DEPTHQUANT: DEPTH ERROR PROPAGATION FOR MIXED-PRECISION BIT ALLOCATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Mixed-precision quantization assigns a bit width to each unit of a large language model under a fixed bit budget, but most current methods decide the allocation from single-layer information and fall well short of full precision. This may be partly because single-layer methods miss information from deeper layers: similar quantization errors can produce different responses at the model output. To address this, we propose DepthQuant, a mixed-precision quantization method that uses depth error propagation as the signal that drives the bit allocation. By measuring the response each precision change induces one layer deeper and how much it changes the model output, DepthQuant spends the bit budget where more precision reduces that measured change the most. Built on this principle, DepthQuant introduces a cohesive multi-scale design: short-depth response scoring ranks precision changes by the response they induce one layer deeper, full-depth allocation then adjusts the resulting allocation by the error's effect on the model output, and a final step sets each row's range from information in its own layer. Experiments show that DepthQuant improves on the compared baselines in most evaluated settings and in every reported average-2-bit setting, reducing perplexity by up to 40.6% at 2.5-bit average precision relative to the best-performing baseline and raising zero-shot accuracy by up to 11.6% at 2-bit average precision, where these methods degrade the most.

1 INTRODUCTION

Large language models (LLMs) now handle a wide range of natural language tasks (Touvron et al., 2023; Grattafiori et al., 2024; Yang et al., 2025), from dialogue to code generation, and their capabilities improve with scale. Scaling, however, drives up memory and inference costs, which are increasingly hard to sustain. Model compression has therefore become a standard tool for reducing these costs while retaining as much accuracy as possible, and weight-only quantization is among the most widely adopted choices (Frantar et al., 2023; Lin et al., 2024).

Quantization methods differ mainly in when they reduce precision and how finely they do so. Post-training quantization (PTQ) stores the weights of a trained model at low precision, using a small calibration set instead of retraining; a uniform bit width is the simplest such choice, and in practice it is often accurate enough (Frantar et al., 2023; Lin et al., 2024). As the budget shrinks, however, one bit does not cost the same everywhere: a uniform choice either spends bits on weights that are robust to quantization, or quantizes sensitive ones as coarsely as the rest. As a more principled solution, mixed-precision quantization (MPQ) instead assigns a bit width per unit, under a constraint on the average precision. Depending on the method, the unit is a layer (Dong et al., 2020; Ma et al., 2023; Lee et al., 2025), a group of columns (Huang et al., 2025), or an output channel (Zheng et al., 2026; Chen et al., 2026). Most MPQ methods follow the same two steps: score how much each unit is harmed by quantization, then allocate bits under the budget.

However, in most of these methods the score is computed where the error is introduced. SliM-LLM (Huang et al., 2025) ranks quantization groups within each layer by an activation-aware salience score, and reuses that score to weight the quantizer calibration. CMPQ (Chen et al., 2026) sets each output channel's precision from activation statistics, so its criterion stays confined to the channel. SignRoundV2 (Cheng et al., 2026) selects layer-wise widths by dynamic programming over a layer-

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.

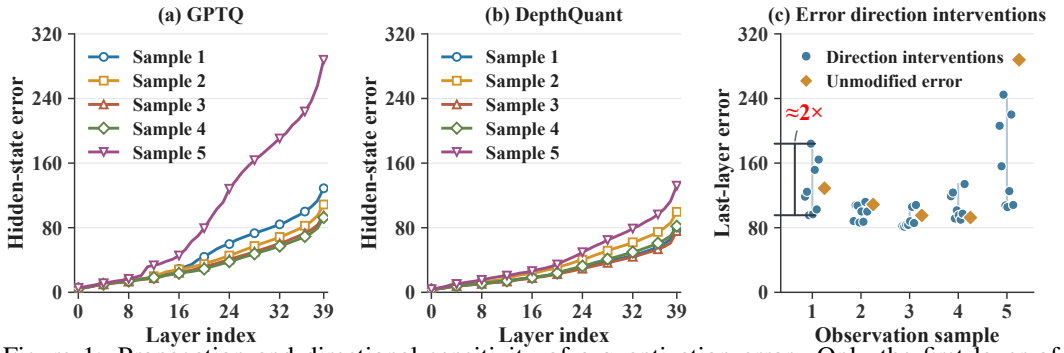


Figure 1: Propagation and directional sensitivity of a quantization error. Only the first layer of Llama-2-13B is quantized, at a requested average of 3 bit. (a) GPTQ and (b) DepthQuant share one backend and the same five calibration samples; the vertical axis is the Frobenius norm of the hidden-state error against the full-precision model. (c) flips the channel signs of the initial error at approximately fixed norm and reports the last-layer error before the final normalization.

local criterion that weighs the quantization-induced reconstruction error by the task-loss gradient at that layer. Our score observes that error at an explicit intermediate point.

However, the assumption behind these criteria shows its limits at the model output: errors of similar magnitude within a layer do not necessarily have similar effects. To visualize this, we quantize only the first layer of Llama-2-13B at a requested average of 3 bit with the GPTQ backend, keep every later layer in full precision, and track the hidden-state error at each layer as the Frobenius norm of its deviation from the full-precision model, for the same five calibration samples. As shown in Figure 1a, the five samples exhibit comparable errors immediately after quantization, yet these errors diverge substantially as they propagate through the network, with their relative ordering also changing across depth. This shows that local reconstruction error can be a poor proxy for the error after propagation. To further isolate the source of this discrepancy, we flip the channel-wise signs of the initial quantization error while keeping its Frobenius norm nearly unchanged, and we repeat the measurement for eight such interventions per sample; as shown in Figure 1c, this changes the last-layer error before the final normalization by nearly $2\times$, demonstrating that errors with similar magnitudes within a layer can produce significantly different propagated effects. These observations expose a limitation of conventional layer-wise objectives: they evaluate quantization error where it is introduced, before it interacts with the following network, and therefore fail to capture how seemingly similar local errors can lead to very different errors at the model output.

Motivated by these observations, we propose DepthQuant, which allocates precision according to how quantization errors propagate through network depth rather than how large they are where they are introduced. DepthQuant first performs Short-Depth Response Scoring: for each candidate precision change, we inject its induced perturbation at the corresponding module output and measure the response after propagation through the immediately following network. This captures how the network transforms the quantization error while avoiding the prohibitive cost of propagating every candidate through the entire model. The resulting scores define one allocation path per layer. Under the same GPTQ backend, this path lowers the later-layer errors of the same five samples and narrows their spread (Figure 1b). Furthermore, DepthQuant performs Full-Depth Allocation: it evaluates the resulting budget states by their effect on the final output distribution and reallocates bits across layers under exactly the same total parameter-bit budget. Finally, with the allocation fixed, we refine each output row's quantization range using local reconstruction. Together, Short-Depth Response Scoring and Full-Depth Allocation progressively extend the observation depth from the immediate propagated response to the final model output, allowing the bit allocation to reflect how quantization errors actually evolve through the network.

We evaluate on eight models at three requested average precisions that include a 2-bit case. At a requested average of 2.5 bit it lowers perplexity by up to 40.6% over the strongest baseline, and in the 2-bit case it improves downstream tasks by up to 11.6 points over the strongest compressed baseline. It comes out ahead of uniform quantization, of recent within-layer allocation methods (Huang et al., 2025; Cheng et al., 2026), and of a method that searches bit-width configurations from the final output (Lee et al., 2025). Ablations confirm that each of the three stages contributes, and DepthQuant leads the compared baselines in most evaluated settings and in every reported average-2-bit setting. The details are reported in Section 4.

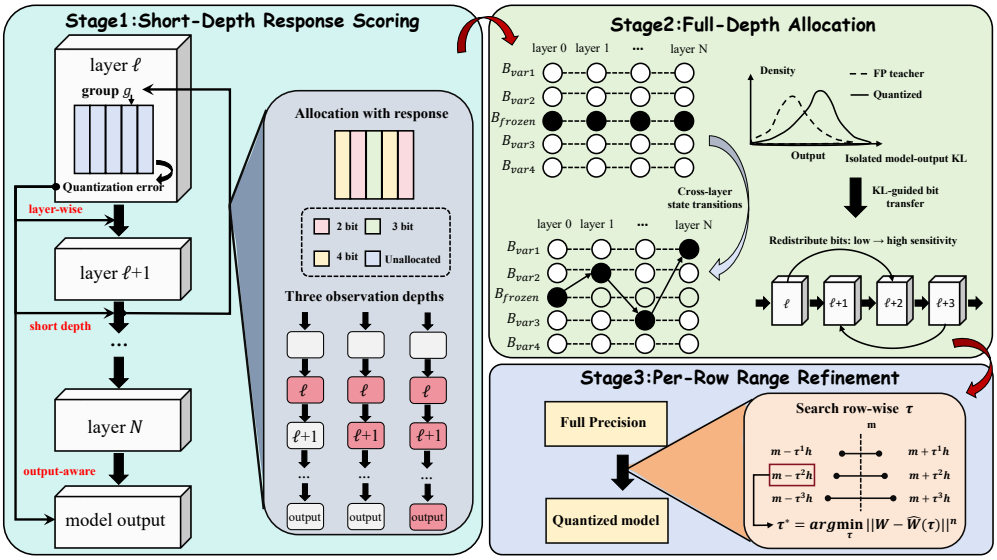


Figure 2: DepthQuant in three stages. (1) **Short-Depth Response Scoring** ranks candidate precision changes by the response they induce one layer deeper and freezes one allocation path per layer; (2) **Full-Depth Allocation** prices those paths' reachable budget states at the model output and moves bits between layers under the same parameter-bit budget; (3) **per-row range refinement** sets each assigned output row's range from its own weights.

2 Related Work

Weight-only quantization. Weight-only post-training quantization stores pretrained weights in low precision, keeps activations in floating point, and needs no retraining. GPTQ (Frantar et al., 2023) quantizes weight columns sequentially and uses approximate inverse-Hessian information to update the remaining unquantized weights, compensating for the resulting layer-output error. AWQ (Lin et al., 2024) rescales channels to protect activation-salient weights from rounding; HQQ (Badri & Shaji, 2023) instead optimizes the weight-reconstruction error without calibration data, and a further line learns the rounding rule or clipping thresholds rather than computing them (Nagel et al., 2020; Shao et al., 2024). These methods allocate precision uniformly and ignore how tolerance to quantization differs across weights. Their reconstruction procedures improve the quantizer at a given bit width, rather than determine where additional precision is most useful.

Mixed-precision quantization. Mixed-precision quantization assigns a bit width to each unit. A common approach scores each unit from its own weights or activations: CMPQ (Chen et al., 2026) uses activation statistics, SliM-LLM (Huang et al., 2025) combines salience-guided group allocation, local output matching, and salience-aware calibration, APTQ (Guan et al., 2024) uses attention-output-aware second-order sensitivity, and SignRoundV2 (Cheng et al., 2026) scores each layer by the gradient weighted by its quantization deviation. Others score units from the loss: HAWQ-V2 (Dong et al., 2020) averages Hessian traces of it and MixLLM (Zheng et al., 2026) measures output-channel salience with respect to it.CLADO (Deng et al., 2025) estimates pairwise cross-layer quantization costs through forward evaluations on calibration data and incorporates them into an integer quadratic program for joint bit assignment. GAMMA (Yao et al., 2026) learns bit-selection preferences under a global budget. The sensitivity criterion determines which units receive more precision under the available budget.

Beyond within-layer signals. Some allocation methods already go beyond single-layer information. MixQuant (Misra et al., 2026) estimates a layer's distortion under quantized upstream contexts, and AMQ (Lee et al., 2025) searches layer-wise configurations through a quantization proxy and a quality predictor. MixFrag (Opi et al., 2026) allocates precision for ViT from the output-distribution change caused by quantizing one component. These methods account for quantized input contexts or changes at the model output, while the criteria above read the signal from a unit's own weights or activations or from the gradient of the final loss. In this work, DepthQuant makes the depth-dependent response the scoring signal, coupling short-depth sensitivity with full-depth output calibration.

3 METHOD

Mixed-precision post-training quantization must decide, under a fixed parameter-bit budget, how many bits each unit receives. Different units tolerate quantization very differently, so a uniform precision cannot spend the budget where it is most needed; the allocation must be driven by a faithful measurement of each unit's response to quantization. In this section, we propose **DepthQuant**, which decides the allocation from how quantization errors propagate through depth. Section 3.1 fixes the quantization and budget language used throughout. Section 3.2 presents Short-Depth Response Scoring (Stage 1), which ranks precision changes by the response they induce one layer deeper; Section 3.3 presents Full-Depth Allocation (Stage 2), which extends the observation to the model output by pricing a few budget states per layer; Section 3.4 presents the per-row range refinement (Stage 3), which adapts the quantization range of each output row once the allocation is fixed; and Section 3.5 gives the complete procedure. Figure 2 summarizes the three stages.

3.1 PRELIMINARIES: QUANTIZATION AND MIXED-PRECISION BUDGETS

Consider a pretrained large language model as a stack of L transformer layers indexed by $\mathcal{L} = \{0, \dots, L-1\}$. Our scope is weight-only post-training quantization: the weights of the linear modules inside them are quantized after training, while embeddings, normalization parameters, and the output head remain in full precision. We use the layer as the basic measure of how far a quantization error propagates.

We first recall the affine uniform quantizer used at a fixed precision. A real weight w is mapped to one of 2^b integer codes by a scale $s > 0$ and an integer zero-point z,

$$q = \mathrm{clip}\big(\lfloor w/s \rceil + z,\ 0,\ 2^b - 1\big), \qquad \hat{w} = s\,(q - z), \tag{1}$$

where $\lfloor \cdot \rceil$ denotes rounding and clip bounds the code to the representable integer range. For a set of values with element range $[w_{\min}, w_{\max}]$, the standard asymmetric min–max choices of the scale and zero point are

$$s = \frac{w_{\max} - w_{\min}}{2^b - 1}, \qquad z = -\left\lfloor \frac{w_{\min}}{s} \right\rceil, \tag{2}$$

so that the interval endpoints determine the reconstruction grid up to the integer rounding of z (at $b = 1$ the grid has two levels, one of which is zero).

Mixed precision decides how many bits each unit receives. The allocation operates on the linear modules of each transformer block, i.e., the attention projections and the feed-forward matrices (for example the $q/k/v/o$ projections and the $up/gate/down$ matrices). For a target module a of layer ℓ, we split its input columns into contiguous column blocks $\mathcal{I}_{\ell,a,j}$ of at most the declared group size. A *precision unit* $g = (a, j)$ is one such column block together with all of its output rows; every weight in the unit shares one bit width. Writing $W_{\ell,a} \in \mathbb{R}^{m \times n}$ for the module weight, the unit's weights and its parameter count are

$$W_{\ell,g} = W_{\ell,a}[:, \mathcal{I}_{\ell,a,j}], \qquad P_{\ell,g} = m\,|\mathcal{I}_{\ell,a,j}|, \tag{3}$$

so $P_{\ell,g}$ counts the number of target weights that the assigned precision applies to, and the column block is the allocation granularity. A mixed-precision map $\mathtt{m}$ assigns to every unit a bit width $b_{\ell,g} \in \mathcal{B}$, where $\mathcal{B}$ is the declared admissible set of integer bit widths for the setting, and has target-weight parameter-bit count

$$\mathsf{B}(\mathtt{m}) = \sum_{\ell,g} P_{\ell,g}\, b_{\ell,g}. \tag{4}$$

For a requested average precision $\bar{b}_{\mathrm{req}}$ over the declared decoder-block linear weights, the nominal parameter-bit budget is $B_{\mathrm{req}} = \bar{b}_{\mathrm{req}} \sum_{\ell,g} P_{\ell,g}$.

3.2 SHORT-DEPTH RESPONSE SCORING

A candidate's local reconstruction error is measured where the error is introduced, so it ignores how the following network interacts with that error. Short-Depth Response Scoring instead scores every candidate by the propagated response of its perturbation, observed at the next layer, and freezes the resulting ranking into one allocation path per layer.

Measuring candidates by their propagated response. A candidate is the concrete intervention of quantizing unit g to b bits. Over N fixed calibration inputs, with $X_{\ell,g}^{(n)}$ the input activations restricted to precision unit (ℓ, g), the candidate perturbation is

$$u_{\ell,g,b}^{(n)} = X_{\ell,g}^{(n)} \big(Q_b^{\mathrm{probe}}(W_{\ell,g}) - W_{\ell,g}\big)^\top, \tag{5}$$

where Q_b^{probe} is the b-bit quantizer that instantiates the intervention. All tensors are vectorized when used in Jacobians or norms. The perturbation is inserted at the module output and replayed through the actual intervening path of the network.

We score each candidate by propagating its perturbation to a fixed observation point e; the network segment between the module output where the perturbation enters the computation and e is the measurement horizon. The horizon is fixed: it covers the remainder of the source layer and then the next layer, which directly receives the perturbation and interacts with it, so the measurement spans one complete Transformer unit and the observation point e is the output of that layer; Appendix D defines the horizon depth and reports its diagnostic. For the last layer, where no next layer exists, the horizon is shortened and continued through the final normalization and language-model head, so that its endpoint is the model output. The propagated response is measured in first order: for each calibration input we run the horizon once in full precision, keep the resulting activations as the linearization point, and let $J_{\ell,g\to e}^{(n)}$ denote the Jacobian of the horizon-endpoint output with respect to the perturbed module's output at this point. The propagated perturbation of the candidate is

$$v_{\ell,g,b}^{(n)} = J_{\ell,g\to e}^{(n)}\, u_{\ell,g,b}^{(n)}, \tag{6}$$

The response cost of candidate (ℓ, g, b) is the mean squared norm of its propagated response,

$$C_{\ell,g}(b) = \frac{1}{N} \sum_{n=1}^{N} \big\| v_{\ell,g,b}^{(n)} \big\|_2^2, \tag{7}$$

which is the quantity this stage ranks. Computing each $C_{\ell,g}(b)$ exactly requires propagating each candidate separately, which is prohibitive at the scale of all candidate precision changes. Instead we draw K independent random ± 1 probe vectors $\xi_{n,k}$ and estimate

$$\widehat{C}_{\ell,g}(b) = \frac{1}{NK} \sum_{n=1}^{N} \sum_{k=1}^{K} \big(\xi_{n,k}^\top v_{\ell,g,b}^{(n)}\big)^2, \tag{8}$$

a standard Hutchinson-type randomized quadratic-form estimator (Hutchinson, 1989) evaluated without materializing any Jacobian: the scalar $\xi^\top J u$ is computed as one vector–Jacobian product (VJP) $(\xi^\top J)u$. The estimate is unbiased for the linearized response energy at the stated linearization points, as the following lemma states; its proof appears in Appendix B.

Lemma 1 (Unbiased propagated-response estimator)**.** *For fixed model weights, calibration examples, candidate perturbations, and linearization points, if the $\xi_{n,k}$ are independent Rademacher vectors, then*

$$\mathbb{E}_\xi \Big[\widehat{C}_{\ell,g}(b) \Big] = C_{\ell,g}(b). \tag{9}$$

The guarantee is per candidate at the linearization points fixed above, and the probe count is part of the measured setting (Table 15).On sampled candidates the ranking induced by the score agrees with the measured response, and the score tracks the model-output divergence of those candidates more closely than the local reconstruction error does (Appendix D).

Constructing the allocation path. For each layer we build the path from its lowest admissible map, repeatedly applying the one-step precision change with the largest estimated reduction in response cost per added parameter bit. A unit can therefore appear more than once, for example in $2 \to 3$ and then $3 \to 4$ changes. Every path state is a concrete mixed-precision map with an exact integer parameter-bit count. The greedy construction yields an allocation path π_ℓ: each later state contains all changes of earlier states, and the allocation order is fixed once and does not depend on the target budget. Once the path is fixed, the Stage-1 state is the reachable state whose integer count is nearest the requested per-layer budget; we write its map and count as $\mathfrak{m}_\ell^{\mathrm{st}}$ and q_ℓ^{st}, and the parameter-bit count of the Stage-1 map as $B_1 = \sum_\ell q_\ell^{\mathrm{st}}$.

3.3 FULL-DEPTH ALLOCATION

Short-Depth Response Scoring measures each precision change at the layer that directly receives the error it creates, so it captures how strongly that error propagates once it has interacted with the following network. It remains a measurement over a fixed horizon, however, and does not observe how a layer's overall assignment affects the model output. Full-Depth Allocation completes the picture by extending the observation to the model output: it prices, for each layer, the budget states admitted by the cap of Eq. (10) against the full-precision model, and then moves bits between layers while keeping the parameter-bit count of the map unchanged.

For layer ℓ, we index the admissible states of its frozen allocation path π_ℓ by s: the state is admitted when its requested per-layer average precision $\widetilde{b}_{\ell,s}$ lies within the cap c of the target $\bar{b}_{\mathrm{req}}$,

$$|\widetilde{b}_{\ell,s} - \bar{b}_{\mathrm{req}}| \le c, \tag{10}$$

that is, within the amount by which the average precision requested of a layer may depart from the target, together with the Stage-1 state. Every such state fixes one concrete mixed-precision map $\mathfrak{m}_{\ell,s}$ and its exact integer parameter-bit count $q_{\ell,s}$.

Every budget state is priced on its own: we quantize only layer ℓ according to that state, keep the others in full precision, and measure how far the output distribution moves from the full-precision one. With $z_{\mathrm{FP},t}(x)$ and $z^{\mathrm{iso}}_{\ell,s,t}(x)$ denoting the reference and state logits at a valid next-token position t, the response cost $k_{\ell,s}$ is the empirical KL divergence between the two softmax distributions,

$$k_{\ell,s} = \frac{1}{|\mathcal{T}_{\mathrm{KL}}|} \sum_{(x,t)\in\mathcal{T}_{\mathrm{KL}}} \mathrm{KL}\big(\mathrm{softmax}(z_{\mathrm{FP},t}(x)) \,\big\|\, \mathrm{softmax}(z^{\mathrm{iso}}_{\ell,s,t}(x))\big)\,, \tag{11}$$

evaluated on calibration slices $\mathcal{T}_{\mathrm{KL}}$ that are disjoint from the scoring set of the first stage, and the allocation compares states by their sum over the frozen path domain. On preselected layer pairs the joint divergence follows this sum to within 0.34% (one adjacent middle pair, 3.27%; Appendix D).

A layer stays at its Stage-1 state when the cap window admits no other state on its path; let $\mathcal{L}_{\mathrm{frz}}$ be those layers and B_{frozen} the count their Stage-1 states contribute. The remaining part $B_{\mathrm{var}} = B_1 - B_{\mathrm{frozen}}$ must be covered by the states selected for the other layers. We choose one state s_ℓ per layer in $\mathcal{L} \setminus \mathcal{L}_{\mathrm{frz}}$ to minimize the total cost,

$$\min_{\{s_\ell\}} \sum_{\ell\notin\mathcal{L}_{\mathrm{frz}}} k_{\ell,s_\ell} \quad \text{subject to} \quad \sum_{\ell\notin\mathcal{L}_{\mathrm{frz}}} q_{\ell,s_\ell} = B_{\mathrm{var}}, \tag{12}$$

a multiple-choice problem solved exactly for the costs of Eq. (11) by dynamic programming over the states that the frozen Stage-1 paths make reachable. The Stage-1 state is admissible for every non-frozen layer and every $q_{\ell,s}$ is an integer, so a solution always exists and never changes the total count. The reported results in Section 4 are measured on the jointly quantized models.

3.4 PER-ROW RANGE REFINEMENT

With the allocation fixed, the per-row range refinement searches the quantization parameters of each assigned output row against local reconstruction, without changing the bit map. For an output row r with weight range $[w_{\min}, w_{\max}]$, midpoint m_r and half-range h, the candidate range is

$$[\, m_r - \tau h,\ m_r + \tau h \,], \qquad \tau \in \mathcal{T}, \tag{13}$$

where $\mathcal{T}$ is the fixed grid $\{0.50, 0.525, \ldots, 1.10\}$; the factor τ scales how much of the row's weight range the grid covers. Each candidate range determines the row's affine grid through the same rule as Eq. (2), so its resolution and the clipping of its extreme weights follow from τ; for each row we take the factor that minimizes its own weight-reconstruction error, searched inside the quantization pass (Appendix E). For low-bit rows with extremely sparse codebooks, this concentrates codes around the dominant weight mass rather than extreme values. Which classes are searched follows the reported configuration: W2/W3/W4 at 2.5/3.0 with candidate bits $\{2, 3, 4\}$, and W1/W2 at 2 bit with $\{1, 2, 3\}$, where W3 keeps the plain asymmetric fit it receives during quantization.

3.5 FULL PROCEDURE

Algorithm 1 in Appendix A summarizes the complete procedure: Stage 1 scores each candidate by its response and freezes one allocation path per layer, Stage 2 prices the reachable budget states at the model output under the frozen parameter-bit count, and Stage 3 sets the range of each assigned output row. The Stage-2 pricing and the deployment share the GPTQ backend family, while the perturbation $u_{\ell,g,b}$ of each candidate is produced with an asymmetric min–max RTN quantizer. At 2 bit Stage 3 covers W1/W2 and at 2.5/3.0 the active W2/W3/W4 classes, each output row choosing its factor on the fixed τ grid by local weight-reconstruction error.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

We evaluate weight-only post-training quantization with group size 128, candidate bits $\{2, 3, 4\}$ (main) and $\{1, 2, 3\}$ (extreme low-bit), and $K = 8$ shared probes. Calibration uses 128 samples of 2048 tokens from the training split of WikiText-2 (Merity et al., 2017). Candidates use asymmetric min–max RTN without sequential compensation; Full-Depth Allocation prices budget states with GPTQ. For every model and precision, Full-Depth Allocation uses cap $c = 0.10$, preserves the Stage-1 parameter-bit count exactly, and prices reachable layer states on disjoint WikiText-2 sequences (Appendix E). Per-row refinement searches the fixed τ grid.

Models and Evaluation. We evaluate on eight models: Qwen2.5-7B, Qwen2.5-14B, Qwen3-4B, Qwen3-8B, Qwen3-32B (Yang et al., 2025), Llama-3.1-8B (Grattafiori et al., 2024), Llama-2-13B (Touvron et al., 2023), and Gemma-2-9B (Gemma Team et al., 2024). All evaluations are zero-shot with the LM-Evaluation-Harness (Gao et al., 2021) under the reporting convention of SignRoundV2 (Cheng et al., 2026), and perplexity is measured on the standard evaluation splits of WikiText-2 and C4 (Raffel et al., 2020). We compare against RTN, GPTQ (Frantar et al., 2023), SliM (Huang et al., 2025), AMQ (Lee et al., 2025), and SignRoundV2 (Cheng et al., 2026); our average-bit labels count target-weight bits and exclude the per-group scale and zero-point overhead, and AMQ is compared at its native effective budgets of 2.75 and 3.25 bit.

4.2 MAIN RESULTS

Tables 1, 2 and 3 report results from 2.0 to 3.0 bit. DepthQuant attains the lowest or second-lowest perplexity on most evaluations, lowering WikiText-2 perplexity by up to 40.6% on Qwen3-4B at 2.5 bit relative to the strongest baseline and by 23.2% on Llama-3.1-8B. The same picture holds for downstream accuracy: DepthQuant ranks first in 13 and second in one of the 16 model–budget settings, improving the six-task average by 2.96 points on Qwen2.5-14B at 2.5 bit and by 1.02 points on Qwen3-32B at 3 bit (1.13 points from its FP16 reference).

We further test the extreme low-bit regime at 2 bit over the candidate set $\{1, 2, 3\}$ and refine the ranges of W1/W2 against uniform RTN, GPTQ, and SliM's native low-bit configuration. SignRoundV2 is compared at the mixed-precision budgets, where its candidate set matches ours: its configuration at this budget uses a single bit width rather than one-bit candidates. Table 4 shows that DepthQuant attains the lowest perplexity on both WikiText-2 and C4 for every model; rela-

Table 1: **Low-bit per-task zero-shot results for Qwen3-32B.**

Avg. bit	Method	Zero-shot accuracy (%)						
		ARC-C	ARC-E	Hella.	PIQA	Wino.	BoolQ	Avg.
16	FP reference	60.84	82.91	82.65	81.77	73.01	86.39	77.93
2	RTN	26.54	30.51	30.35	53.32	50.67	39.63	38.50
2	GPTQ	27.05	32.58	40.44	55.44	51.07	52.72	43.22
2	SliM	35.41	49.54	61.76	67.08	60.77	72.81	57.90
2	DepthQuant	**48.89**	**71.63**	**70.04**	**74.10**	**66.30**	**85.96**	**69.49**
2.5	SignRoundV2	54.78	78.54	72.98	77.53	**71.67**	**87.06**	73.76
2.5	AMQ	54.44	**78.62**	74.06	77.64	65.27	74.65	70.78
2.5	DepthQuant	**57.42**	78.28	**78.42**	**78.56**	71.27	86.42	**75.06**

Table 2: **WikiText-2 and C4 perplexity (↓) for the main mixed-precision results.**

Bit	Method	Qwen3-4B		Qwen3-8B		Qwen3-32B		Qwen2.5-7B	
		Wiki	C4	Wiki	C4	Wiki	C4	Wiki	C4
16	FP reference	13.64	19.83	9.72	15.42	7.61	12.45	6.85	11.88
2.5	SignRoundV2(Cheng et al., 2026)	33.30	35.27	21.07	24.04	13.19	21.12	11.10	17.18
	AMQ(Lee et al., 2025)	27.10	34.93	16.18	23.27	10.56	16.02	10.10	16.02
	DepthQuant	**16.10**	**25.27**	**12.13**	**20.20**	**8.94**	**15.40**	**8.31**	**15.53**
3	RTN	22.51	33.23	13.50	20.82	24.30	34.81	12.09	18.40
	GPTQ(Frantar et al., 2023)	15.42	22.72	11.03	17.72	8.93	14.94	7.83	13.66
	SliM(Huang et al., 2025)	15.52	22.42	11.50	18.38	8.43	13.90	7.67	13.44
	SignRoundV2(Cheng et al., 2026)	19.43	26.82	13.11	19.68	10.42	15.93	9.30	15.36
	AMQ(Lee et al., 2025)	17.40	24.18	11.98	18.05	8.79	13.87	7.95	13.22
	DepthQuant	**14.50**	**21.54**	**10.64**	**17.09**	**8.19**	**13.68**	**7.45**	**13.10**
Bit	Method	Qwen2.5-14B		Llama3.1-8B		Llama2-13B		Gemma2-9B	
		Wiki	C4	Wiki	C4	Wiki	C4	Wiki	C4
16	FP reference	5.29	10.35	6.24	9.54	4.88	6.73	6.80	11.32
2.5	SignRoundV2(Cheng et al., 2026)	9.72	14.88	12.18	18.02	6.70	9.17	9.36	14.41
	AMQ(Lee et al., 2025)	8.46	13.87	11.69	17.79	6.17	**8.48**	9.60	15.22
	DepthQuant	**7.07**	**12.97**	**8.98**	**16.30**	**5.78**	8.73	**8.06**	**14.21**
3	RTN	7.63	12.97	10.87	17.05	5.52	7.58	8.68	13.74
	GPTQ(Frantar et al., 2023)	6.69	12.03	14.56	15.67	5.33	7.58	7.49	12.73
	SliM(Huang et al., 2025)	6.45	11.80	7.58	13.27	5.26	7.51	7.39	12.58
	SignRoundV2(Cheng et al., 2026)	8.57	13.65	10.23	15.70	5.51	7.55	7.93	12.66
	AMQ(Lee et al., 2025)	6.60	11.50	7.98	**12.57**	5.34	**7.30**	7.88	12.74
	DepthQuant	**6.22**	**11.35**	**7.33**	13.01	**5.23**	7.43	**7.24**	**12.28**

Notes. The FP reference is the unquantized model and is excluded from ranking. Bit denotes the average bit. AMQ is reported at its native effective budgets of 2.75 and 3.25 bit. Bold and underlined mark the best and second-best values per model and budget.

tive to the stronger mixed-precision baseline SliM it lowers WikiText-2 perplexity by 29.0% on Qwen3-8B and C4 perplexity by 34.8% on Qwen2.5-7B. The downstream results are consistent (Table 5): DepthQuant attains the highest six-task average on every reported model, improving over the strongest compressed baseline by 4.57–11.59 points; on Qwen3-32B it rises from 57.90% to 69.49%, retaining 89.2% of its FP16 average, and a gap to full precision remains at this budget.

Table 3: **Main mixed-precision results**: zero-shot average accuracy (%).

Avg. bit	Method	Qwen3			Qwen2.5		Llama		Gemma
		4B	8B	32B	7B	14B	3.1-8B	2-13B	2-9B
16	FP reference	71.22	74.06	77.93	74.10	77.25	75.91	73.30	79.21
2.5	SignRoundV2	60.43	66.07	73.76	67.20	70.86	**66.85**	69.12	72.51
	AMQ	58.03	64.70	70.78	67.50	71.31	66.80	**69.71**	70.64
	DepthQuant	**64.86**	**66.88**	**75.06**	**69.52**	**74.27**	66.73	68.83	**73.22**
3	RTN	58.86	66.57	68.97	67.00	73.50	67.11	71.03	75.40
	GPTQ	63.13	69.51	73.98	70.34	75.28	66.59	71.62	76.90
	SliM	64.95	67.52	75.71	71.62	75.86	67.07	70.55	76.71
	SignRoundV2	65.52	71.64	75.78	71.65	72.65	70.82	71.75	76.65
	AMQ	64.59	70.51	75.17	70.92	76.39	71.96	**72.12**	76.08
	DepthQuant	**67.37**	**71.95**	**76.80**	**72.98**	**76.48**	**72.54**	72.10	**77.04**

Notes. Notation follows Table 2. Bold and underlined mark the best and second-best values per model and budget; per-task values are in Appendix C.

4.3 Ablation studies

Table 6 adds Full-Depth Allocation and per-row refinement to Short-Depth Response Scoring on Qwen3-8B, Qwen3-32B, and Llama-2-13B at 3 bit. The complete method leads each block, raising six-task accuracy by 2.16, 3.54, and 0.84 points. Appendix D then varies the allocation settings one at a time: moving the probe count K and the cap c changes the six-task average by at most 1.9 and 1.4 points and the C4 perplexity by at most 0.69 (0.03 on Qwen2.5-7B), so the reported $K = 8$ and

Table 4: **WikiText-2 and C4 perplexity (↓) at 2-bit average precision.**

Method	Qwen3-4B		Qwen3-8B		Qwen3-32B		Qwen2.5-7B	
	Wiki	C4	Wiki	C4	Wiki	C4	Wiki	C4
RTN	2.51e5	2.65e5	1.05e6	1.08e6	1.13e5	8.35e4	5.04e5	5.21e5
GPTQ	150.03	471.25	59.03	120.75	29.36	69.99	28.63	122.57
SliM	<u>29.37</u>	<u>55.89</u>	<u>26.28</u>	<u>56.07</u>	<u>13.81</u>	<u>26.97</u>	<u>14.84</u>	<u>48.81</u>
DepthQuant	**21.72**	**37.60**	**18.67**	**40.39**	**12.04**	**23.48**	**12.69**	**31.83**
Method	Qwen2.5-14B		Llama3.1-8B		Llama2-13B		Gemma2-9B	
	Wiki	C4	Wiki	C4	Wiki	C4	Wiki	C4
RTN	8.12e4	8.98e4	7.52e5	2.96e5	122.35	131.75	1.69e4	6.19e3
GPTQ	26.46	77.49	275.50	8.35e3	14.04	26.23	20.65	60.39
SliM	<u>13.63</u>	<u>25.87</u>	<u>30.12</u>	<u>171.87</u>	<u>9.26</u>	<u>18.60</u>	<u>13.62</u>	<u>31.20</u>
DepthQuant	**11.32**	**22.69**	**26.18**	**124.92**	**8.17**	**13.84**	**12.48**	**26.51**

Notes. RTN/GPTQ use uniform 2-bit weights; SliM uses its native 2-bit configuration, and DepthQuant refines low-bit ranges. AMQ and SignRoundV2 are compared at mixed-precision budgets. Values $\geq 10^3$ use exponent notation; bold/underlined indicate best/second-best.

Table 5: **Zero-shot average accuracy (%) at 2-bit average precision.**

Method	Qwen3-4B	Qwen3-8B	Qwen3-32B	Qwen2.5-7B	Qwen2.5-14B	Llama3.1-8B	Llama2-13B	Gemma2-9B
RTN	37.11	37.51	38.50	38.33	37.70	35.80	39.36	40.33
GPTQ	37.40	37.47	43.22	41.29	42.57	<u>36.58</u>	39.50	48.33
SliM	<u>44.10</u>	<u>46.05</u>	<u>57.90</u>	<u>48.65</u>	<u>55.11</u>	36.16	<u>48.97</u>	<u>55.93</u>
DepthQuant	**51.18**	**57.12**	**69.49**	**57.75**	**63.82**	**41.15**	**54.49**	**60.58**

Notes. The same setting as Table 4; per-task values are in Appendix C. Bold and underlined mark the best and second-best values within each model.

Table 6: **Component analysis at a requested average of 3 bit.**

Model	Configuration	Wiki		C4		ZS (%)	
Qwen3-8B	Short-Depth Response Scoring	10.80		17.55		69.79	
	+ Full-Depth Allocation	10.79	(−0.01)	17.23	(−0.32)	69.99	(+0.20)
	+ per-row range refinement (DepthQuant)	**10.64**	(−0.16)	**17.09**	(−0.46)	**71.95**	(+2.16)
Qwen3-32B	Short-Depth Response Scoring	8.65		14.66		73.26	
	+ Full-Depth Allocation	8.56	(−0.09)	14.31	(−0.35)	74.55	(+1.29)
	+ per-row range refinement (DepthQuant)	**8.19**	(−0.46)	**13.68**	(−0.98)	**76.80**	(+3.54)
Llama-2-13B	Short-Depth Response Scoring	5.27		7.50		71.26	
	+ Full-Depth Allocation	5.27	(0.00)	7.48	(−0.02)	71.32	(+0.06)
	+ per-row range refinement (DepthQuant)	**5.23**	(−0.04)	**7.43**	(−0.07)	**72.10**	(+0.84)

Notes. Wiki and C4 are perplexity (↓) and ZS the six-task average (↑). Each block adds one refinement to the row above it; bold marks the best value of each block, and the shaded row is the complete method.

$c = 0.10$ are near-best (Tables 15 and 16); the same appendix reports the scoring-signal control and the range and horizon diagnostics. At 2 bit the scoring comparison is sharper: the response score leads the local reconstruction error and random priorities by more than ten points of six-task average on both models (Table 21).

5 Conclusion

DepthQuant allocates weight-only mixed precision using quantization errors' propagated responses rather than local magnitude alone. Short-Depth Response Scoring constructs within-layer allocation paths from one-layer-deeper responses. Full-Depth Allocation uses model-output KL divergence to redistribute precision along these paths while preserving the total parameter-bit budget. Per-row range refinement then minimizes local weight-reconstruction error with the bit map fixed. This design combines short-depth sensitivity, full-depth allocation, and local reconstruction. Across eight models, DepthQuant ranks first or second in six-task accuracy in 14 of 16 reported model–budget settings, with gains up to 11.6 points at 2 bit, supporting effective quantization at very low precision.

AI USE STATEMENT

Generative AI tools were used solely for language polishing, including improving grammar, clarity, and presentation. They were not used for generating research ideas, designing the methodology. All AI-assisted text was carefully reviewed and revised by the authors. The authors take full responsibility for the final content of this work.

ETHICS STATEMENT

This work studies weight-only mixed-precision quantization for large language models and does not involve human-subject studies, collection of personal data, or deployment in sensitive application domains. We use publicly available pretrained models, datasets, and evaluation benchmarks following their respective licenses and standard research protocols. We do not identify additional ethical concerns specific to the proposed methodology.

REPRODUCIBILITY STATEMENT

We provide the complete formulation of DepthQuant, including the scoring rule, the allocation process, and the per-row range search, in Section 3, together with its full procedure as Algorithm 1 in Appendix A and the proof of our theoretical result in Appendix B. All calibration and evaluation choices, including the group size, the candidate bit widths, the probe count, the redistribution cap, and the calibration sets, are documented in Section 4; Appendix C reports the full per-task results for every model and budget, Appendix D the parameter and horizon diagnostics behind the sensitivity analysis, and Appendix E the configuration and measured cost of the reported runs. Source code is included in the supplementary materials to facilitate reproduction.

REFERENCES

Hicham Badri and Appu Shaji. Half-quadratic quantization of large machine learning models, November 2023. URL https://dropbox.github.io/hqq_blog/.

Zihan Chen, Bike Xie, Jundong Li, and Cong Shen. Adapting to any bit-width: Channel-wise mixed-precision quantization for LLMs. *Transactions on Machine Learning Research(TMLR)*, 2026. ISSN 2835-8856.

Wenhua Cheng, Weiwei Zhang, Heng Guo, Haihao Shen, and Zaner Ma. SignRoundV2: Toward closing the performance gap in extremely low-bit post-training quantization for LLMs. arXiv preprint arXiv:2512.04746, 2026.

Zihao Deng, Sayeh Sharify, Xin Wang, and Michael Orshansky. Mixed-precision quantization for deep vision models with integer quadratic programming. In *2025 62nd ACM/IEEE Design Automation Conference (DAC)*, pp. 1–7, 2025.

Zhen Dong, Zhewei Yao, Daiyaan Arfeen, Amir Gholami, et al. HAWQ-V2: Hessian aware trace-weighted quantization of neural networks. In *Advances in Neural Information Processing Systems(NIPS)*, volume 33, pp. 18518–18529. Curran Associates, Inc., 2020.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. GPTQ: Accurate post-training quantization for generative pre-trained transformers. arXiv preprint arXiv:2210.17323, 2023.

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, Jason Phang, Laria Reynolds, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation. Zenodo, 2021.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, et al. Gemma 2: Improving open language models at a practical size. arXiv preprint arXiv:2408.00118, 2024.

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. In *Proceedings of the 61st ACM/IEEE Design Automation Conference(DAC)*, 2024. doi: 10.1145/3649329.3658498.

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 *Proceedings of the International Conference on Machine Learning (ICML)*, volume 267, pp. 25672–25692, 2025.

Michael F. Hutchinson. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics - Simulation and Computation*, 18(3):1059–1076, 1989.

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 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 35532–35550, 2025.

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 LLM compression and acceleration. In *Proceedings of Machine Learning and Systems(MLSys)*, volume 6, pp. 87–100, 2024.

Yuexiao Ma, Taisong Jin, Xiawu Zheng, Yan Wang, Huixia Li, Yongjian Wu, Guannan Jiang, Wei Zhang, and Rongrong Ji. OMPQ: Orthogonal mixed precision quantization. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 37, pp. 9029–9037, 2023.

Stephen Merity, Caiming Xiong, James Bradbury, et al. Pointer sentinel mixture models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

Ashitabh Misra, Madhav Agrawal, Arham Jain, and Tarek Abdelzaher. MixQuant: Adaptive mixed-precision quantization for large language models. arXiv preprint arXiv:2607.23047, 2026.

Markus Nagel, Rana Ali Amjad, Mart Van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? adaptive rounding for post-training quantization. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 119, pp. 7197–7206, 2020.

Md. Mehrab Hossain Opi, Robiul Islam Ryad, and Md. Umar Faruk. MixFrag: Fragility-guided mixed-precision post-training quantization for vision transformers. arXiv preprint arXiv:2607.28589, 2026.

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(JMLR)*, 21(140):1–67, 2020.

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 *Proceedings of the International Conference on Learning Representations (ICLR)*, volume 2024, pp. 45472–45496, 2024.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288, 2023.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. arXiv preprint arXiv:2505.09388, 2025.

Zhangyang Yao, Haiyan Zhao, Haoyu Wang, and Xu Han. GAMMA: Global bit allocation for mixed-precision models under arbitrary budgets. arXiv preprint arXiv:2605.18475, 2026.

Zhen Zheng, Xiaonan Song, and Chuanjie Liu. MixLLM: LLM quantization with global mixed-precision between output-features and highly-efficient system design. In *Proceedings of Machine Learning and Systems(MLSys)*, volume 8, pp. 1771–1785, 2026.

APPENDIX CONTENTS

- A **Full Procedure.** Three-stage pseudocode.
- B **Proof of Lemma 1.**
- C **Per-task Zero-shot Results.** Detailed results across eight models.
- D **Additional Ablation Results.** Parameter studies and diagnostic experiments.
- E **Implementation and Cost.** Implementation details and computational costs.

A FULL PROCEDURE

Algorithm 1 lists the three stages of the method in the order in which they are executed. Short-Depth Response Scoring (Stage 1) scores every candidate by the propagated response of its perturbation and freezes one allocation path per layer together with the Stage-1 state; Full-Depth Allocation (Stage 2) prices the reachable budget states at the model output and moves bits between layers, keeping the parameter-bit count of the resulting map exactly equal to the count frozen in Stage 1; the per-row range refinement (Stage 3) searches the quantization parameters of each assigned output row on the fixed range grid.

Algorithm 1: **DepthQuant**

Input: FP model f; calibration data $\mathcal{D}$; candidate bits $\mathcal{B}$; requested average $\bar{b}_{\mathrm{req}}$; probe count K; cap c; range grid $\mathcal{T}$

Output: quantized model and its parameter-bit count

Stage 1: Short-Depth Response Scoring
foreach *layer* ℓ **do**
  foreach *precision unit* g *and candidate bit* $b \in \mathcal{B}$ **do**
    build the candidate perturbation $u_{\ell,g,b}$
    propagate it over the measurement horizon
    estimate its propagated response $\widehat{C}_{\ell,g}(b)$ with K probes
  build the allocation path π_ℓ by greedy gain per added bit
  select the Stage-1 state of π_ℓ nearest the requested budget
freeze the paths and their parameter-bit count B_1

Stage 2: Full-Depth Allocation at the model output
foreach *layer* ℓ **do**
  collect the reachable states on π_ℓ with $|\widetilde{b}_{\ell,s} - \bar{b}_{\mathrm{req}}| \le c$, together with the Stage-1 state
  foreach *such state* s **do**
    measure the response $k_{\ell,s}$ at the model output, with all other layers in full precision
solve the exact-budget selection (12) and apply the chosen maps

Stage 3: per-row range refinement
foreach *output row* r *of each assigned unit whose bit class is searched in this configuration* **do**
  pick $\tau \in \mathcal{T}$ minimizing the row's local weight-reconstruction error
  refit that row's affine scale, zero-point, and codes

B PROOF OF LEMMA 1

Throughout this appendix, model weights, calibration examples, candidate perturbations, and linearization points are fixed. Expectations are therefore only over the Rademacher probes.

Fix a candidate (ℓ, g, b) and a calibration example n, and abbreviate

$$u^{(n)} = u^{(n)}_{\ell,g,b}, \qquad J^{(n)} = J^{(n)}_{\ell,g\to e}, \qquad v^{(n)} = J^{(n)}u^{(n)}. \tag{14}$$

The coordinates of a Rademacher vector $\xi_{n,k}$ are independent, have zero mean, and have unit second moment. Consequently, $\mathbb{E}[\xi_{n,k}\xi_{n,k}^\top] = I$, and

$$\begin{aligned}\mathbb{E}_\xi\left[\left(\xi_{n,k}^\top v^{(n)}\right)^2\right] &= (v^{(n)})^\top \mathbb{E}_\xi[\xi_{n,k}\xi_{n,k}^\top]v^{(n)} \\ &= \|v^{(n)}\|_2^2.\end{aligned} \tag{15}$$

Linearity of expectation, rather than probe independence, is sufficient to average this identity over the N examples and K probes. Hence

$$\mathbb{E}_\xi[\widehat{C}_{\ell,g}(b)] = \frac{1}{NK}\sum_{n=1}^{N}\sum_{k=1}^{K}\mathbb{E}_\xi\left[\left(\xi_{n,k}^\top v^{(n)}\right)^2\right] = \frac{1}{N}\sum_{n=1}^{N}\|v^{(n)}\|_2^2 = C_{\ell,g}(b), \tag{16}$$

which is Equation 9.

The projected response does not require materializing either $J^{(n)}$ or $v^{(n)}$, because

$$\xi_{n,k}^\top v^{(n)} = \xi_{n,k}^\top J^{(n)} u^{(n)} = \left((J^{(n)})^\top \xi_{n,k}\right)^\top u^{(n)}. \tag{17}$$

Reverse-mode differentiation evaluates $(J^{(n)})^\top \xi_{n,k}$ as one vector–Jacobian product. Sharing a probe between candidates therefore changes their covariance but not this candidate-wise expectation.

C Per-task Zero-shot Results

The panels below give the per-task detail of the main results: one panel per model, grouped by model family, each covering every budget at which that model is reported. Rows are grouped by requested average precision and the first row of a panel is the full-precision reference, which does not participate in the ranking. The requested averages follow Section 4, where AMQ is reported at its native effective budgets of 2.75 and 3.25 bit. Within a model and budget, bold and underlined values are the best and second-best results and the shaded row is DepthQuant.The average-2-bit rows belong to the stress test of Section 4; their method set comprises methods that allocate mixed precision over a candidate set which includes one-bit weights, while SignRoundV2 do not include one-bit candidates. The Avg. column is the unweighted mean of the six task scores, computed before rounding. Corresponding perplexities are in Tables 2 and 4.

Table 7: Per-task zero-shot results for Qwen3-4B.

Avg. bit	Method	Zero-shot accuracy (%)						
		ARC-C	ARC-E	Hella.	PIQA	Wino.	BoolQ	Avg.
16	FP reference	54.10	78.62	68.45	74.92	66.14	85.11	71.22
2	RTN	<u>26.02</u>	27.06	26.64	52.07	50.28	40.61	37.11
2	GPTQ	25.17	26.60	26.45	52.56	<u>52.17</u>	41.47	37.40
2	SliM	25.68	<u>39.35</u>	<u>39.70</u>	<u>60.28</u>	52.09	<u>47.49</u>	<u>44.10</u>
2	DepthQuant	**29.69**	**49.54**	**44.63**	**64.96**	**56.35**	**61.93**	**51.18**
2.5	SignRoundV2	<u>38.91</u>	<u>63.09</u>	52.23	<u>69.91</u>	<u>59.59</u>	<u>78.87</u>	<u>60.43</u>
2.5	AMQ	35.75	56.27	<u>52.24</u>	68.39	57.38	78.13	58.03
2.5	DepthQuant	**45.65**	**68.94**	**58.30**	**71.93**	**61.17**	**83.18**	**64.86**
3	RTN	37.20	60.06	55.35	69.53	56.51	74.53	58.86
3	GPTQ	41.98	65.19	62.37	72.36	61.33	75.57	63.13
3	SliM	44.28	66.16	<u>62.87</u>	<u>72.85</u>	61.48	82.05	64.95
3	SignRoundV2	<u>45.82</u>	66.20	61.57	72.80	**64.56**	<u>82.17</u>	<u>65.52</u>
3	AMQ	44.62	<u>66.41</u>	61.25	72.80	60.77	81.65	64.59
3	DepthQuant	**47.70**	**71.00**	**64.24**	**74.27**	<u>63.06</u>	**83.94**	**67.37**

Table 8: Per-task zero-shot results for Qwen3-8B.

Avg. bit	Method	Zero-shot accuracy (%)						
		ARC-C	ARC-E	Hella.	PIQA	Wino.	BoolQ	Avg.
16	FP reference	56.48	80.93	74.93	77.69	67.72	86.61	74.06
2	RTN	26.79	24.58	26.24	52.07	50.43	44.95	37.51
2	GPTQ	26.37	27.44	29.44	51.52	47.83	42.23	37.47
2	SliM	26.96	37.54	43.09	60.17	52.96	55.57	46.05
2	DepthQuant	**33.62**	**57.70**	**51.15**	**67.52**	**58.33**	**74.40**	**57.12**
2.5	SignRoundV2	44.80	**73.57**	59.62	**73.67**	**64.48**	80.31	66.07
2.5	AMQ	41.55	66.33	62.50	72.36	60.93	**84.56**	64.70
2.5	DepthQuant	**46.08**	73.15	**65.72**	73.56	63.77	78.99	**66.88**
3	RTN	47.95	71.30	66.08	73.50	61.25	79.36	66.57
3	GPTQ	50.77	72.73	70.40	75.08	64.25	83.85	69.51
3	SliM	44.37	65.53	69.87	74.76	66.14	84.43	67.52
3	SignRoundV2	**54.10**	**79.08**	68.09	75.63	67.09	**85.87**	71.64
3	AMQ	51.37	75.97	70.50	76.71	63.77	84.71	70.51
3	DepthQuant	53.07	77.36	**71.68**	**77.42**	**67.40**	84.74	**71.95**

Table 9: Per-task zero-shot results for Qwen3-32B.

Avg. bit	Method	Zero-shot accuracy (%)						
		ARC-C	ARC-E	Hella.	PIQA	Wino.	BoolQ	Avg.
16	FP reference	60.84	82.91	82.65	81.77	73.01	86.39	77.93
2	RTN	26.54	30.51	30.35	53.32	50.67	39.63	38.50
2	GPTQ	27.05	32.58	40.44	55.44	51.07	52.72	43.22
2	SliM	35.41	49.54	61.76	67.08	60.77	72.81	57.90
2	DepthQuant	**48.89**	**71.63**	**70.04**	**74.10**	**66.30**	**85.96**	**69.49**
2.5	SignRoundV2	54.78	78.54	72.98	77.53	**71.67**	**87.06**	73.76
2.5	AMQ	54.44	**78.62**	74.06	77.64	65.27	74.65	70.78
2.5	DepthQuant	**57.42**	78.28	**78.42**	**78.56**	71.27	86.42	**75.06**
3	RTN	52.30	73.32	67.57	74.70	64.96	80.98	68.97
3	GPTQ	55.89	75.88	78.61	78.84	68.67	86.02	73.98
3	SliM	59.64	79.97	80.44	**80.69**	70.95	82.60	75.71
3	SignRoundV2	58.11	80.09	76.75	79.00	**72.77**	**87.95**	75.78
3	AMQ	56.66	80.09	79.14	79.33	71.82	83.98	75.17
3	DepthQuant	**59.98**	**81.19**	**80.83**	79.38	72.61	86.82	**76.80**

Table 10: Per-task zero-shot results for Qwen2.5-7B.

Avg. bit	Method	Zero-shot accuracy (%)						
		ARC-C	ARC-E	Hella.	PIQA	Wino.	BoolQ	Avg.
16	FP reference	50.85	77.44	78.96	79.76	72.93	84.68	74.10
2	RTN	25.68	25.29	25.27	50.00	51.30	52.45	38.33
2	GPTQ	25.00	33.38	36.23	55.71	51.30	46.15	41.29
2	SliM	27.39	42.80	46.42	61.86	55.72	57.68	48.65
2	DepthQuant	**34.47**	**60.82**	**55.72**	**70.46**	**61.01**	**64.01**	**57.75**
2.5	SignRoundV2	44.20	69.44	66.45	75.19	**68.75**	**79.14**	67.20
2.5	AMQ	45.73	71.30	70.87	76.99	65.51	74.62	67.50
2.5	DepthQuant	**47.95**	**74.24**	**71.37**	**77.97**	68.03	77.55	**69.52**
3	RTN	46.93	68.77	69.22	75.24	64.72	77.13	67.00
3	GPTQ	47.44	72.81	75.30	78.29	66.85	81.35	70.34
3	SliM	**52.65**	**79.42**	74.71	78.07	67.40	77.49	71.62
3	SignRoundV2	49.40	76.73	72.98	77.75	70.40	82.63	71.65
3	AMQ	47.35	72.60	75.54	79.11	67.64	83.27	70.92
3	DepthQuant	50.68	76.56	**76.24**	**79.16**	**71.03**	**84.22**	**72.98**

Table 11: Per-task zero-shot results for Qwen2.5-14B.

Avg. bit	Method	Zero-shot accuracy (%)						
		ARC-C	ARC-E	Hella.	PIQA	Wino.	BoolQ	Avg.
16	FP reference	58.87	79.12	82.90	82.05	75.37	85.20	77.25
2	RTN	27.22	25.13	26.67	51.96	50.28	44.92	37.70
2	GPTQ	24.49	34.60	35.93	56.75	53.35	50.31	42.57
2	SliM	32.76	54.25	53.51	67.85	59.27	63.00	55.11
2	DepthQuant	**39.76**	**66.08**	**63.29**	**73.34**	**64.72**	**75.75**	**63.82**
2.5	SignRoundV2	49.23	76.30	71.02	77.09	71.98	79.51	70.86
2.5	AMQ	51.54	77.78	75.18	79.43	71.35	72.57	71.31
2.5	DepthQuant	**52.65**	**78.70**	**76.93**	**79.81**	**73.40**	**84.13**	**74.27**
3	RTN	54.78	77.10	77.96	79.05	70.95	81.16	73.50
3	GPTQ	54.18	81.06	79.30	79.54	73.88	83.70	75.28
3	SliM	54.95	80.35	79.91	**81.50**	**75.61**	82.84	75.86
3	SignRoundV2	49.66	74.62	75.13	79.16	74.59	82.72	72.65
3	AMQ	57.25	82.28	79.91	80.58	74.43	**83.91**	76.39
3	DepthQuant	**58.02**	**82.58**	**80.24**	80.96	75.53	81.56	**76.48**

Table 12: Per-task zero-shot results for Llama-3.1-8B.

Avg. bit	Method	Zero-shot accuracy (%)						
		ARC-C	ARC-E	Hella.	PIQA	Wino.	BoolQ	Avg.
16	FP reference	54.95	82.45	79.33	81.12	74.51	83.12	75.91
2	RTN	23.81	26.09	26.27	50.92	48.15	39.57	35.80
2	GPTQ	**26.45**	25.17	25.93	50.76	48.86	42.32	36.58
2	SliM	24.66	26.09	25.91	51.96	50.51	37.83	36.16
2	DepthQuant	21.33	**34.13**	**32.13**	**55.88**	**52.33**	**51.10**	**41.15**
2.5	SignRoundV2	43.77	70.54	66.27	74.92	67.09	**78.53**	**66.85**
2.5	AMQ	**44.11**	**70.96**	**68.25**	**75.19**	**69.06**	73.24	66.80
2.5	DepthQuant	42.75	70.54	66.44	74.70	68.03	77.92	66.73
3	RTN	45.05	70.75	71.01	75.19	67.25	73.43	67.11
3	GPTQ	42.58	70.24	70.53	73.94	69.38	72.84	66.59
3	SliM	41.89	68.94	71.39	76.50	68.59	75.14	67.07
3	SignRoundV2	50.26	73.86	71.01	76.66	71.67	**81.47**	70.82
3	AMQ	50.60	77.57	**74.75**	**78.84**	70.72	79.30	71.96
3	DepthQuant	**52.05**	**77.69**	74.41	78.18	**72.93**	80.00	**72.54**

Table 13: Per-task zero-shot results for Llama-2-13B.

Avg. bit	Method	Zero-shot accuracy (%)						
		ARC-C	ARC-E	Hella.	PIQA	Wino.	BoolQ	Avg.
16	FP reference	48.81	76.47	79.66	80.25	72.45	82.14	73.30
2	RTN	23.89	29.76	34.10	55.11	48.78	44.53	39.36
2	GPTQ	26.28	27.31	32.00	52.94	47.91	50.55	39.50
2	SliM	25.51	43.01	45.54	62.02	53.43	64.31	48.97
2	DepthQuant	**32.08**	**54.21**	**51.62**	**65.40**	**54.30**	**69.33**	**54.49**
2.5	SignRoundV2	42.41	**71.89**	72.22	77.04	**71.03**	80.15	69.12
2.5	AMQ	**43.94**	70.92	**75.18**	**77.48**	70.80	79.94	**69.71**
2.5	DepthQuant	43.17	71.04	71.91	76.61	69.77	**80.46**	68.83
3	RTN	46.25	73.48	77.24	78.84	71.59	78.81	71.03
3	GPTQ	**48.38**	74.83	76.83	**79.54**	72.30	77.86	71.62
3	SliM	44.71	73.78	76.48	79.05	70.95	78.35	70.55
3	SignRoundV2	47.18	75.08	77.36	79.11	70.40	**81.35**	71.75
3	AMQ	47.44	74.33	**77.76**	79.49	**72.38**	81.31	**72.12**
3	DepthQuant	47.95	**76.73**	77.13	79.16	71.19	80.46	72.10

Table 14: Per-task zero-shot results for Gemma-2-9B.

Avg. bit	Method	Zero-shot accuracy (%)						
		ARC-C	ARC-E	Hella.	PIQA	Wino.	BoolQ	Avg.
16	FP reference	65.70	88.17	79.86	83.03	74.11	84.37	79.21
2	RTN	23.55	31.19	34.12	55.66	49.33	48.13	40.33
2	GPTQ	28.50	41.92	46.83	62.57	56.99	53.15	48.33
2	SliM	29.95	54.50	52.62	67.25	**63.69**	67.55	55.93
2	DepthQuant	**35.15**	**67.72**	**57.68**	**71.27**	63.30	**68.38**	**60.58**
2.5	SignRoundV2	51.37	**80.72**	72.21	**79.11**	70.96	80.70	72.51
2.5	AMQ	51.88	79.12	71.58	78.29	70.17	72.81	70.64
2.5	DepthQuant	**52.05**	79.63	**73.22**	78.24	**71.82**	**84.34**	**73.22**
3	RTN	60.15	83.88	75.06	79.82	73.01	80.49	75.40
3	GPTQ	**61.77**	85.27	77.27	80.69	73.24	83.18	76.90
3	SliM	59.90	84.97	76.72	81.50	74.03	83.15	76.71
3	SignRoundV2	59.98	**86.41**	76.54	81.50	72.22	83.24	76.65
3	AMQ	58.62	83.92	76.62	80.85	73.64	82.81	76.08
3	DepthQuant	59.56	84.47	**77.81**	**81.61**	**74.11**	**84.68**	**77.04**

D Additional Ablation Results

This appendix records the panels behind the sensitivity analysis of Section 4. They share the protocol of the main results: group size 128, the same calibration sets, and the same evaluation harness. In every panel, Wiki and C4 are WikiText-2 and C4 perplexity (↓) and ZS is the six-task zero-shot average in percent (↑); Tables 18 and 19 report ZS only.

We write d for the depth of the response measurement, counted in layers. The reported configuration fixes $K = 8$ shared probes, a redistribution cap of $c = 0.10$, a horizon of $d = 2$, and a per-row range search. The panels below vary one setting at a time at a requested average of 3 bit: first the probe count and the redistribution cap on Qwen2.5-7B and Llama-3.1-8B (Tables 15 and 16), then the horizon and the range rule (Table 17), which are kept separate because they change different quantities.

Table 15: Probe count K on two models at a requested average of 3 bit. The shaded rows are the settings used for the reported results.

Model	K	Wiki	C4	ZS (%)
Qwen2.5-7B	1	7.45	13.10	72.03
	4	7.45	13.10	72.13
	8 (reported)	7.45	13.10	**72.98**
	12	7.45	**13.09**	71.16
Llama-3.1-8B	1	**7.33**	**13.00**	71.88
	4	7.43	13.27	72.03
	8 (reported)	**7.33**	13.01	**72.54**
	12	7.35	13.69	72.32

Notes. Wiki and C4 are WikiText-2 and C4 perplexity (↓); ZS is the six-task zero-shot average (↑); bold marks the best value of each column. The four rows of a model come from one probe-count batch and share its range configuration, so the spread across the rows measures the sensitivity to the probe count as a setting.

Table 18 separates the two refinements on the three models of Table 6: each stage contributes on its own and again on top of the other, so the ordering of Table 6 holds whichever stage is applied first. Table 19 asks whether the scoring signal itself matters. With the range refinement fixed and Full-Depth Allocation disabled, replacing the short-depth response score by the local reconstruction error of the same candidates lowers the six-task average of Llama-3.1-8B by 6.9 points at the same budget, and replacing it by random priorities lowers it by 11.5. The reference row of Table 19 is the reported scoring signal with the range refinement and without the allocation stage, and the three rows share the calibration set, the evaluation harness, and the realized parameter-bit total to within 0.001.

Table 16: Redistribution cap c on two models at a requested average of 3 bit. The cap bounds the set of layer-budget states that Full-Depth Allocation may price (Section 3.3); the shaded rows are the settings used for the reported results.

Model	c	Wiki	C4	ZS (%)
Qwen2.5-7B	0.05	7.48	13.12	71.79
	0.10 (reported)	**7.45**	**13.10**	72.98
	0.15	**7.45**	**13.10**	**73.14**
Llama-3.1-8B	0.05	**7.31**	**12.65**	71.66
	0.10 (reported)	7.33	13.01	**72.54**
	0.15	7.34	13.07	71.55

Notes. Wiki and C4 are WikiText-2 and C4 perplexity (↓); ZS is the six-task zero-shot average (↑); bold marks the best value of each column. The three rows of a model come from one cap batch and share its range configuration.

Table 17: Measurement-horizon and range-rule diagnostics at a requested average of 3 bit.

(A) Horizon d of the response measurement, counted in layers, with the reported design at $d = 2$ (the remainder of the source layer and the next layer).

Model	d	Wiki	C4	ZS (%)
Qwen3-4B	1	14.64	21.60	67.79
	2 (reported)	14.50	21.54	67.37
	4	14.56	21.54	67.77
Qwen2.5-7B	1	7.41	13.04	72.21
	2 (reported)	7.45	13.10	72.98
	4	7.46	13.12	71.60

(B) Row-wise search against fixed per-bit range factors on Qwen2.5-7B: each factor is the mean selection of the per-row search, rounded to the grid; both rows keep the same bit map.

Range rule	Wiki	C4	ZS (%)
Fixed factor per bit ($.625/.825/.950$)	7.55	13.82	72.86
Per-row search (reported)	7.45	13.10	72.98

Notes. Wiki and C4 are perplexity (↓) and ZS is the six-task zero-shot average (↑). In (A) the reported rows are the main-table entries for the same model and budget; the panel characterizes the horizon rather than establishing a universal optimum, since the reported $d = 2$ has the highest six-task average on Qwen2.5-7B but not on Qwen3-4B, where $d = 1$ and $d = 4$ are higher. In (B) the comparison is between a per-row choice and one shared factor per bit class.

Table 20 measures the same comparison at the model output: for each configuration it reports the divergence between the full-precision output distribution and that of the jointly quantized model, so the ordering of the two stages can be read on the deployed object and not only on a single layer.

Table 21 repeats the comparison at a requested average of 2 bit, where the per-row range refinement is fixed on W1/W2 and only the signal that ranks the candidates changes: on both models the response score leads the local reconstruction error and the average of three random draws by more than ten points of six-task average.

The allocation ranks maps by the priced objective, a sum of isolated per-layer divergences in which only one layer is quantized at a time, whereas the maps we deploy quantize every layer at once, so the quantity of interest is their joint divergence. Table 22 checks whether the cheap objective orders maps in the same way as that measured divergence. On Qwen3-32B at a requested average of 3 bit we build five maps on the same frozen allocation path, using four random priority orders and one reversed order; the path admits its candidate changes one at a time, so a different order gives a different map at the same budget. For each map we report the priced objective and the divergence of the quantized model. The five maps share one candidate domain, the same GPTQ backend, and the same parameter-bit total ($93{,}627{,}613{,}184$ bits, realized 3.0003); the range search is left off, so both scores are read on the same objects. Within this domain the two measures order the five maps identically (Spearman and Kendall both 1.0, with no discordant pair), so the cheap objective does

Table 18: Component analysis at a requested average of 3 bit with the two stages separated. Each block starts from Short-Depth Response Scoring and adds the stage or stages named in its configuration column; Δ is the complete method minus Short-Depth Response Scoring alone.

Model	Configuration	ZS (%)	Δ
Qwen3-8B	Short-Depth Response Scoring	69.79	
	+ Full-Depth Allocation	69.99	
	+ per-row range refinement	70.94	
	+ Full-Depth Allocation and per-row range refinement (DepthQuant)	**71.95**	+2.16
Qwen3-32B	Short-Depth Response Scoring	73.26	
	+ Full-Depth Allocation	74.55	
	+ per-row range refinement	76.04	
	+ Full-Depth Allocation and per-row range refinement (DepthQuant)	**76.80**	+3.54
Llama-2-13B	Short-Depth Response Scoring	71.26	
	+ Full-Depth Allocation	71.32	
	+ per-row range refinement	71.35	
	+ Full-Depth Allocation and per-row range refinement (DepthQuant)	**72.10**	+0.84

Notes. All entries are the six-task zero-shot average in percent (↑). The four configurations of a block share the requested average of 3 bit and the total parameter-bit budget: Full-Depth Allocation moves bits between layers under a fixed total, and the per-row range refinement leaves the bit map unchanged. The complete method is the main-table entry for the same model and budget, and the other three configurations use the same calibration set and evaluation protocol.

Table 19: Scoring-signal control on Llama-3.1-8B at a requested average of 3 bit. All three rows hold the per-row range refinement fixed with Full-Depth Allocation disabled, so the comparison isolates the signal that ranks the candidate precision.

Scoring signal	ZS (%)	Δ
Short-Depth Response Scoring	**71.95**	—
Local reconstruction error	65.03	−6.92
Random priorities	60.44	−11.51

Notes. Six-task zero-shot average (↑); Δ is relative to the reported signal. The three rows share the calibration set, the evaluation harness, and the realized parameter-bit total to within 0.001. "Local reconstruction error" scores a candidate by the reconstruction error of the corresponding module output, and "random priorities" draws a fixed random priority per group, so both rank the candidates from information available within the module.

Table 20: Output-distribution divergence at a requested average of 3 bit. Each block lists the four configurations of the two stages.

Model	Configuration	KL (nats/token)
Qwen3-32B	Short-Depth Response Scoring	0.2206
	+ Full-Depth Allocation	0.2064
	+ per-row range refinement	0.1564
	+ Full-Depth Allocation and per-row range refinement (DepthQuant)	**0.1495**
Qwen3-8B	Short-Depth Response Scoring	0.1840
	+ Full-Depth Allocation	0.1759
	+ per-row range refinement	0.1377
	+ Full-Depth Allocation and per-row range refinement (DepthQuant)	**0.1360**

Notes. The divergence is measured from the full-precision model to the quantized model over the full vocabulary, on 64 held-out WikiText-2 validation windows (131,008 predicted positions) whose normalized lines do not overlap the calibration set; lower is better. The four configurations of a block share the parameter-bit total, the calibration set, and the evaluation protocol, and the complete method attains the lowest divergence in each block.

Table 21: Scoring-signal comparison at a requested average of 2 bit. Each block keeps the range refinement fixed and varies only the signal that ranks the candidate precision.

Model	Scoring signal	Wiki	C4	ZS (%)	Bits
Qwen3-4B	Short-Depth Response Scoring	**22.44**	**38.59**	**51.25**	2.0010
	Local reconstruction error	411.28	1689.81	36.36	2.0010
	Random priorities (three draws)	286.87	788.76	36.31	2.0011
Llama-2-13B	Short-Depth Response Scoring	**8.27**	**14.34**	**47.67**	2.0002
	Local reconstruction error	138.92	780.43	36.52	2.0006
	Random priorities (three draws)	250.37	570.01	36.79	2.0006

Notes. At this budget the configuration searches candidate precisions $\{1, 2, 3\}$, disables the output-based allocation, and keeps the per-row range refinement on W1/W2, so only the ranking signal changes; realized average bits differ by at most 0.001 between arms. The random rows average three draws (35.63–36.80 and 36.46–37.28 in the two models), reported without selecting one; all arms share the calibration set, the harness, and the tokenizer settings, and the first row of each block is the same-protocol reference, not the complete method.

not mislead the selection here. The reported allocation is not in the ranking, because it minimizes the priced objective by construction; its measured divergence is 0.2064, below all five probes.

Table 22: Ordering check for the priced objective on Qwen3-32B at a requested average of 3 bit.

Map	Priced objective	Output divergence
random1	0.4301	0.2150
random3	0.4313	0.2191
random0	0.4313	0.2198
random2	0.4402	0.2203
reverse	0.4790	0.2412

Notes. The two columns are not on one scale: the objective sums one isolated cost per layer, whereas the divergence is measured once on the whole model, so only their order is compared. The divergence is measured as in Table 20. The five maps are pre-registered probes chosen to spread the priced objective, so the table checks the ordering within one candidate domain rather than a population correlation.

Because the allocation minimizes a sum of isolated layer costs, Table 23 measures how closely that sum follows the joint value when the corresponding layers are quantized together.

Table 23: Additivity of the isolated costs on preselected layer pairs of Llama-2-13B. Divergences are in 10^{-3} nats/token.

Budget	Layer pair	Sum of isolated	Joint	Relative deviation
3 bit	0, 1	0.5761	0.5758	-0.055%
	19, 20	3.6714	3.6837	$+0.335\%$
	0, 39	2.2422	2.2398	-0.108%
2.5 bit	0, 1	1.2652	1.2663	$+0.094\%$
	19, 20	9.9951	10.3221	$+3.271\%$
	0, 39	6.6666	6.6664	-0.004%

Notes. For each pair the two layers are quantized independently under a full-precision prefix with the quantized weights held fixed, and the output divergence (full vocabulary, 64 held-out windows) is measured for each layer alone and for the two together. Five of the six pairs agree to within 0.34%; the middle adjacent pair at the tighter budget differs by 3.27%. Because each layer is quantized on its own here, unlike the sequential pass that produces the reported models, and the divergence is averaged over windows, the agreement supports the sum as an ordering signal for small groups of layers rather than a bound on the deviation of a fully quantized network.

Stage 1 ranks candidates by an estimated quantity rather than a measured one: the probe estimate, in first order, of how far quantizing one group moves the response at the observation point. Table 24 checks that estimate against a direct measurement. On an early and a late layer of each reported configuration we sample candidate groups—16 random groups per layer, plus up to four that the reported allocation actually raises—and record for every sampled candidate both its probe estimate and the change that the same depth-2 horizon produces when only that group is quantized by the

same rule and the other weights stay in full precision. The allocation uses these numbers in two ways, so the audit reports two agreement measures. The first is the ordering: whether the candidates that the score ranks higher are also the ones that move the response more, which is what the allocation compares within a layer. Panel (A) gives this ordering for the response energy and for the marginal gain per added parameter bit, as Spearman ρ together with the overlap of the best three and five candidates, and panel (B) repeats the energy ρ separately for each bit width. The second is the magnitude, where a first-order estimate is loosest: the median relative error of the energy ranges from 11% to 56%. The ordering of the marginal gain, which is what the allocation ranks on, stays within ρ 0.979–0.998 in every row, and the energy ordering is 0.972–0.999. Panel (B) separates the energy ρ by bit width: it is 0.935 or above for the one-, two-, and three-bit classes and weakens only in the four-bit class, where the perturbation is smallest (ρ down to 0.669), so the property that the allocation relies on most is the one that the audit supports best. The audit is local—one group at a time, with the other weights in full precision—and every row is a separate configuration, so candidates are compared within a layer and rows not with one another.

Table 24: Agreement between the short-depth score and the measured forward change at the observation point, on sampled candidates. ρ is Spearman and each row is one reported configuration.

(A) Response energy and marginal gain over all sampled candidates of a layer.

Model	Layer	Energy ρ	Gain ρ	Gain top-3	Gain top-5	Rel. error
Qwen3-4B (2 bit)	9	0.998	0.992	3/3	5/5	11.4%
	27	0.999	0.994	2/3	4/5	16.7%
Qwen2.5-7B (3 bit)	7	0.984	0.994	3/3	4/5	34.1%
	21	0.985	0.989	3/3	5/5	36.7%
Qwen3-8B (3 bit)	8	0.992	0.995	2/3	5/5	29.3%
	24	0.994	0.998	3/3	5/5	27.9%
Llama-3.1-8B (3 bit)	8	0.988	0.995	3/3	4/5	35.1%
	24	0.972	0.979	1/3	5/5	55.5%
Gemma-2-9B (2 bit)	10	0.997	0.995	2/3	5/5	13.3%
	31	0.996	0.992	2/3	3/5	18.4%
Gemma-2-9B (3 bit)	10	0.985	0.994	3/3	4/5	37.8%
	31	0.990	0.997	3/3	5/5	36.2%

(B) Energy ρ by bit width; a dash marks a bit class that the requested budget does not use.

Model	Layer	W1	W2	W3	W4
Qwen3-4B	9	0.991	0.989	0.995	—
	27	1.000	0.998	0.998	—
Qwen2.5-7B	7	—	0.989	0.962	0.826
	21	—	0.991	0.935	0.770
Qwen3-8B	8	—	0.988	0.986	0.925
	24	—	0.985	0.988	0.956
Llama-3.1-8B	8	—	0.992	0.986	0.782
	24	—	0.983	0.980	0.669
Gemma-2-9B (2 bit)	10	0.994	0.983	0.958	—
	31	0.979	0.993	0.982	—
Gemma-2-9B (3 bit)	10	—	0.991	0.958	0.865
	31	—	0.994	0.982	0.890

Notes. Predicted values are the recorded probe estimates; measured values re-run the same depth-2 horizon with the group quantized by the same rule. Kendall's τ is 0.887–0.982 for the energy and 0.905–0.972 for the gain, with at most 5.6% of pairs discordant; the energy τ falls to 0.55 in the 4-bit subset, where the perturbation is smallest, and every sampled step reduces the measured response, so the audit concerns magnitudes and ordering. "Rel. error" is the median of $|\mathrm{pred} - \mathrm{meas}|/\mathrm{meas}$ (quartiles 5–34%, 11–56%, 31–76%); the gain ordering stays within 0.929–1.000 across bit-width pairs, and the output-injection path reproduces it to within 0.0021 in ρ.

Table 25 asks the same question one step further out. The previous table compared the score with the change measured at the observation point; here it is compared with the divergence of the model output, which is the quantity Stage 2 prices. For each sampled candidate we quantize only that candidate's group with the same quantizer as the audit, keep the other weights in full precision, and measure the divergence of the output distribution from the full-precision model on 64 held-

out WikiText-2 windows. Each layer contributes 20 groups at three bit widths, so 60 candidates, and the second panel normalizes the adjacent-width difference by the parameters that the extra bit adds, which is the marginal gain per added parameter bit that the allocation ranks on (40 per layer). Both panels make the same comparison between two scores, the local reconstruction error of the candidate and its response score: panel (A) correlates each of them with the measured divergence for the energy, as Spearman ρ and Kendall τ, and panel (B) does the same for the marginal gain, as ρ together with the overlap of the five highest-ranked candidates. The response score leads in all four rows. For the energy, ρ is 0.975–0.990 against 0.637–0.895 and τ is 0.867–0.924 against 0.473–0.741; for the marginal gain, ρ is 0.967–0.980 against 0.476–0.881 and the top-5 overlap reaches 1.0 against 0.0–0.8. Four layers in two models are audited, each at its reported budget, so the rows are separate configurations and are not comparable with one another; the advantage is also not uniform in every subset, since on Qwen3-4B layer 27 the top-10 overlap is 0.8 for the response score against 0.9 for the local error. One group is perturbed at a time with the other weights in full precision, so this is a local diagnostic rather than a joint-quantization result.

Table 25: Score against the whole-network output divergence, per sampled candidate.

(A) Response energy against the output divergence.

Model	Layer	Spearman ρ		Kendall τ	
		Local error	Response score	Local error	Response score
Qwen3-4B (2 bit)	9	0.803	**0.989**	0.611	**0.922**
	27	0.895	**0.990**	0.741	**0.924**
Qwen3-8B (3 bit)	8	0.637	**0.976**	0.473	**0.867**
	24	0.848	**0.975**	0.678	**0.881**

(B) Marginal gain per added parameter bit against the output divergence.

Model	Layer	Spearman ρ		Top-5 overlap	
		Local error	Response score	Local error	Response score
Qwen3-4B (2 bit)	9	0.514	**0.967**	0.6	**1.0**
	27	0.714	**0.980**	0.8	0.8
Qwen3-8B (3 bit)	8	0.476	**0.967**	0.0	**0.8**
	24	0.881	**0.978**	0.6	**1.0**

Notes. The candidate quantizer and the single-group protocol are those of Table 24, and the correlations of a row are pooled over the candidates of one layer.

E Implementation and Cost

This appendix records the configuration behind the reported results and the measured cost of the pipeline.

Configuration. DepthQuant uses group size 128, weight precisions 2, 3, 4 for the main comparison and 1, 2, 3 at a requested average of 2 bit, GPTQ with block size 128, damping 0.01, asymmetric zero points, sequential updates, and BF16 weights. The RTN and GPTQ baselines use uniform RTN and GPTQ at the same group size, and the AMQ, SliM and SignRoundV2 rows are produced with the released configuration of each method, AMQ with its native AWQ quantizer. Calibration uses 128 WikiText-2 training sequences of 2048 tokens, and the allocation stage prices its states against the full-precision model on 96 additional WikiText-2 sequences of 512 tokens that do not overlap the Stage-1 calibration samples. Perplexity is evaluated on the full corpora, and zero-shot accuracy on the six tasks of Section 4 with the native tokenizer configuration of each model. Model checkpoints are the public releases of the families in Tables 2 and 3, loaded through their native configuration files.

All zero-shot numbers use a single harness revision (`008fc2a2`), and every baseline is run with its released configuration under the shared evaluation protocol. Accuracy is the normalized accuracy for ARC-Challenge, ARC-Easy, HellaSwag, and PIQA and the accuracy for WinoGrande and BoolQ, following the reporting convention of SignRoundV2 (Cheng et al., 2026), with the six-task average taken as their unweighted mean.

One-bit units follow the same affine rule as Eq. (2), with the fitted interval anchored at zero ($w_{\min} \leq 0 \leq w_{\max}$) and the two codes $\{0, 1\}$: the grid then has two levels, one of which is zero, so a unit keeps a single nonzero magnitude on the side that dominates its fitted range and maps the remaining weights to zero, and a constant unit is stored as zeros.

Realized precision. Average bits are parameter-weighted counts of quantized weight bits. Within a model and budget, the maps we allocate realize counts that agree to within 0.001 bit; AMQ is reported at its own native effective budgets and the two labels are not interchangeable. Table 26 lists the measured values of representative configurations.

Table 26: Realized average bit of the reported configurations, to four decimal places.

Model	Requested	Realized average bit
Llama-2-13B	3	3.0001
Qwen3-32B	3	3.0003
Qwen2.5-7B	2.5	2.5007–2.5008
Gemma-2-9B	2.5	2.5009
	3	3.0008

Order of the range search. The per-row range refinement runs inside the GPTQ sequential pass. For each group, the reconstruction reference is the group state at that point, which already carries the compensation applied to earlier columns. For every output row of the group we evaluate the multipliers that are admissible for its bit class on the fixed grid, refit the scale and zero point from the candidate endpoints, re-derive the integer codes, and keep the multiplier with the lowest local reconstruction error. The selected endpoints then fix the row's grid, the compensation for that group proceeds with the resulting quantized weights, and the bit map is unchanged; rows outside the searched classes keep the plain asymmetric fit. The allocation prices its states before this search is applied, so the priced quantity is the sum of the isolated per-layer divergences rather than the joint divergence of the final quantized model. This order is the one recorded by the reported runs: the range of each row is decided from the weight state it receives, the codes follow from the refit endpoints, and the error compensation that follows uses those quantized weights.

Measured cost. Table 27 lists the measured cost of the pipeline. The probe pass is a one-time cost for a fixed model, candidate set, and calibration protocol; the per-configuration figure covers quantization and evaluation on one GPU with the probe scoring and the allocation reused, and the allocation solve covers the CPU dynamic program only, excluding the GPU cost of computing the per-layer prices. Changing the candidate set, for example from $\{2, 3, 4\}$ to $\{1, 2, 3\}$, requires re-running the probe pass and the pricing, with no downstream result reused. The audit of the short-depth score fixes the candidate quantizer of the scoring path, whereas the pricing and the deployment use the GPTQ backend, so the transfer between the two is measured end-to-end (Tables 19 and 20) rather than by the audit. We report parameter-bit budgets and quantization cost, and do not claim inference-time memory or latency gains.

Table 27: Measured cost of the reported pipeline.

Quantity	Measured value
Budget states within the cap, per layer ($c = 0.10$, 7B-class)	5
Reachable budget states, 3 bit ($c = 0.10$, 32B-class)	42,433
Allocation solve, CPU (32B-class)	4.07 s
Quantization throughput, 13B-class	≈29 s per layer
Quantization throughput, 32B-class	≈46 s per layer
Quantization and evaluation, 8B-class, allocation assets cached	68–70 min on one GPU
Peak GPU memory, 8B-class	≈7.6 GB
One-time probe pass, 8B-class, 128 samples	≈10.4 h