FROM ATTENTION SENSITIVITY TO LAYER ROLE: REVISITING MIXED-PRECISION QUANTIZATION OF TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Most post-training quantization pipelines keep each quantized weight matrix close to its pretrained counterpart, one matrix at a time, and stop there. Whether that proxy tracks what the attention block actually computes, or how errors in the Q, K and V projections compound once they meet inside the softmax, is rarely checked. The obvious fix is to write the objective on the attention output itself, over all three projections at once, and to use that same objective wherever the pipeline needs a signal. That is what **JAB** does. It defines one scalar loss over the joint Q, K, V weights of a block, evaluated against the block's real causally-masked attention output, and uses it both to fit the quantized weights (GPTQ warm start, then STE with learnable scales) and to score the block for a multiple-choice knapsack allocation.

On attention-only quantization of Mistral-7B this pays off: at 3 bits JAB recovers 77–90% of the gap between uniform GPTQ and full precision, and its sensitivity estimate tracks an oracle costing 73 forward passes to within a fraction of a perplexity point. Once MLP layers enter the allocation, though, JAB is beaten by a role-aware offset rule that needs no sensitivity estimate at all, both on GPT-2 with QKV and MLP quantized together and on the full Mistral-7B model. With a 3-bit floor, this rule quantizes 96.4% of Mistral-7B's weights to 4.5 bits per parameter at 6.933 perplexity — within 4.4% of full precision (6.643) at $3.56\times$ compression, 14 GB to 3.93 GB — against 7.158 for JAB at the same budget, and to 3.5 bits at 7.875 ($4.57\times$). Whichever matrix a weight belongs to inside its block matters more than any sensitivity estimate we managed to compute.

Two things came out sideways. Block-local reconstruction turned out to be an unreliable proxy for end-to-end perplexity: in one run a $4.6\times$ improvement in a block's own objective came with a $32\times$ increase in perplexity, which is why every allocation in this paper is validated end-to-end rather than on local objectives alone. And on attention-only quantization, fine-tuning consistently moved weights farther from their pretrained values while pulling attention outputs closer, with net gains. Post-training seems to recover attention behavior, not weights.

1 INTRODUCTION

Large language models (LLMs) face severe memory bottlenecks during inference, and quantization mitigates these costs by mapping real-valued weights to discrete integers (Rakka et al., 2026). Because quantization-aware training (QAT) remains prohibitively resource-intensive, post-training quantization (PTQ) has become the dominant paradigm for efficient compression (Cao & Aref, 2025).

Early quantization methods relied on the inverse Hessian matrix of the Optimal Brain Surgeon (OBS) algorithm, which proved computationally intractable for modern deep networks (Hassibi & Stork, 1990; Liu et al., 2025). GPTQ resolved this by processing weights in blocks via Cholesky decomposition, which made it possible to compress 175-billion-parameter LLMs such as GPT-3, models that are prohibitively expensive to deploy in standard FP16 precision, to 3 or 4 bits efficiently. Building on OBS, GPTQ minimizes a layer-wise weight-reconstruction loss; this local

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.

weight-space objective can, however, fail to capture end-to-end Transformer fidelity (Frantar et al., 2023).

Pushing below 4 bits exposes structural vulnerabilities to heavy-tailed activation outliers. Outlier-Aware Weight Quantization (OWQ) addresses this by preserving a subset of sensitive weights in higher precision (Lee et al., 2024). To automate variable bit-width allocation under strict budgets, HAWQ-V2 sets bit widths from the average Hessian trace of the task loss, which serves as a measure of parameter sensitivity (Dong et al., 2020), and the resulting assignment is optimized with Multiple-Choice Knapsack Problem (MCKP) solvers (Sinha & Zoltners, 1979; Lee et al., 2026). HAQ learns the allocation instead (Wang et al., 2019).

Moving beyond independent layer-wise optimization, BRECQ introduced block-wise reconstruction to capture cross-layer dependencies (Li et al., 2021). Attention-Aware PTQ (APTQ) narrows the gap for attention layers by folding attention information — the gradients of the non-linear softmax — into its sensitivity estimate to guide optimization, but that criterion is still a weight-space proxy, not the reconstruction objective itself (Guan et al., 2024).

In most PTQ pipelines, then, the objective that reconstructs the quantized weights is not the criterion that sets their bit widths (Appendix A). This disconnect between the sensitivity-estimation criterion and the actual reconstruction loss, which we term the *proxy gap*, motivates the Joint Attention-Based (JAB) framework: we set out to test whether one objective could do both jobs, produce the weights and set the budget.

JAB (Joint Attention-Based allocation) uses a single loss for both steps. We define $\mathcal{L}_{\mathrm{JAB}}$ over the concatenated $[W_Q \mid W_K \mid W_V]$ parameters of a block and evaluate it against that block's own causally-masked attention output and attention map. We then use it twice. Minimizing it directly produces the weights: a GPTQ warm start, then straight-through fine-tuning with learnable scales. Differentiating it twice, through Hutchinson probes of the Hessian trace, scores the block for an MCKP allocation. We ask whether the same machinery transfers once allocation extends past attention to the MLP and to the whole model.

Findings Attention-only and heterogeneous full-model quantization separate cleanly. On attention-only Mistral-7B, JAB closes most of the gap between uniform GPTQ and full precision at 3 bits, recovering 77–90% of it across all four calibration/evaluation corpus pairings. Its Hessian-trace criterion tracks the end-to-end oracle closely too, and at a small fraction of the model evaluations. The same advantage does not carry over to the MLP or the full model. On GPT-2's MLP the oracle allocates bits mainly by matrix role rather than depth, and the Hessian criterion does not pick that structure up. On the full Mistral-7B model a seven-parameter role offset rule reaches lower perplexity than JAB at the same parameter-weighted budget, with no sensitivity estimation at all. With a 3-bit floor on every unit, the same rule reaches 6.933 perplexity at 4.5 bits per parameter, within 4.4% of full precision (6.643) at $3.56\times$ compression, and 7.875 at 3.5 bits ($4.57\times$). Two further results bear on block-local refinement. Improving a block's local reconstruction objective does not guarantee better end-to-end quality: in one experiment a $4.6\times$ reduction in the local objective came with a $32\times$ increase in end-to-end perplexity. And on attention-only quantization, fine-tuning moves the weights farther from their floating-point values while making the attention outputs more accurate, which points to functional recovery mattering more than weight-space fidelity.

Contributions

- **Joint objective for reconstruction and allocation.** We introduce JAB: a single joint QKV attention loss used for both quantized-weight refinement and sensitivity-based bit allocation. We also separate the joint from the per-matrix treatment of QKV. The two agree in expectation, but the per-matrix form can underestimate the variability of the true joint loss (Section 2.3).
- **Evidence for role-dependent allocation.** MLP and full-model experiments show that matrix role can provide a stronger allocation signal than the curvature criterion JAB uses, which motivates a Hessian-free role-based prior (Sections 5 and 6).
- **Local-to-global failure mode.** We isolate a case where improving a block-local reconstruction objective substantially worsens end-to-end perplexity. Local refinement objectives alone do not guarantee global quality (Section 6.4).

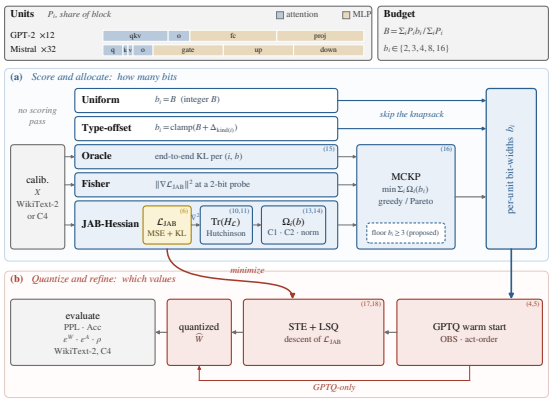

Figure 1: Overview of the proposed mixed-precision PTQ pipeline. The method separates the process into two stages. **(a) Bit allocation:** candidate sensitivity criteria determine how many bits each unit receives under a parameter-weighted budget. **(b) Quantization and refinement:** GPTQ provides the initial quantized weights, after which the joint attention loss $\mathcal{L}_{\mathrm{JAB}}$ is minimized through the quantizer using STE+LSQ.

- **Attention-space interpretation of refinement.** We introduce the amplification factor $\rho = \varepsilon^A/\varepsilon^W$ to separate weight fidelity from functional fidelity. Attention-aware refinement can improve the latter while moving the weights farther from their floating-point values (Section 4).

A parallel line of PTQ work improves the quantizer itself rather than the allocation, through activation-aware scaling (Lin et al., 2024; Xiao et al., 2023), learned clipping (Shao et al., 2024), or incoherence processing (Tseng et al., 2024). These are orthogonal to bit allocation and compose with it; we hold the quantizer fixed and vary only how bits are assigned.

2 Method

2.1 Layer-wise PTQ

Let $X \in \mathbb{R}^{B\times T\times d}$ be a calibration input to one block and $y = xW$ a sublayer with $W \in \mathbb{R}^{d_{\text{in}}\times d_{\text{out}}}$; the QKV projection fuses $W = [W_Q \mid W_K \mid W_V]$, and attention is $A(X) = \mathrm{softmax}\left(QK^\top/\sqrt{d_{\text{head}}} + M\right)V$ with causal mask M. GPTQ minimizes the local surrogate

$$\mathcal{L}(\widehat{W}) = \textstyle\sum_{x\in X} \|xW - x\widehat{W}\|_2^2, \qquad H = 2X^\top X, \tag{1}$$

placing each scalar on a symmetric grid with a per-(row, group) scale s,

$$Q(w) = \mathrm{clip}\big(\mathrm{round}(w/s), -q_{\max}, q_{\max}\big)\cdot s, \qquad q_{\max} = 2^{b-1}-1, \tag{2}$$

and compensating not-yet-quantized columns with the OBS update (Appendix B.1). H is accumulated by a forward hook on the model in its *current* state, so block i's Hessian already reflects blocks $0,\dots,i-1$ being quantized. The solve runs in float64; in float32 the column loop's repeated division by the inverse-Hessian diagonal makes perplexity non-monotone in the bit-width.

2.2 The joint attention loss

Let $A(X), P(X)$ be the attention output and post-softmax map of the unmodified block, and $\widehat{A}(X;w), Q(X;w)$ those of a differentiable re-implementation evaluated at the quantized $w = [\mathrm{vec}(\widehat{W}_Q); \mathrm{vec}(\widehat{W}_K); \mathrm{vec}(\widehat{W}_V)]$. The joint loss is

$$\mathcal{L}_{\mathrm{JAB}}(w) = \big\|A(X) - \widehat{A}(X;w)\big\|_F^2 + \lambda_{\mathrm{KL}}\, D_{\mathrm{KL}}\big(P \,\|\, Q(w)\big). \tag{3}$$

Because the two terms are normalized differently, the effective ratio of KL to MSE is 10^3–10^4 at $\lambda_{\mathrm{KL}} = 0.1$: what is optimized is attention-map distillation with an output-MSE regularizer (Appendix B.2).

2.3 WHY ONE JOINT OBJECTIVE RATHER THAN THREE

Attention is *bilinear* in (W_Q, W_K): the scores depend on the pair only through $W_Q W_K^\top$, whose perturbation contains a cross term $\delta_Q \delta_K^\top$ that no per-matrix model represents. Treating the three separately is the block-diagonal model H_{sep} of $H_{\mathcal{L}} = \nabla_w^2 \mathcal{L}_{\text{JAB}}$, discarding cross-blocks $\Gamma = H_{\mathcal{L}} - H_{\text{sep}}$ that are structurally non-zero: $\mathcal{L}_{\text{JAB}}$ is invariant under $W_Q \mapsto W_Q R$, $W_K \mapsto W_K R^{-\top}$ per head, so $H_{\mathcal{L}}$ has a null space of dimension at least $n_{\text{head}} d_{\text{head}}^2$ (49,152 per GPT-2 block) on which the separable model reports damage where there is none. With a zero-mean rounding residual of variance σ^2, the two agree in expectation but not in spread:

$$\mathbb{E}[\Delta\mathcal{L}_{\text{JAB}}] = \mathbb{E}[\Delta\mathcal{L}_{\text{JAB}}^{\text{sep}}], \qquad \text{Var}[\Delta\mathcal{L}_{\text{JAB}}] - \text{Var}[\Delta\mathcal{L}_{\text{JAB}}^{\text{sep}}] = \tfrac{1}{2}\sigma^4 \|\Gamma\|_F^2 \;>\; 0. \tag{4}$$

A per-matrix model is therefore unbiased on average but over-confident about the tail — the wrong profile for allocation, since perplexity is convex and effectively unbounded in the low-bit direction. The argument bears on reconstruction and fine-tuning; for scoring, a trace sees only diagonal blocks and the two coincide (Appendix B.3).

2.4 SENSITIVITY AND ALLOCATION

The criterion is the Hessian trace of the *same* loss w.r.t. the *same* vector, estimated with Hutchinson's method (Hutchinson, 1990), $\text{Tr}(H_{\mathcal{L}}) \approx \frac{1}{n}\sum_i v_i^\top H_{\mathcal{L}} v_i$ with Rademacher v_i, each product computed by two chained backward passes (Pearlmutter, 1994). Because $\mathcal{L}_{\text{JAB}}$ is zero at the float point w^*, the Hessian there reduces to

$$H_{\mathcal{L}}(w^*) = 2J_A^\top J_A + \lambda_{\text{KL}}\, J_Q^\top \mathcal{I}(P)\, J_Q, \tag{5}$$

a squared attention-output Jacobian plus an attention-map Fisher information, not a generic curvature probe. Following HAWQ-V2, the allocator scores each candidate width by

$$\textbf{(C1)}\;\; \Omega_i(b) = \text{Tr}(H_i)\, \big\|Q_b(W_i) - W_i\big\|_F^2, \qquad \textbf{(C2)}\;\; \Omega_i^{\text{prop}}(b) = (L - i)\, \Omega_i(b), \tag{6}$$

where C2 re-weights by the number of layers an error propagates through. As a bound, the oracle **(C3)** measures each (i, b) pair's end-to-end logit KL by quantizing only layer i (≈ 73 model loads). Each layer then chooses $b_i \in \mathcal{B} = \{2, 3, 4, 8, 16\}$ by

$$\min \textstyle\sum_i \Omega_i(b_i) \quad \text{s.t.} \quad \sum_i c(b_i) \le B, \tag{7}$$

solved greedily or on the pooled Pareto frontier (Appendix B.5).

2.5 JOINT FINE-TUNING THROUGH THE QUANTIZER

GPTQ solves equation 1, not equation 3. The final stage takes GPTQ's solution as a warm start and descends $\mathcal{L}_{\text{JAB}}$ through the quantizer, using the straight-through estimator on weights (Bengio et al., 2013) and the LSQ gradient on the per-(row, group) scales (Esser et al., 2020). Because GPTQ's output sits exactly on grid points, learning rates are set in units of the mean grid step $\bar{s}$ ($\eta_w = \alpha_w \bar{s}$, $\eta_s = \alpha_s \bar{s}$) over 200 Adam steps; teacher and student see the same partially-quantized input; a held-out batch gates every write-back; and we log the *flip rate*, the fraction of weights whose grid position changed, since a stage that changes nothing and one that changes everything can leave perplexity equally unmoved (Appendix B.6).

2.6 METRICS

We report perplexity and next-token accuracy under a sliding window, the weight-space error $\varepsilon^W = \|W - \widehat{W}\|_F / \|W\|_F$, the attention-space error ε^A, and the amplification factor $\rho = \overline{\varepsilon^A} / \overline{\varepsilon^W}$, which separates weight fidelity from functional fidelity (Appendix B.7).

3 EXPERIMENTAL SETUP

We study GPT-2 small (Radford et al., 2019) (124M parameters, 12 blocks) and Mistral-7B-v0.1 (Jiang et al., 2023) (32 layers, grouped-query attention, rotary embeddings). Both use group

Table 1: Attention-only Mistral-7B at 3 bits, all four calibrate→evaluate pairings (W=WikiText-2), with the share of the uniform-versus-fp16 gap closed. Full sweep: Appendix E.

Pairing	fp16	Uniform	Adaptive	Gap closed	Δ Acc.
W → W	4.818	8.738	**5.228**	89.5%	+0.031
C4 → C4	7.492	13.506	**8.898**	76.6%	+0.014
W → C4	7.492	13.575	**8.622**	81.4%	+0.018
C4 → W	4.818	8.875	**5.428**	85.0%	+0.028

size 128, activation ordering, $\lambda_{\text{KL}} = 0.1$ unless ablated (except the MLP study, where it is 0 by construction), and 10 Hutchinson probes per layer. The **attention-only** study quantizes the QKV projection: on GPT-2 it sweeps four criteria over seven budgets with and without fine-tuning, calibrating on WikiText-2 (Merity et al., 2017) or C4 (Raffel et al., 2020); on Mistral-7B it covers all four calibrate→evaluate pairings. The **MLP-only** study quantizes GPT-2's `c_fc` and `mlp.c_proj`. The **full-model** study quantizes all seven linear modules of Mistral-7B. We evaluate GPT-2 on the full WikiText-2 test split and Mistral-7B on 32 windows of 512 tokens, so Mistral perplexities are exact within a study but not comparable to published full-split figures. All results are single-seed. Complete configurations are in Appendix C.

4 Attention-Only Quantization

At 7B scale, allocation carries the result On Mistral-7B, adaptive allocation dominates uniform GPTQ at 3 bits in all four corpus pairings, recovering 77–90% of the uniform-versus-fp16 gap and 2–3 accuracy points (Table 1). By 4 bits every method is within 1 -2% of fp16, and refinement is second-order: at 3 bits allocation is worth 3–5 perplexity points and fine-tuning at most 0.17 in either direction. The allocation transfers across corpus mismatch, and the GPT-2 trace ranking is nearly calibration-invariant (Appendix H).

A cheap criterion matches an expensive oracle Across the four GPT-2 budgets ≥ 3.5 bits, the Hutchinson criterion and the $\approx$ 73-load oracle differ by 0.17 perplexity on average (Table 8). On Mistral, second-order scoring beats a first-order Fisher proxy at 3 bits (5.268 vs. 5.302), while greedy and Pareto solvers are indistinguishable: the score matters, the solver does not.

Mixed precision is not uniformly safe, and the objective says why On GPT-2 at exactly 3.0 bits, adaptive allocation reaches 86.36 against uniform's 38.35. This follows from the objective. At an average of exactly 3 bits, an assignment with no block below 3 must place every block at 3, which is the uniform solution; the adaptive assignment differs from uniform, so at least one block sits at 2 bits, and one is enough to dominate the total (uniform 2-bit scores 5137.6). Minimizing a *sum* of $\Omega_i(b_i)$, the allocator cannot see that perplexity is convex and unbounded below. Depth re-weighting C2 halves the damage (43.93) at no cost. Mixed precision earns its keep at fractional budgets instead, beating a geometric interpolation between neighboring uniform solutions by 2.2× at 2.5 bits and 1.11× at 3.5 (Appendix E).

Refinement recovers attention behavior, not weights On GPT-2, joint fine-tuning improves 18 of 21 configurations, and does so by moving the weights *away* from float: at 4 bits ε^W rises 19% (0.174 → 0.208) while ε^A falls 33% (0.238 → 0.160), and both perplexity and accuracy improve. Every GPTQ-only arm has $\rho > 1$ and every fine-tuned arm $\rho < 1$, with the ordering $\rho_{\text{GPTQ}} > \rho_{\text{MSE-only}} > \rho_{\text{MSE+KL}}$ holding at every budget on both corpora (Appendix G). The attention *map* is what must be preserved: the KL term supplies 88% of the gain, and MSE-only refinement lowers ε^A yet still degrades perplexity.

5 Beyond Attention: the MLP

The JAB machinery is not intrinsically tied to the softmax. To test whether it generalizes beyond attention, we apply the same allocation-and-refinement pipeline to the GPT-2 MLP, quantizing `mlp.c_fc` and `mlp.c_proj` with attention kept in fp32; the MLP holds 66.6% of the non-

Table 2: GPT-2 MLP, WikiText-2 perplexity (fp32 control 24.357) at the two budgets with a uniform baseline. Full sweep: Appendix I, Table 11.

B (bits)	Uniform	JAB (C1)	Oracle (C3)
3.0	53.71	120.70	**46.15**
4.0	**26.23**	27.46	**26.23**

Table 3: GPT-2 with QKV and MLP quantized together (C4 calibration, GPTQ-only): WikiText-2 perplexity (fp32 control 24.357) and accuracy at the two budgets that separate the allocators. Full table: Appendix J.5; in-domain C4, same ranking: Appendix J.6.

	$B = 3.5$		$B = 4.5$	
Allocator	PPL	Acc	PPL	Acc
JAB (C1)	75.009	0.296	29.041	0.394
JAB+prop (C2)	67.337	0.307	28.694	**0.397**
Type-offset	**64.896**	**0.310**	**28.211**	0.395

embedding weights against 25.0% for QKV. Two structural differences follow. The two matrices are sequential rather than a jointly coupled projection, so their Hessians are collected in execution order; and there is no attention map to distill, so the KL component is absent ($\lambda_{\mathrm{KL}} = 0$) and $\mathcal{L}_{\mathrm{JAB}}$ reduces to output MSE. The experiment therefore tests which parts of JAB transfer once its attention-specific signal is removed.

The Hessian criterion does not improve over uniform allocation, whereas the oracle does At the two budgets for which a uniform baseline is available, the Hessian criterion performs worse than uniform quantization: 120.70 vs. 53.71 perplexity at 3 bits and 27.46 vs. 26.23 at 4 bits (Table 2). This failure differs qualitatively from the QKV collapse, since the 4-bit allocation contains no unit below 3 bits. The oracle, in contrast, improves on uniform at 3 bits, reaching 46.15, a 14% reduction. Thus, useful allocation structure exists in the MLP, but the Hessian criterion fails to identify it.

The oracle allocates by role, not depth At every fractional budget the oracle returns the same *flat* allocation in every block, split only by matrix role: `mlp.c_fc` at $b{+}1$ and `mlp.c_proj` at $b{-}1$ (Appendix I, Table 12). The criterion cannot find that structure, by construction. One trace over the concatenated block vector is identical for both units of a block, so the only thing that can tell them apart is the perturbation factor of Eq. (6). What the criterion sees instead is a depth gradient rising $4.5\times$, unlike the QKV trace on the same network, which peaks at block 3 with a $41\times$ spread. Sensitivity profiles are a property of the sublayer, not of the architecture alone.

6 Full-Model Quantization

We now put QKV and MLP under one shared budget, first on GPT-2 small, then on every linear module of Mistral-7B.

6.1 GPT-2: QKV and MLP Together

The fused QKV projection and both MLP matrices of every GPT-2 block now share one budget: 36 units, 91.7% of block weights, giving $2.21\times$ end-to-end compression at 4 bits against $1.66\times$ for the MLP alone. Four allocators are compared at five budgets, calibrated on C4, GPTQ-only, evaluated on WikiText-2 (fp32 control 24.357) and a disjoint C4 slice with the same ranking (Appendix J.6). Sensitivity tables are normalized per block type first, since raw JAB scores are not commensurate across types. The shared block trace was also replaced by an unbiased per-unit trace in one of the sweeps, but no improvement was obtained.

Role beats curvature once both sublayers share the budget Type-offset needs no scoring pass at all and still leads at the fractional budgets: 64.896 at 3.5 bits and 28.211 at 4.5, against JAB+prop's

Table 4: Mistral-7B, all seven modules quantized: WikiText-2 perplexity (fp16 control 6.643) and compression of the quantized weights. Budgets B are bits per parameter (Eq. (8)). **Left:** criteria at fractional budgets; Type-offset needs no scoring pass. **Right:** uniform quantization, defined only at integer budgets, and the adaptive JAB arm. Best per budget in bold.

B (bits/param)	2.5	3.5	4.5
Compression	6.40×	4.57×	3.56×
Type-offset	6624.3	32.33	**6.970**
JAB (C1)	1799.6	**8.570**	7.250
JAB-norm	**648.0**	109.6	10.77

Arm	B	Compr.	PPL
Uniform	2.0	8.00×	64779.9
Uniform	3.0	5.33×	9.126
Uniform	4.0	4.00×	6.998
JAB adaptive	4.3	3.72×	7.373

67.337 and 28.694 (Table 3). JAB's attention-defined sensitivity over-invests in QKV and underfunds the MLP, which now dominates the parameter count; only near saturation, at 6 bits, does uniform win. Two observations carry over to Mistral (Appendix J.5): ε^A stops tracking perplexity once the MLP shares the budget, and the joint STE stage raises perplexity at every budget, by 16.35× at 3 bits.

6.2 Mistral-7B: All Seven Modules Together

We now quantize every linear module of all 32 Mistral-7B layers under one budget: 224 units, 6.98B parameters, 96.4% of the model.

A parameter-weighted budget The sublayer studies had units of equal size, so per-unit and per-parameter costs coincided. Here grouped-query attention makes `k_proj` and `v_proj` (4.2M) a quarter of `q_proj` and `o_proj` (16.8M), while the three SwiGLU matrices hold 58.7M each — a 14× span, with the MLP carrying 80.8% of the weights. A per-unit cost would price those equally, so we define the budget per parameter,

$$B \;=\; \frac{\sum_i P_i\, b_i}{\sum_i P_i}, \qquad c(b_i) = P_i\, b_i, \tag{8}$$

with P_i the parameter count of unit i. On the quantized weights this makes storage $16/B$ times smaller than fp16: 13.96 GB → 3.93 GB (3.56×) at $B = 4.5$.

Type-offset: a Hessian-free role prior Motivated by the MLP oracle's role rule, Type-offset gives each unit kind κ a fixed integer offset Δ_κ and assigns

$$b_i \;=\; \mathrm{nearest}_{\mathcal{B}}\big(t^\star + \Delta_{\kappa(i)}\big)\,, \qquad t^\star = \arg\min_t \Big|\textstyle\sum_i c(b_i) - B\sum_i P_i\Big|, \tag{9}$$

followed by single-unit moves meeting the budget exactly: seven integers, no scoring pass. We set $\Delta = 0$ for q, k, v, o, $+1$ for `gate` and `up`, -1 for `down`, transposed from the GPT-2 oracle rather than tuned, and also test *JAB-norm*, which divides each trace by $\overline{\mathrm{diag}(H_i)}$.

6.3 Layer Role Outperforms Curvature

The role prior wins at a matched budget At $B = 4.5$ bits per parameter, Type-offset reaches 6.970 perplexity against 7.250 for JAB and 10.770 for JAB-norm (Table 4). All three realize the same budget to three decimals, 4.500 bits per parameter, but Type-offset computes no Hessian and runs no Hutchinson probes. The curvature information JAB pays for does not buy a better allocation than seven integers. The MLP experiment showed the same thing: there too the end-to-end oracle preferred a flat allocation set by matrix role rather than depth. That it recurs in Mistral-7B, with grouped-query attention, SwiGLU and 58× the parameters, is what makes us read it as a property of heterogeneous transformers rather than of one architecture.

Why the JAB criterion misses the role structure JAB computes one Hessian trace for the concatenated parameter vector of each decoder block, so all seven linear modules in a block receive the same trace value. The criterion can separate them only through the perturbation term in Eq. (6).

Table 5: Joint refinement of the adaptive B=4.3 allocation. Block 0 is the only block whose weights change, so refining blocks 0–3 reproduces the full-depth result exactly. Full table: Appendix J.

Refinement	Blocks refined	Blocks changed	PPL
None (GPTQ only)	–	–	**7.373**
Joint FT	0–31	1 (block 0)	239.52
Joint FT	0–3	1 (block 0)	239.52

What it does vary over is depth: the trace rises $381\times$ from block 0 to block 31. The oracle's allocation has no depth structure at all, only a role split, so JAB spends bits along the wrong axis. It distinguishes *where* a layer sits rather than lemphwhich matrix it is. A per-unit trace would remove this limitation (Appendix J). Section 6.5 removes this limitation and finds the ranking unchanged.

Type-offset is useful, but its offsets are not universally transferable Type-offset degrades sharply below 4.5 bits: at $B = 3.5$ the fixed $\Delta_{\text{down}} = -1$ offset assigns 2 bits to 30 of the 32 $\text{down}_{\text{proj}}$ matrices, giving 32.33 perplexity. The GPT-2 MLP experiment pointed to this direction, but Mistral's $\text{down}_{\text{proj}}$ receives heavier-tailed inputs, so the same offset need not apply; the floor of Section 6.5 removes the failure entirely. Each candidate offset costs one evaluation, so we report the 4.5-bit result as a floor for the role-prior family, not a tuned choice.

6.4 Block-Local Objectives Can Diverge from End-to-End Quality

A locally improved block can substantially degrade the complete model Joint refinement of the adaptive $B = 4.3$ allocation reduces the block-local objective of block 0 from 1.85 to 0.41×10^{-3}, a $4.6\times$ improvement. End-to-end perplexity increases from 7.373 for GPTQ-only to 239.52 after refinement. Refining only blocks 0–3 gives exactly the same perplexity and the same error metrics (Table 5), and in both runs block 0 is the only block whose quantized weights move at all; the other 31 return to their GPTQ grid points.

This is not a failure of the local optimization. The refinement reduces the objective it was given and passes the held-out checkpoint gate. One refined block still raises perplexity $32.5\times$.

Why block 0 is especially vulnerable Two properties coincide in this block. The adaptive allocator assigns 2 bits to six of its seven linear modules, giving a weight-space error of $\varepsilon^W = 1.015$: the quantized block is already farther from its floating-point weights than the zero matrix is, in relative Frobenius error. And the output of block 0 passes through all 31 downstream decoder blocks. Refinement is therefore working on a severely quantized representation at the point of maximum downstream leverage.

We read the failure as follows. Minimizing a block-local objective on a small calibration set can move the weights toward calibration-specific behavior while losing what the downstream network needs. The held-out gate cannot catch this, because it evaluates the same local objective rather than the end-to-end metric.

Granularity matters too. Under uniform 4-bit quantization the same refinement stage changes three blocks and raises perplexity by $1.6\times$ rather than $32.5\times$, so in these two runs the mismatch is far more severe when low-bit assignments are concentrated.

Implication for refinement The problem is not specific to $\mathcal{L}_{\text{JAB}}$. Any refinement procedure that optimizes a block-local target and validates against that same target gives no guarantee of better end-to-end quality. At minimum, refinement should track a global metric alongside a per-block flip-rate diagnostic (Appendix B.6, Eq. (15)), which is what identifies a locally accepted update with outsized downstream effect. The same block also generalizes the 2-bit pathology we prove for GPT-2 in Section 4. The additive objective equation 7 cannot see it, and it appears whenever the allocator is free to concentrate low widths on a single block.

Table 6: Mistral-7B, all seven modules, floor $b_i \geq 3$ and per-unit traces (GPTQ-only; fp16 6.643; uniform 4-bit 6.938; adaptive B=4.3 7.211). Compression is on the quantized weights. Not comparable to Table 4 (64 vs 128 Hessian batches); see Appendix J.7.

B	Compr.	JAB (C1)	JAB-norm	Type-offset
3.0	5.33×	9.713	9.713	9.713
3.5	4.57×	**7.514**	8.215	7.875
4.5	3.56×	7.158	7.592	**6.933**

6.5 A 3-Bit Floor with Per-Unit Curvature

Each diagnostic suggests a change: the additive objective cannot see the low-bit cliff, so we impose the floor $b_i \geq 3$; and the shared block trace cannot rank units within a block, so we replace it with an unbiased *per-unit* trace at no extra cost, partitioning $v^\top H v$ over each unit's coordinate range (Appendix J.7).

The floor removes the catastrophic regime No arm now exceeds 9.72, against 32.33 and 109.6 at $B = 3.5$ without it; at $B = 3.0$ the floor saturates the budget, so all four criteria return the uniform assignment and the identical 9.713. The usable operating points are Type-offset at $B = 3.5$ (7.875, 4.57×) and $B = 4.5$ (6.933, 3.56×): 14 GB reduced to 3.93 GB for 4.4% higher perplexity than fp16.

Per-unit curvature does not change the ranking The traces now discriminate strongly within a block, yet Type-offset still leads at matched $B = 4.5$, 6.933 against 7.158: q and k take 77% of the trace mass while holding 19.2% of the weights, since $\mathcal{L}_{\text{JAB}}$ is defined on the attention output and map. The limitation is the objective the curvature comes from, not the shared trace this experiment removes (Appendix J.7).

7 Conclusion

Using one loss for both reconstruction and allocation lets us analyze the two stages of mixed-precision PTQ in a single frame, and on attention it works: adaptive allocation recovers most of the uniform-to-fp16 gap at 7B. Beyond attention it does not transfer. Where units are heterogeneous, matrix role carries more signal than the curvature we computed, and per-unit resolution does not change that, since the curvature comes from an attention objective. Two lessons hold independently of JAB. An additive objective cannot see the low-bit cliff, and a floor $b_i \geq 3$ repairs it at no cost. And a certified local improvement is not evidence of a global one (Appendix K).

Both regimes matter, but not equally. Compression is the reason to quantize, and the ratio moves fastest at the low end: $16/3$ is 5.3× against 3.56× at 4.5 bits. Large models are the ones forced to live there, and 3.5 bits is where the curvature criterion is ahead of the role prior (7.51 against 7.87). The regime JAB serves is therefore the one that grows in importance with model size, although we test only to 7B.

AI use statement

In this work, we used generative AI tools (a large language model assistant) for editing and condensing the manuscript text, drafting and checking LaTeX, writing experiment, analysis and plotting code, and cross-checking the numbers reported in the text against our result files. We have not used generative AI tools to produce any experimental result: every number in the paper is computed by our code from the models' outputs, and every AI-drafted script was run and checked by the authors. We have reviewed all AI-assisted work and take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

Reproducibility Statement

Code, configurations and seeds: anonymous.4open.science/r/...-6080. All runs use seed 42 and are single-seed; full hyperparameters are in Appendix C.

References

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Deyu Cao and Samin Aref. Enhancing ultra-low-bit quantization ofnbsp;large language models through saliency-aware partial retraining. In *Modeling Decisions for Artificial Intelligence: 22nd International Conference*, pp. 354–383, 2025. doi: 10.1007/978-3-032-00891-6_28.

George B. Dantzig. Discrete-variable extremum problems. *Operations Research*, 5(2):266–288, 1957.

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.

Steven K. Esser, Jeffrey L. McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S. Modha. Learned step size quantization. In *International Conference on Learning Representations (ICLR)*, 2020.

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 (ICLR)*, 2023.

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. In *ACM/IEEE Design Automation Conference (DAC)*, 2024.

Babak Hassibi and David G. Stork. Second order derivatives for network pruning: Optimal brain surgeon. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 164–171, 1990.

Michael F. Hutchinson. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics – Simulation and Computation*, 19(2):433–450, 1990.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Changhun Lee, Jungyu Jin, Taesu Kim, Hyungjun Kim, and Eunhyeok Park. Owq: outlier-aware weight quantization for efficient fine-tuning and inference of large language models. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence*, 2024. doi: 10.1609/aaai.v38i12.29237.

Deokjae Lee, Sihun Chu, and Hyun Oh Song. Q-strata: Hierarchical bit allocation for mixed-precision quantization of mixture-of-experts llms, 2026. URL https://arxiv.org/abs/2608.30564.

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

Jing Liu, Toshiaki Koike-Akino, Ye Wang, Hassan Mansour, and Matthew Brand. Awp: Activation-aware weight pruning and quantization with projected gradient descent, 2025. URL https://arxiv.org/abs/2506.10205.

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

Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Barak A. Pearlmutter. Fast exact multiplication by the Hessian. *Neural Computation*, 6(1):147–160, 1994.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019.

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.

Mariam Rakka, Marios Fournarakis, Olga Krestinskaya, Jinane Bazzi, Khaled Salama, Fadi Kurdahi, Ahmed Eltawil, and Mohammed Fouda. Mixed-precision quantization for language models: Techniques and prospects. In *ACM Comput. Surv.*, pp. 0360–0300, 2026.

Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Peng Gao, Yu Qiao, and Ping Luo. OmniQuant: Omnidirectionally calibrated quantization for large language models. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2308.13137.

Prabhakant Sinha and Andris A. Zoltners. The multiple-choice knapsack problem. *Operations Research*, 27(3):503–515, 1979.

Albert Tseng, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa. QuIP#: Even better LLM quantization with hadamard incoherence and lattice codebooks. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, 2024. URL https://arxiv.org/abs/2402.04396.

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)*, pp. 8612–8620, 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.