occurs where the preset incurs the greatest observed degradation (Llama-3.2-1B at `Q2_K_S`, where the preset reaches 18.78 and ours 11.28), and narrows to almost nothing at `Q4_K_S`, where measured allocation gains are smaller (Section D). At each preset's perplexity, interpolation estimates that our files are smaller than the

preset that reaches it at 17 of 19 such points, by up to 14%; these estimates do not represent additional builds. For each preset, we interpolate our size–perplexity curve in log–log space between the two builds bracketing its measured perplexity. Presets outside the interpolation range are excluded, yielding fewer comparisons than the matched-size analysis.

Figure 4 summarizes independent-span assignment agreement, resulting quality margins, off-domain evaluation, and effective-width comparisons with published quantizers.

F STORAGE AND RANKING ABLATIONS

Table 7: Separating the byte budget from the average-bit-width convention. All four arms share the sensitivity map, importance matrix and level set. Nominal arms change the cost model; the raw arm changes the ranking procedure. NOMINAL is built once with no size correction, illustrating uncorrected nominal accounting; the other arms are retargeted to the preset's measured size. Its perplexity is not a matched-size comparison because the file is larger.

			NOMINAL		NOM.-MATCHED	RAW
Model	Preset	Ours PPL	size	PPL	PPL	PPL
Qwen2.5-1.5B	Q2_K_S	12.3852	+15.41%	10.8290	12.3561	13.8285
Qwen2.5-1.5B	Q2_K	11.5760	+14.47%	10.5612	11.5720	12.8815
Qwen2.5-1.5B	Q3_K_S	10.6702	+13.01%	10.1728	10.6376	11.3566
Qwen2.5-1.5B	Q3_K_M	10.2886	+12.32%	10.0164	10.3007	10.8569
Qwen2.5-1.5B	Q3_K_L	10.1356	+11.64%	9.9298	10.1184	10.6224
Qwen2.5-1.5B	Q4_K_S	10.0093	+10.93%	9.8843	10.0198	10.4236
Qwen2.5-7B	Q2_K_S	8.5134	+16.90%	7.5704	8.1988	8.4127
Qwen2.5-7B	Q2_K	7.9018	+15.78%	7.4901	7.8655	8.1384
Qwen2.5-7B	Q3_K_S	7.4905	+14.13%	7.3536	7.4747	7.6712
Qwen2.5-7B	Q3_K_M	7.3804	+13.04%	7.3221	7.3751	7.5565
Qwen2.5-7B	Q3_K_L	7.3594	+12.14%	7.3067	7.3537	7.5344
Qwen2.5-7B	Q4_K_S	7.3206	+10.76%	7.2752	7.3188	7.4334
Llama-3.2-1B	Q2_K_S	11.2837	+14.49%	10.2964	11.3372	17.6470
Llama-3.2-1B	Q2_K	10.9452	+14.12%	10.0047	10.9661	15.1743
Llama-3.2-1B	Q3_K_S	10.2840	+13.16%	9.6503	10.2784	11.8314
Llama-3.2-1B	Q3_K_M	9.8795	+12.33%	9.4783	9.8787	10.7801
Llama-3.2-1B	Q3_K_L	9.6429	+11.64%	9.3557	9.6478	9.9531
Llama-3.2-1B	Q4_K_S	9.4727	+10.58%	9.2740	9.4834	9.6702
Llama-3.2-3B	Q2_K_S	8.9002	+15.30%	8.0468	8.9171	10.3734
Llama-3.2-3B	Q2_K	8.3714	+14.32%	7.7958	8.3702	9.3072
Llama-3.2-3B	Q3_K_S	7.8097	+13.19%	7.5649	7.8246	8.1213
Llama-3.2-3B	Q3_K_M	7.6324	+12.26%	7.4518	7.6334	7.8222
Llama-3.2-3B	Q3_K_L	7.5135	+11.49%	7.3808	7.5124	7.6122
Llama-3.2-3B	Q4_K_S	7.4342	+10.84%	7.2978	7.4401	7.4831
Mistral-7B-v0.3	Q2_K_S	6.3761	+17.85%	5.9076	6.3754	6.5417
Mistral-7B-v0.3	Q2_K	6.0971	+15.69%	5.8146	6.0978	6.2748
Mistral-7B-v0.3	Q3_K_S	5.8013	+13.55%	5.7208	5.8025	5.9050
Mistral-7B-v0.3	Q3_K_M	5.7402	+12.56%	5.6703	5.7382	5.8151
Mistral-7B-v0.3	Q3_K_L	5.6760	+11.84%	5.6365	5.6744	5.7781
Mistral-7B-v0.3	Q4_K_S	5.6494	+10.94%	5.6215	5.6450	5.6929

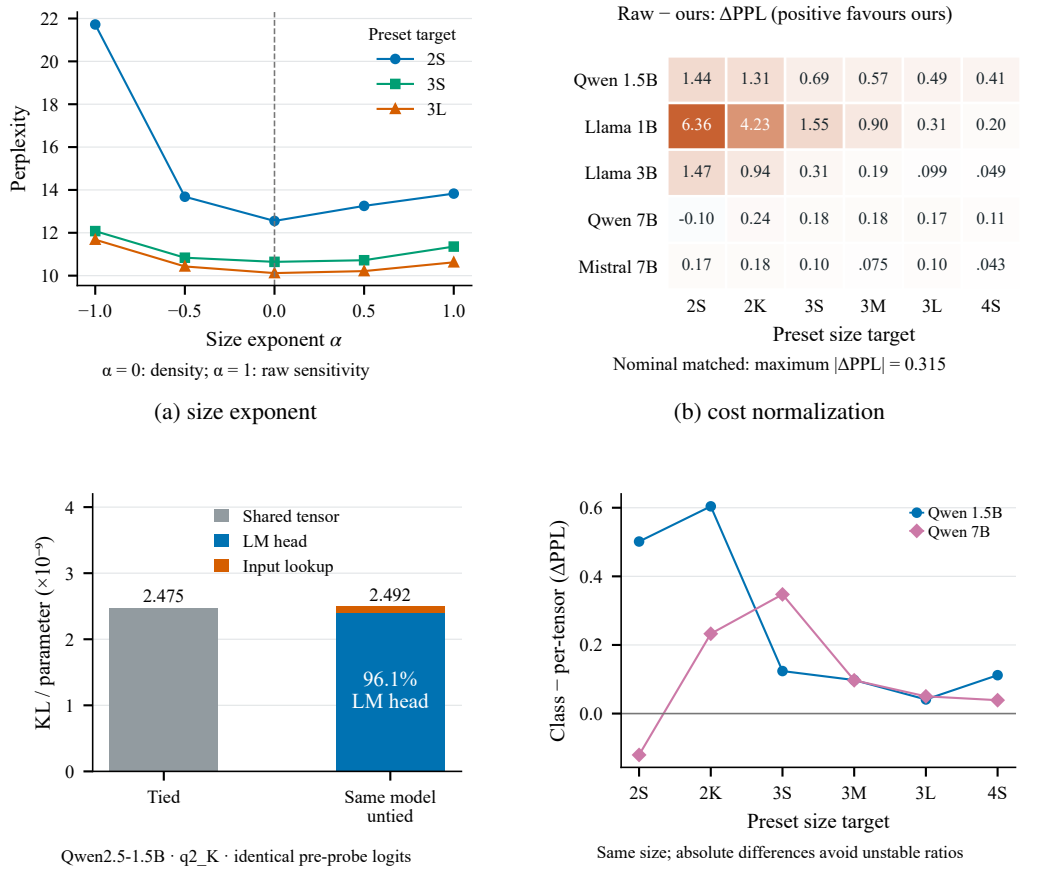
G PROFILING GRANULARITY

Class-level versus per-tensor profiling. Every model is profiled per tensor. To evaluate a cheaper approximation, we rebuild the twelve matched-size points on the two Qwen models using class means. This approximation retains a median of 57% of the margin, ranging from 30% to 86%, over the 7 budgets where that margin exceeds 1% of the preset's perplexity, with 5 further budgets where the per-tensor margin is itself under that threshold. It still beats the preset at 9 of 12 matched-size points,

with 3 points where it loses to the preset outright. Class-level profiling retains some of the benefit but loses a substantial fraction of the per-tensor gain on the two evaluated models.

H THE SIZE EXPONENT

The size exponent. We study a scalar ranking family that scales per-parameter sensitivity density by a power of tensor size. This family excludes joint, hardware-aware and learned allocators, as well as some scalar rules. It ranks tensors by $d_i n_i^{\alpha}$, where $\alpha = 1$ gives total-sensitivity ranking and $\alpha = 0$ gives per-parameter density ranking (Section P). We build and score five exponent settings at three budgets on Qwen2.5-1.5B. The minimum falls at $\alpha = 0$ at every budget, and moving to $\alpha = 1$ costs 0.50 to 1.28 perplexity. This result supports density ranking within the tested family; other signals or format transitions may favor different rules. Figure 5 and Section R present the audits alongside negative and null results.



(a) size exponent

(b) cost normalization

(c) weight tying

(d) profiling granularity

Figure 5: Component audits. (a) The size-exponent sweep on Qwen2.5-1.5B at 2S, 3S and 3L. (b) Raw-ranking perplexity minus ours at matched size; positive values favour ours. (c) The same-model untying control at `q2_K`. (d) Class-level minus per-tensor perplexity on the two evaluated models. Absolute differences avoid unstable ratios when per-tensor gains approach zero.

I SENSITIVITY PATTERNS AND PROFILING COST

All models use per-tensor profiling, summarized by class in Table 1. Class-level profiling reduces the number of probes; Section G quantifies its accuracy cost on the same models. Here we report profiling costs and allocation patterns.

Profiling cost. Sensitivity profiling is the principal additional computational cost of the method. Profiling one model per tensor at four levels takes 22 to 113 GPU-minutes per model on one V100, or 3.0 to 8.6 seconds per probe, on the same hardware the artifacts are later built on. This cost is incurred once per model and amortized across budgets: the six matched-size points in Table 2 share one sweep, and so do the ablation arms, the size-exponent ladder. Allocation uses scalar bisection and greedy filling over a few hundred tensors. The build–measure–retarget loop uses `llama-quantize --dry-run` to reduce full builds; the measured artifact determines acceptance. Section G reports the trade-off from reducing probe counts through class-level profiling.

Allocation pattern. On Qwen2.5-7B, the input embedding stays at `q2_K` at all six budgets. The LM head uses `q5_K` at five budgets and `q3_K` at the tightest, `Q2_K_S`. On Mistral-7B-v0.3, the head stays at `q5_K` throughout, while the input embedding ranges from `q2_K` to `q4_0`.

J Reproduction details

Allocator artifacts are produced by one `llama.cpp` build via

```
llama-quantize --imatrix I.dat --pure --tensor-type-file T.txt src.gguf
                                  out.gguf Q5_K
```

where `T.txt` lists one anchored regular expression per tensor with its assigned level, and unlisted tensors fall back to the base type. Anchoring matters: `llama.cpp` matches with `regex_search`, so an unanchored `blk.2` also matches `blk.20`. Perplexity is measured with `llama-perplexity -c 512` on the `eval` span. Importance matrices are computed once per model with `llama-imatrix -c 512` on the `calib` span and shared by every configuration of that model, ours and the production presets alike. The academic baselines of Section O follow their published calibration procedures.

Evaluation datasets and slices. The following dataset identifiers and character ranges specify the evaluation slices. The WikiText-2 spans are disjoint character ranges of the concatenated `test` split of `wikitext-2-raw-v1`: `calib` is [0, 120K), `val` is [200K, 360K) and `eval` is [600K, 760K), giving files of 120,233, 160,235 and 160,290 bytes. Drawing `calib` from the `test` split departs from the usual convention of calibrating on `train`. The comparison targets `llama.cpp` production presets, whose published recipes are tuned on WikiText-2. Both methods receive the same calibration information through a shared importance matrix and calibration span. Although character-disjoint, the spans share a document collection and may overlap in style and vocabulary. We therefore also rescore on The Pile and C4, neither used for calibration or allocation (Table 3). A pipeline calibrated on the WikiText-2 `train` split is not evaluated here. The off-domain slices are the first 400K characters of `NeelNanda/pile-10k` (`train`) and of `allenai/c4`, shard `en/c4-validation.00000-of-00008.json.gz`, taken document by document in file order and truncated to the same 160K-character length as `eval`: 161,017 and 160,960 bytes. All configurations use `llama-perplexity -c 512` with identical token windows.

The downstream tasks are run by `llama.cpp`'s own multiple-choice evaluator at `-c 2048` with no few-shot prefix, on fixed item subsets in a fixed order: HellaSwag 1200, WinoGrande 1266, ARC-Easy 1200, ARC-Challenge 1172 (its full test set), OpenBookQA 500 and MMLU 1200 items. The subsets are drawn as follows. ARC-Challenge (1172 of 1172) and OpenBookQA (500 of 500) are the complete `test` splits. ARC-Easy takes the first 1200 items of its `test` split, and HellaSwag and WinoGrande the first 1200 and 1266 of their `validation` splits, in native dataset order. MMLU's native order groups questions by subject, so we sample rather than truncate: `cais/mmlu` configuration `all`, split `test`, is shuffled with seed 1234 and the first 1200 items taken, yielding a uniform sample of the 14,042-item set that covers all 57 subjects, with representation approximately proportional to subject size. Taking only the first items would cover only the alphabetically earliest subjects. Both methods use identical deterministic item order, permitting paired analysis (Tables 10 and 11). Paired statistics use the runtime's running-accuracy trace without separate rescoring.

Allocator implementation details. The complete allocation procedure is stated in Section B. The bisection bound grows by factors of four because λ spans many orders of magnitude; 64 steps take it below the resolution at which any tensor changes level.

Dry-run estimates and artifact verification. The build–measure–retarget loop of Section B needs the realized size of a candidate assignment, and quantizing a 7B model to obtain it costs minutes. We use

`llama-quantize --dry-run`, which reports the payload a build would write without writing it, to search the budget, and build the candidate assignment for verification, repeating if the measured file remains outside tolerance. The measured file is still what closes the loop: the difference between the file on disk and the dry-run payload is the GGUF metadata, the alignment padding and the tensors outside the level set, and we record it per configuration rather than assuming it.

Environment. All results come from one `llama.cpp` build (commit `67a17c1`), compiled with CUDA 12.2 and GCC 11.3, on a cluster of NVIDIA V100-32GB and A100-40GB nodes. The cluster could not access `huggingface.co` or `github.com` directly. Checkpoints were retrieved through `hf-mirror.com`, and repositories through a proxy. Llama-3.2 weights were retrieved from a public mirror of the gated upstream files. We verify checkpoint completeness by reading every shard listed in the safetensors index.

Sensitivity measurement without reimplementing K-quants. Bit-exact PyTorch reimplementations of the K-quant formats, including their imatrix-aware weight search, would be substantial and error-prone. Instead, we build one complete GGUF per level and read its weights through `gguf.quants.dequantize`. OAT perturbations therefore preserve the deployed quantized weights and their importance-matrix calibration.

Runtime checks verify tensor mapping and quantify departures from sensitivity monotonicity. The tensor-name mapping must be correct: GGUF and Hugging Face use different tensor names. For Llama-family architectures, conversion also permutes query and key projections for rotary positional embeddings. The probe compares FP16 GGUF weights against checkpoint weights, testing both layouts and selecting the matching permutation. It aborts unless the largest absolute difference across probed tensors is below `2e-3`. Sensitivity is expected to decrease as precision increases. Across the five models and the untying control, 98% of tensors are strictly monotone; exceptions account for 1.9% of the palette-wide effect. Greedy filling rejects any upgrade whose measured sensitivity gain is non-positive.

This profiling harness produces sensitivity maps rather than deployment scores; every perplexity, accuracy and size the paper reports for our allocator and for the production presets is measured by `llama.cpp` on a real GGUF.

K FULL SENSITIVITY TABLES

Table 8 gives the per-class summary for Qwen2.5-1.5B; compare with Table 1 for 7B. The conspicuous difference is `token_embd`, which is tied in the 1.5B model and separate in the 7B one, and which the untying control of Section 8 accounts for. Per-parameter sensitivity and total quality cost induce different rankings. The tied `token_embd` dominates total degradation but ranks second of eight classes by density, at 1.5 times the next class. Byte-constrained allocation preserves this distinction.

Table 8: Qwen2.5-1.5B class shares and per-parameter sensitivity at `q3_K` and `q2_K` (in units of 10^{-9}), with the rank taken at `q2_K`, mirroring Table 1. The tied `token_embd` combines input-embedding and LM-head roles. Shares are rounded independently and sum to 100.1%.

Class	Role	Params	% of total	q3_K	q2_K	Rank by s/n
attn_v	attention value proj.	11 M	0.7	2.099	4.894	1 (most sensitive)
token_embd	input embedding lookup	233 M	15.1	0.695	2.475	2
attn_k	attention key proj.	11 M	0.7	0.656	1.606	3
attn_output	attention output proj.	66 M	4.3	0.411	0.938	4
ffn_down	feed-forward down	385 M	25.0	0.260	0.356	5
ffn_up	feed-forward up	385 M	25.0	0.136	0.353	6
ffn_gate	feed-forward gate	385 M	25.0	0.093	0.278	7
attn_q	attention query proj.	66 M	4.3	0.096	0.238	8 (least sensitive)

L PAIRED STATISTICS AND MULTIPLE TESTING

Table 9: Matched-size downstream accuracy on the deployed GGUF files. Paired significance tests appear in Table 10.

Model	Preset	HellaSwag		WinoGrande		ARC-E		ARC-C		OBQA		MMLU	
		Preset	Ours	Preset	Ours	Preset	Ours	Preset	Ours	Preset	Ours	Preset	Ours
Qwen2.5-1.5B	Q2_K_S	54.3	58.6	60.3	61.8	59.1	65.9	36.9	41.6	38.8	37.0	34.2	36.5
Qwen2.5-1.5B	Q2_K	59.0	60.0	61.7	60.8	63.2	65.8	38.2	40.6	41.6	40.8	34.8	37.1
Qwen2.5-1.5B	Q3_K_S	61.8	62.3	62.2	61.5	63.2	65.0	42.8	43.2	41.4	41.4	38.8	37.0
Qwen2.5-1.5B	Q3_K_M	63.5	62.8	62.7	62.3	63.9	65.9	41.6	43.7	40.4	41.4	38.2	38.2
Qwen2.5-1.5B	Q3_K_L	63.7	63.3	63.3	63.7	65.0	65.2	41.6	42.5	40.4	40.0	38.2	38.2
Qwen2.5-1.5B	Q4_K_S	64.2	64.1	63.0	63.8	64.8	65.2	41.8	42.2	41.4	41.0	38.2	38.0
Qwen2.5-7B	Q2_K_S	70.5	72.8	69.1	69.5	65.0	68.0	44.0	45.7	39.4	40.6	39.8	40.7
Qwen2.5-7B	Q2_K	73.4	74.8	70.6	71.2	70.6	66.4	48.5	46.6	42.2	41.2	43.0	42.2
Qwen2.5-7B	Q3_K_S	74.6	76.2	70.9	70.1	67.8	68.8	47.6	47.0	42.8	41.8	43.2	43.7
Qwen2.5-7B	Q3_K_M	75.7	75.9	70.1	71.4	70.4	70.3	49.0	48.3	42.4	42.2	43.9	44.5
Qwen2.5-7B	Q3_K_L	76.4	77.1	72.0	71.8	69.8	67.7	48.5	47.3	42.4	43.6	43.9	44.2
Qwen2.5-7B	Q4_K_S	76.8	77.5	73.2	71.5	67.8	68.1	48.2	46.8	42.6	43.0	45.1	44.8
Llama-3.2-1B	Q2_K_S	45.5	55.8	55.8	59.3	43.4	52.5	27.6	34.0	32.2	37.4	29.2	30.3
Llama-3.2-1B	Q2_K	48.3	56.3	57.3	59.6	46.2	53.2	28.1	34.6	35.2	36.2	28.7	30.2
Llama-3.2-1B	Q3_K_S	53.6	58.1	58.6	61.8	47.0	53.8	32.8	35.8	36.6	36.2	28.5	31.1
Llama-3.2-1B	Q3_K_M	58.7	59.3	61.0	61.6	53.2	53.6	35.1	35.3	38.0	37.8	31.8	31.0
Llama-3.2-1B	Q3_K_L	57.9	59.8	61.8	62.3	52.8	53.6	33.9	36.8	36.8	37.8	31.2	30.8
Llama-3.2-1B	Q4_K_S	60.7	60.4	61.8	61.4	53.2	53.5	35.1	35.8	38.2	37.4	30.7	31.7
Llama-3.2-3B	Q2_K_S	57.4	65.5	62.4	66.3	53.3	58.2	34.6	38.8	38.2	40.4	33.2	34.0
Llama-3.2-3B	Q2_K	64.6	68.6	64.5	66.8	57.8	57.0	37.6	39.5	39.4	40.2	33.5	34.9
Llama-3.2-3B	Q3_K_S	68.9	71.1	67.0	67.9	59.5	61.2	39.9	41.1	40.6	42.4	35.1	37.1
Llama-3.2-3B	Q3_K_M	71.8	71.7	69.4	69.5	61.2	63.1	42.0	42.4	42.4	44.0	37.3	36.5
Llama-3.2-3B	Q3_K_L	71.5	72.6	69.8	69.1	61.8	63.1	41.5	43.3	43.0	43.0	37.2	37.2
Llama-3.2-3B	Q4_K_S	72.0	72.3	68.9	68.0	62.2	62.4	42.7	42.9	42.4	42.2	36.8	35.9
Mistral-7B-v0.3	Q2_K_S	74.1	75.2	71.9	71.0	66.0	66.6	46.6	47.0	42.2	43.6	38.4	38.8
Mistral-7B-v0.3	Q2_K	75.6	75.1	72.4	71.7	66.8	66.9	46.8	46.9	44.2	43.2	39.3	38.2
Mistral-7B-v0.3	Q3_K_S	78.1	77.2	72.1	73.4	68.4	68.8	48.9	49.7	45.0	45.4	40.2	40.8
Mistral-7B-v0.3	Q3_K_M	77.0	78.1	73.3	73.9	69.4	69.5	50.3	49.7	44.0	45.2	42.4	40.4
Mistral-7B-v0.3	Q3_K_L	77.3	78.2	73.3	73.7	69.0	70.3	50.1	50.3	45.2	44.8	41.6	42.2
Mistral-7B-v0.3	Q4_K_S	77.7	78.2	73.7	73.5	70.0	70.2	51.4	50.5	45.6	46.0	42.3	42.1

Table 10: Paired downstream differences in percentage points; bold marks $p < 0.05$ in our favour under an unadjusted exact McNemar test.

Model	Preset	HellaSwag	WinoGrande	ARC-E	ARC-C	OBQA	MMLU
Qwen2.5-1.5B	Q2_K_S	**+4.2**	+1.6	**+6.8**	**+4.7**	-1.8	**+2.2**
Qwen2.5-1.5B	Q2_K	+1.0	-0.9	**+2.6**	**+2.4**	-0.8	**+2.2**
Qwen2.5-1.5B	Q3_K_S	+0.5	-0.7	+1.8	+0.3	+0.0	-1.8
Qwen2.5-1.5B	Q3_K_M	-0.8	-0.4	**+2.0**	**+2.0**	+1.0	+0.0
Qwen2.5-1.5B	Q3_K_L	-0.3	+0.3	+0.2	+0.9	-0.4	+0.0
Qwen2.5-1.5B	Q4_K_S	-0.2	+0.8	+0.3	+0.3	-0.4	-0.2
Qwen2.5-7B	Q2_K_S	**+2.2**	+0.4	**+3.0**	+1.7	+1.2	+0.9
Qwen2.5-7B	Q2_K	+1.3	+0.6	-4.2	-1.9	-1.0	-0.8
Qwen2.5-7B	Q3_K_S	**+1.7**	-0.8	+0.9	-0.6	-1.0	+0.5
Qwen2.5-7B	Q3_K_M	+0.2	+1.3	-0.1	-0.7	-0.2	+0.6
Qwen2.5-7B	Q3_K_L	+0.7	-0.2	-2.1	-1.2	+1.2	+0.3
Qwen2.5-7B	Q4_K_S	+0.7	-1.7	+0.2	-1.5	+0.4	-0.3
Llama-3.2-1B	Q2_K_S	**+10.3**	**+3.5**	**+9.1**	**+6.4**	**+5.2**	+1.2
Llama-3.2-1B	Q2_K	**+8.0**	+2.4	**+7.1**	**+6.5**	+1.0	+1.6
Llama-3.2-1B	Q3_K_S	**+4.5**	**+3.2**	**+6.8**	**+3.1**	-0.4	**+2.6**

Table 10: (continued).

Model	Preset	HellaSwag	WinoGrande	ARC-E	ARC-C	OBQA	MMLU
Llama-3.2-1B	Q3_K_M	+0.7	+0.6	+0.4	+0.3	-0.2	-0.8
Llama-3.2-1B	Q3_K_L	**+1.8**	+0.6	+0.8	**+2.9**	+1.0	-0.4
Llama-3.2-1B	Q4_K_S	-0.2	-0.4	+0.2	+0.8	-0.8	+1.0
Llama-3.2-3B	Q2_K_S	**+8.1**	**+3.9**	**+4.8**	**+4.2**	+2.2	+0.8
Llama-3.2-3B	Q2_K	**+4.0**	+2.4	-0.8	+1.9	+0.8	+1.4
Llama-3.2-3B	Q3_K_S	**+2.2**	+0.9	+1.7	+1.2	+1.8	**+2.0**
Llama-3.2-3B	Q3_K_M	-0.2	+0.2	**+1.8**	+0.4	+1.6	-0.8
Llama-3.2-3B	Q3_K_L	+1.1	-0.7	+1.3	**+1.8**	+0.0	-0.1
Llama-3.2-3B	Q4_K_S	+0.3	-0.9	+0.2	+0.3	-0.2	-0.8
Mistral-7B-v0.3	Q2_K_S	+1.2	-0.9	+0.6	+0.4	+1.4	+0.3
Mistral-7B-v0.3	Q2_K	-0.5	-0.6	+0.1	+0.2	-1.0	-1.2
Mistral-7B-v0.3	Q3_K_S	-0.9	+1.3	+0.4	+0.9	+0.4	+0.5
Mistral-7B-v0.3	Q3_K_M	**+1.1**	+0.6	+0.1	-0.7	+1.2	-2.0
Mistral-7B-v0.3	Q3_K_L	**+0.8**	+0.4	**+1.3**	+0.2	-0.4	+0.6
Mistral-7B-v0.3	Q4_K_S	+0.5	-0.2	+0.2	-0.9	+0.4	-0.2

Table 11: All 180 downstream comparisons, ordered by McNemar p. Δ is ours minus preset in percentage points, with a paired 95% item-bootstrap CI. Discordant counts (disc.) are $n_{\text{ours only}}/n_{\text{preset only}}$. BH and BY denote Benjamini–Hochberg and Benjamini–Yekutieli adjustments across the full family. BH identifies 24 discoveries at $q < 0.05$.

Model	Preset	Task	Δ (pp)	95% CI	disc.	McNemar p	BH q	BY q
Llama-3.2-1B	Q2_K_S	HellaSwag	+10.33	[+8.08, +12.58]	165/41	7.9×10^{-19}	1.4×10^{-16}	8.3×10^{-16}
Llama-3.2-3B	Q2_K_S	HellaSwag	+8.08	[+6.00, +10.17]	133/36	2.8×10^{-14}	2.5×10^{-12}	1.5×10^{-11}
Llama-3.2-1B	Q2_K	HellaSwag	+8.00	[+5.83, +10.17]	142/46	1.4×10^{-12}	8.1×10^{-11}	4.7×10^{-10}
Llama-3.2-1B	Q2_K_S	ARC-e	+9.08	[+6.58, +11.67]	182/73	6.5×10^{-12}	2.9×10^{-10}	1.7×10^{-9}
Llama-3.2-1B	Q3_K_S	ARC-e	+6.83	[+4.58, +9.17]	144/62	1.1×10^{-8}	3.8×10^{-7}	2.2×10^{-6}
Qwen2.5-1.5B	Q2_K_S	ARC-e	+6.83	[+4.50, +9.08]	145/63	1.3×10^{-8}	3.8×10^{-7}	2.2×10^{-6}
Llama-3.2-1B	Q2_K	ARC-e	+7.08	[+4.50, +9.50]	164/79	5.2×10^{-8}	1.3×10^{-6}	7.8×10^{-6}
Llama-3.2-1B	Q2_K	ARC-c	+6.48	[+3.92, +9.04]	156/80	8.5×10^{-7}	1.9×10^{-5}	0.0001
Llama-3.2-1B	Q3_K_S	HellaSwag	+4.50	[+2.67, +6.33]	93/39	2.9×10^{-6}	5.5×10^{-5}	0.0003
Llama-3.2-1B	Q2_K_S	ARC-c	+6.40	[+3.75, +9.04]	165/90	3.1×10^{-6}	5.5×10^{-5}	0.0003
Qwen2.5-1.5B	Q2_K_S	HellaSwag	+4.25	[+2.50, +6.08]	86/35	4.0×10^{-6}	6.6×10^{-5}	0.0004
Llama-3.2-3B	Q2_K	HellaSwag	+4.00	[+2.33, +5.75]	82/34	9.7×10^{-6}	0.0001	0.0008
Qwen2.5-7B	Q2_K	ARC-e	-4.17	[-6.08, -2.25]	44/94	2.5×10^{-5}	0.0003	0.0020
Llama-3.2-3B	Q2_K_S	ARC-e	+4.83	[+2.50, +7.25]	136/78	8.9×10^{-5}	0.0011	0.0066
Qwen2.5-1.5B	Q2_K_S	ARC-c	+4.69	[+2.39, +6.91]	125/70	1.0×10^{-4}	0.0012	0.0069
Qwen2.5-7B	Q2_K_S	ARC-e	+3.00	[+1.42, +4.67]	68/32	0.0004	0.0046	0.0265
Llama-3.2-3B	Q2_K_S	ARC-c	+4.18	[+1.71, +6.57]	127/78	0.0008	0.0081	0.0465
Llama-3.2-3B	Q3_K_S	HellaSwag	+2.17	[+0.92, +3.42]	43/17	0.0011	0.0107	0.0615
Qwen2.5-7B	Q2_K_S	HellaSwag	+2.25	[+0.92, +3.58]	47/20	0.0013	0.0124	0.0715
Llama-3.2-1B	Q2_K_S	OpenBookQA	+5.20	[+2.00, +8.40]	47/21	0.0022	0.0197	0.1136
Llama-3.2-1B	Q3_K_L	ARC-c	+2.90	[+1.11, +4.69]	78/44	0.0027	0.0218	0.1261
Mistral-7B-v0.3	Q3_K_M	MMLU	-2.00	[-3.25, -0.75]	18/42	0.0027	0.0218	0.1261
Llama-3.2-3B	Q2_K_S	Winogrande	+3.87	[+1.34, +6.40]	158/109	0.0032	0.0253	0.1461
Qwen2.5-7B	Q3_K_L	ARC-e	-2.08	[-3.50, -0.75]	24/49	0.0046	0.0347	0.2003
Llama-3.2-1B	Q3_K_S	ARC-c	+3.07	[+0.85, +5.29]	106/70	0.0082	0.0587	0.3388
Llama-3.2-3B	Q3_K_S	MMLU	+2.00	[+0.50, +3.50]	53/29	0.0106	0.0712	0.4110
Llama-3.2-1B	Q3_K_S	MMLU	+2.58	[+0.67, +4.50]	85/54	0.0107	0.0712	0.4110
Llama-3.2-1B	Q3_K_L	HellaSwag	+1.83	[+0.50, +3.25]	47/25	0.0128	0.0821	0.4741
Mistral-7B-v0.3	Q3_K_M	HellaSwag	+1.08	[+0.33, +1.92]	19/6	0.0146	0.0908	0.5243
Qwen2.5-7B	Q3_K_S	HellaSwag	+1.67	[+0.42, +3.00]	42/22	0.0169	0.0980	0.5655
Llama-3.2-1B	Q3_K_S	Winogrande	+3.24	[+0.71, +5.77]	161/120	0.0169	0.0980	0.5655
Qwen2.5-1.5B	Q3_K_M	ARC-e	+2.00	[+0.42, +3.67]	61/37	0.0197	0.1108	0.6394
Qwen2.5-1.5B	Q2_K	ARC-e	+2.58	[+0.58, +4.67]	100/69	0.0207	0.1131	0.6531

continued on the next page

Model	Preset	Task	Δ (pp)	95% CI	disc.	McNemar p	BH q	BY q
Llama-3.2-3B	Q3_K_M	ARC-e	+1.83	[+0.25, +3.42]	57/35	0.0280	0.1483	0.8563
Llama-3.2-1B	Q2_K_S	Winogrande	+3.48	[+0.47, +6.48]	217/173	0.0293	0.1508	0.8705
Qwen2.5-1.5B	Q2_K_S	MMLU	+2.25	[+0.25, +4.25]	88/61	0.0328	0.1633	0.9429
Qwen2.5-1.5B	Q2_K	ARC-c	+2.39	[+0.26, +4.52]	95/67	0.0336	0.1633	0.9429
Qwen2.5-1.5B	Q2_K	MMLU	+2.25	[+0.25, +4.25]	90/63	0.0352	0.1668	0.9629
Mistral-7B-v0.3	Q3_K_L	ARC-e	+1.33	[+0.17, +2.50]	34/18	0.0365	0.1684	0.9721
Qwen2.5-1.5B	Q3_K_S	MMLU	-1.83	[-3.50, -0.17]	41/63	0.0390	0.1747	1.0000
Qwen2.5-1.5B	Q3_K_M	ARC-c	+2.05	[+0.26, +3.92]	75/51	0.0400	0.1747	1.0000
Mistral-7B-v0.3	Q3_K_L	HellaSwag	+0.83	[+0.08, +1.58]	15/5	0.0414	0.1747	1.0000
Llama-3.2-3B	Q3_K_L	ARC-c	+1.79	[+0.17, +3.41]	59/38	0.0417	0.1747	1.0000
Mistral-7B-v0.3	Q2_K_S	HellaSwag	+1.17	[+0.08, +2.25]	30/16	0.0541	0.2212	1.0000
Qwen2.5-7B	Q4_K_S	ARC-c	-1.45	[-2.90, +0.00]	27/44	0.0568	0.2212	1.0000
Qwen2.5-1.5B	Q3_K_S	ARC-e	+1.75	[+0.08, +3.50]	66/45	0.0572	0.2212	1.0000
Llama-3.2-3B	Q2_K	Winogrande	+2.37	[+0.00, +4.74]	132/102	0.0578	0.2212	1.0000
Qwen2.5-7B	Q2_K	HellaSwag	+1.33	[+0.00, +2.58]	41/25	0.0640	0.2368	1.0000
Llama-3.2-3B	Q3_K_S	ARC-e	+1.67	[+0.00, +3.33]	63/43	0.0645	0.2368	1.0000
Mistral-7B-v0.3	Q3_K_S	HellaSwag	-0.92	[-1.83, +0.00]	10/21	0.0708	0.2547	1.0000
Qwen2.5-7B	Q4_K_S	Winogrande	-1.74	[-3.55, +0.08]	63/85	0.0840	0.2964	1.0000
Qwen2.5-7B	Q2_K	ARC-c	-1.88	[-4.01, +0.17]	65/87	0.0882	0.3052	1.0000
Llama-3.2-3B	Q3_K_L	HellaSwag	+1.08	[-0.08, +2.25]	33/20	0.0984	0.3341	1.0000
Mistral-7B-v0.3	Q3_K_S	Winogrande	+1.26	[-0.16, +2.69]	51/35	0.1052	0.3508	1.0000
Qwen2.5-7B	Q2_K_S	ARC-c	+1.71	[-0.26, +3.67]	80/60	0.1080	0.3509	1.0000
Mistral-7B-v0.3	Q3_K_M	OpenBookQA	+1.20	[+0.00, +2.40]	8/2	0.1094	0.3509	1.0000
Llama-3.2-3B	Q2_K	ARC-c	+1.88	[-0.26, +4.10]	98/76	0.1111	0.3509	1.0000
Mistral-7B-v0.3	Q2_K	MMLU	-1.17	[-2.50, +0.17]	28/42	0.1196	0.3712	1.0000
Llama-3.2-1B	Q2_K	Winogrande	+2.37	[-0.55, +5.29]	196/166	0.1273	0.3879	1.0000
Llama-3.2-3B	Q3_K_L	ARC-e	+1.33	[-0.33, +2.92]	57/41	0.1293	0.3879	1.0000
Llama-3.2-3B	Q3_K_M	OpenBookQA	+1.60	[-0.20, +3.40]	15/7	0.1338	0.3948	1.0000
Llama-3.2-1B	Q4_K_S	MMLU	+1.00	[-0.25, +2.33]	37/25	0.1619	0.4572	1.0000
Llama-3.2-3B	Q3_K_S	OpenBookQA	+1.80	[-0.40, +4.00]	21/12	0.1628	0.4572	1.0000
Mistral-7B-v0.3	Q4_K_S	ARC-c	-0.85	[-1.96, +0.26]	16/26	0.1641	0.4572	1.0000
Llama-3.2-3B	Q2_K	MMLU	+1.42	[-0.50, +3.25]	75/58	0.1651	0.4572	1.0000
Qwen2.5-7B	Q3_K_L	ARC-c	-1.19	[-2.82, +0.34]	38/52	0.1702	0.4643	1.0000
Qwen2.5-7B	Q3_K_M	Winogrande	+1.34	[-0.47, +3.16]	81/64	0.1837	0.4936	1.0000
Mistral-7B-v0.3	Q2_K_S	OpenBookQA	+1.40	[-0.40, +3.20]	14/7	0.1892	0.4964	1.0000
Llama-3.2-1B	Q2_K	MMLU	+1.58	[-0.67, +3.83]	104/85	0.1903	0.4964	1.0000
Qwen2.5-7B	Q4_K_S	HellaSwag	+0.67	[-0.25, +1.58]	19/11	0.2005	0.5155	1.0000
Llama-3.2-3B	Q2_K_S	OpenBookQA	+2.20	[-1.00, +5.20]	37/26	0.2074	0.5257	1.0000
Llama-3.2-3B	Q3_K_S	ARC-c	+1.19	[-0.51, +2.90]	61/47	0.2108	0.5269	1.0000
Llama-3.2-3B	Q3_K_M	MMLU	-0.83	[-2.08, +0.42]	23/33	0.2288	0.5566	1.0000
Llama-3.2-3B	Q4_K_S	MMLU	-0.83	[-2.00, +0.42]	23/33	0.2288	0.5566	1.0000
Mistral-7B-v0.3	Q3_K_S	ARC-c	+0.85	[-0.43, +2.13]	34/24	0.2370	0.5689	1.0000
Qwen2.5-7B	Q3_K_S	ARC-e	+0.92	[-0.50, +2.33]	45/34	0.2604	0.6152	1.0000
Mistral-7B-v0.3	Q4_K_S	HellaSwag	+0.50	[-0.25, +1.25]	13/7	0.2632	0.6152	1.0000
Qwen2.5-7B	Q3_K_L	HellaSwag	+0.67	[-0.42, +1.75]	25/17	0.2800	0.6461	1.0000
Qwen2.5-7B	Q2_K_S	MMLU	+0.92	[-0.67, +2.50]	50/39	0.2891	0.6587	1.0000
Qwen2.5-1.5B	Q2_K_S	OpenBookQA	-1.80	[-4.80, +1.20]	25/34	0.2976	0.6696	1.0000
Mistral-7B-v0.3	Q3_K_M	ARC-c	-0.68	[-1.79, +0.43]	19/27	0.3020	0.6711	1.0000
Qwen2.5-1.5B	Q2_K_S	Winogrande	+1.58	[-1.34, +4.43]	191/171	0.3180	0.6980	1.0000
Llama-3.2-3B	Q4_K_S	Winogrande	-0.87	[-2.45, +0.79]	47/58	0.3291	0.7138	1.0000
Llama-3.2-1B	Q2_K_S	MMLU	+1.17	[-1.00, +3.42]	98/84	0.3353	0.7184	1.0000
Qwen2.5-1.5B	Q2_K	HellaSwag	+1.00	[-0.83, +2.83]	73/61	0.3420	0.7243	1.0000
Mistral-7B-v0.3	Q2_K_S	Winogrande	-0.87	[-2.53, +0.79]	52/63	0.3511	0.7349	1.0000
Qwen2.5-1.5B	Q3_K_M	HellaSwag	-0.75	[-2.17, +0.67]	33/42	0.3557	0.7359	1.0000
Qwen2.5-7B	Q2_K_S	OpenBookQA	+1.20	[-1.00, +3.40]	18/12	0.3616	0.7396	1.0000
Qwen2.5-7B	Q3_K_L	OpenBookQA	+1.20	[-1.00, +3.40]	19/13	0.3771	0.7626	1.0000
Qwen2.5-1.5B	Q3_K_L	ARC-c	+0.85	[-0.94, +2.65]	61/51	0.3952	0.7759	1.0000
Mistral-7B-v0.3	Q3_K_M	Winogrande	+0.63	[-0.63, +1.90]	38/30	0.3961	0.7759	1.0000
Llama-3.2-1B	Q3_K_M	MMLU	-0.75	[-2.33, +0.75]	40/49	0.3966	0.7759	1.0000
Mistral-7B-v0.3	Q2_K	HellaSwag	-0.50	[-1.50, +0.50]	15/21	0.4050	0.7839	1.0000
Qwen2.5-7B	Q2_K	MMLU	-0.83	[-2.67, +0.92]	55/65	0.4114	0.7855	1.0000
Llama-3.2-1B	Q4_K_S	ARC-c	+0.77	[-0.85, +2.39]	53/44	0.4168	0.7855	1.0000

continued on the next page

Model	Preset	Task	Δ (pp) 95% CI	disc.	McNemar p	BH q	BY q
Llama-3.2-3B	Q3_K_S	Winogrande	+0.87 [-1.03, +2.77]	82/71	0.4189	0.7855	1.0000
Mistral-7B-v0.3	Q2_K	OpenBookQA	-1.00 [-3.00, +0.80]	10/15	0.4244	0.7875	1.0000
Llama-3.2-1B	Q3_K_L	ARC-e	+0.75 [-0.92, +2.42]	57/48	0.4351	0.7967	1.0000
Llama-3.2-1B	Q3_K_M	HellaSwag	+0.67 [-0.83, +2.17]	45/37	0.4397	0.7967	1.0000
Mistral-7B-v0.3	Q2_K_S	ARC-e	+0.58 [-0.67, +1.92]	34/27	0.4426	0.7967	1.0000
Qwen2.5-7B	Q3_K_M	ARC-c	-0.68 [-2.22, +0.85]	39/47	0.4505	0.8030	1.0000
Mistral-7B-v0.3	Q3_K_L	MMLU	+0.58 [-0.67, +1.92]	36/29	0.4570	0.8065	1.0000
Llama-3.2-3B	Q3_K_L	Winogrande	-0.71 [-2.45, +0.95]	57/66	0.4709	0.8206	1.0000
Qwen2.5-7B	Q3_K_S	Winogrande	-0.79 [-2.76, +1.11]	74/84	0.4741	0.8206	1.0000
Llama-3.2-3B	Q2_K_S	MMLU	+0.75 [-1.17, +2.58]	70/61	0.4847	0.8235	1.0000
Qwen2.5-1.5B	Q4_K_S	Winogrande	+0.79 [-1.18, +2.84]	88/78	0.4850	0.8235	1.0000
Mistral-7B-v0.3	Q2_K	Winogrande	-0.63 [-2.21, +1.03]	50/58	0.5008	0.8424	1.0000
Qwen2.5-7B	Q3_K_M	MMLU	+0.58 [-0.92, +2.08]	45/38	0.5104	0.8507	1.0000
Llama-3.2-1B	Q3_K_L	OpenBookQA	+1.00 [-1.40, +3.40]	22/17	0.5224	0.8526	1.0000
Qwen2.5-7B	Q3_K_S	OpenBookQA	-1.00 [-3.40, +1.40]	17/22	0.5224	0.8526	1.0000
Mistral-7B-v0.3	Q3_K_S	MMLU	+0.50 [-0.75, +1.83]	34/28	0.5258	0.8526	1.0000
Qwen2.5-7B	Q3_K_S	ARC-c	-0.60 [-2.22, +0.94]	43/50	0.5341	0.8583	1.0000
Qwen2.5-1.5B	Q3_K_M	OpenBookQA	+1.00 [-1.60, +3.60]	24/19	0.5424	0.8640	1.0000
Qwen2.5-7B	Q2_K	OpenBookQA	-1.00 [-3.60, +1.60]	20/25	0.5515	0.8708	1.0000
Llama-3.2-3B	Q2_K	ARC-e	-0.75 [-3.08, +1.58]	93/102	0.5668	0.8868	1.0000
Mistral-7B-v0.3	Q3_K_S	ARC-e	+0.42 [-0.75, +1.58]	28/23	0.5758	0.8868	1.0000
Qwen2.5-1.5B	Q3_K_S	HellaSwag	+0.50 [-0.92, +2.00]	43/37	0.5764	0.8868	1.0000
Qwen2.5-1.5B	Q2_K	Winogrande	-0.87 [-3.71, +1.90]	163/174	0.5860	0.8908	1.0000
Qwen2.5-1.5B	Q3_K_S	Winogrande	-0.71 [-3.00, +1.50]	105/114	0.5889	0.8908	1.0000
Llama-3.2-1B	Q4_K_S	OpenBookQA	-0.80 [-3.00, +1.40]	14/18	0.5966	0.8941	1.0000
Mistral-7B-v0.3	Q3_K_L	Winogrande	+0.39 [-0.79, +1.58]	32/27	0.6029	0.8941	1.0000
Qwen2.5-7B	Q2_K	Winogrande	+0.63 [-1.50, +2.76]	98/90	0.6098	0.8941	1.0000
Llama-3.2-1B	Q3_K_M	Winogrande	+0.63 [-1.50, +2.84]	102/94	0.6172	0.8941	1.0000
Llama-3.2-1B	Q2_K	OpenBookQA	+1.00 [-2.20, +4.20]	35/30	0.6201	0.8941	1.0000
Qwen2.5-7B	Q3_K_S	MMLU	+0.50 [-1.25, +2.17]	57/51	0.6306	0.8941	1.0000
Llama-3.2-3B	Q2_K	OpenBookQA	+0.80 [-1.60, +3.20]	22/18	0.6358	0.8941	1.0000
Llama-3.2-3B	Q4_K_S	HellaSwag	+0.33 [-0.67, +1.33]	22/18	0.6358	0.8941	1.0000
Qwen2.5-7B	Q4_K_S	MMLU	-0.33 [-1.33, +0.67]	18/22	0.6358	0.8941	1.0000
Mistral-7B-v0.3	Q2_K_S	ARC-c	+0.43 [-1.11, +1.96]	44/39	0.6609	0.9222	1.0000
Llama-3.2-1B	Q3_K_L	Winogrande	+0.55 [-1.58, +2.76]	105/98	0.6738	0.9296	1.0000
Llama-3.2-1B	Q3_K_L	MMLU	-0.42 [-2.00, +1.17]	44/49	0.6785	0.9296	1.0000
Llama-3.2-3B	Q3_K_M	ARC-c	+0.43 [-1.19, +2.05]	50/45	0.6817	0.9296	1.0000
Qwen2.5-1.5B	Q2_K	OpenBookQA	-0.80 [-3.80, +2.40]	29/33	0.7035	0.9510	1.0000
Mistral-7B-v0.3	Q2_K_S	MMLU	+0.33 [-1.00, +1.67]	34/30	0.7080	0.9510	1.0000
Llama-3.2-1B	Q3_K_M	ARC-e	+0.42 [-1.42, +2.25]	68/63	0.7269	0.9525	1.0000
Qwen2.5-1.5B	Q4_K_S	ARC-e	+0.33 [-1.08, +1.75]	39/35	0.7275	0.9525	1.0000
Qwen2.5-1.5B	Q3_K_L	HellaSwag	-0.33 [-1.83, +1.08]	36/40	0.7310	0.9525	1.0000
Qwen2.5-7B	Q3_K_L	MMLU	+0.33 [-1.08, +1.75]	42/38	0.7376	0.9525	1.0000
Qwen2.5-7B	Q2_K_S	Winogrande	+0.39 [-1.50, +2.37]	82/77	0.7512	0.9525	1.0000
Mistral-7B-v0.3	Q4_K_S	OpenBookQA	+0.40 [-0.80, +1.60]	6/4	0.7539	0.9525	1.0000
Llama-3.2-1B	Q4_K_S	Winogrande	-0.39 [-2.37, +1.58]	79/84	0.7542	0.9525	1.0000
Qwen2.5-1.5B	Q4_K_S	ARC-c	+0.34 [-1.28, +1.96]	48/44	0.7547	0.9525	1.0000
Qwen2.5-7B	Q3_K_M	HellaSwag	+0.25 [-0.83, +1.42]	25/22	0.7709	0.9525	1.0000
Mistral-7B-v0.3	Q4_K_S	MMLU	-0.25 [-1.42, +0.92]	24/27	0.7798	0.9525	1.0000
Qwen2.5-1.5B	Q3_K_M	Winogrande	-0.39 [-2.69, +1.90]	103/108	0.7831	0.9525	1.0000
Llama-3.2-1B	Q4_K_S	HellaSwag	-0.25 [-1.42, +0.92]	25/28	0.7838	0.9525	1.0000
Qwen2.5-7B	Q4_K_S	ARC-e	+0.25 [-0.92, +1.42]	28/25	0.7838	0.9525	1.0000
Qwen2.5-1.5B	Q3_K_S	ARC-c	+0.34 [-1.45, +2.13]	62/58	0.7843	0.9525	1.0000
Qwen2.5-1.5B	Q4_K_S	MMLU	-0.25 [-1.50, +1.00]	28/31	0.7948	0.9525	1.0000
Mistral-7B-v0.3	Q3_K_L	OpenBookQA	-0.40 [-2.00, +1.20]	7/9	0.8036	0.9525	1.0000
Mistral-7B-v0.3	Q3_K_S	OpenBookQA	+0.40 [-1.20, +2.00]	9/7	0.8036	0.9525	1.0000
Mistral-7B-v0.3	Q4_K_S	Winogrande	-0.24 [-1.50, +1.03]	31/34	0.8043	0.9525	1.0000
Llama-3.2-3B	Q4_K_S	ARC-c	+0.26 [-1.19, +1.71]	38/35	0.8151	0.9590	1.0000
Qwen2.5-7B	Q4_K_S	OpenBookQA	+0.40 [-1.40, +2.20]	13/11	0.8388	0.9690	1.0000
Llama-3.2-1B	Q4_K_S	ARC-e	+0.25 [-1.33, +1.83]	50/47	0.8392	0.9690	1.0000
Qwen2.5-1.5B	Q3_K_L	Winogrande	+0.32 [-1.90, +2.69]	112/108	0.8398	0.9690	1.0000
Qwen2.5-1.5B	Q3_K_L	ARC-e	+0.25 [-1.50, +1.92]	56/53	0.8482	0.9712	1.0000

continued on the next page

Model	Preset	Task	Δ (pp) 95% CI	disc.	McNemar p	BH q	BY q
Qwen2.5-1.5B	Q4_K_S	OpenBookQA	-0.40 [-2.60, +1.80]	15/17	0.8601	0.9712	1.0000
Mistral-7B-v0.3	Q4_K_S	ARC-e	+0.17 [-0.75, +1.17]	18/16	0.8642	0.9712	1.0000
Llama-3.2-1B	Q3_K_M	ARC-c	+0.26 [-1.71, +2.22]	70/67	0.8644	0.9712	1.0000
Qwen2.5-7B	Q3_K_L	Winogrande	-0.24 [-2.13, +1.66]	75/78	0.8716	0.9712	1.0000
Llama-3.2-3B	Q3_K_M	HellaSwag	-0.17 [-1.25, +0.92]	22/24	0.8830	0.9712	1.0000
Mistral-7B-v0.3	Q3_K_L	ARC-c	+0.17 [-1.02, +1.37]	26/24	0.8877	0.9712	1.0000
Qwen2.5-1.5B	Q3_K_L	OpenBookQA	-0.40 [-3.20, +2.40]	24/26	0.8877	0.9712	1.0000
Llama-3.2-1B	Q3_K_S	OpenBookQA	-0.40 [-3.40, +2.60]	28/30	0.8957	0.9712	1.0000
Qwen2.5-1.5B	Q4_K_S	HellaSwag	-0.17 [-1.42, +1.08]	28/30	0.8957	0.9712	1.0000
Llama-3.2-3B	Q4_K_S	ARC-e	+0.17 [-1.25, +1.58]	40/38	0.9099	0.9773	1.0000
Mistral-7B-v0.3	Q2_K	ARC-c	+0.17 [-1.37, +1.71]	42/40	0.9122	0.9773	1.0000
Llama-3.2-3B	Q3_K_M	Winogrande	+0.16 [-1.58, +1.90]	64/62	0.9291	0.9895	1.0000
Llama-3.2-1B	Q3_K_M	OpenBookQA	-0.20 [-3.00, +2.60]	24/25	1.0000	1.0000	1.0000
Llama-3.2-3B	Q3_K_L	MMLU	-0.08 [-1.33, +1.25]	31/32	1.0000	1.0000	1.0000
Llama-3.2-3B	Q3_K_L	OpenBookQA	+0.00 [-2.00, +2.00]	14/14	1.0000	1.0000	1.0000
Llama-3.2-3B	Q4_K_S	OpenBookQA	-0.20 [-2.20, +1.80]	13/14	1.0000	1.0000	1.0000
Mistral-7B-v0.3	Q2_K	ARC-e	+0.08 [-1.25, +1.42]	35/34	1.0000	1.0000	1.0000
Mistral-7B-v0.3	Q3_K_M	ARC-e	+0.08 [-1.08, +1.25]	25/24	1.0000	1.0000	1.0000
Qwen2.5-1.5B	Q3_K_L	MMLU	+0.00 [-1.50, +1.50]	42/42	1.0000	1.0000	1.0000
Qwen2.5-1.5B	Q3_K_M	MMLU	+0.00 [-1.50, +1.50]	41/41	1.0000	1.0000	1.0000
Qwen2.5-1.5B	Q3_K_S	OpenBookQA	+0.00 [-2.20, +2.00]	15/15	1.0000	1.0000	1.0000
Qwen2.5-7B	Q3_K_M	ARC-e	-0.08 [-1.42, +1.25]	32/33	1.0000	1.0000	1.0000
Qwen2.5-7B	Q3_K_M	OpenBookQA	-0.20 [-2.40, +2.00]	15/16	1.0000	1.0000	1.0000

Table 12: All 30 matched-size perplexity tests. Δ is ours minus preset. Two-sided paired sign-flip tests use 10^5 Monte Carlo draws and $p = (r + 1)/(n + 1)$, where r counts equally or more extreme draws. Signs are flipped by blocks of $L = 4$ among 70–78 consecutive evaluation windows; mean absolute lag-one autocorrelation is 0.12. Validity requires invariance under the block sign changes; blocking alone does not establish that assumption. CIs use paired percentile moving-block bootstrap at the same L. BH and BY adjust all cells; BY permits arbitrary dependence for valid marginal tests.

Model	Preset	Δ PPL	95% CI	permutation p	BH q	BY q
Qwen2.5-1.5B	Q2_K	-1.2109	[-1.4412, -1.0016]	1.0×10^{-5}	2.3×10^{-5}	9.2×10^{-5}
Qwen2.5-1.5B	Q3_K_S	-0.6863	[-0.7874, -0.6029]	1.0×10^{-5}	2.3×10^{-5}	9.2×10^{-5}
Qwen2.5-7B	Q2_K	-0.4980	[-0.5824, -0.4041]	1.0×10^{-5}	2.3×10^{-5}	9.2×10^{-5}
Qwen2.5-7B	Q3_K_S	-0.4948	[-0.5396, -0.4394]	1.0×10^{-5}	2.3×10^{-5}	9.2×10^{-5}
Llama-3.2-1B	Q2_K_S	-7.4962	[-7.8588, -6.9876]	1.0×10^{-5}	2.3×10^{-5}	9.2×10^{-5}
Llama-3.2-1B	Q2_K	-6.9109	[-7.3961, -6.3260]	1.0×10^{-5}	2.3×10^{-5}	9.2×10^{-5}
Llama-3.2-1B	Q3_K_S	-3.1718	[-3.4846, -2.8687]	1.0×10^{-5}	2.3×10^{-5}	9.2×10^{-5}
Llama-3.2-1B	Q3_K_M	-0.4754	[-0.5443, -0.3862]	1.0×10^{-5}	2.3×10^{-5}	9.2×10^{-5}
Llama-3.2-1B	Q3_K_L	-0.5028	[-0.5736, -0.4181]	1.0×10^{-5}	2.3×10^{-5}	9.2×10^{-5}
Llama-3.2-3B	Q2_K_S	-2.5933	[-2.8226, -2.3803]	1.0×10^{-5}	2.3×10^{-5}	9.2×10^{-5}
Llama-3.2-3B	Q2_K	-1.6533	[-1.8954, -1.4151]	1.0×10^{-5}	2.3×10^{-5}	9.2×10^{-5}
Llama-3.2-3B	Q3_K_S	-0.6419	[-0.7325, -0.5493]	1.0×10^{-5}	2.3×10^{-5}	9.2×10^{-5}
Mistral-7B-v0.3	Q3_K_S	-0.0799	[-0.1011, -0.0555]	1.0×10^{-5}	2.3×10^{-5}	9.2×10^{-5}
Qwen2.5-1.5B	Q2_K_S	-1.5267	[-1.7854, -1.2338]	2.0×10^{-5}	3.7×10^{-5}	0.0001
Qwen2.5-1.5B	Q3_K_L	-0.2905	[-0.3544, -0.2120]	2.0×10^{-5}	3.7×10^{-5}	0.0001
Mistral-7B-v0.3	Q2_K_S	-0.1030	[-0.1344, -0.0687]	2.0×10^{-5}	3.7×10^{-5}	0.0001
Mistral-7B-v0.3	Q2_K	-0.0828	[-0.1162, -0.0542]	1.0×10^{-4}	0.0002	0.0007
Qwen2.5-1.5B	Q3_K_M	-0.2267	[-0.3079, -0.1305]	0.0001	0.0002	0.0009
Mistral-7B-v0.3	Q3_K_L	-0.0481	[-0.0674, -0.0266]	0.0002	0.0003	0.0011
Llama-3.2-3B	Q3_K_L	-0.0955	[-0.1335, -0.0523]	0.0003	0.0005	0.0020
Llama-3.2-3B	Q3_K_M	-0.0682	[-0.1042, -0.0276]	0.0014	0.0021	0.0082
Qwen2.5-7B	Q3_K_M	-0.0649	[-0.1037, -0.0184]	0.0138	0.0189	0.0754
Qwen2.5-7B	Q4_K_S	+0.0469	[+0.0157, +0.0795]	0.0247	0.0323	0.1289
Qwen2.5-7B	Q3_K_L	-0.0651	[-0.1181, -0.0085]	0.0583	0.0728	0.2910

Table 12: (continued).

Model	Preset	Δ PPL	95% CI	permutation p	BH q	BY q
Llama-3.2-1B	Q4_K_S	-0.0674	[-0.1306, +0.0149]	0.1379	0.1655	0.6613
Mistral-7B-v0.3	Q4_K_S	+0.0095	[-0.0042, +0.0271]	0.3170	0.3657	1.0000
Llama-3.2-3B	Q4_K_S	+0.0136	[-0.0251, +0.0606]	0.5054	0.5616	1.0000
Mistral-7B-v0.3	Q3_K_M	+0.0035	[-0.0138, +0.0220]	0.6949	0.7446	1.0000
Qwen2.5-1.5B	Q4_K_S	-0.0080	[-0.0708, +0.0691]	0.8218	0.8502	1.0000
Qwen2.5-7B	Q2_K_S	+0.0095	[-0.0824, +0.1097]	0.8729	0.8729	1.0000

M THE VALIDATION SPAN

Table 13: Validation-span perplexity and differences on both held-out spans. The `val` span is disjoint from calibration and `eval`; neither held-out span selects assignments. One fixed allocator produces one final artifact per budget after size-only retargeting. The signs agree at 27 of 30 cells; at the three disagreements the `eval`-span margin is a few thousandths of a PPL point (-0.0080, $+0.0095$ and $+0.0035$), while the `val`-span margin is larger, up to -0.0987. These results test span sensitivity, not the independence of the research process from all prior evaluation.

		`val` perplexity		Δ	
Model	Preset	`llama.cpp`	Ours	on `val`	on `eval`
Qwen2.5-1.5B	Q2_K_S	12.9357	11.1267	-1.8090	-1.5267
Qwen2.5-1.5B	Q2_K	11.6713	10.4063	-1.2650	-1.2109
Qwen2.5-1.5B	Q3_K_S	10.2202	9.4307	-0.7895	-0.6863
Qwen2.5-1.5B	Q3_K_M	9.3491	9.0759	-0.2732	-0.2267
Qwen2.5-1.5B	Q3_K_L	9.2637	8.9668	-0.2969	-0.2905
Qwen2.5-1.5B	Q4_K_S	8.7782	8.8463	+0.0681	-0.0080
Qwen2.5-7B	Q2_K_S	7.6008	7.5021	-0.0987	+0.0095
Qwen2.5-7B	Q2_K	7.3208	6.9406	-0.3802	-0.4980
Qwen2.5-7B	Q3_K_S	6.8601	6.4998	-0.3603	-0.4948
Qwen2.5-7B	Q3_K_M	6.5089	6.4545	-0.0544	-0.0649
Qwen2.5-7B	Q3_K_L	6.4672	6.4285	-0.0387	-0.0651
Qwen2.5-7B	Q4_K_S	6.3499	6.3820	+0.0321	+0.0469
Llama-3.2-1B	Q2_K_S	19.3278	10.9945	-8.3333	-7.4962
Llama-3.2-1B	Q2_K	17.8678	10.5991	-7.2687	-6.9109
Llama-3.2-1B	Q3_K_S	13.0043	10.0536	-2.9507	-3.1718
Llama-3.2-1B	Q3_K_M	10.0049	9.5911	-0.4138	-0.4754
Llama-3.2-1B	Q3_K_L	9.8304	9.3657	-0.4647	-0.5028
Llama-3.2-1B	Q4_K_S	9.2342	9.1581	-0.0761	-0.0674
Llama-3.2-3B	Q2_K_S	11.4471	8.8487	-2.5984	-2.5933
Llama-3.2-3B	Q2_K	9.9163	8.1983	-1.7180	-1.6533
Llama-3.2-3B	Q3_K_S	8.3738	7.6794	-0.6944	-0.6419
Llama-3.2-3B	Q3_K_M	7.5460	7.4307	-0.1153	-0.0682
Llama-3.2-3B	Q3_K_L	7.4788	7.3098	-0.1690	-0.0955
Llama-3.2-3B	Q4_K_S	7.2146	7.2417	+0.0271	+0.0136
Mistral-7B-v0.3	Q2_K_S	6.0139	5.9468	-0.0671	-0.1030
Mistral-7B-v0.3	Q2_K	5.7250	5.6211	-0.1039	-0.0828
Mistral-7B-v0.3	Q3_K_S	5.4090	5.3458	-0.0632	-0.0799
Mistral-7B-v0.3	Q3_K_M	5.2827	5.2818	-0.0009	+0.0035
Mistral-7B-v0.3	Q3_K_L	5.2623	5.2283	-0.0340	-0.0481
Mistral-7B-v0.3	Q4_K_S	5.1808	5.1826	+0.0018	+0.0095

N DEPLOYMENT MEASUREMENTS

Peak resident set size (RSS) is reported in MiB.

Table 14: Hardware cost at matched file size. Throughput is reported as a ratio to the preset measured inside the same `llama-bench` invocation; energy is GPU board energy over a fixed generation workload, sampled at 10 Hz. `cpu` rows run with no GPU offload, a CPU-only configuration on the evaluated server. A dagger marks a throughput ratio whose repeat range exceeds 0.15. Three repeats on a shared node cannot separate artifact effects from machine variability. These ratios remain visible here but are excluded from the range in Section 6.

Model	Preset	Device	Peak RSS (MiB)		Throughput ratio		Energy
			preset	ours	prefill	decode	ratio
Qwen2.5-1.5B	Q2_K_S	cpu	987	989	0.959	1.592†	–
Qwen2.5-1.5B	Q2_K_S	gpu	908	910	1.033	0.838	0.998
Qwen2.5-1.5B	Q2_K	cpu	1024	1022	0.972	0.965	–
Qwen2.5-1.5B	Q2_K	gpu	942	942	1.071	0.895	1.036
Qwen2.5-1.5B	Q3_K_S	cpu	1105	1104	1.033	1.244†	–
Qwen2.5-1.5B	Q3_K_S	gpu	1022	1023	1.054	1.083	0.892
Qwen2.5-1.5B	Q3_K_M	cpu	1165	1164	1.010	0.979	–
Qwen2.5-1.5B	Q3_K_M	gpu	1082	1083	1.052	1.152	0.974
Qwen2.5-1.5B	Q3_K_L	cpu	1216	1219	1.012	0.983	–
Qwen2.5-1.5B	Q3_K_L	gpu	1139	1136	1.022	1.380	0.796
Qwen2.5-1.5B	Q4_K_S	cpu	1274	1273	1.017	1.020†	–
Qwen2.5-1.5B	Q4_K_S	gpu	1193	1194	1.102	0.978	0.980
Qwen2.5-7B	Q2_K_S	cpu	3090	3097	1.286	1.050†	–
Qwen2.5-7B	Q2_K_S	gpu	3001	3007	1.002	0.927	1.077
Qwen2.5-7B	Q2_K	cpu	3265	3270	1.169†	1.044†	–
Qwen2.5-7B	Q2_K	gpu	3175	3181	1.021	0.977	1.024
Qwen2.5-7B	Q3_K_S	cpu	3716	3723	1.077†	1.028†	–
Qwen2.5-7B	Q3_K_S	gpu	3627	3634	1.013	1.175	0.910
Qwen2.5-7B	Q3_K_M	cpu	4020	4024	1.111†	1.039†	–
Qwen2.5-7B	Q3_K_M	gpu	3930	3935	1.010	1.168	0.944
Qwen2.5-7B	Q3_K_L	cpu	4288	4289	0.995	0.973	–
Qwen2.5-7B	Q3_K_L	gpu	4198	4201	0.978	1.396	0.783
Qwen2.5-7B	Q4_K_S	cpu	4636	4647	1.069	0.958	–
Qwen2.5-7B	Q4_K_S	gpu	4550	4558	1.059	0.981	1.047

Scope of these measurements. Deployment benchmarking covers Qwen2.5-1.5B and Qwen2.5-7B only. Llama-3.2 and Mistral-7B-v0.3, which carry the largest perplexity margins, were not benchmarked, so no runtime claim in this paper extends to them. The exclusion rule affects configurations unevenly: on Qwen2.5-7B CPU it removes 7 of 12 throughput ratios, and at `Q2_K`, `Q3_K_S` and `Q3_K_M` it removes both ratios, leaving those three budgets with no usable CPU throughput figure; on Qwen2.5-1.5B CPU it removes 3 of 12 (Table 16). The throughput range quoted in Section 6 therefore rests on partial CPU coverage at 7B. Dedicated, core-pinned nodes and additional repetitions are needed, with 7B CPU throughput a priority.

Runtime variability. The first `cpu` row has a daggered decode ratio of 1.59 in Table 14: the preset's three repeats read 22–43 tokens per second while ours read 49–50. We treat this ratio as unresolved because the shared-node repeats cannot isolate the source of the variability.

Table 15: Absolute throughput behind the ratios of Table 14: the median of each set of repeats, with the observed range in brackets, in tokens per second. Shared-node runs use 8 threads, 256-token prefill (`pp`) and 64-token generation (`tg`). CPU rows use `-ngl 0`; GPU rows use full offload.

				prefill (tok/s)		decode (tok/s)	
Model	Preset	Device	Reps	preset	ours	preset	ours
Qwen2.5-1.5B	Q2_K_S	cpu	3	1656 [1632–1678]	1594 [1569–1597]	36 [22–43]	50 [49–50]
Qwen2.5-1.5B	Q2_K_S	gpu	3	8007 [7832–8021]	8213 [8211–8215]	296 [296–296]	248 [248–248]
Qwen2.5-1.5B	Q2_K	cpu	3	1534 [1529–1585]	1514 [1483–1518]	49 [49–50]	48 [48–48]
Qwen2.5-1.5B	Q2_K	gpu	3	7920 [7285–7929]	8244 [8239–8252]	264 [259–264]	235 [235–235]
Qwen2.5-1.5B	Q3_K_S	cpu	3	1365 [1330–1433]	1421 [1416–1423]	46 [26–47]	45 [45–46]
Qwen2.5-1.5B	Q3_K_S	gpu	3	7716 [7692–7745]	8137 [8130–8142]	191 [191–191]	207 [207–207]
Qwen2.5-1.5B	Q3_K_M	cpu	3	1325 [1287–1326]	1326 [1321–1329]	46 [42–47]	44 [44–44]
Qwen2.5-1.5B	Q3_K_M	gpu	3	7628 [7622–7634]	8028 [8016–8036]	203 [203–203]	234 [234–234]
Qwen2.5-1.5B	Q3_K_L	cpu	3	1251 [1239–1251]	1260 [1258–1268]	45 [43–45]	43 [43–43]
Qwen2.5-1.5B	Q3_K_L	gpu	3	7816 [7767–7824]	7975 [7948–7995]	199 [199–200]	275 [275–275]
Qwen2.5-1.5B	Q4_K_S	cpu	3	1179 [1153–1187]	1196 [1187–1197]	45 [34–46]	42 [42–42]
Qwen2.5-1.5B	Q4_K_S	gpu	3	7447 [6883–7449]	7992 [7970–8000]	297 [295–298]	290 [290–291]
Qwen2.5-7B	Q2_K_S	cpu	3	323 [312–331]	414 [410–418]	8 [7–9]	9 [7–9]
Qwen2.5-7B	Q2_K_S	gpu	3	2432 [2427–2434]	2448 [2413–2448]	139 [139–139]	129 [129–129]
Qwen2.5-7B	Q2_K	cpu	3	311 [310–391]	391 [387–392]	8 [6–8]	8 [7–9]
Qwen2.5-7B	Q2_K	gpu	3	2392 [2364–2393]	2435 [2431–2436]	116 [116–116]	114 [114–114]
Qwen2.5-7B	Q3_K_S	cpu	3	343 [275–517]	343 [340–514]	7 [7–8]	7 [7–7]
Qwen2.5-7B	Q3_K_S	gpu	3	2297 [2293–2314]	2342 [2307–2342]	82 [82–82]	96 [96–96]
Qwen2.5-7B	Q3_K_M	cpu	3	275 [260–315]	313 [311–315]	7 [6–7]	7 [7–7]
Qwen2.5-7B	Q3_K_M	gpu	3	2274 [2273–2276]	2315 [2256–2319]	92 [92–92]	107 [107–107]
Qwen2.5-7B	Q3_K_L	cpu	3	295 [293–296]	293 [292–294]	6 [6–7]	6 [6–7]
Qwen2.5-7B	Q3_K_L	gpu	3	2305 [2277–2374]	2284 [2226–2291]	89 [89–89]	124 [124–124]
Qwen2.5-7B	Q4_K_S	cpu	3	231 [213–257]	243 [231–273]	6 [6–7]	6 [6–6]
Qwen2.5-7B	Q4_K_S	gpu	3	2195 [2116–2203]	2311 [2265–2317]	132 [132–132]	130 [130–130]

Table 16: Measurement ledger for Table 14. *Reps* counts repeated `llama-bench` invocations; *ratios* counts the throughput ratios produced. *Kept* and *cut* apply the stated repeat-range threshold of 0.15. Excluded ratios remain daggered in Table 14 but do not enter the summary range in Section 6. *Energy* says whether a board-energy reading was available, which it is on GPU rows only. No invocation was rerun or dropped for any other reason.

Model	Preset	Device	Reps	Ratios	Kept	Cut	Energy
Qwen2.5-1.5B	Q2_K_S	cpu	3	2	1	1	–
Qwen2.5-1.5B	Q2_K_S	gpu	3	2	2	0	yes
Qwen2.5-1.5B	Q2_K	cpu	3	2	2	0	–
Qwen2.5-1.5B	Q2_K	gpu	3	2	2	0	yes
Qwen2.5-1.5B	Q3_K_S	cpu	3	2	1	1	–
Qwen2.5-1.5B	Q3_K_S	gpu	3	2	2	0	yes
Qwen2.5-1.5B	Q3_K_M	cpu	3	2	2	0	–
Qwen2.5-1.5B	Q3_K_M	gpu	3	2	2	0	yes
Qwen2.5-1.5B	Q3_K_L	cpu	3	2	2	0	–
Qwen2.5-1.5B	Q3_K_L	gpu	3	2	2	0	yes
Qwen2.5-1.5B	Q4_K_S	cpu	3	2	1	1	–
Qwen2.5-1.5B	Q4_K_S	gpu	3	2	2	0	yes
Qwen2.5-7B	Q2_K_S	cpu	3	2	1	1	–
Qwen2.5-7B	Q2_K_S	gpu	3	2	2	0	yes
Qwen2.5-7B	Q2_K	cpu	3	2	0	2	–
Qwen2.5-7B	Q2_K	gpu	3	2	2	0	yes
Qwen2.5-7B	Q3_K_S	cpu	3	2	0	2	–
Qwen2.5-7B	Q3_K_S	gpu	3	2	2	0	yes
Qwen2.5-7B	Q3_K_M	cpu	3	2	0	2	–
Qwen2.5-7B	Q3_K_M	gpu	3	2	2	0	yes
Qwen2.5-7B	Q3_K_L	cpu	3	2	2	0	–

Table 16: (continued).

Model	Preset	Device	Reps	Ratios	Kept	Cut	Energy
Qwen2.5-7B	Q3_K_L	gpu	3	2	2	0	yes
Qwen2.5-7B	Q4_K_S	cpu	3	2	2	0	–
Qwen2.5-7B	Q4_K_S	gpu	3	2	2	0	yes

O ACADEMIC-BASELINE PROTOCOL AND LIMITATIONS

GPTQ, AWQ, QuaRot and round-to-nearest use uniform widths for the tensors they quantize. Our method instead allocates formats and uses the importance-matrix calibration supplied by `llama.cpp`. These approaches address distinct quantization decisions and may be combined.

Evaluation and storage matching. Baseline weights are dequantized to FP16 and evaluated by one Hugging Face harness on `eval`. Allocator artifacts are evaluated directly in `llama.cpp`. The harnesses differ by approximately 20% even without quantization. We therefore report each method's PPL divided by its own harness's FP16 reference. This normalization reduces scale differences but does not establish equivalent degradation measurements. Context construction, token counting and windowing can still affect the comparison.

Each allocator artifact targets the baseline's effective width rather than the nearest production-preset size. For grouped quantization, the storage model is $b + (16 + b)/g$, including an FP16 scale and packed b-bit zero point per group. For per-output-channel quantization, the corresponding overhead is $(16+b)/d_{\text{in}}$, which is a few thousandths of a bit at these tensor shapes. Tables round these per-channel widths to the nominal value. This accounting covers quantized tensors; the full checkpoint ledger separately includes embeddings, heads and normalization parameters retained in FP16.

Published implementations. We use the authors' repositories (IST-DASLab/gptq, mit-han-lab/llm-awq and spcl/QuaRot), their hyperparameters, and 128×2048-token calibration data. Two repositories import packing kernels at module load: `awq_inference_engine` and `fast_hadamard_transform`. The dequantized evaluation does not use the packing kernels, so compatible stand-ins satisfy those imports. The Hadamard stand-in implements the required fast Walsh–Hadamard transform.

QuaRot exclusion. The released code rejects `Qwen2ForCausalLM`. The attempted Llama-3.2-1B integration also encounters tied input and output weights, which layer-norm fusion and rotation would otherwise transform twice. Untying the head resolves that duplication but does not restore expected behavior. Using the released `fake_quant` pipeline, including online Hadamard transforms, still gives 38.2 PPL at four bits, versus 21.1 for per-channel round-to-nearest. We treat this as unresolved integration failure and exclude QuaRot from comparative conclusions. It is not evidence about the published method's quality. The academic-baseline comparison covers models with at most 1.5B parameters, making Llama-3.2-1B the only candidate for this integration. The released QuaRot code supports Mistral-7B-v0.3, a `MistralForCausalLM` with untied embeddings. Evaluating that supported model is a priority for resolving the integration issue.

Integration checks. AWQ's search updates block scales in place and returns them for replay on a fresh model. Replaying them on the searched model applies scaling twice. The diagnostic PPL values were 11.63 for FP16, 11.68 after search, and 21533 after erroneous replay. Similarly, rotating and rounding QuaRot weights without the online Hadamard transforms produced invalid outputs. These were integration errors, not measured failures of the published algorithms. Reported GPTQ and AWQ results use corrected pipelines; QuaRot remains excluded.

Checkpoint footprint. The comparison matches *effective linear-layer* width, not whole-model storage. The baseline drivers quantize only the internal block linear layers and leave token embeddings, norms and the LM head in FP16, so at the same effective width their estimated checkpoint footprints are 1.43 to 1.92 times the measured GGUF sizes (Table 18). The Qwen2.5-1.5B control assigns 96.1% of decomposed tied-tensor sensitivity to the head (Section 8). Keeping the head in FP16 preserves precision at additional storage cost. This asymmetry prevents the four losses at 4.16 bits per weight from isolating estimator quality. The same asymmetry applies at the other end: the configurations we mark unreachable ask for effective widths below our palette floor while their estimated complete

checkpoints, 733.0–801.4 MiB, are larger than the `Q2_K_S` artifacts we do build. An ablation holding embeddings, norms and head at FP16 on our side, or matching on total checkpoint bytes, would help separate quantizer quality from whole-model allocation and storage cost.

Scope. The comparison measures algorithmic degradation under different harnesses, rather than matched deployment performance. Baselines are evaluated on dequantized weights and would require their own kernels for packed execution. The results do not establish that mixed-precision allocation supersedes GPTQ or AWQ. A common harness supporting both their packed models and our GGUFs would provide a stronger comparison.

Table 17: Published weight-only quantizers, run from the authors' own repositories and scored on dequantized weights by one Hugging Face harness, against an allocation of ours built to that baseline's own effective width, and a note where the width is below what four executable levels can reach. The two harnesses disagree in absolute value even at `fp16`, so we report the degradation ratio relative to each harness's *own* unquantized reference; this does not remove all harness effects. Effective width is $b + (16 + b)/g$ for a grouped quantizer, since it stores a scale and a zero point per group, and b where the method quantizes per output channel and takes no group size; our artifact is retargeted until its realized size is within the same 0.35% the size matching tolerates elsewhere in this paper, and the width it actually reached is printed beside it.

Model	Method	Bits	Group	Effective bpw	PPL	PPL / `fp16`	Ours, built to that width
Qwen2.5-1.5B	AWQ	4	128	4.156	12.3678	1.06×	10.4036, 1.08× at 4.157 bpw
Qwen2.5-1.5B	GPTQ	4	128	4.156	12.2614	1.05×	10.4036, 1.08× at 4.157 bpw
Qwen2.5-1.5B	RTN	4	128	4.156	13.1910	1.13×	10.4036, 1.08× at 4.157 bpw
Qwen2.5-1.5B	AWQ	3	128	3.148	16.6913	1.44×	13.2129, 1.37× at 3.151 bpw
Qwen2.5-1.5B	GPTQ	3	128	3.148	16.4298	1.41×	13.2129, 1.37× at 3.151 bpw
Qwen2.5-1.5B	RTN	3	128	3.148	28.0381	2.41×	13.2129, 1.37× at 3.151 bpw
Qwen2.5-1.5B	AWQ	2	64	2.281	1.1×10^{8}	9.6×10^{6}	*below the palette floor*
Qwen2.5-1.5B	GPTQ	2	64	2.281	427.4504	36.75×	*below the palette floor*
Qwen2.5-1.5B	RTN	2	64	2.281	2.1×10^{5}	1.8×10^{4}	*below the palette floor*
Llama-3.2-1B	AWQ	4	128	4.156	13.4292	1.11×	10.2840, 1.13× at 4.154 bpw
Llama-3.2-1B	GPTQ	4	128	4.156	13.1077	1.08×	10.2840, 1.13× at 4.154 bpw
Llama-3.2-1B	RTN	4	128	4.156	14.4814	1.19×	10.2840, 1.13× at 4.154 bpw
Llama-3.2-1B	RTN	4	–	4.000	21.0808	1.74×	10.4848, 1.15× at 4.002 bpw
Llama-3.2-1B	AWQ	3	128	3.148	20.6824	1.70×	14.3921, 1.58× at 3.149 bpw
Llama-3.2-1B	GPTQ	3	128	3.148	19.5518	1.61×	14.3921, 1.58× at 3.149 bpw
Llama-3.2-1B	RTN	3	128	3.148	41.7975	3.45×	14.3921, 1.58× at 3.149 bpw
Llama-3.2-1B	RTN	3	–	3.000	2278.2	187.81×	16.6004, 1.82× at 3.001 bpw
Llama-3.2-1B	AWQ	2	64	2.281	1.3×10^{6}	1.1×10^{5}	*below the palette floor*
Llama-3.2-1B	GPTQ	2	64	2.281	1132.8	93.39×	*below the palette floor*
Llama-3.2-1B	RTN	2	64	2.281	2.6×10^{4}	2123×	*below the palette floor*
Llama-3.2-1B	RTN	2	–	2.000	5.3×10^{5}	4.4×10^{4}	*below the palette floor*

Table 18: What an effective width implies for a whole checkpoint. An effective width prices the tensors a method quantizes, and the published drivers quantize the linears inside the transformer blocks while leaving the embedding, the LM head and the norms in `fp16`, whose parameter share is reported explicitly in the table. The model-wide figure and the checkpoint size are arithmetic over the tensor inventory, not a measured packed file, because these baselines are never packed. Our column reports measured GGUF size at the matched effective width. Our allocation also quantizes embeddings and heads, while tensors outside the palette remain separately accounted for. The final columns illustrate why effective width over selected tensors should not be interpreted as the storage cost of a complete checkpoint.

Model	Effective bpw	Quantized share	Model-wide bpw	Checkpoint (MiB)	Our GGUF (MiB)
Qwen2.5-1.5B	4.156	84.9%	5.947	1094.3	764.9
Qwen2.5-1.5B	3.148	84.9%	5.091	936.9	579.8

Table 18: (continued).

Model	Effective bpw	Quantized share	Model-wide bpw	Checkpoint (MiB)	Our GGUF (MiB)
Qwen2.5-1.5B	2.281	84.9%	4.355	801.4	*unreachable*
Llama-3.2-1B	4.156	78.7%	6.674	983.1	612.0
Llama-3.2-1B	4.000	78.7%	6.551	965.0	589.5
Llama-3.2-1B	3.148	78.7%	5.880	866.2	464.0
Llama-3.2-1B	3.000	78.7%	5.763	849.0	442.0
Llama-3.2-1B	2.281	78.7%	5.197	765.6	*unreachable*
Llama-3.2-1B	2.000	78.7%	4.976	733.0	*unreachable*

P THE SIZE-EXPONENT LADDER

We study a scalar ranking family that scales per-parameter sensitivity density by a power of tensor size. This family excludes joint, hardware-aware and learned allocators, as well as some scalar rules. It ranks tensors by $d_i \cdot n_i^{\alpha}$, where d_i is a per-parameter sensitivity density and n_i the parameter count. Within this family, $\alpha = +1$ ranks by total sensitivity and $\alpha = 0$ ranks by density. Density ranking is proportional to loss per saved byte only when comparing the same format transition. A metric that is natively *intensive*, such as a normalized spectral entropy, sits at $\alpha = 0$ in its native form and is pushed to $\alpha = -1$ by dividing it by cost.

The ranking is turned into a multi-level assignment by greedy demotion, not by quantile cuts, and the same routine serves the RAW arm of Section 7. Every tensor starts at the top level `q5_K`. While the assignment exceeds the byte budget, we consider for each tensor its single one-level demotion, score that move by $\big(s(i, L_{\downarrow}) - s(i, L_i)\big) \cdot n_i^{\alpha-1}$, apply the lowest-scoring move, and subtract the bytes it saves; we stop when the budget is met or no demotion saves bytes. With $\alpha = 1$ the key is the raw cost increase, which is the RAW control; $\alpha = 0$ divides it by n_i. All four levels remain reachable through sequential demotions. A tensor is demoted again only when its next step has the lowest available score. Unlike the main solver, this sweep uses greedy demotion without Lagrangian initialization. At $\alpha = 0$, it divides by parameter count rather than each transition's byte saving; different transitions have different width increments.

We sweep $\alpha \in \{-1, -\frac{1}{2}, 0, \frac{1}{2}, 1\}$ at three Qwen2.5-1.5B budgets. Every setting is built and scored as a GGUF under the same file-size tolerance. Holding the cost signal fixed at the measured KL density, the minimum falls at $\alpha = 0$ at every budget, and moving to the protect-the-sensitive-layer end, $\alpha = 1$, costs 0.50 to 1.28 perplexity. The curve is not flat on either side: $\alpha = -1$ gives larger tensors lower demotion scores and can over-demote them, while $\alpha = +1$ protects a handful of expensive tensors and pushes almost everything else to `q2_K` (Figure 2).

The sweep has two limitations. First, fixing the cost signal identifies its preferred exponent, not a universal exponent: a natively intensive metric such as a normalized spectral entropy already sits at $\alpha = 0$ in its own units and would be pushed to $\alpha = -1$ by dividing it by cost. Second, the single-model evaluation does not establish generality across architectures or scales.

Q INTERPRETATIONS NOT SUPPORTED BY THE EVIDENCE

The following interpretations are not supported by the current measurements. We retain them explicitly to distinguish the evidence from stronger claims.

Ranking has little effect. A weak association between perplexity and one size exponent does not establish that ranking is unimportant. At matched size, cost-normalized allocation outperforms the raw-ranking control at 29 of 30 matched-size points, by up to 6.36 PPL. The control holds sensitivity, palette and size target fixed but does not reimplement a published joint optimizer.

Tied embeddings imply intrinsically fragile input lookup. The shared `token_embd` has 11 times the total sensitivity of the `attn_v` class, yet ranks 25 of 197 tensors per parameter. Total sensitivity is not density. The same-model untying control attributes most sensitivity to the output role (Section 8).

Absolute cost differences establish failure to transfer across devices. Different absolute cost coefficients do not alone establish that a resulting allocation fails to transfer. A common positive rescaling of sensitivity preserves the ordering and the constrained optimum. Changes in relative costs require a separate analysis. No cross-ARM transfer conclusion is drawn from this evaluation.

R NEGATIVE AND NULL RESULTS

Profiling granularity. Substituting class-mean density for per-tensor sensitivity and rebuilding every matched-size point retains a median of 57% of the margin, ranging from 30% to 86%, over the 7 budgets where that margin exceeds 1% of the preset's perplexity, with 5 further budgets where the per-tensor margin is itself under that threshold; over all twelve it still beats the presets at 9 of 12 matched-size points, with 3 points where it loses to the preset outright. The production results use per-tensor profiling throughout; class means are used only in this ablation.

Proxies for measured sensitivity. We test Frobenius rounding error $\|W - Q(W)\|_F^2$ and spectral weight summaries as inexpensive proxies for OAT sensitivity. Their weak rank agreement with measured sensitivity does not support replacing the OAT sweep in these experiments. The rounding-error family in particular has a per-parameter density that is nearly constant across tensors, so ranking by $d_i n_i^\alpha$ reduces to ranking by size and α merely selects how strong the size bias is.

S EVALUATION LIMITATIONS

Evaluation covers three English corpora and six likelihood-scored multiple-choice tasks. It does not test free generation, instruction following, multilingual text, code or long-context behavior. Sensitivity and allocation behavior in these settings remain untested.

Paired bootstrap intervals and hypothesis tests quantify uncertainty conditional on the chosen text spans or task subsets. They do not measure variation across independently sampled corpora, training seeds or model checkpoints. Consecutive windows may remain dependent after blocking, so block-length sensitivity is reported alongside family-level corrections.

Allocator and production-preset scores come from executable GGUFs in `llama.cpp`. Academic-baseline scores use dequantized weights in a different harness. Normalizing by each harness's FP16 reference reduces scale differences without making the protocols equivalent. Hardware evaluation uses shared server nodes and does not establish performance on mobile devices.

T ADDITIONAL PROTOCOL AND RESULT DETAILS

T.1 EXPERIMENTAL PROTOCOL

Every primary allocator configuration targets the shipped preset's *measured* file size. Dry-run payload estimates guide retargeting; the built file must be within 0.35% of the target, with further retargeting if necessary. Both sides share the importance matrix, calibration span, and evaluation settings. Their assignments differ, and the four-bit format is asymmetric: ours uses `q4_0`, whereas the presets use `q4_K`. Section 7 reports a controlled substitution on Qwen2.5-1.5B.

Sensitivity to the profiling span. We rebuild the whole map on a disjoint span and rerun the pipeline. Per-byte density rank correlations between the maps are 0.969 to 0.998, and pre-retargeting assignments agree on 85–100% of tensors. After rebuilding and rescoring with the independent map, 26 of 30 matched-size points remain improvements (Table 6).

T.2 RESULTS

Paired uncertainty and block length. Each main-table comparison pairs the same 70–78 evaluation windows between allocator and preset artifacts. Llama models use 70 windows, Qwen models 72, and Mistral 78; the count depends on tokenization. Their mean absolute lag-one autocorrelation is 0.12, motivating blocks of $L = 4$ for bootstrap intervals and sign-flip tests. The interval is a paired moving-block percentile interval; the sign-flip test requires invariance under block sign changes.

Counts of resolved intervals, BH discoveries and BY discoveries are 25, 25 and 23 at L=1; 24, 23 and 21 at L=4; 25, 23 and 21 at L=8. The sweep supports reporting the $L = 4$ results while retaining dependence as an evaluation limitation. Table 12 lists every p-value and adjustment.

Downstream tests. Across 180 comparisons, 121 point estimates favour ours and 55 favour the preset; four are ties. At unadjusted $p < 0.05$, 39 gains and four losses are significant. The largest loss is Qwen2.5-7B at `Q2_K` on ARC-Easy, at -4.17 percentage points. BH correction identifies 21 gains and three losses, while BY retains 17 discoveries. Of the 24 BH discoveries, 21 occur at the three tightest budgets. Tables 10 and 11 provide all paired tests, bootstrap intervals and discordant counts.

Runtime qualification. Peak resident memory differs by at most 11 MiB at matched file size. Throughput ratios span 0.84 to 1.40, and energy-per-token ratios span 0.78 to 1.08, after the stated variability exclusions. Tables 15 and 16 give absolute rates, repeat counts and excluded ratios. At attained preset quality, measured cross-budget comparisons save 35 to 447 MiB of peak resident memory of peak resident memory. These server measurements do not establish mobile performance gains.

Academic-baseline qualification. Allocator artifacts are built to each baseline's effective width and compared using harness-normalized PPL degradation. The allocation improves 10 of 14 reachable comparisons. All four losses occur at 4.16 bits per weight, against AWQ and GPTQ. Seven additional baseline settings lie below our palette floor of 2.5625 realized bits per weight. We mark them as unreachable rather than approximate their target widths. The baseline drivers retain embeddings, heads and normalization parameters in FP16, comprising 21% of the smaller model's parameters. The resulting checkpoint size can differ substantially from its quantized-tensor width: at 3.000 bits per weight on Llama-3.2-1B the checkpoint that width implies is 849 MiB, against 442 MiB for our GGUF at the same width, or 48% less. Section O and Table 18 distinguish measured GGUF sizes from estimated baseline footprints.

T.3 Separating storage accounting from sensitivity ranking

RAW demotes in increasing order of raw quality cost rather than of cost per byte, under the same byte budget and the same measured sensitivities. This is the $\alpha = 1$ end of the size-exponent family of Section P: the rule that ranks demotions by total sensitivity increase without normalizing by storage savings. It is worse at 29 of 30 matched-size points, by up to 6.36 perplexity, and better at the one remaining point by 0.10. Averaged over all 30 matched-size points that penalty is 79% of the mean margin our method holds over the production presets: a ratio of two means, not a per-cell figure. The resulting assignments illustrate the difference: at the tightest 1.5B budget the cost-normalized rule keeps 38, 28, 60 and 71 tensors at `q5_K`, `q4_0`, `q3_K` and `q2_K`, whereas the raw rule keeps 5 tensors at `q4_0` or `q5_K`, 20 at `q3_K`, and 172 of 197 at `q2_K` (Figure 2). Most tensors therefore remain at the lowest level.

T.4 Interpreting sensitivity under weight tying

Smaller language models often tie the input embedding to the LM head to reduce parameter count. Qwen2.5-1.5B does; Qwen2.5-7B does not. A sensitivity score on the shared tensor therefore combines two computational roles, and two quantities are easy to conflate. On Qwen2.5-1.5B the shared `token_embd` dominates the model in *total* quality cost: demoting it alone to `q2_K` costs 0.578, which is 11 times the whole `attn_v` class combined. Per parameter, it ranks 25th among 197 tensors. In the class-level comparison, its density is 1.5 times that of the next class. Its total cost establishes neither exceptional sensitivity density nor the contribution of input lookup.

At 7B the two roles occupy separate, identically shaped tensors, and their measured sensitivities differ by a factor of 76 to 112, depending on the level; the allocator keeps `output` at `q3_K` or `q5_K`, above `token_embd` at `q2_K`. But that comparison is confounded by scale, vocabulary, depth and training, so it cannot on its own attribute the difference to the role.

T.5 Related work

Mixed-precision methods use Hessian, geometric or activation-derived sensitivity (Dong et al., 2019; 2020; Shen et al., 2020; Guan et al., 2024; Xiao et al., 2025). HAWQ-V2 selects bit configurations using an additive sensitivity score and a Pareto frontier over model size. Our contribution is the

executable instantiation and controlled evaluation, not the knapsack formulation itself. Proposition 1 gives sufficient conditions for equivalent nominal and realized budgets, plus a counterexample. The raw-ranking and exponent controls test cost normalization with a fixed sensitivity signal, palette and realized-size target. They do not establish superiority over published joint optimizers.