TWO-PASS TRAJECTORY CALIBRATION FOR COMPENSATION-BASED POST-TRAINING QUANTIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

In post-training quantization (PTQ), learned per-row codebooks are typically fitted once on the weights where an error compensation engine such as GPTQ or LDLQ starts from. The engine, however, does not round those weights: after each column is rounded, the remaining columns are updated to compensate for its rounding error. By the time a later column is rounded, its values have therefore already been shifted by the compensation from every column rounded before it. We call this sequence of shifted values the layer's *trajectory*. We propose 2-Pass Trajectory Calibration (2PTC), a drop-in correction for the mismatch between the static weights a codebook is fit on and the shifted trajectory the engine actually rounds: run the engine once to record the trajectory, refit the codebook on it, and rerun the same engine on the same starting weights with the refitted grid. 2PTC stores a single codebook of the same memory size and leaves inference unchanged; it only costs one additional offline pass. Across six open LLMs and two compensation engines, on top of QEP inter-layer correction, per-channel 2PTC lowers perplexity at every bit-width with GPTQ and at 2 and 3 bits with LDLQ. The gains are largest at 2 bits: WikiText-2 perplexity decreases in 36 of 36 paired runs, by 15.5% on average with GPTQ and 5.7% with LDLQ, and average zero-shot accuracy improves by up to 9.4 points.

1 INTRODUCTION

Post-training quantization (PTQ) compresses a trained model to b bits per weight using a small calibration set, without retraining. Layer-wise PTQ does so one linear layer at a time, minimizing the error at the layer's *output*. The current PTQ methods differ mainly in terms of their engine, starting point, and grid. The **engine** decides how rounding errors interact: GPTQ (Frantar et al., 2023) sweeps the columns in a fixed order and pushes each column's rounding error into the columns not yet processed through the inverse Hessian of the layer input. LDLQ (Chee et al., 2023) generalizes this update: it derives the update from an LDL decomposition of the Hessian rather than from a fixed column order. The **starting point** decides what object the engine is fitted against: QEP (Arai & Ichikawa, 2025) corrects $W \to W^*$ before rounding so that each layer is fitted to the activations it will actually receive from already-quantized layers, thus accounting for propagated error in these layers. The **grid** decides where the 2^b representable values sit for b-bit quantization: a uniform grid stores a scale and zero-point per row, whereas a learned code-

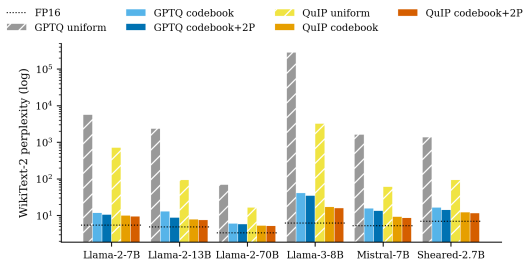


Figure 1: **The 2-bit regime, per model** (per-channel, WikiText-2 perplexity, mean of 3 seeds, log scale). Within each engine lane, replacing the uniform grid with a learned codebook lowers perplexity by up to $\sim$6800$\times$; the trajectory refit (+2P), *our contribution*, lowers it further on every model, in both the GPTQ and QuIP groups of experiments. Dotted lines: FP16 baselines. Seed-level statistics are shown in Table 4.

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.

book (Kim et al., 2024; Zhang & Shrivastava, 2025) gives each row its own optimized 2^b levels via weighted k-means. Since optimized codebooks are not restricted to a uniform placement of levels, they generally achieve lower quantization error than uniform grids.

This paper focuses on jointly optimizing the starting point and the grid to improve the downstream perplexity. A learned codebook is fitted once, before rounding begins, on the starting point the engine receives (i.e., W or W^*). However, the compensation engine does not round W^* itself: by the time column i reaches the rounding step, its weights have already been shifted by the compensation applied for every column rounded before it, so what column i's rounding actually sees is not $W^*_{:,i}$ but a displaced version of it. We call the collection of these actual inputs to the quantizer the layer's *trajectory* W^+. When inspecting layer trajectories obtained before quantizing W^* with error compensation engines (Figure 4, Appendix C.1), W^+ is displaced from W^* by 6–10% in relative norm, while its variance changes by at most 2.2%. The weights move; the codebook is therefore fitted on a distribution different from the one it is applied to. Weight drift at compensation time has been observed before—most recently by Li et al. (2026), who account for the discrepancy directly in the engine's column-wise compensation update. In this work, we leave the update untouched and move the grid instead.

To align codebook fitting with error compensation, we introduce 2-Pass Trajectory Calibration (2PTC). 2PTC runs the compensation engine once to record the trajectory W^+, refits the codebook on W^+, and reruns the same engine from the same starting weights W^* with the refitted grid (Algorithm 1). The core idea behind this is *to fit the grid where the engine rounds rather than where it starts*. 2PTC stores one codebook of the same size as a single-pass codebook, and leaves inference unchanged; its cost is one additional offline quantization pass. Because it changes only what the grid is fitted on, 2PTC can flexibly adopt any engine that rounds column by column onto a learned grid with any inter-layer corrections such as QEP.

Summary of contributions.

- *A drop-in refinement of codebook fitting in compensation-based PTQ*: 2PTC (§4) fits learned scalar codebooks on the compensation trajectory, without memory or inference overhead.
- *Theoretical motivation*: a layer-level analysis (§4.1) showing that the engine's output error equals a weighted quantization error of the trajectory, not that of the starting weights.
- *Ablation of the fitting target*: keeping everything else fixed, we change only the tensor the grid is fitted on ($W \to W^* \to W^+$) and show that our improvements come from fitting on the trajectory (§5.2).
- *Seed-level evaluation* on six open LLMs, two engines and three bit-widths with paired confidence intervals (§5): In the challenging setting of 2 bits per channel (Figure 1), 2PTC lowers perplexity in all 36 paired runs, with a mean change of -15.5% with GPTQ (95% CI $[-19.8, -11.7]$) and -5.7% with LDLQ ($[-6.7, -4.9]$).

2 Related work

Second-order compression spans from Optimal Brain Damage and Surgeon (LeCun et al., 1989; Hassibi & Stork, 1992) through OBC (Frantar & Alistarh, 2022) to GPTQ (Frantar et al., 2023), which fixed the column order and batched the update. QuIP (Chee et al., 2023) added incoherence preprocessing and the LDLQ rounding, a generalization of GPTQ algorithm. QuIP# (Tseng et al., 2024) replaced the random rotations by Hadamard transforms and the scalar grid by lattice codebooks. On the grid axis, recent codebook-fitting methods for quantization grids such as SqueezeLLM (Kim et al., 2024) and LeanQuant (Zhang & Shrivastava, 2025) fit per-row codebooks by Fisher-weighted k-means and an OBS-style loss-error weighting with an equispaced initialization, respectively. VPTQ (Liu et al., 2024), AQLM (Egiazarian et al., 2024) and QuIP# instead quantize vectors of weights jointly and fine-tune after quantization, a direction complementary to the scalar grids we study. Every scalar codebook method above fits its grid on the weights the layer starts from; that is the choice we revisit.

Closest to our premise is the line of work on *what the column-by-column compensation engine sweep is aligned to*. GPTAQ (Li et al., 2025) minimizes $\|WX - \widehat{W}\widehat{X}\|$ with the full-precision input

on one side and the quantized-model input on the other. Li et al. (2026) observe that GPTQ and GPTAQ align each step against the output of the *already compensated* weights rather than of the original layer, and add a compensation-aware term to the update. QEP (Arai & Ichikawa, 2025) corrects the starting point $W \to W^*$ in closed form before the engine runs. All three intervene on the *weights* — where the sweep starts, what it aligns to, and how it updates. All these methods are evaluated with grids fitted in advance (uniform, group-wise in the case of Li et al. (2026)). 2PTC intervenes on the grid instead, and is in principle compatible with each of them. We use 2PTC together with QEP throughout and leave its composition with the compensation-aware update to future work (Appendix 7). The complete related work is provided in Appendix B.

3 Preliminaries and notation

Layer-wise PTQ compresses a network one linear layer at a time. For a linear layer with unquantized weights $W \in \mathbb{R}^{R\times C}$ and calibration inputs $X \in \mathbb{R}^{C\times N}$, compensation-based methods minimize the squared error at the layer's output,

$$\min_{\widehat{W}} \left\|WX - \widehat{W}X\right\|_F^2 \;=\; \min_{\widehat{W}} \operatorname{tr}\big[(W-\widehat{W})\,H\,(W-\widehat{W})^\top\big], \qquad H \propto XX^\top, \tag{1}$$

where every entry of row r of $\widehat{W}$ takes a value in a finite grid $\mathcal{G}_r$, and $Q_{\mathcal{G}_r}(z)$ rounds z to the nearest level of $\mathcal{G}_r$.

Grid. A *uniform* grid places 2^b equispaced levels between a minimum and a maximum per row. Because LLM weight distributions are heavy-tailed, the range is set by a few extreme values, and most levels are wasted. A *learned codebook* instead assigns each row its own 2^b levels, fitted by weighted 1-D k-means (Kim et al., 2024; Zhang & Shrivastava, 2025):

$$\min_{\mathcal{G}_r,\,|\mathcal{G}_r|=2^b} \sum_j \omega_j \left(w_{rj} - Q_{\mathcal{G}_r}(w_{rj})\right)^2, \qquad \omega_j = H_{jj} \propto \mathbb{E}[x_j^2], \tag{2}$$

initialized at the empirical quantiles of the row. In standard pipelines (Kim et al., 2024; Zhang & Shrivastava, 2025) the codebook is fitted once, before any rounding or compensation takes place.

Engine. Round-to-nearest (RtN) rounds each weight independently. An error-compensation engine $\mathrm{Engine}(W, \mathcal{G})$ instead sweeps the columns in a fixed order (in blocks of 128 columns for efficiency) and, after rounding column i, updates the columns not yet processed to compensate for the output error:

$$\epsilon = -\frac{W_{:,\,i} - Q_{\mathcal{G}}(W_{:,\,i})}{[H^{-1}]_{ii}}\, H^{-1}_{i,\,i+1:}, \qquad W_{:,\,i+1:} \leftarrow W_{:,\,i+1:} + \epsilon. \tag{3}$$

The values being rounded therefore change during the sweep. We use two engines: GPTQ (Frantar et al., 2023), and LDLQ (Chee et al., 2023), which generalizes GPTQ's update and which we run with QuIP's incoherence processing (we label this lane QuIP).

Starting point. Both engines calibrate a layer against activations of the unquantized model, whereas at inference the layer receives activations from already-quantized layers. QEP (Arai & Ichikawa, 2025) corrects the starting weights before rounding, $W^* = W + \alpha\, W\delta\hat{X}^\top\hat{H}^{-1}$, where $\hat{X}$ is the layer input in the partially quantized model, $\delta = X - \hat{X}$, $\hat{H} = \hat{X}\hat{X}^\top$ and $\alpha = \frac{1}{2}$. QEP moves where the sweep starts; the drift of (3) still occurs from W^*.

Notation. We distinguish four weight matrices: the original weights W, the QEP-corrected starting weights W^*, the *trajectory* W^+, whose column i holds the values of column i at the moment they are rounded, after absorbing the compensation of all preceding columns; and $W^{+\prime}$, the trajectory of a second pass. Standard pipelines fit the codebook on W or W^*; §4 fits it on W^+.

4 METHOD

4.1 THEORETICAL MOTIVATION: THE TRAJECTORY OBJECTIVE

To motivate what trajectory we fit our codebook to, we first rewrite the layer objective (1) in terms of the values the engine actually rounds. Following Chee et al. (2023), factor the Hessian as $H = (I + U)\, D\, (I + U)^\top$, with U strictly upper triangular and $D = \mathrm{diag}(d_1, \ldots, d_C)$, with $d_k > 0$ for all k.[1] For a codebook $\mathcal{G}$, let $\widehat{W}(\mathcal{G}) = \mathrm{Engine}(W^*, \mathcal{G})$ be the quantized matrix, and define the trajectory as the matrix of pre-rounding values

$$W^+(\mathcal{G}) \coloneqq W^* + \big(W^* - \widehat{W}(\mathcal{G})\big)\, U, \tag{4}$$

so that $\widehat{W}(\mathcal{G}) = Q_{\mathcal{G}}\big(W^+(\mathcal{G})\big)$ entrywise. Because U is strictly upper triangular, column k of $W^+(\mathcal{G})$ depends only on the rounding errors of columns $< k$, so the recursion is resolved by the sweep. With $E = \widehat{W}(\mathcal{G}) - W^*$ one has $\widehat{W}(\mathcal{G}) - W^+(\mathcal{G}) = E(I + U)$, and hence

$$\big\|\big(W^* - \widehat{W}(\mathcal{G})\big)X\big\|_F^2 = \mathrm{tr}\big(EHE^\top\big) = \sum_{r=1}^{R}\sum_{k=1}^{C} d_k\ \mathrm{dist}\big(W^+(\mathcal{G})_{rk},\ \mathcal{G}_r\big)^2 =: \mathcal{L}(\mathcal{G}), \tag{5}$$

where $\mathrm{dist}(z, \mathcal{G}_r) = \min_{c \in \mathcal{G}_r} |z - c|$. We verify (5) numerically on a dumped layer to seven significant digits (Appendix C.3). The layer objective is thus a d-weighted quantization error of the trajectory, not of W or W^*: a codebook fitted on W^* places its levels for values that the quantizer does not receive once the sweep has started.

Refitting on a frozen trajectory. Let $W^+ \coloneqq W^+(\mathcal{G}_1)$ be the trajectory of a first pass with grid $\mathcal{G}_1$, and let $\Phi(\mathcal{G} \mid W^+) \coloneqq \sum_{r,k} d_k\, \mathrm{dist}(W^+_{rk}, \mathcal{G}_r)^2$ be the loss of a grid on that trajectory held fixed, so that $\Phi(\mathcal{G}_1 \mid W^+) = \mathcal{L}(\mathcal{G}_1)$.

Proposition 1 (Refitting on a frozen trajectory) *If a refitted grid $\mathcal{G}_2$ minimizes the weighted error over a fixed trajectory W_1^+ such that $\Phi(\mathcal{G}_2 \mid W_1^+) \le \Phi(\mathcal{G}_1 \mid W_1^+)$, then:*

$$\Phi(\mathcal{G}_2 \mid W_1^+) \le \mathcal{L}(\mathcal{G}_1). \tag{6}$$

Proposition 1 identifies the trajectory as the tensor on which the objective is evaluated. It concerns a frozen trajectory and is not a guarantee for the rerun. A second pass with $\mathcal{G}_2$ produces a new trajectory $W^{+\prime} \coloneqq W^+(\mathcal{G}_2)$, and since $z \mapsto \mathrm{dist}(z, \mathcal{G})$ is 1-Lipschitz, Minkowski's inequality in the D-weighted norm (the norm that weights column k's squared entries by d_k before summing, the same weights as in the loss (5)) gives

$$\sqrt{\mathcal{L}(\mathcal{G}_2)} \le \sqrt{\Phi(\mathcal{G}_2 \mid W^+)} + \big\|W^{+\prime} - W^+\big\|_D. \tag{7}$$

This bound is loose in practice: on a replayed layer the trajectory moves between passes by 0.5–$1.3\sqrt{\mathcal{L}(\mathcal{G}_1)}$ in the D-norm, of the order of the loss itself, yet the refit lowers the true objective in all four measured cells, by 0.6–10.8% (Appendix C.3). Whether $\mathcal{L}(\mathcal{G}_2) < \mathcal{L}(\mathcal{G}_1)$ holds after the rerun is therefore an empirical question, which §5 answers at scale.

Weighting. Equation (5) weights column k by the pivot d_k, where $d_k = H_{kk} - h_k^\top H_{>k}^{-1} h_k$, and h_k and $H_{>k}$ denote the blocks of H for the columns rounded after k. Thus, H_{kk} represents the marginal diagonal curvature, whereas d_k is the corresponding conditional LDL decomposition pivot after accounting for the coupling to the remaining columns. Our implementation uses the simpler diagonal weight $\omega_j = H_{jj}$ of (2) as a deliberate approximation to the exact pivot-weighted objective. The condition of Proposition 1 still holds for this refit on the replayed layer: across the four (bit-width, block) cells we measure, $\Phi(\mathcal{G}_2 \mid W^+)/\mathcal{L}(\mathcal{G}_1)$ ranges from 0.938 to 0.994, i.e. the surrogate loss decreases in every cell, as the proposition guarantees. We test alternative codebook solvers — different column weightings, initializations, and k-means iteration budgets — on the same replayed layer in Appendix C.3.

[1] Under QEP, X and H are those of the partially quantized model ($\hat{X}$, $\hat{H}$). We omit the hats in this section.

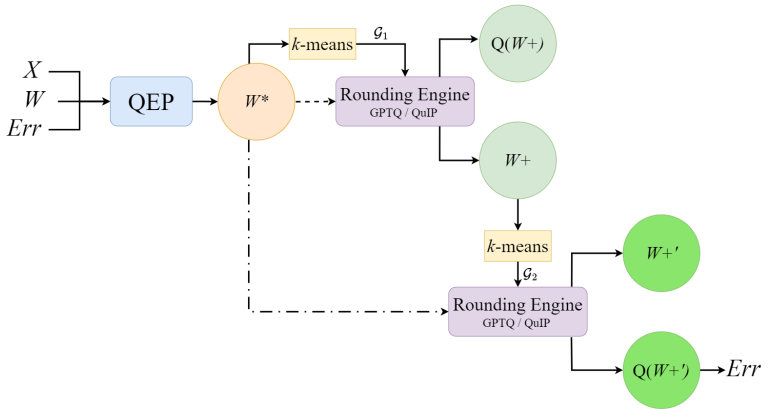


Figure 2: **The two-pass pipeline.** Pass 1 records the trajectory W^+ produced by the engine and discards its quantized output. The grid is refitted on W^+. Pass 2 restarts from the *same* starting weights W^* with the refitted grid $\mathcal{G}_2$. Only $\mathcal{G}_2$ is stored for inference.

Algorithm 1 2-Pass Trajectory Calibration.

Require: weights W, calibration inputs $\hat{X}$, inherited error δ (QEP), bit-width b, Engine GPTQ or LDLQ

1: $\hat{H} \leftarrow \hat{X}\hat{X}^\top; \quad W^* \leftarrow W + \alpha\, W\delta\hat{X}^\top\hat{H}^{-1}$ (starting weights, once)
2: $\omega \leftarrow \mathrm{diag}(\hat{H}); \quad \mathcal{G}_1 \leftarrow \mathrm{wkmeans}(W^*,\ \omega,\ 2^b)$ (initial grid on W^*)
3: $(\,\cdot\,,\ W^+) \leftarrow \mathrm{Engine}(W^*,\ \hat{H},\ \mathcal{G}_1)$ (pass 1: output discarded, trajectory kept)
4: $\mathcal{G}_2 \leftarrow \mathrm{wkmeans}(W^+,\ \omega,\ 2^b)$ (refit on the trajectory)
5: $(\widehat{W},\ \cdot\,) \leftarrow \mathrm{Engine}(W^*,\ \hat{H},\ \mathcal{G}_2)$ (pass 2: restart from the *same* W^*)
6: **return** $\widehat{W}$, codebook $\mathcal{G}_2$

4.2 Two-pass trajectory calibration (2PTC)

Motivated by (5), 2PTC fits the grid on the trajectory rather than on the starting weights (Figure 2, Algorithm 1):

1. **Starting point and initial grid.** The starting weights W^* are computed once (QEP, $W \to W^*$), and an initial codebook $\mathcal{G}_1$ is fitted on W^* with the weighted k-means in (2).
2. **Pass 1 (recording).** The engine runs with $\mathcal{G}_1$ and records the pre-rounding values, i.e. the trajectory W^+. Its quantized output is discarded.
3. **Refit.** The codebook is refitted on W^+ with the same weights, solver and initialization, resulting in $\mathcal{G}_2$.
4. **Pass 2 (final).** The engine restarts from the *same* starting weights W^* with $\mathcal{G}_2$ and returns $\widehat{W} = Q_{\mathcal{G}_2}(W^{+\prime})$.

Operational considerations. The second pass does not quantize W^+ directly: it reruns the full sweep from W^*, so inter-column compensation is preserved and only the grid changes. Only $\mathcal{G}_2$ and the quantized indices are stored. Since $\mathcal{G}_2$ has the same size as $\mathcal{G}_1$ (2^b levels per row), storage and inference are identical to those of a single-pass codebook, and the only cost is one additional offline pass per layer. We show that two passes suffice and an additional pass does not improve the results any further (Appendix D.3).

4.3 Applicability

2PTC changes only the tensor that the grid is fitted on, so it can be used on the fly with any engine that rounds column by column onto a learned grid. We evaluate 2PTC with GPTQ and LDLQ engines.

Table 1: **Main results, per-channel** (WikiText-2 perplexity, mean of 3 seeds. We highlight the best-performing method per lane in bold). Lanes: QEP+RtN, QEP+GPTQ, QEP+QuIP. "+2P" is the trajectory refit (2PTC). Seed-level standard deviations for the 2-bit rows: Table 11 (Appendix D.2). Group-wise results: Appendix D.4. The remaining configurations — engines run without QEP, the LeanQuant-style grid, the fitting-target ladder, and the third-pass refit — are reported in full in Table 9 (Appendix D.1).

		QEP + RtN		QEP + GPTQ			QEP + QuIP		
Model	FP16	unif.	codebk	unif.	codebk	**+2P**	unif.	codebk	**+2P**
4-bit									
Llama-2-7B	5.472	6.012	5.752	5.964	5.717	**5.689**	5.719	5.634	**5.625**
Llama-2-13B	4.884	5.166	5.134	5.114	5.110	**5.084**	5.071	**5.004**	5.005
Llama-2-70B	3.321	3.624	3.505	3.569	3.497	**3.487**	3.529	3.426	**3.425**
Llama-3-8B	6.136	7.967	6.906	10.17	6.791	**6.754**	6.834	6.535	**6.530**
Mistral-7B	5.252	5.868	5.520	5.537	5.491	**5.485**	5.444	**5.368**	5.371
Sheared-2.7B	6.906	7.411	7.203	7.289	7.155	**7.141**	7.153	**7.064**	7.068
3-bit									
Llama-2-7B	5.472	17.31	6.241	8.236	6.202	**6.178**	6.933	6.061	**6.041**
Llama-2-13B	4.884	7.521	5.587	6.259	5.509	**5.455**	5.940	5.308	**5.299**
Llama-2-70B	3.321	5.645	3.845	4.732	3.838	**3.821**	4.446	3.713	**3.706**
Llama-3-8B	6.136	86.73	8.626	38.39	8.310	**8.161**	9.973	7.518	**7.509**
Mistral-7B	5.252	10.02	6.069	7.351	6.001	**5.968**	6.428	5.713	**5.702**
Sheared-2.7B	6.906	12.35	7.817	9.234	7.704	**7.676**	8.362	7.521	**7.507**
2-bit									
Llama-2-7B	5.472	1.2×10^5	18.80	5 732	11.95	**10.55**	723.8	10.08	**9.534**
Llama-2-13B	4.884	84 299	124.8	2 375	13.09	**8.754**	94.90	7.942	**7.561**
Llama-2-70B	3.321	25 999	8.094	70.09	6.131	**5.807**	16.50	5.381	**5.192**
Llama-3-8B	6.136	7.2×10^5	438.6	2.8×10^5	41.18	**34.93**	3 262	17.40	**15.91**
Mistral-7B	5.252	53 808	42.70	1 622	15.83	**13.54**	61.68	9.366	**8.729**
Sheared-2.7B	6.906	1.2×10^5	27.43	1 389	16.58	**14.29**	93.92	12.37	**11.73**

The effects of the second pass are most pronounced under three conditions: At 2 bits, each row has only four levels, so levels placed for the wrong distribution are the costliest to get wrong. A learned codebook can also move its levels to where the trajectory concentrates its mass, whereas a uniform grid can only widen its range to span the tails of W^+ distribution, which costs resolution everywhere. In our experiments, the uniform two-pass degrades sharply (Appendix D.6). Finally, the refit is designed for per-channel codebooks: with group-wise codebooks ($g = 128$), we find that it is consistently harmful in our experiments (Appendix D.4).

5 Experiments and results

We evaluate 2PTC experimentally on six open-source LLMs, across multiple bit-widths and both compensation engines (GPTQ and QuIP's LDLQ). Models, protocol, hyperparameters, hardware, and the validation of every component against its reference implementation are detailed in Appendix A. A per-channel codebook stores 2^b half-precision values per row, an overhead of 0.016, 0.031 and 0.063 bits per weight for $C = 4096$ and $b = 2, 3, 4$, respectively. Following Arai & Ichikawa (2025), the results are compared between configurations that share the same engine (RtN, GPTQ and QuIP, each with QEP). Perplexity is measured on WikiText-2, calibration uses C4, and every table cell is the mean of three calibration seeds. Paired statistics are computed at the (model, seed) level on runs sharing the calibration sample, with 95% bootstrap confidence intervals (CIs) of the mean relative change and a one-sided Wilcoxon signed-rank test (Wilcoxon, 1945). Our method is labelled E8 (GPTQ) and Q8 (QuIP) in the appendix tables. This main configuration was fixed when the experimental grid was designed, before the ablations were run (Appendix A.1).

5.1 MAIN RESULTS

Table 1 reports the per-channel perplexity on the Wikitext-2 dataset for every model, bit-width, and engine. Compared with the single-pass codebook of the same engine, 2PTC lowers perplexity in all 24 cells at 2 and 3 bits (six models, two engines) and in 9 of the 12 cells at 4 bits, including all six experiments using GPTQ. The three remaining QuIP cells change by less than 0.1%. *The gain grows as the bit-width shrinks and peaks at the challenging setting of 2 bits*, where each row has only four levels, with perplexity improvement of 5–33% when using GPTQ (Llama-2-13B: $13.09 \rightarrow 8.75$) and 3.5–8.6% when using QuIP (Llama-3-8B: $17.40 \rightarrow 15.91$). Table 4 tests these differences seed by seed.

The baselines behave as expected. Learned codebooks outperform uniform grids in every cell (Kim et al., 2024; Zhang & Shrivastava, 2025), by up to $\sim 6800\times$ in perplexity for the 2 bit setting, where uniform grids are unusable on every model except for the Llama-2-70B model when using QuIP. QuIP outperforms GPTQ on every model and bit-width. QEP itself lowers perplexity in 70 of the 72 single-pass cells where we ran the engines both with and without it (Table 9). Because of this consistent benefit, our main results always include QEP.

The perplexity gains carry over to zero-shot accuracy (Table 2). 2PTC improves the average accuracy over ARC-Easy (Clark et al., 2018), PIQA (Bisk et al., 2020) and StoryCloze (Mostafazadeh et al., 2016) in 11 of 12 model–engine pairs, by up to 9.4 points: with GPTQ, Llama-2-13B climbs from 59.1% with the single-pass codebook to 68.5% (FP16: 79.3%). The only decrease is Llama-3-8B with QuIP ($65.1 \rightarrow 62.3$). With uniform grids, 2-bit accuracy stays close to chance (42–49%, chance 41.7%) except for Llama-2-70B with QuIP (58.0%), and the single-pass codebook recovers 45–87% of the gap to FP16. Per-task results at every bit-width are in Appendix D.9.

Within the Llama-2 family, larger models tolerate 2-bit quantization better. With 2PTC, the zero-shot gap to FP16 shrinks from 9.0 points for the 7B model to 6.8 for 13B and 2.6 for 70B with QuIP (11.4, 10.8 and 4.3 with GPTQ). On Llama-2-70B, GPTQ perplexity improves from 70.1 (uniform) to 6.13 (single-pass codebook) to 5.81 (2PTC). With QuIP, 2PTC reaches $\mathbf{5.19}$, against 3.32 for FP16. This is a 2-bit checkpoint of roughly 17.5 GB of nominal weight storage, and its zero-shot accuracy, $\mathbf{79.1}\%$, is within 0.2 points of FP16 Llama-2-13B.

Table 4 compares 2PTC with the same engine with a single-pass codebook, pair by pair (Figure 3 shows every pair). At 2 bits per channel the refit lowers perplexity in all 36 (model, seed) pairs: the mean change is -15.5% with GPTQ (95% CI $[-19.8, -11.7]$) and -5.7% with QuIP ($[-6.7, -4.9]$); both are significant under a one-sided Wilcoxon signed-rank test ($p < 10^{-4}$). At 3 and 4 bits the gains are smaller but systematic with GPTQ (-0.75% in 18 of 18 pairs and -0.35% in 17 of 18, both $p < 10^{-4}$) and at 3 bits with QuIP (-0.20%, 14 of 18, $p = 0.001$). With QuIP at 4 bits the refit leaves perplexity unchanged (-0.02%, $p = 0.26$).

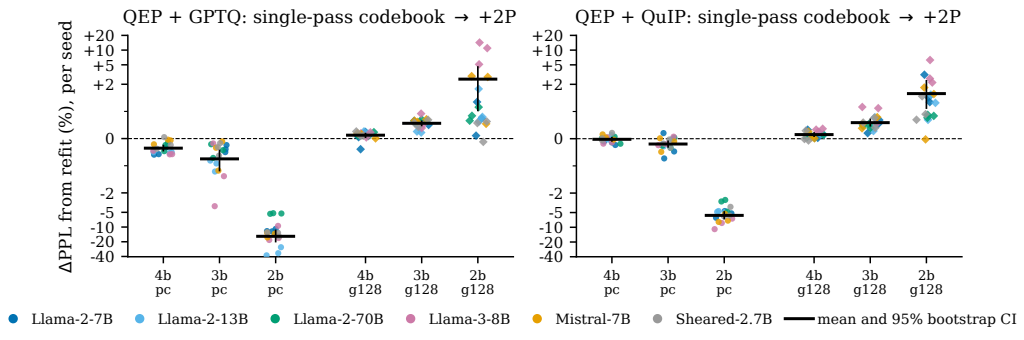


Figure 3: **Effect of the refit per seed** (relative change in WikiText-2 perplexity from the single-pass codebook to 2PTC in the same experiment group; symlog scale). Circles: per-channel; diamonds: g=128. Black: mean and 95% bootstrap CI over the 18 pairs. The effect is largest at 2 bits per channel; with 128-column groups the refit yields slightly higher perplexity at every width.

Table 2: **Zero-shot accuracy at 2 bits, per-channel** (% average of ARC-Easy, PIQA and StoryCloze. Mean of 3 seeds; chance ≈41.7%; bold = best quantized method per experiment group). Per-task tables at all widths: Appendix D.9.

		QEP + GPTQ			QEP + QuIP		
Model	FP16	unif.	codebk	**+2P**	unif.	codebk	**+2P**
Llama-2-7B	77.4	43.7	64.6	**66.0**	45.9	67.9	**68.4**
Llama-2-13B	79.3	42.5	59.1	**68.5**	48.8	71.9	**72.5**
Llama-2-70B	81.7	49.1	76.7	**77.4**	58.0	78.7	**79.1**
Llama-3-8B	79.6	42.2	63.4	**63.5**	43.2	**65.1**	62.3
Mistral-7B	80.3	43.5	63.8	**66.1**	47.9	70.5	**71.2**
Sheared-2.7B	72.8	44.4	58.1	**60.0**	46.0	62.3	**63.2**

Table 3: **Ablating the fitting target.** WikiText-2 perplexity (mean over 6 models and 3 seeds, per-channel) when the *only* change is the tensor the codebook is fitted on; ω, codebook solver, engine, calibration and rotation are identical. Seed-level: Table 13.

bits	on W	on W^*	on W^+	$W^* \to W^+$
GPTQ				
4	5.630	5.627	**5.607**	-0.36%
3	6.264	6.261	**6.210**	-0.81%
2	21.646	17.462	**14.645**	-16.14%
QuIP				
4	5.504	5.505	**5.504**	-0.02%
3	5.971	5.972	**5.961**	-0.19%
2	10.229	10.423	**9.775**	-6.22%

Table 4: **The refit at seed level**: single-pass codebook → 2PTC in the same experiment group, per-channel. Each pair is one (model, seed) with the same calibration sample ($n = 18$ per row). Mean relative change in WikiText-2 perplexity with 95% bootstrap CI, median, range, and one-sided Wilcoxon signed-rank p (H_1: refit lower). Group-wise (g=128) rows are shown in Table 12.

bits	wins	mean Δ	95% CI	median	min	max	p
QEP + GPTQ							
4	17/18	-0.35%	[-0.44, -0.26]	-0.41%	-0.59%	+0.05%	$<10^{-4}$
3	18/18	-0.75%	[-1.18, -0.44]	-0.46%	-3.74%	-0.11%	$<10^{-4}$
2	18/18	-15.47%	[-19.83, -11.70]	-13.11%	-37.91%	-5.21%	$<10^{-4}$
QEP + QuIP							
4	10/18	-0.02%	[-0.08, +0.03]	-0.02%	-0.23%	+0.21%	0.261
3	14/18	-0.20%	[-0.30, -0.09]	-0.20%	-0.73%	+0.21%	0.001
2	18/18	-5.71%	[-6.65, -4.87]	-5.25%	-10.92%	-2.76%	$<10^{-4}$

5.2 Ablation studies

Ablating the fitting target. To isolate which weight tensor is responsible for 2PTC's perplexity improvement over a single-pass codebook, we hold the weights ω, solver, engine, calibration and rotation fixed and change only the tensor the grid is fitted on (Table 3; seed-level statistics in Table 13). Moving the fit from W to the QEP-corrected W^* yields no consistent improvement: the proposed method outperforms the baselines in at most 10 of 18 pairs at any width on either engine and is never significant ($p \geq 0.08$). The largest mean change when applying the proposed method, -9.3% with GPTQ at 2 bits, comes from a minority of pairs (7 of 18 wins; CI $[-22.1, +1.0]$), and with QuIP at 2 bits perplexity increases slightly instead ($+1.5\%$). Moving the fit from W^* to the trajectory W^+, in contrast, lowers perplexity at every width with GPTQ (17–18 of 18 pairs, $p < 10^{-4}$) and at 2 and 3 bits with QuIP (18 and 14 of 18). Since W^* differs from W but fitting on it does not help, the gain is not explained by fitting on a more recently corrected tensor, it is specific to the trajectory. Nor does it require QEP: without it, the refit gains as much (Appendix D.5).

Ablating grid design: weighting and initialization. Our grid differs from LeanQuant's (Zhang & Shrivastava, 2025) in two components, which we ablate one at a time inside our pipeline (Appendix D.7, D.8)): the weighting, $(1/[H^{-1}]_{jj})^p$ instead of $\mathrm{diag}(H)_j$, and the initialization, equispaced instead of row quantiles. We implement the weighting with the hyperparameter $p = 4$ as used in Zhang & Shrivastava (2025), and $p = 3$. We call the result LeanQuant-style. Our $\mathrm{diag}(H)$ weighting for a fixed initialization in all 18 models tested–the refit destabilizes width cells, and the LeanQuant-style grid: at 2 bits, 2PTC outperforms the LeanQuant-style grid with two passes on every model with both engines (mean perplexity 14.6 against 40.9 with GPTQ, 9.8 against 24.1

with QuIP). The equispaced initialization, conversely, is better at fixed weighting in 17 of the 18 model–width cells (all six at 4 bits and at 3 bits, five of six at 2 bits).

Interaction between initialization and refit. We also test our weighting with the equispaced initialization, keeping the refit (E16 in the appendix tables). This combination improves on our main configuration at 3–4 bits, winning all 12 model–width cells. The gain comes largely from a better single-pass grid. At 2 bits, the regime where the refit matters most, the two initializations tie with GPTQ (9 of 18 pairs, median $+0.1\%$, with two divergent E16 seeds) and our quantile initialization is better with QuIP (13 of 18 pairs). We therefore keep the pre-specified quantile initialization as the main configuration and report E16 as the best-performing variant at 3–4 bits.

Ablating the codebook solver budget. 2PTC calls the codebook solver — the weighted k-means that fits each row's levels — twice: once to fit $\mathcal{G}_1$, and once more to fit $\mathcal{G}_2$ on the trajectory. So part of its gain might be related to the extra solver work rather than the trajectory. We test this on one layer (`mlp.gate_proj` of Llama-2-7B, first and last blocks) by replaying the quantization and measuring the layer objective $\mathcal{L}$ of (5) directly (Appendix C.3). At 2 bits, a single pass with 200 k-means iterations, five times the iterations of 2PTC, changes $\mathcal{L}$ by at most 0.2% because the 20-iteration solver has already converged, whereas the refit, in contrast, lowers $\mathcal{L}$ by 10.8% and 1.3% in the two blocks, and keeps that margin when both passes use 200 iterations. On this layer, the 2-bit gain is therefore not a solver-budget effect. At 4 bits the 20-iteration solver has not converged: a longer single pass lowers $\mathcal{L}$ more than the refit does, and with 200 iterations in both passes the refit still helps in the first block and ties in the last.

Ablating granularity. Our main results use per-channel codebooks, whose storage overhead is negligible. With one codebook per 128 columns, which adds 0.5 to 2 bits per weight, the refit does not help: it raises perplexity slightly at every width ($+0.1\%$ to $+2.6\%$; Table 12). Such group-wise codebooks are in any case Pareto-dominated by per-channel codebooks with 2PTC once storage is counted (Figure 6). Appendix D.4 reports the group-wise results in full and discusses a possible explanation.

6 Conclusion

Learned codebooks in compensation-based PTQ are fitted on the weights the engine starts from, while the engine rounds a trajectory of progressively compensated values. 2PTC records that trajectory, refits the codebook on it, and reruns the same engine from the same starting point. The correction is simple, composes with inter-layer correction, and is free at inference. Its effect is largest where the grid is most constrained: for per-channel codebooks at 2 bits it lowers perplexity in every paired run, by 15.5% with GPTQ and 5.7% with LDLQ on average, and improves zero-shot accuracy in 11 of 12 model–engine pairs. At 3 and 4 bits its gains are smaller but systematic with GPTQ, and at 3 bits with LDLQ. Controlled ablations attribute the gain to the trajectory itself rather than to a more corrected weight tensor, and the analysis identifies the trajectory as the argument of the engine's objective. Because 2PTC changes only what the grid is fitted on, it can be applied as is to other column-wise engines; evaluating it there, with the LDL-prescribed weighting that already improves the objective at layer scale, and in combination with compensation-aware updates (Li et al., 2026) are natural next steps.

7 Limitations

Our evaluation targets the 2-bit regime, where post-training quantization error is largest and codebook design therefore has the most room to help: learning a codebook on the observed compensation trajectory yields consistent gains precisely because the rounding error it must absorb is large (§5.1). At 3 and 4 bits that error is already small, so a better-fitted grid has correspondingly less to correct, and the improvement shrinks accordingly—vanishing with QuIP at 4 bits. The same reasoning explains why grouped codebooks (g=128) and a uniform, non-learned grid are not suitable substitutes: neither can represent the trajectory's structure finely enough for a correction to matter (Appendices D.4, D.6).

LLM USAGE STATEMENT

In this work, we used generative AI (large language model) assistants for writing and debugging the code used in our experiments, for drafting and editing prose, and for reviewing experimental results. All LLM-generated code was verified and tested by the authors, and all experimental design decisions, result interpretation, and final text were reviewed by the authors and are their responsibility. No LLM is an author of this work. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We describe the full experimental protocol — models, calibration and evaluation datasets, hyperparameters, and hardware — in Appendix A, with all hyperparameters and random seeds listed in Table 5 (Appendix A.2). Every component of our implementation is validated against its reference implementation in the same appendix. Calibration seeds are fixed and reported per configuration, but bitwise reproducibility across independent executions does not hold, since the weighted k-means uses non-deterministic GPU kernels (Appendix A). The paired statistics we report for every comparison (bootstrap confidence intervals and Wilcoxon tests) are computed across independent executions and therefore already include this variability. Code to reproduce all experiments will be released upon acceptance.

REFERENCES

Yamato Arai and Yuma Ichikawa. Quantization error propagation: Revisiting layer-wise post-training quantization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

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

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

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

Vage Egiazarian, Andrei Panferov, Denis Kuznedelev, Elias Frantar, Artem Babenko, and Dan Alistarh. Extreme compression of large language models via additive quantization. In *International Conference on Machine Learning (ICML)*, 2024.

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

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

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

Babak Hassibi and David G. Stork. Second order derivatives for network pruning: Optimal brain surgeon. In *Advances in Neural Information Processing Systems (NIPS)*, 1992.

Itay Hubara, Yury Nahshan, Yair Hanani, Ron Banner, and Daniel Soudry. Improving post training neural quantization: Layer-wise calibration and integer programming. *arXiv preprint arXiv:2006.10518*, 2020.

Itay Hubara, Yury Nahshan, Yair Hanani, Ron Banner, and Daniel Soudry. Accurate post training quantization with small calibration sets. In *International Conference on Machine Learning (ICML)*, 2021.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

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

Yann LeCun, John S. Denker, and Sara A. Solla. Optimal brain damage. In *Advances in Neural Information Processing Systems (NIPS)*, 1989.

Shuaiting Li, Juncan Deng, Kedong Xu, Rongtao Deng, Hong Gu, Minghan Jiang, Haibin Shen, and Kejie Huang. Rethinking residual errors in compensation-based LLM quantization. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2604.07955.

Yuhang Li, Ruokai Yin, Donghyun Lee, Shiting Xiao, and Priyadarshini Panda. GPTAQ: Efficient finetuning-free quantization for asymmetric calibration. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 36690–36706, 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 on-device LLM compression and acceleration. In *Proceedings of Machine Learning and Systems (MLSys)*, 2024.

Yifei Liu, Jicheng Wen, Yang Wang, Shengyu Ye, Li Lyna Zhang, Ting Cao, Cheng Li, and Mao Yang. VPTQ: Extreme low-bit vector post-training quantization for large language models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2024.

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

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of NAACL-HLT*, 2016.

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

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)*, 2020.

Hugo Touvron, Louis Martin, Kevin Stone, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Albert Tseng, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa. QuIP#: Even better LLM quantization with Hadamard incoherence and lattice codebooks. In *International Conference on Machine Learning (ICML)*, 2024.

Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945.

Mengzhou Xia, Tianyu Gao, Zhiyuan Zeng, and Danqi Chen. Sheared LLaMA: Accelerating language model pre-training via structured pruning. In *International Conference on Learning Representations (ICLR)*, 2024.

Tianyi Zhang and Anshumali Shrivastava. LeanQuant: Accurate and scalable large language model quantization with loss-error-aware grid. In *International Conference on Learning Representations (ICLR)*, 2025.

A EXPERIMENTAL SETUP AND IMPLEMENTATION

A.1 PROTOCOL, MODELS AND REPORTING

Protocol. Calibration always uses C4 (Raffel et al., 2020) (128 sequences of 2048 tokens (Hubara et al., 2021)); evaluation uses WikiText-2 perplexity (Merity et al., 2017) with the standard 2048-token sliding evaluation, and zero-shot accuracy on ARC-Easy (Clark et al., 2018), PIQA (Bisk et al., 2020) and StoryCloze (Mostafazadeh et al., 2016). Calibration and evaluation data are never drawn from the same corpus.

Models and grid. Llama-2-7B/13B/70B (Touvron et al., 2023), Llama-3-8B (Grattafiori et al., 2024), Mistral-7B (Jiang et al., 2023) and Sheared-LLaMA-2.7B (Xia et al., 2024). All are released in 16-bit precision except Sheared-LLaMA-2.7B, whose published checkpoint is `float32`; we include it so that the conclusions are not drawn exclusively from checkpoints that have been through a half-precision cast. It is also the only pruned-and-retrained model in the set, and the smallest. Bit-widths and granularities: $b \in \{4, 3, 2\}$, per-channel and group-wise (g=128), three calibration seeds per configuration. Every table cell is the **mean over the three seeds**; paired statistics are computed at the (model, bits, granularity, seed) level.

Statistics. For a comparison between two configurations $a \to b$, a pair is one (model, seed) cell run with the same calibration sample, and the statistic is the relative change $100\,(\mathrm{PPL}_b - \mathrm{PPL}_a)/\mathrm{PPL}_a$. We report the number of pairs in which b is lower, the mean relative change with a 95% percentile-bootstrap confidence interval (20,000 resamples over the pairs), the median and range, and the one-sided Wilcoxon signed-rank p-value for $\mathrm{PPL}_b < \mathrm{PPL}_a$. With six models and three seeds there are $n = 18$ pairs per (engine, bits, granularity) row, fewer when a member of a pair has no finite perplexity or did not complete, in which case n is reported; "wins" alone is not a test, which is why the interval is the primary quantity. The two members of a pair share the calibration sample but are independent executions of the k-means, so the intervals include its GPU non-determinism.

Lanes. Following QEP's reporting style, we organize comparisons as *lanes* — RtN, GPTQ and QuIP, each preceded by the QEP correction — and ask what each ingredient contributes *within* a lane. GPTQ and QuIP are also run without QEP, single-pass (cases E3, E4, Q3 and Q4 in Table 9) and, as an ablation, with the refit (E8-noQEP and Q8-noQEP, Appendix D.5); every other configuration of 2PTC, including the main one, is run with QEP. Cross-lane statements are made only when explicitly marked as such.

Effective bits. Comparisons account for parameter storage: $\mathrm{bits}_{\mathrm{eff}} = b + \text{overhead}$. A per-channel codebook stores 2^b half-precision values per row of C weights, i.e. $2^b \cdot 16/C$ bits/weight: 0.016, 0.031 and 0.063 at $b = 2, 3, 4$ for $C = 4096$. Group-wise codebooks store the same 2^b values per 128 weights: 0.5, 1.0 and 2.0 bits/weight, the basis of Appendix D.4.

Naming convention. Configurations are labelled with a short code: **E**n for cases built on the GPTQ engine and **Q**n for cases built on QuIP's LDLQ, with matching numbers denoting the same configuration on the two engines up to 11; Q12 and Q13 are the QuIP counterparts of E15 and E16. **E8 and Q8 are our method** — QEP, a learned codebook, and the trajectory refit. Rows added during revision keep the number of the configuration they modify and carry a suffix instead of a new number: -noQEP disables the QEP correction (E8-noQEP, Q8-noQEP), and the superscript p4 replaces the LeanQuant-style weighting by LeanQuant's weighting as published (E10$^{\mathrm{p4}}$, E11$^{\mathrm{p4}}$, E13$^{\mathrm{p4}}$, Q10$^{\mathrm{p4}}$, Q11$^{\mathrm{p4}}$). The tag names the published exponent, but the weight is a different quantity, not the same one at another exponent (Appendix A.2). Tables 9 and 10 list every code with its description; the revision rows are per-channel only.

Pre-specification. The 18 cases of the grid (E1–E12, Q3–Q8), and E8/Q8 as the main configuration, were fixed when the grid was designed. The LeanQuant decomposition (E13, E14, Q10, Q11, and E16, Q13 with the equispaced initialization) and the ladder (E15, Q12) were added afterwards as ablations, and the rows without QEP and with LeanQuant's weighting as published (-noQEP, p4) during revision; the main configuration was not changed in response to any of them (§5.2).

Validation, in three separate senses. (i) *Equivalence with the reference*: the GPTQ engine is an adaptation of the reference implementation whose only change is a pluggable rounding step, and with uniform rounding its output is bit-identical to that reference (largest absolute difference over all weights exactly zero); the same holds for the uniform quantizer, the RtN path and the two-

pass wrapper. The QuIP rotation is orthogonal to 3×10^{-7} and round-trips W to 2.9×10^{-6}, and the batched zero-shot evaluator agrees with its unbatched reference to 0.0 on all 11 tasks; 126 numerical and zero-shot checks pass. (ii) *Determinism of the replay*: given the stored W^*, codebooks and H^{-1}, a GPTQ pass can be replayed offline exactly (Appendix C.1). (iii) *Bitwise reproducibility across independent runs*: this does *not* hold, because the weighted k-means uses non-deterministic GPU kernels; the paired statistics of §5 are computed across independent executions and include this variability.

A.2 Implementation details and hyperparameters

Codebase. The pipeline is built directly on the reference QEP codebase, which itself extends the GPTQ reference implementation. The quantization engine keeps the reference GPTQ helper unchanged in every numerical respect — same Hessian damping (1% of the mean diagonal), same Cholesky factorization, same block size (128 columns) — and modifies only the rounding step, which becomes pluggable (uniform, codebook, or codebook-with-refit). The GPTQ Hessian factorization is performed in FP32 throughout. And the QEP correction $W \rightarrow W^*$ ($\alpha = \frac{1}{2}$, `mlp.down_proj` exempt) is called from the reference implementation unchanged.

Weighted k-means (per row). Initialization by empirical quantiles of the row; assignment by `searchsorted` against level midpoints; update by ω-weighted means; empty clusters retain their level; all rows of a layer solved in parallel on GPU; at most 20 iterations, stopping earlier when the levels change by less than a relative 10^{-6}. Weights $\omega_j = H_{jj}$. The same solver, with the same budget, fits $\mathcal{G}_1$ and $\mathcal{G}_2$; Appendix C.3 examines what a larger budget buys at layer scale.

LeanQuant grids. LeanQuant (Zhang & Shrivastava, 2025) initializes the k-means on 2^b equispaced points over the row's range and weights it by an OBS loss-error term raised to a power p. We implement that weight in two forms, which are different quantities, not one weight at two exponents; both are computed from the same damped H the engine factorizes, in the engine's coordinates. *LeanQuant-style* (E10, E11, E13, Q10, Q11) follows their Eq. (6) as written, $\omega_j = (1/[H^{-1}]_{jj})^p$ with $p = 3$, where $1/[H^{-1}]_{jj} = H_{jj} - h_{-j}^\top H_{-j}^{-1} h_{-j}$ is the curvature of column j once *all* other columns have compensated (h_{-j} and H_{-j} being the blocks of H for the other columns). *As published* (the p4 rows) follows their Algorithm 1, which evaluates the weight on the Cholesky factor of the damped H^{-1} that GPTQ uses, $\omega_j = [\mathrm{chol}(H^{-1})]_{jj}^{-p}$ with their $p = 4$. Since $[\mathrm{chol}(H^{-1})]_{jj}^{-2} = d_j$, the pivot of (5) (Appendix C.3), this weight is d_j^2: the curvature of column j once only the columns rounded *after* it have compensated, squared. We did not run the released code; the comparisons in Appendices D.7 and D.8 are between components implemented in our pipeline.

QuIP incoherence. Random butterfly orthogonal matrices U, V are constructed on GPU, and the rounding step reuses the validated GPTQ engine, which is exact since LDLQ generalizes GPTQ. Two deliberate, documented differences from the reference QuIP code: $\mathrm{diag}(W^\top W)$ is computed as $\sum_r (W \odot W)_r$. (identical result, avoids a 3.3 GB allocation on the 70B), and rounding uses our per-row quantizer rather than the global $2.4 \cdot \mathrm{rms}$ scale of the reference `qfn=b`. The codebook is fitted, and the trajectory recorded, in the rotated coordinates in which LDLQ runs; the analysis of §4.1 is written for the GPTQ/LDLQ update in whatever coordinates the engine operates, and we make no claim about invariance of the codebook itself under the rotation. In the ladder experiments the original W travels through the *same* rotation as W^*, so that the comparison isolates the fitting target.

Numerical precision. The 70B is quantized in `bfloat16`: in `float16` the calibration forward overflows at 80 layers and produces `nan`. All other models use `float16`. Hessian and activation statistics therefore differ in precision between the 70B and the rest, and we have not checked whether any of the scale trend in Table 1 is attributable to that.

Calibration and evaluation. C4, 128 sequences $\times$ 2048 tokens, with seeds 2026, 1530 and 2307 controlling the calibration sample; WikiText-2 perplexity with the standard sliding evaluation; zero-shot scoring with a batched length-normalized evaluator.

Hardware and cost. Every cell runs as an independent job on a single NVIDIA GH200 Grace Hopper Superchip with a 96 GB HBM3 Hopper GPU with 8 CPU cores; host memory 96 GB for the 7–8B models, 128 GB for the 13B and 360 GB for the 70B. A job quantizes one (model, width,

granularity, seed) cell with all the cases of its family, reloading the model for each case so that case order cannot influence the result. Wall-clock is 1–7 hours per job for the 7–13B models and up to 24 hours for the 70B. The complete grid is $\sim$650 jobs; this is the cost of the *ablation*, not of the method, which costs two engine passes.

Table 5: Hyperparameters (identical across all lanes unless stated).

Hessian damping	1% of mean diagonal
GPTQ block size	128 columns
QEP α	$\frac{1}{2}$ (`down_proj` exempt)
Calibration	C4, 128 seq. $\times$ 2048 tokens
Seeds	2026, 1530, 2307
k-means weight ω	$\mathrm{diag}(H)$
k-means init / budget	row quantiles (LeanQuant-style cases: equispaced) / 20 iterations
LeanQuant-style weight exponent	$p = 3$ or $p = 4$ (specified)
Bit-widths / granularity	$\{4, 3, 2\}$ / per-channel and g=128

B Extended related work

Hessian-guided compression. Second-order information has been used to decide how to perturb a trained network since Optimal Brain Damage (LeCun et al., 1989), which prunes weights by a saliency computed from a diagonal approximation of the Hessian; Optimal Brain Surgeon (Hassibi & Stork, 1992) drops the diagonal assumption. Optimal Brain Compression (Frantar & Alistarh, 2022) made that update tractable for layer-wise post-training compression, and GPTQ (Frantar et al., 2023) scaled it to LLMs with a fixed column order, block-wise updates and a single Cholesky factorization in place of repeated inversions. In parallel, AdaRound (Nagel et al., 2020) and AdaQuant (Hubara et al., 2021) established that nearest rounding is not optimal for the layer output; both predate the LLM scale, but the small-calibration-set protocol they introduced (Hubara et al., 2020) is the one we inherit.

Incoherence processing. QuIP (Chee et al., 2023) multiplies weights and Hessian by random orthogonal matrices so that the largest entry is bounded relative to the average and the outliers dominating the quantization error are spread across coordinates. QuIP also introduces LDLQ, a rounding algorithm derived from an LDL decomposition of the Hessian which provably generalizes GPTQ; LDLQ is the second engine in our experiments, run with the incoherence preprocessing that accompanies it. QuIP# (Tseng et al., 2024) obtains the same guarantees with randomized Hadamard transforms at $O(n \log n)$ and pairs them with E_8 lattice codebooks and end-to-end fine-tuning.

Non-uniform grids. SqueezeLLM (Kim et al., 2024) fits a per-row grid by k-means weighted by Fisher information and stores a small set of extreme outliers separately in a dense-and-sparse decomposition. LeanQuant (Zhang & Shrivastava, 2025) keeps the weighted k-means but replaces the Fisher weight by the OBS loss error raised to a power p — written $(1/[H^{-1}]_{jj})^p$ in their Eq. (6) and evaluated on the Cholesky factor of H^{-1} in their Algorithm 1 (Appendix A.2) — and initializes the k-means on an equispaced grid; the two are presented together, and we separate them in Appendices D.7 and D.8. VPTQ (Liu et al., 2024) leaves the scalar setting altogether and quantizes vectors of weights jointly against a learned codebook, with fine-tuning of the centroids and normalization layers; such vector quantizers are complementary to the scalar grids we study. What all scalar codebook methods share, and what we change, is that the grid is fitted on the weights the layer starts from. Orthogonally, AWQ (Lin et al., 2024) protects salient channels by folding a per-channel scale into the surrounding operations rather than by compensating error.

Inter-layer and intra-layer target drift. GPTAQ (Li et al., 2025) intervenes in the objective, minimizing $\|WX - \widehat{W}\widehat{X}\|$ with the full-precision input on one side and the quantized-model input on the other, so that the sweep compensates the inherited error as it advances. Li et al. (2026) observe that this is not the only place the target drifts: at each step, GPTQ and GPTAQ align against the output of the *already compensated* weights rather than of the original layer, and they add a compensation-aware term $(W^{(0)} - W^{(q)})\widetilde{X}$ to the update, evaluated with uniform group-wise (g=128) grids and composed with rotations. Their premise and ours coincide — the compensated weights are no longer the weights the method was fitted for — and the interventions are on different objects: they

correct what the update is aligned to, we refit the grid on the drifted values. The two should compose; we have not measured the composition. QEP (Arai & Ichikawa, 2025) makes the inter-layer correction explicit and prior to quantization: two activation streams, one clean and one quantized, are propagated in parallel, and before the engine touches the layer QEP solves for a corrected W^* minimizing the layer output error given the error already accumulated upstream. We build on QEP unchanged, taking W^* as the starting point of both engines and inheriting its lane-based reporting. Each of these methods intervenes on the weights while the quantization grid stays fixed and fitted in advance; 2PTC leaves the weights alone and moves the grid, and the ablation of the fitting target (§5.2) separates the two effects.

C LAYER-LEVEL ANALYSIS

This appendix looks at single weight matrices. §C.1 measures what the two passes do to one matrix; §C.2 derives, from the layer-wise error bound of Chee et al. (2023), why the QEP correction grows with depth; §C.3 replays both passes on the dumped matrix under alternative codebook solvers, which is as far as this paper takes the weighting question of §4.1. Everything here comes from one matrix of one model, dumped from a single run: it makes the mechanism visible and tests the alternatives at layer scale; the quantitative claims of the paper rest on §5.

C.1 THE FOUR STATES OF A WEIGHT MATRIX

We dump W, W^* and W^+ — and reconstruct a fourth state, $W^{+\prime}$ — for a single dense matrix of Llama-2-7B (`mlp.gate_proj`, 11008×4096, 4-bit per-channel codebook), at the first and the last transformer block. Figure 4 plots one output channel (the row whose L_2 norm is the median of the matrix, a rule fixed before looking at any histogram); every number below is measured on the full matrix.

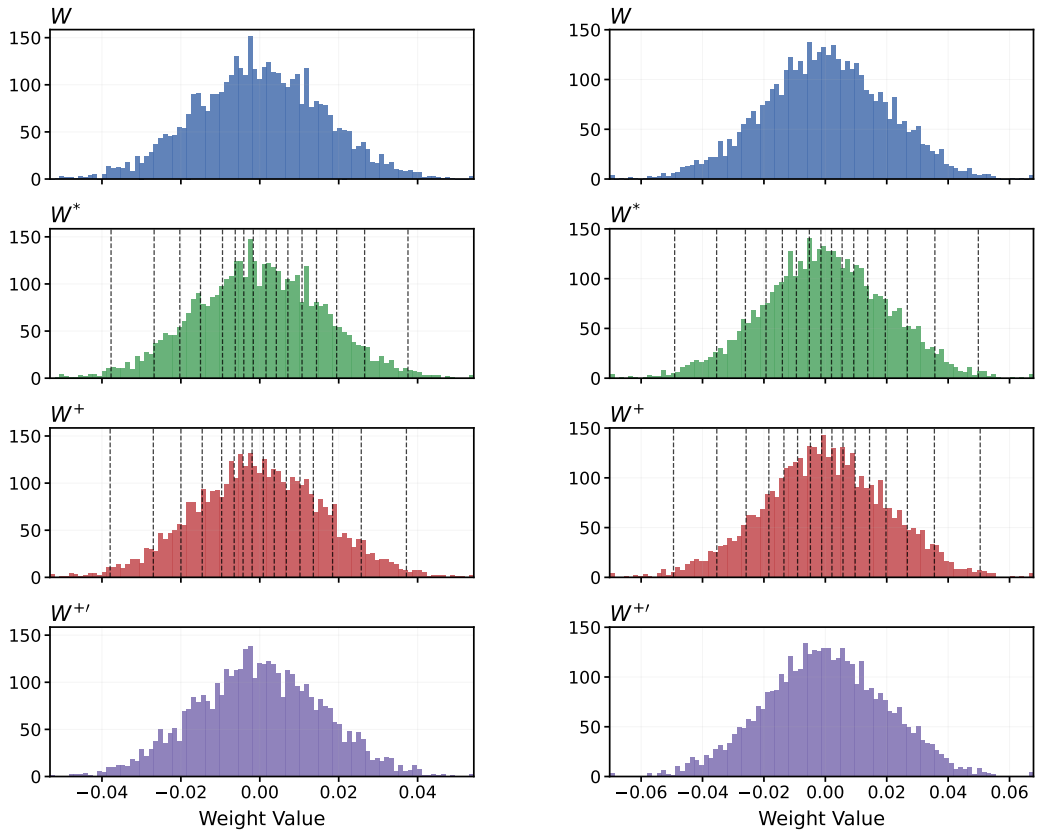
The moments barely move; the weights do. Across both blocks no variance changes by more than 2.2%, while relative displacements reach 10.3% (Table 6). The passes move individual weights, not the ensemble — the kind of change a second-order summary cannot see and a codebook refit can exploit.

The QEP correction is the only displacement that grows with depth. Every other quantity in Table 6 shrinks between block 0 and block 31. The QEP displacement goes from 1.07% to 1.84%, as the bound of §C.2 predicts: the size of the correction is governed by $\|W_l \delta_l\|$, and the inherited mismatch δ_l accumulates with depth. The decay of the compensation displacement we report without explanation; two blocks of one matrix establish no relationship between the two trends.

The refit is not a rescaling. Averaged over all 11008 channels, in units of each channel's own $\sigma(W^*)$, the centroids move 4.86% between the two passes at block 0 and 2.37% at block 31. Scaling each first-pass grid by its own $\sigma(W^+)/\sigma(W^*)$ — the best a pure dilation can do — leaves 98% of that movement in place (4.74% and 2.33%). The second pass relocates levels rather than scaling them.

Directions. Define $\Delta_{\mathrm{QEP}} := W^* - W$, $\Delta_1 := W^+ - W^*$, $\Delta_2 := W^{+\prime} - W^*$ and $\Delta_{\mathrm{refit}} := W^{+\prime} - W^+ = \Delta_2 - \Delta_1$, the last being the tensor whose D-norm enters the bound (7). Table 7 gives the angles between them. Every angle involving Δ_{QEP} lies between 88.9° and 91.8°: the inter-layer correction and the intra-layer sweep draw on disjoint information ($\delta = X - \hat{X}$ on one side, this layer's rounding errors propagated by H^{-1} on the other) and move the weights in very nearly orthogonal directions, which is consistent with their gains being additive. The two passes, which share the sweep and differ only in the grid, are 64° and 51° apart with $\|\Delta_2\|$ within 1% of $\|\Delta_1\|$: the refit changes the destination of the trajectory rather than its length. We report these as measurements on one matrix and draw no further geometric conclusion from them.

Where $W^{+\prime}$ comes from, and what is exact. Three of the four states are dumped during quantization; $W^{+\prime}$ is reconstructed offline. GPTQ is row-independent given the inverse Hessian — the update $W_{:,i:} \mathrel{-}= e \otimes H^{-1}_{i,i:}$ never mixes rows — so W^*, the codebooks and H^{-1} suffice to replay a pass, and replaying pass 1 under this procedure reproduces the stored W^+ bit for bit on both blocks. That exactness is a property of the replay (deterministic arithmetic on stored inputs), not of the pipeline (Appendix A.1, sense (iii)): between two executions of the identical configuration the



(a) **Block 0** (channel 3795). The activations entering this block are exact, so QEP has almost nothing to correct: W^* is a 1.1% displacement of W. The grid moves visibly between the two passes.

(b) **Block 31** (channel 10900). Behind 31 already-quantized blocks, QEP has real inherited error to absorb and its displacement grows by 72% while GPTQ's shrinks.

Figure 4: **The four states of a weight matrix, and the grids fitted on them.** One output channel of `mlp.gate_proj` in Llama-2-7B at 4 bits, per-channel, first and last transformer block. From top to bottom: W; W^* with the 16 centroids of the first pass (dashed); W^+ with the centroids of the second pass, refitted on W^+; and $W^{+\prime}$, on which no grid is fitted. Both codebooks are the H_{jj}-weighted k-means the method uses. To the eye the four distributions are almost the same — variance changes by at most 2.2% — yet the weights travel up to 10% in relative norm (Table 6).

Table 6: **Moments and displacement of the four states.** Relative change of the variance with respect to W and relative L_2 displacement of each step. "channel" is the single row plotted in Figure 4; "matrix" is the full 11008×4096 tensor (variance ratios on the 2×10^6 entries retained by the dump, displacements and angles on every weight).

		block 0		block 31	
		channel	matrix	channel	matrix
$\mathrm{Var}(W^*)/\mathrm{Var}(W) - 1$		-0.01%	$+0.28\%$	-0.40%	-0.56%
$\mathrm{Var}(W^+)/\mathrm{Var}(W) - 1$		$+1.61\%$	$+2.22\%$	$+0.05\%$	$+0.03\%$
$\mathrm{Var}(W^{+\prime})/\mathrm{Var}(W) - 1$		$+1.58\%$	$+2.14\%$	$+0.37\%$	-0.00%
$\|W^* - W\|/\|W\|$	(QEP)	0.72%	1.07%	1.45%	1.84%
$\|W^+ - W^*\|/\|W^*\|$	(pass 1)	9.06%	9.93%	6.33%	6.09%
$\|W^{+\prime} - W^*\|/\|W^*\|$	(pass 2)	9.06%	9.84%	6.42%	6.06%
$\|W^{+\prime} - W^+\|/\|W^+\|$	(trajectory)	8.73%	10.35%	4.86%	5.23%

Table 7: **Angles between the elementary displacements**, in degrees, for the same layer as Table 6. "matrix" uses the 2×10^6 entries the dump retains for pairs involving W (validated to $0.13°$ on the three pairs where the exact angle is available). The last two rows follow from $\angle(\Delta_1, \Delta_2)$ and the norms.

		block 0		block 31	
angle	between	channel	matrix	channel	matrix
$\angle(\Delta_{\mathrm{QEP}}, \Delta_1)$	QEP and pass 1	$91.8°$	$88.9°$	$90.9°$	$91.0°$
$\angle(\Delta_{\mathrm{QEP}}, \Delta_2)$	QEP and pass 2	$90.8°$	$89.1°$	$90.2°$	$90.9°$
$\angle(\Delta_{\mathrm{QEP}}, \Delta_{\mathrm{refit}})$	QEP and the refit	$88.9°$	$90.2°$	$89.2°$	$89.9°$
$\angle(\Delta_1, \Delta_2)$	the two passes	$58.1°$	$63.7°$	$44.9°$	$51.0°$
derived from the rows above					
$\angle(\Delta_1, \Delta_{\mathrm{refit}})$	pass 1 and the refit	—	$122°$	—	$116°$
$\angle(\Delta_2, \Delta_{\mathrm{refit}})$	pass 2 and the refit	—	$59°$	—	$65°$

Hessian diagonal moves by 0.2–0.4% in median, and the centroid movements above are the most sensitive quantity and should be read as magnitudes.

C.2 Why the QEP correction grows with depth

Setup. Consider an L-layer network $X_1 = X$, $X_{l+1} = \sigma_l(W_l X_l)$ and its quantized counterpart $\widehat{X}_{l+1} = \sigma_l(\widehat{W}_l \widehat{X}_l)$, each σ_l being γ_l-Lipschitz in the Frobenius norm, with end-to-end outputs $f(X) := X_{L+1}$ and $h(X) := \widehat{X}_{L+1}$. Write $\delta_l := X_l - \widehat{X}_l$ for the inherited activation mismatch at the input of layer l; it is zero at the embedding and grows as layers are quantized. Let $\widehat{H}_l := \widehat{X}_l \widehat{X}_l^\top$ and $P_l := \widehat{X}_l^\top \widehat{H}_l^{-1} \widehat{X}_l$, the orthogonal projector onto the row space of $\widehat{X}_l$ (rank at most $C = 4096$ inside an $N = 262{,}144$-dimensional token space in our setting). QEP replaces W_l by $W_l^* = W_l + W_l \delta_l \widehat{X}_l^\top \widehat{H}_l^{-1}$, i.e. the QEP update of §3 with $\alpha = 1$ as in the derivations of Arai & Ichikawa (2025); the practical $\alpha = \frac{1}{2}$ scales the correction without affecting the argument.

What QEP removes. Inserting $\pm W_L^* \widehat{X}_L$ into $\|X_{L+1} - \widehat{X}_{L+1}\| \le \gamma_L \|W_L X_L - \widehat{W}_L \widehat{X}_L\|$ gives

$$\|X_{L+1} - \widehat{X}_{L+1}\| \le \underbrace{\gamma_L \|W_L \delta_L (I - P_L)\|}_{\text{inherited, after QEP}} + \underbrace{\gamma_L \|W_L^* \widehat{X}_L - \widehat{W}_L \widehat{X}_L\|}_{\text{rounding}}, \tag{8}$$

since $W_L X_L - W_L^* \widehat{X}_L = W_L \delta_L (I - P_L)$. Without QEP the first term would be $\gamma_L \|W_L \delta_L\|$; the correction replaces it by its projection onto the complement of the calibration row space. Since $I - P_L$ is non-expansive, $\|W_L \delta_L (I - P_L)\| \le \|W_L\| \, \|X_L - \widehat{X}_L\|$, and unrolling the recursion to the input leaves one rounding term per layer:

$$\|f(X) - h(X)\| \le \sum_{l=1}^{L} \Big(\prod_{k=l+1}^{L} \gamma_k \|W_k\| \Big) \gamma_l \big\| W_l^* \widehat{X}_l - \widehat{W}_l \widehat{X}_l \big\|. \tag{9}$$

The recursion uses the weakened $\|W_l \delta_l\|$ rather than $\|W_l \delta_l (I - P_l)\|$, so (9) does not credit QEP for its projection and is conservative about it.

What this locates. The size of the correction QEP must apply at layer l is governed by $\|W_l \delta_l\|$, and δ_l accumulates with depth by construction — the trend measured in §C.1. Every summand of (9) is a rounding error of exactly the form (5) minimizes, in which W_l^* is fixed and the only quantity the grid controls is $\widehat{W}_l$: the codebook can act on this bound only through the rounding term, and QEP only through the inherited term, which is why the two compose. The bound does not rank codebooks fitted on W, W^* or W^+; that ranking is the empirical content of the ablation of the fitting target (§5.2). This analysis is the standard Lipschitz unrolling of a layer-wise network specialized to QEP-corrected weights and is not a contribution of this work.

C.3 ALTERNATIVE CODEBOOK SOLVERS, REPLAYED AT LAYER SCALE

Three objections to §4.1 can be tested on the dumped matrix, because a GPTQ pass can be replayed exactly from W^*, the codebook and H^{-1} (§C.1). We use that replay to evaluate the *true* layer objective $\mathcal{L}(\mathcal{G})$ of (5) — not perplexity — for codebooks obtained in different ways, on `mlp.gate_proj` of Llama-2-7B at blocks 0 and 31, at 4 and 2 bits. Before anything is measured, two checks pass: the identity $\mathrm{tr}(EHE^\top) = \sum_{r,k} d_k \,\mathrm{dist}(W^+_{rk}, \mathcal{G}_r)^2$ of (5) holds to seven significant digits with $d_k = 1/[\mathrm{chol}(H^{-1})]^2_{kk}$, and the CPU reimplementation of the k-means reproduces the dumped 4-bit codebook to $7{\times}10^{-4}$ of its range (the residual is the GPU non-determinism noted in Appendix A.1). All variants below use the same W^*, H^{-1} and replay; $\mathcal{L}$ is reported relative to the single-pass codebook of the pipeline ($\mathcal{G}_1$: H_{jj} weighting, quantile initialization, 20 iterations).

Table 8: **Alternative codebook solvers at layer scale.** True layer objective $\mathcal{L}(\mathcal{G})$ of (5) after a full GPTQ replay, relative to the pipeline's single-pass codebook $\mathcal{G}_1$ (lower is better), on `mlp.gate_proj` of Llama-2-7B. "1P" fits on W^*; "2P" refits on the trajectory $W^+(\mathcal{G}_1)$ and reruns the pass; "3P" refits again on the trajectory of that second pass. The frozen-trajectory loss $\Phi(\mathcal{G} \mid W^+)$ of Proposition 1 is given alongside for the refits.

		block 0		block 31	
bits	codebook	$\mathcal{L}/\mathcal{L}_1$	$\Phi/\mathcal{L}_1$	$\mathcal{L}/\mathcal{L}_1$	$\Phi/\mathcal{L}_1$
4-bit					
	1P, H_{jj}, 20 it. (pipeline $\mathcal{G}_1$)	1.000	1.000	1.000	1.000
	1P, H_{jj}, 200 it. (compute-matched)	0.949	0.954	0.875	0.878
	1P, H_{jj}, best of 5 random restarts	1.000	1.000	1.000	1.000
	1P, H_{jj}, equispaced init. (E14)	0.982	0.993	1.091	1.092
	1P, d_k weighting	0.995	0.893	1.006	0.996
	1P, d_k weighting, 200 it.	0.866	0.794	0.866	0.865
	2P, H_{jj} refit (pipeline $\mathcal{G}_2$)	0.977	0.959	0.994	0.991
	2P, H_{jj} refit, 200 it.	0.924	0.912	0.876	0.865
	2P, unweighted (Lloyd) refit	0.975	0.920	0.999	0.989
	2P, H_{jj} refit, equispaced init. (E16)	1.012	0.991	1.104	1.096
	2P, d_k refit (weighting of eq. 5)	0.919	0.848	0.998	0.985
	2P, d_k refit, 200 it.	0.811	0.749	0.866	0.849
	2P, d_k on both passes	0.871	0.875	0.992	0.988
	2P, H_{jj} refit on the trajectory of an unrelated (equispaced) $\mathcal{G}_1$	0.994	0.992	0.999	0.999
	2P, H_{jj} refit on the trajectory of a d_k-weighted $\mathcal{G}_1$	0.982	0.984	0.996	0.996
	3P, H_{jj} refit on the second-pass trajectory	0.987	0.957	0.995	0.991
2-bit					
	1P, H_{jj}, 20 it. (pipeline $\mathcal{G}_1$)	1.000	1.000	1.000	1.000
	1P, H_{jj}, 200 it. (compute-matched)	1.000	1.000	1.002	1.001
	1P, H_{jj}, best of 5 random restarts	1.000	1.000	1.000	1.000
	1P, H_{jj}, equispaced init. (E14)	0.940	0.969	1.011	1.009
	1P, d_k weighting	1.077	0.940	1.036	0.997
	1P, d_k weighting, 200 it.	1.073	0.939	1.036	0.997
	2P, H_{jj} refit (pipeline $\mathcal{G}_2$)	0.892	0.938	0.987	0.994
	2P, H_{jj} refit, 200 it.	0.897	0.938	0.987	0.995
	2P, unweighted (Lloyd) refit	1.496	0.912	0.983	0.988
	2P, H_{jj} refit, equispaced init. (E16)	0.901	0.942	1.008	1.012
	2P, d_k refit (weighting of eq. 5)	0.924	0.877	0.982	0.987
	2P, d_k refit, 200 it.	0.922	0.877	0.982	0.987
	2P, d_k on both passes	0.869	0.897	0.986	0.994
	2P, H_{jj} refit on the trajectory of an unrelated (equispaced) $\mathcal{G}_1$	0.895	0.946	0.986	0.995
	2P, H_{jj} refit on the trajectory of a d_k-weighted $\mathcal{G}_1$	0.914	0.959	0.990	0.999
	3P, H_{jj} refit on the second-pass trajectory	0.899	0.881	0.987	0.984

Table 8 carries six messages, each confined to this layer.

The refit lowers the true objective, most where the grid is most constrained. The pipeline's own refit (2P, H_{jj}) lowers $\mathcal{L}$ by 2.3% and 0.6% at 4 bits and by 10.8% and 1.3% at 2 bits, at blocks 0 and 31 respectively — the same asymmetry between widths as the grid-level result, and the same depth

trend as Table 6. A third pass is worse than the second in three of the four cells ($+1.0$, $+0.05$ and $+0.7$ points) and 0.3 points better in the fourth, as at grid scale (Appendix D.3).

The frozen objective is not a proxy for the rerun. Every refit started from the quantile initialization lowers $\Phi(\cdot \mid W^+)$, as the hypothesis of Proposition 1 requires (the equispaced refit at block 31 does not even do that, $\Phi/\mathcal{L}_1 = 1.10$: k-means finds a local optimum), but the ranking by Φ and the ranking by $\mathcal{L}$ after the rerun disagree repeatedly. The unweighted (Lloyd) refit at 2 bits, block 0, has $\Phi/\mathcal{L}_1 = 0.91$ and $\mathcal{L}/\mathcal{L}_1 = 1.50$; the equispaced refit at 4 bits, block 0, has $\Phi/\mathcal{L}_1 = 0.99$ and $\mathcal{L}/\mathcal{L}_1 = 1.01$. This is the gap that §4.1 leaves open, measured: the trajectory of the second pass moves by 0.5–$1.3\sqrt{\mathcal{L}_1}$ in the D-norm, of the same order as the loss itself, and where the refitted grid lands relative to that moved trajectory is what decides $\mathcal{L}$.

The LDL-prescribed weighting d_k is not uniformly better as a refit, but is never worse when used on both passes. Refitting with d_k instead of H_{jj} lowers Φ further in every cell, as it must, and the true objective at 4 bits (0.919 against 0.977 at block 0) and at 2 bits on block 31 (0.982 against 0.987), but not at 2 bits on block 0 (0.924 against 0.892). Used on the first pass alone it is neutral or harmful (0.995–1.077). Used on *both* passes — a d_k-weighted $\mathcal{G}_1$, its own trajectory, a d_k-weighted refit — it matches or beats the pipeline in all four cells (0.871, 0.992, 0.869, 0.986 against 0.977, 0.994, 0.892, 0.987). The weighting question is therefore open at grid scale, with "d_k on both passes" as the variant worth running first.

More k-means effort on W^* is the binding constraint at 4 bits, not the refit. With the pipeline's 20-iteration budget the k-means is not converged at 4 bits: 200 iterations on W^*, single pass, lower $\mathcal{L}$ by 5.1% and 12.5% — more than the refit does at that width. At 2 bits the 20-iteration solver is converged (200 iterations change $\mathcal{L}$ by 0.0% and $+0.2\%$), and the refit is the remaining lever. Random restarts never beat the quantile initialization on $\Phi(\cdot \mid W^*)$. We read this as a caveat on the 3–4-bit rows of Table 4 — a converged solver is a cheaper improvement there, which we have not run at grid scale — and as evidence that the 2-bit gain is not a solver-budget effect. The equispaced initialization is better single-pass at block 0 (as E14 is at grid scale) and worse at block 31, and the refit on top of it is worse than on the quantile grid at 4 bits, which matches the inconsistent E16 rows of Table 20.

Weighting matters for the refit at 2 bits. The unweighted refit diverges at 2 bits, block 0 ($+50\%$), where the H_{jj}-weighted refit gains 10.8%: with four levels, where the levels go is decided by the weights, and a density-only fit follows the tails of the trajectory to the wrong place. This is the layer-scale counterpart of the sensitivity to the weighting that §5.2 reports.

At matched solver budget the refit still helps. The compute-matched objection is that a second pass buys the same as twice the k-means effort. It does not on this layer: with 200 iterations on both the first fit and the refit, the two-pass is still below the single pass at 4 bits (0.924 against 0.949 at block 0; 0.876 against 0.875 at block 31, i.e. equal) and at 2 bits, where the budget is irrelevant, by the full refit margin (0.897 and 0.987 against 1.000 and 1.002). With the d_k weighting and 200 iterations the refit lowers the objective to 0.812 at 4 bits, block 0, the best value in the table.

Whose trajectory? The ladder of §5.2 cannot separate "the trajectory induced by the same codebook" from "any trajectory of the sweep". Refitting $\mathcal{G}_2$ on the trajectory produced by an *unrelated* first pass (an equispaced-initialized grid, or a d_k-weighted one) answers it at layer scale, and the answer depends on the width. At 4 bits the own trajectory matters: refitting on an unrelated one recovers little (0.994 and 0.982 at block 0 against 0.977; 0.999 and 0.996 at block 31 against 0.994). At 2 bits it does not: the refit on the equispaced first pass's trajectory is as good as on the pipeline's own (0.895 against 0.892; 0.986 against 0.987). With four levels the drift is dominated by the sweep rather than by which grid produced it, which is consistent with the refit being a 2-bit phenomenon at grid scale and suggests that, at 2 bits, the first pass need not even use the final grid — a cheaper first pass is a natural follow-up.

D Additional results

D.1 Complete grid, all cases

The two tables below are the raw material for the rest of the paper. Rows are configuration codes — En on the GPTQ engine, Qn on QuIP's LDLQ (naming convention in Appendix A.1); E8 and Q8

are our method. Cells are the mean of three calibration seeds; the markers are defined in the caption of Table 9. In Table 9 each bit-width block opens with the two rows without QEP (Appendix D.5) and closes with the five rows with LeanQuant's weighting as published (Appendix D.8); these seven rows were added during revision and are per-channel only. The two tables are not comparable cell by cell: at g=128 a codebook stores one set of levels per 128 columns rather than one per row, so its parameter overhead is two orders of magnitude larger (§D.4).

Table 9: All cases, **per-channel**, WikiText-2 perplexity (mean of 3 seeds). Bold = best method per model within each bit-width block among the rows without a suffix; the seven rows added during revision (-noQEP, p4) are not considered. †: mean over the seeds with finite perplexity; div.: no seed with finite perplexity; ‡: mean of two seeds, the third run did not complete (a job failure on a shared GPU, not a divergence); empty cell: not shown because of numerical divergence in at least one seed. "LeanQuant grid" (E10, Q10) and "Lean weight" (E13) denote the LeanQuant-style weighting of Appendix A.2; the p4 rows use LeanQuant's weighting as published.

Case	Llama-2-7B	Llama-2-13B	Llama-2-70B	Llama-3-8B	Mistral-7B	Sheared-2.7B
4-bit						
E1: QEP+RtN, uniform	6.012	5.166	3.624	7.967	5.868	7.411
E2: QEP+RtN, codebook	5.752	5.134	3.505	6.906	5.520	7.203
E3: GPTQ, uniform	6.079	5.155	3.590	274.9	5.620	7.474
E4: GPTQ, codebook	5.761	5.179	3.482	6.943	5.560	7.215
E5: QEP+GPTQ, uniform	5.964	5.114	3.569	10.17	5.537	7.289
E6: QEP+GPTQ, codebook	5.717	5.110	3.497	6.791	5.491	7.155
E7: QEP+GPTQ, uniform, 2P	6.778	5.129	3.636	8.621	5.893	7.805
E8: QEP+GPTQ, codebook, 2P (ours)	5.689	5.084	3.487	6.754	5.485	7.141
E9: QEP+GPTQ, codebook, 3P	5.711	5.089	3.481	6.761	5.490	7.142
E10: QEP+GPTQ, LeanQuant grid	5.706	5.039	3.484	6.763	5.438	7.167
E11: E10 + two-pass	5.708	5.044	3.482	6.772	5.445	7.173
E12: E7 with MSE refit	26.45†	7.046	3.641	7.731	6.526	14.00
Q3: QuIP, uniform	5.758	5.095	3.536	6.921	5.465	7.182
Q4: QuIP, codebook	5.652	5.029	3.428	6.575	5.376	7.088
Q5: QEP+QuIP, uniform	5.719	5.071	3.529	6.834	5.444	7.153
Q6: QEP+QuIP, codebook	5.634	5.004	3.426	6.535	5.368	7.064
Q7: QEP+QuIP, uniform, 2P	5.721	5.075	3.527	6.844	5.441	7.152
Q8: QEP+QuIP, codebook, 2P (ours)	5.625	5.005	**3.425**	6.530	5.371	7.068
E13: Lean weight + our init	6.004	8.463	3.612	7.292	5.687	7.463
E14: our weight + Lean init	5.655	5.017	3.449	6.673	5.412	7.110
E15: E6 but grid fitted on W	5.716	5.120	3.493	6.790	5.502	7.158
E16: E14 + two-pass	5.660	5.013	3.450	6.645	5.414	7.127
Q10: QEP+QuIP, LeanQuant grid	5.618	**4.994**	3.434	6.554	**5.362**	7.055
Q11: Q10 + two-pass	5.623	5.001	3.434	6.554	5.365	**7.054**
Q12: Q6 but grid fitted on W	5.629	5.005	3.425	**6.529**	5.371	7.067
Q13: Q8 with Lean init	**5.612**	5.000	3.434	6.555	5.367	7.068
3-bit						
E1: QEP+RtN, uniform	17.31	7.521	5.645	86.73	10.02	12.35
E2: QEP+RtN, codebook	6.241	5.587	3.845	8.626	6.069	7.817
E3: GPTQ, uniform	10.95	6.577	4.832	177.2	8.368	11.11
E4: GPTQ, codebook	6.357	5.718	3.843	8.602	6.189	7.911
E5: QEP+GPTQ, uniform	8.236	6.259	4.732	38.39	7.351	9.234
E6: QEP+GPTQ, codebook	6.202	5.509	3.838	8.310	6.001	7.704
E7: QEP+GPTQ, uniform, 2P	20.72	6.408	5.338	28.15	26.46	14.26
E8: QEP+GPTQ, codebook, 2P (ours)	6.178	5.455	3.821	8.161	5.968	7.676
E9: QEP+GPTQ, codebook, 3P	6.165	5.453	3.832	8.277	5.974	7.666
E10: QEP+GPTQ, LeanQuant grid	6.257	5.425	3.838	8.104	5.908	7.845
E11: E10 + two-pass	6.251	5.424	3.828	8.074	5.926	7.790
E12: E7 with MSE refit	div.	70.95	4.980	12.40	18.84	40.79
Q3: QuIP, uniform	7.390	6.280	4.526	11.64	6.609	8.776
Q4: QuIP, codebook	6.128	5.374	3.730	7.631	5.739	7.587
Q5: QEP+QuIP, uniform	6.933	5.940	4.446	9.973	6.428	8.362
Q6: QEP+QuIP, codebook	6.061	5.308	3.713	7.518	5.713	7.521
Q7: QEP+QuIP, uniform, 2P	6.987	5.969	4.480	10.11	6.467	8.424
Q8: QEP+QuIP, codebook, 2P (ours)	6.041	5.299	3.706	7.509	5.702	7.507
E13: Lean weight + our init	6.648†	16.77	4.047	9.795	6.543	8.352
E14: our weight + Lean init	6.134	5.365	3.760	7.761	5.802	7.683
E15: E6 but grid fitted on W	6.199	5.512	3.836	8.351	6.000	7.686
E16: E14 + two-pass	6.105	5.358	3.755	7.792	5.812	7.671
Q10: QEP+QuIP, LeanQuant grid	**6.019**	5.290	3.692	7.467	5.669	7.479
Q11: Q10 + two-pass	6.023	5.287	3.689	7.467	5.669	**7.460**
Q12: Q6 but grid fitted on W	6.062	5.307	3.709	7.496	5.717	7.534
Q13: Q8 with Lean init	6.022	**5.280**	**3.689**	**7.454**	**5.664**	7.487
2-bit						
E1: QEP+RtN, uniform	1.2×10^5	84 299	25 999	7.2×10^5	53 808	1.2×10^5

Table 9 continued

Case	Llama-2-7B	Llama-2-13B	Llama-2-70B	Llama-3-8B	Mistral-7B	Sheared-2.7B
E2: QEP+RtN, codebook	18.80	124.8	8.094	438.6	42.70	27.43
E3: GPTQ, uniform	10 871†	1 520	168.7	3.1×10^5	2 104	2 769
E4: GPTQ, codebook	14.49	14.38	7.522	44.46	26.62	19.39
E5: QEP+GPTQ, uniform	5 732	2 375	70.09	2.8×10^5	1 622	1 389
E6: QEP+GPTQ, codebook	11.95	13.09	6.131	41.18	15.83	16.58
E7: QEP+GPTQ, uniform, 2P	1.0×10^5	69 654	10 013	5.4×10^5	54 355	91 647†
E8: QEP+GPTQ, codebook, 2P (ours)	10.55	8.754	5.807	34.93	13.54	14.29
E9: QEP+GPTQ, codebook, 3P	10.82	8.968	5.783	37.09	13.94	14.45
E10: QEP+GPTQ, LeanQuant grid	14.06	11.54	6.002	53.05	16.24	18.97
E11: E10 + two-pass	14.24	133.2	5.852	54.85	15.89	21.22
E12: E7 with MSE refit	div.	2 299	10.79	172.3	93.91	1 568
Q3: QuIP, uniform	988.9	277.6	23.34	14 344	251.1	423.9
Q4: QuIP, codebook	11.38	8.674	5.545	17.90	10.31	13.69
Q5: QEP+QuIP, uniform	723.8	94.90	16.50	3 262	61.68	93.92
Q6: QEP+QuIP, codebook	10.08	7.942	5.381	17.40	9.366	12.37
Q7: QEP+QuIP, uniform, 2P	8 370	582.7	29.96	1.5×10^5	156.9	168.7
Q8: QEP+QuIP, codebook, 2P (ours)	**9.534**	7.561	**5.192**	15.91	**8.729**	**11.73**
E13: Lean weight + our init	17.08†	378.0	6.703	103.4	28.08	30.76
E14: our weight + Lean init	12.27	10.00	5.975	33.53	13.97	15.61
E15: E6 but grid fitted on W	11.77	40.91	6.046	37.87	16.05	17.23
E16: E14 + two-pass	11.20	13.91	5.723	33.99	13.60	14.23
Q10: QEP+QuIP, LeanQuant grid	9.963	7.965	5.404	17.17	9.424	12.22
Q11: Q10 + two-pass	41.71	7.677	8.309	16.99	22.66	47.36
Q12: Q6 but grid fitted on W	9.814	7.911	5.354	16.59	9.252	12.45
Q13: Q8 with Lean init	9.635	**7.558**	5.205	**15.86**	8.819	11.76

Table 10: All cases, **group-wise** g=128, WikiText-2 perplexity (mean of 3 seeds). Bold = best method per model within each bit-width block. Codebook cases at g=128 carry a large parameter overhead, §D.4; comparisons within this table are *not* at equal storage budget.

Case	Llama-2-7B	Llama-2-13B	Llama-2-70B	Llama-3-8B	Mistral-7B	Sheared-2.7B
4-bit (nominal)						
E1: QEP+RtN, uniform	5.682	4.964	3.435	6.588	5.373	7.150
E2: QEP+RtN, codebook	5.661	5.006	3.431	6.624	5.383	7.113
E3: GPTQ, uniform	5.724	4.986	3.422	12.76	5.387	7.085
E4: GPTQ, codebook	5.665	5.024	3.434	6.678	5.386	7.105
E5: QEP+GPTQ, uniform	5.639	4.963	3.416	6.767	5.351	7.048
E6: QEP+GPTQ, codebook	5.652	5.016	3.431	6.583	5.376	7.081
E7: QEP+GPTQ, uniform, 2P	5.666	4.964	3.418	6.828	5.353	7.049
E8: QEP+GPTQ, codebook, 2P (ours)	5.650	5.026	3.436	6.593	5.382	7.092
E9: QEP+GPTQ, codebook, 3P	5.656	5.025	3.436	6.600	5.386	7.093
E10: QEP+GPTQ, LeanQuant grid	5.563	**4.951**	3.394	6.401	5.330	7.013
E11: E10 + two-pass	5.571	4.954	3.394	6.411	5.330	7.021
E12: E7 with MSE refit	5.982	4.995	3.425	7.087	5.707	7.067
Q3: QuIP, uniform	5.603	4.978	3.410	6.527	5.348	7.062
Q4: QuIP, codebook	5.633	5.005	3.419	6.539	5.359	7.070
Q5: QEP+QuIP, uniform	5.587	4.966	3.408	6.477	5.345	7.036
Q6: QEP+QuIP, codebook	5.621	4.991	3.418	6.499	5.355	7.052
Q7: QEP+QuIP, uniform, 2P	5.593	4.967	3.407	6.480	5.342	7.035
Q8: QEP+QuIP, codebook, 2P (ours)	5.628	5.000	3.422	6.519	5.361	7.057
E13: Lean weight + our init	5.750	5.110	3.466	6.932	5.453	7.294
E14: our weight + Lean init	**5.554**	4.953	3.390	**6.386**	5.328	7.021
E15: E6 but grid fitted on W	5.671	5.030	3.442	6.632	5.396	7.105
E16: E14 + two-pass	5.562	4.952	3.391	6.394	5.325	7.015
Q10: QEP+QuIP, LeanQuant grid	5.569	4.954	3.389	6.389	5.323	**7.008**
Q11: Q10 + two-pass	5.567	4.956	3.391	6.397	**5.322**	7.009
Q12: Q6 but grid fitted on W	5.635	5.006	3.426	6.529	5.371	7.074
Q13: Q8 with Lean init	5.563	4.953	**3.389**	6.398	5.322	7.013
3-bit (nominal)						
E1: QEP+RtN, uniform	6.346	5.403	3.876	10.41	5.912	7.903
E2: QEP+RtN, codebook	6.072	5.315	3.694	7.753	5.807	7.652
E3: GPTQ, uniform	6.416	5.462	3.867	29.31	6.027	7.930
E4: GPTQ, codebook	6.108	5.339	3.712	7.774	5.824	7.586
E5: QEP+GPTQ, uniform	6.187	5.370	3.838	9.149	5.797	7.679
E6: QEP+GPTQ, codebook	6.049	5.321	3.702	7.520	5.779	7.541
E7: QEP+GPTQ, uniform, 2P	6.234	5.369	3.841	10.52	5.801	7.683
E8: QEP+GPTQ, codebook, 2P (ours)	6.084	5.338	3.725	7.566	5.817	7.584
E9: QEP+GPTQ, codebook, 3P	6.105	5.348	3.730	7.604	5.822	7.581
E10: QEP+GPTQ, LeanQuant grid	5.956	5.240	3.682	7.358	5.657	7.460
E11: E10 + two-pass	5.946	5.226	3.680	7.371	5.656	7.457
E12: E7 with MSE refit	7.916	5.496	3.861	9.438	8.189	7.827
Q3: QuIP, uniform	6.217	5.435	3.820	8.136	5.782	7.719

Table 10 continued

Case	Llama-2-7B	Llama-2-13B	Llama-2-70B	Llama-3-8B	Mistral-7B	Sheared-2.7B
Q4: QuIP, codebook	6.050	5.321	3.678	7.470	5.670	7.514
Q5: QEP+QuIP, uniform	6.089	5.327	3.796	7.794	5.740	7.565
Q6: QEP+QuIP, codebook	5.982	5.268	3.664	7.345	5.654	7.450
Q7: QEP+QuIP, uniform, 2P	6.085	5.337	3.794	7.824	5.743	7.586
Q8: QEP+QuIP, codebook, 2P (ours)	6.010	5.293	3.681	7.414	5.687	7.494
E13: Lean weight + our init	6.207	5.468	3.786	8.374	6.054	7.869
E14: our weight + Lean init	5.922	5.214	3.660	7.295	5.633	7.416
E15: E6 but grid fitted on W	6.126	5.367	3.745	7.661	5.854	7.621
E16: E14 + two-pass	**5.912**	**5.206**	3.659	7.305	5.610	7.391
Q10: QEP+QuIP, LeanQuant grid	5.959	5.227	**3.646**	7.288	5.618	7.401
Q11: Q10 + two-pass	5.933	5.222	3.649	**7.288**	5.602	**7.382**
Q12: Q6 but grid fitted on W	6.031	5.323	3.694	7.464	5.727	7.529
Q13: Q8 with Lean init	5.940	5.215	3.649	7.298	**5.596**	7.391
2-bit (nominal)						
E1: QEP+RtN, uniform	34.84	12.81	8.802	1 599	24.93	44.08
E2: QEP+RtN, codebook	11.49	8.485	5.263	34.83	11.76	16.48
E3: GPTQ, uniform	460.7	16.56	8.165	1 689	24.59	47.86
E4: GPTQ, codebook	10.51	8.201	5.405	19.99	11.70	12.82
E5: QEP+GPTQ, uniform	22.18	10.91	6.958	101.5	15.29	20.24
E6: QEP+GPTQ, codebook	9.430	7.564	5.184	16.82	9.280	11.79
E7: QEP+GPTQ, uniform, 2P	33.58	11.03	7.112	117.7	17.05	23.27
E8: QEP+GPTQ, codebook, 2P (ours)	9.500	7.645	5.230	18.54	9.476	11.83
E9: QEP+GPTQ, codebook, 3P	9.476	7.623	5.263	17.97	9.485	11.97
E10: QEP+GPTQ, LeanQuant grid	9.131	7.397	5.161	15.41	8.738	11.53
E11: E10 + two-pass	9.166	7.348	5.136	15.02	8.723	11.32
E12: E7 with MSE refit	22.93	284.4	9.395	57.83	47.88	21.81
Q3: QuIP, uniform	36.62	15.25	6.936	290.4	16.67	29.92
Q4: QuIP, codebook	9.926	7.718	5.085	15.33	8.679	11.78
Q5: QEP+QuIP, uniform	15.32	10.06	6.459	50.05	11.65	17.20
Q6: QEP+QuIP, codebook	8.843	7.245	5.009	14.40	8.259	10.90
Q7: QEP+QuIP, uniform, 2P	15.60	10.05	6.461	48.57	11.74	17.70
Q8: QEP+QuIP, codebook, 2P (ours)	9.020	7.333	5.051	14.93	8.354	11.01
E13: Lean weight + our init	10.16	8.398	5.380	25.74	11.85	13.23
E14: our weight + Lean init	8.978	7.313	5.067	14.59	8.404	11.09
E15: E6 but grid fitted on W	9.728	8.123	5.460	18.05	10.15	12.36
E16: E14 + two-pass	8.927	7.243	5.048	14.44	8.340	11.02
Q10: QEP+QuIP, LeanQuant grid	**8.777**	7.269	5.015	14.40	8.216	**10.76**
Q11: Q10 + two-pass	8.782	**7.195**	**4.984**	14.44	**8.107**	10.78
Q12: Q6 but grid fitted on W	9.447	7.715	5.200	15.42	8.936	11.73
Q13: Q8 with Lean init	8.852	7.212	4.994	**14.36**	8.109	10.80

D.2 SEED-LEVEL STATISTICS

Table 11: **2-bit per-channel cells with seed-level standard deviation** (WikiText-2 perplexity, mean $\pm$ std over 3 seeds). The standard deviation combines calibration-sample variation and the GPU non-determinism of the k-means.

		QEP + GPTQ		QEP + QuIP	
Model	FP16	codebook	**+2P**	codebook	**+2P**
Llama-2-7B	5.472	11.95$\pm$0.35	10.55$\pm$0.26	10.08$\pm$0.27	9.53$\pm$0.34
Llama-2-13B	4.884	13.09$\pm$1.46	8.75$\pm$0.23	7.94$\pm$0.10	7.56$\pm$0.10
Llama-2-70B	3.321	6.13$\pm$0.16	5.81$\pm$0.15	5.38$\pm$0.07	5.19$\pm$0.05
Llama-3-8B	6.136	41.18$\pm$4.31	34.93$\pm$2.05	17.40$\pm$0.17	15.91$\pm$0.43
Mistral-7B	5.252	15.83$\pm$1.16	13.54$\pm$0.72	9.37$\pm$0.07	8.73$\pm$0.09
Sheared-2.7B	6.906	16.58$\pm$0.38	14.29$\pm$0.36	12.37$\pm$0.39	11.73$\pm$0.31

D.3 TWO-PASS SUFFICIENCY: A THIRD PASS DOES NOT HELP

A third pass, refitting on the trajectory of the second, adds nothing at 4 and 3 bits (6/18 and 9/18 wins, CIs straddling zero) and is slightly *worse* at 2 bits ($+2.5\%$, 95% CI $[+1.2, +3.9]$, 4/18 wins). Two passes are therefore an empirical design choice justified by this table, not a convergence claim: the grids do not settle. Measured on `mlp.gate_proj` of Llama-2-7B in units of each channel's own $\sigma(W^*)$, the second grid sits 4.86% from the first at block 0 and 2.37% at block 31, and a third grid sits 4.78% and 2.10% from the second — essentially the same step; per channel the third step is *larger* than the second in 49% and 39% of channels, which a contraction would not produce. The

Table 12: **The refit at seed level, group-wise** g=128: same statistics as Table 4. The refit is slightly harmful at every width.

bits	wins	mean Δ	95% CI	median	min	max	p
QEP + GPTQ							
4	1/18	+0.12%	[+0.05, +0.18]	+0.16%	-0.39%	+0.26%	0.999
3	0/18	+0.56%	[+0.49, +0.64]	+0.59%	+0.22%	+0.92%	1.000
2	1/18	+2.55%	[+1.03, +4.47]	+0.82%	-0.12%	+14.26%	1.000
QEP + QuIP							
4	2/18	+0.15%	[+0.09, +0.21]	+0.15%	-0.06%	+0.37%	1.000
3	0/18	+0.59%	[+0.48, +0.71]	+0.54%	+0.21%	+1.17%	1.000
2	1/18	+1.66%	[+1.13, +2.36]	+1.43%	-0.02%	+6.27%	1.000

Table 13: **Ablating the fitting target at seed level.** Each step is a paired comparison over the 18 (model, seed) cells of a width: wins, mean relative change with 95% bootstrap CI, and one-sided Wilcoxon p (H_1: right-hand tensor lower). The first step is never significant; the second is significant at every width on GPTQ and at 3 and 2 bits on QuIP. Figure 5 plots the corresponding means.

	fit on $W \to$ fit on W^*			fit on $W^* \to$ fit on W^+		
bits	wins	mean [95% CI]	p	wins	mean [95% CI]	p
QEP + GPTQ						
4	10/18	-0.05% [-0.12, +0.02]	0.084	17/18	-0.35% [-0.44, -0.26]	$<10^{-4}$
3	8/18	-0.03% [-0.22, +0.12]	0.617	18/18	-0.75% [-1.17, -0.44]	$<10^{-4}$
2	7/18	-9.25% [-22.09, +1.03]	0.433	18/18	-15.47% [-19.75, -11.73]	$<10^{-4}$
QEP + QuIP						
4	8/18	+0.02% [-0.02, +0.05]	0.752	10/18	-0.02% [-0.08, +0.03]	0.261
3	8/18	+0.03% [-0.07, +0.13]	0.584	14/18	-0.20% [-0.30, -0.10]	0.001
2	3/18	+1.53% [+0.57, +2.67]	0.998	18/18	-5.71% [-6.66, -4.88]	$<10^{-4}$

refit is the one-off correction of a specific mismatch — a grid fitted on W^* used to round values drawn from W^+ — and once it is closed further refits chase a trajectory that moves as much as ever without improving the objective (Appendix C.3 measures the same at layer scale).

D.4 Group-wise codebooks and effective bits

Comparisons at *nominal* bit-width are misleading once a codebook is stored per group rather than per row. A per-channel codebook stores 2^b half-precision levels for every C weights (0.016–0.063 bits/weight at C=4096); a g=128 codebook stores the same 2^b levels for every 128 weights, an overhead of $2^b \cdot 16/128$ bits/weight: 0.5 bits at b=2, 1.0 at b=3 and 2.0 at b=4. A "4-bit" group-wise codebook is a 6-bit method in storage.

Table 15 reports the group-wise results at nominal bits; Figure 6 replots them against effective bits. Once the overhead is charged, the per-channel codebook lane Pareto-dominates. The refit is not neutral at g=128: it is slightly harmful at every width (Table 12; $+0.1$–0.2% at 4 bits, $+0.6\%$ at 3, $+1.7$–2.6% at 2, losing 16–18 of 18 pairs). We do not have a tested explanation; the candidate we would test first is that a codebook fitted on 128 trajectory values is fitted on a sample small enough that the second pass's trajectory moves it by more than the refit gained, which is the perturbation term of (7) dominating. We therefore report per-channel throughout the body and treat the group-wise grid as an ablation.

D.5 The refit without QEP

2PTC is evaluated on top of QEP throughout the body. To test whether the refit depends on that correction, E8-noQEP and Q8-noQEP run E8 and Q8 with QEP disabled: the codebook is fitted on W, the engine sweeps from W, the codebook is refitted on that trajectory, and the second pass restarts from W. Their single-pass references are E4 and Q4; calibration samples, seeds and all

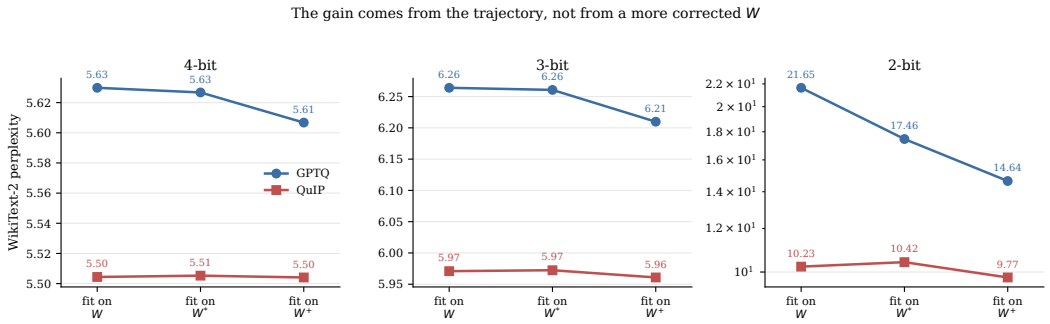


Figure 5: **Ablating the fitting target**, mean WikiText-2 perplexity over six models and three seeds as the *only* change is the tensor the codebook is fitted on (per-channel; log scale at 2 bits); the same numbers as Table 3. Fitting on W^* instead of W gives no consistent improvement on either engine; fitting on the trajectory W^+ carries the effect.

Table 14: Two versus three passes (QEP+GPTQ lane, per-channel). Left: per-model means of 3 seeds. Right: seed-level comparison E8 $\rightarrow$ E9 (wins for the third pass, mean relative change with 95% CI, median, one-sided Wilcoxon p for E9 lower).

Model	2 passes (E8)	3 passes (E9)	Δ
4-bit			
Llama-2-7B	5.689	5.711	+0.38%
Llama-2-13B	5.084	5.089	+0.11%
Llama-2-70B	3.487	3.481	-0.19%
Llama-3-8B	6.754	6.761	+0.10%
Mistral-7B	5.485	5.490	+0.10%
Sheared-2.7B	7.141	7.142	+0.01%
3-bit			
Llama-2-7B	6.178	6.165	-0.22%
Llama-2-13B	5.455	5.453	-0.04%
Llama-2-70B	3.821	3.832	+0.29%
Llama-3-8B	8.161	8.277	+1.42%
Mistral-7B	5.968	5.974	+0.09%
Sheared-2.7B	7.676	7.666	-0.13%
2-bit			
Llama-2-7B	10.55	10.82	+2.60%
Llama-2-13B	8.754	8.968	+2.45%
Llama-2-70B	5.807	5.783	-0.40%
Llama-3-8B	34.93	37.09	+6.20%
Mistral-7B	13.54	13.94	+2.91%
Sheared-2.7B	14.29	14.45	+1.14%

bits	wins	mean	95% CI	median	p
4	6/18	+0.08%	[-0.02, +0.19]	+0.07%	0.909
3	9/18	+0.23%	[-0.02, +0.52]	-0.01%	0.649
2	4/18	+2.45%	[+1.16, +3.90]	+1.77%	0.999

other settings are those of the rest of the grid. Table 16 gives the four combinations averaged over models, Table 17 the paired statistics, and Table 9 the per-model values.

The refit does not depend on QEP. Without QEP the refit lowers 2-bit perplexity in all 36 pairs, by 16.1% with GPTQ (95% CI $[-20.6, -12.1]$) and 6.4% with QuIP ($[-7.4, -5.5]$), against 15.5% and 5.7% with QEP (Table 4). At 3 and 4 bits the gains are small, as in Table 4: significant with GPTQ (-1.10% and -0.47%) and not with QuIP ($p = 0.13$ at 3 bits and 0.054 at 4), whereas with QEP the QuIP refit is significant at 3 bits. Pair by pair, the difference between the refit's gain without and with QEP has a 95% CI containing zero at every width on both engines; the largest mean difference is 0.7 points, at 2 bits.

Nor does QEP depend on the refit. Adding QEP to the two-pass configuration lowers 2-bit perplexity by 17.0% with GPTQ and 6.7% with QuIP, against 17.6% and 7.4% in the single pass (E4 $\rightarrow$ E6, Q4 $\rightarrow$ Q6), and helps in 34 of the 36 two-pass cells; the exceptions are Llama-2-70B with GPTQ

Table 15: Group-wise (g=128) results at *nominal* bits (mean of 3 seeds). Codebook columns carry $+2^b \cdot 16/128$ bits/weight of overhead.

	QEP + GPTQ			QEP + QuIP		
Model	unif.	codebk	+2P	unif.	codebk	+2P
4-bit (nominal)						
Llama-2-7B	5.639	5.652	5.650	**5.587**	5.621	5.628
Llama-2-13B	**4.963**	5.016	5.026	4.966	4.991	5.000
Llama-2-70B	3.416	3.431	3.436	**3.408**	3.418	3.422
Llama-3-8B	6.767	6.583	6.593	**6.477**	6.499	6.519
Mistral-7B	5.351	5.376	5.382	**5.345**	5.355	5.361
Sheared-2.7B	7.048	7.081	7.092	**7.036**	7.052	7.057
3-bit (nominal)						
Llama-2-7B	6.187	6.049	6.084	6.089	**5.982**	6.010
Llama-2-13B	5.370	5.321	5.338	5.327	**5.268**	5.293
Llama-2-70B	3.838	3.702	3.725	3.796	**3.664**	3.681
Llama-3-8B	9.149	7.520	7.566	7.794	**7.345**	7.414
Mistral-7B	5.797	5.779	5.817	5.740	**5.654**	5.687
Sheared-2.7B	7.679	7.541	7.584	7.565	**7.450**	7.494
2-bit (nominal)						
Llama-2-7B	22.18	9.430	9.500	15.32	**8.843**	9.020
Llama-2-13B	10.91	7.564	7.645	10.06	**7.245**	7.333
Llama-2-70B	6.958	5.184	5.230	6.459	**5.009**	5.051
Llama-3-8B	101.5	16.82	18.54	50.05	**14.40**	14.93
Mistral-7B	15.29	9.280	9.476	11.65	**8.259**	8.354
Sheared-2.7B	20.24	11.79	11.83	17.20	**10.90**	11.01

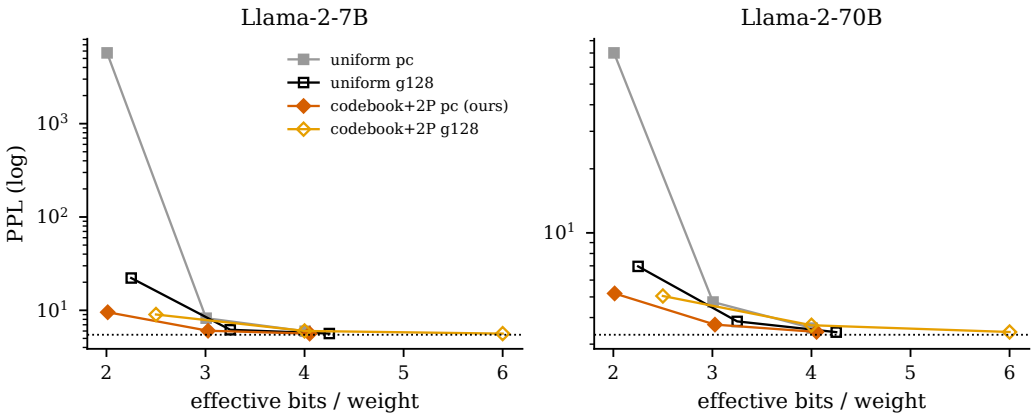


Figure 6: Perplexity versus **effective** bits per weight (storage overhead included), Llama-2-7B and Llama-2-70B. The per-channel codebook+2P lane Pareto-dominates; the g=128 codebook (hollow diamonds) is shifted right by its parameter cost.

at 3 and 4 bits, and at 4 bits QEP does not help that single pass either (§5.1). The two corrections therefore compose approximately additively — together they lower 2-bit perplexity by 30.7% with GPTQ and 12.7% with QuIP relative to E4 and Q4 — which is consistent with the near-orthogonal displacements measured on one layer (Table 7).

D.6 Negative result: the two-pass on a uniform grid

Nothing in the definition of the refit requires a codebook. Replacing the weighted k-means by a (scale, zero-point) pair in Algorithm 1 gives a uniform two-pass, and under our implementation it fails badly (Table 18). The mechanism we propose is that a uniform grid is determined by the *range*

Table 16: **The refit with and without QEP** (per-channel, WikiText-2 perplexity, mean over six models and three seeds; bold = best in the row). Without QEP the codebook is fitted on W and both passes start from W (E4, E8-noQEP; Q4, Q8-noQEP); with QEP they start from W^* (E6, E8; Q6, Q8).

	without QEP		with QEP	
bits	1 pass	+2P	1 pass	**+2P**
GPTQ				
4	5.690	5.664	5.627	**5.607**
3	6.437	6.370	6.261	**6.210**
2	21.142	18.153	17.462	**14.645**
QuIP				
4	5.525	5.522	5.505	**5.504**
3	6.032	6.023	5.972	**5.961**
2	11.249	10.529	10.423	**9.775**

Table 17: **The refit without QEP, at seed level.** Paired comparisons over the 18 (model, seed) cells of a width, as in Table 4: wins, mean relative change with 95% bootstrap CI, and one-sided Wilcoxon p (H_1: right-hand configuration lower). Left: the refit when QEP is disabled (E4 $\rightarrow$ E8-noQEP, Q4 $\rightarrow$ Q8-noQEP); right: QEP when the refit is applied (E8-noQEP $\rightarrow$ E8, Q8-noQEP $\rightarrow$ Q8). The refit with QEP is in Table 4.

	refit without QEP			QEP with the refit		
bits	wins	mean [95% CI]	p	wins	mean [95% CI]	p
GPTQ						
4	17/18	-0.47% [-0.58, -0.34]	$<10^{-4}$	15/18	-0.88% [-1.28, -0.47]	$<10^{-4}$
3	16/18	-1.10% [-1.50, -0.69]	$<10^{-3}$	15/18	-2.19% [-2.91, -1.47]	$<10^{-4}$
2	18/18	-16.14% [-20.57, -12.05]	$<10^{-4}$	17/18	-17.02% [-22.76, -11.82]	$<10^{-4}$
QuIP						
4	12/18	-0.05% [-0.11, +0.01]	0.054	17/18	-0.30% [-0.38, -0.22]	$<10^{-4}$
3	10/18	-0.14% [-0.30, +0.02]	0.132	18/18	-0.99% [-1.22, -0.75]	$<10^{-4}$
2	18/18	-6.38% [-7.38, -5.47]	$<10^{-4}$	18/18	-6.72% [-8.37, -5.07]	$<10^{-4}$

it must span; W^+ has heavier tails than W^*, so refitting on the trajectory widens the range, and a wider range at fixed bit-width means coarser steps everywhere, including in the dense region near zero, with nothing bought in return because a uniform grid cannot place levels where the trajectory concentrates. We note an alternative we have not excluded: at 2 bits the uniform first pass is already close to divergence, and the second pass may simply amplify an unstable starting point. Either way the conclusion is the one stated in §4.3: trajectory refitting is effective with learned codebooks and destabilizes uniform grids in our setting.

D.7 LeanQuant-style grids inside our pipeline

Their weighting does not survive the refit. The two-pass that improves our grid on every model at 2 bits produces isolated blow-ups on theirs — Llama-2-13B from 11.5 to 133 on the GPTQ engine (E10 vs E11) — and on the QuIP engine the failure is systematic: three of six models exceed perplexity 22, while our configuration stays below 12 on the same three (Q11 vs Q8, Table 22). The mechanism we propose is the one behind the uniform failure: refitting on a wider-tailed trajectory needs a grid that spreads levels by density, and a $p = 3$ weight concentrates them so aggressively that the refitted grid destabilizes. In aggregate at 2 bits our full method reaches 14.65 against 40.87 for their grid under GPTQ, and 9.78 against 24.12 under QuIP.

Their weighting is worse. At fixed initialization (E6 vs E13 in Table 20), our weighting is better in all 18 model–width cells, and theirs alone degrades sharply or diverges: at 2 bits Llama-2-13B reaches perplexity 378 and Llama-3-8B 103.

Table 18: The same two-pass refit on a uniform grid and on a learned codebook (QEP+GPTQ lane, per-channel, mean of 3 seeds). The refit helps the codebook and destroys the uniform grid.

	uniform grid		learned codebook	
Model	1 pass (E5)	2 passes (E7)	1 pass (E6)	2 passes (E8)
4-bit				
Llama-2-7B	5.964	6.778	5.717	5.689
Llama-2-13B	5.114	5.129	5.110	5.084
Llama-2-70B	3.569	3.636	3.497	3.487
Llama-3-8B	10.17	8.621	6.791	6.754
Mistral-7B	5.537	5.893	5.491	5.485
Sheared-2.7B	7.289	7.805	7.155	7.141
3-bit				
Llama-2-7B	8.236	20.72	6.202	6.178
Llama-2-13B	6.259	6.408	5.509	5.455
Llama-2-70B	4.732	5.338	3.838	3.821
Llama-3-8B	38.39	28.15	8.310	8.161
Mistral-7B	7.351	26.46	6.001	5.968
Sheared-2.7B	9.234	14.26	7.704	7.676
2-bit				
Llama-2-7B	5 732	1.0×10^5	11.95	10.55
Llama-2-13B	2 375	69 654	13.09	8.754
Llama-2-70B	70.09	10 013	6.131	5.807
Llama-3-8B	2.8×10^5	5.4×10^5	41.18	34.93
Mistral-7B	1 622	54 355	15.83	13.54
Sheared-2.7B	1 389	91 647†	16.58	14.29

Table 19: **Our grid versus the LeanQuant-style grid**, with and without the trajectory refit (per-channel, mean over six models and three seeds; bold = best in the row). Their grid is stronger at 4 bits and comparable at 3; at 2 bits ours dominates, and their weighting is actively harmful under the refit.

	our grid		LeanQuant-style grid	
bits	1 pass	**+2P**	1 pass	+2P
GPTQ				
4	5.627	5.607	**5.599**	5.604
3	6.261	**6.210**	6.230	6.216
2	17.462	**14.645**	19.976	40.871
QuIP				
4	5.505	5.504	**5.503**	5.505
3	5.972	5.961	5.936	**5.932**
2	10.423	**9.775**	10.357	24.117

Their initialization is better. At fixed weighting (E6 vs E14), the equispaced initialization wins in 17 of 18 model–width cells: in all six at 3 bits (-2.7% in mean) and at 4 bits (-1.4%), and in five of six at 2 bits, where the effect is less consistent (-9.7% in mean, 15 of 18 pairs). Table 21 isolates it under the refit.

Their weighting does not survive the refit. The two-pass that improves our grid on every model at 2 bits produces isolated blow-ups on theirs — Llama-2-13B from 11.5 to 133 on the GPTQ engine (E10 vs E11) — and on the QuIP engine the failure is systematic: three of six models exceed perplexity 22, while our configuration stays below 12 on the same three (Q11 vs Q8, Table 22). The mechanism we propose is the one behind the uniform failure: refitting on a wider-tailed trajectory needs a grid that spreads levels by density, and a $p = 3$ weight concentrates them so aggressively that the refitted grid destabilizes. In aggregate at 2 bits our full method reaches 14.65 against 40.87 for their grid under GPTQ, and 9.78 against 24.12 under QuIP.

The configuration we did not adopt. Our weighting with *their* initialization, refit included (E16), beats E8 in 15 of 18 model–width cells (all six at 4 bits, all six at 3, three of six at 2, where Q8 beats Q13 on the QuIP engine). §5.2 discusses why E8 remains the main configuration.

Table 20: Every combination of the two LeanQuant-style changes, **per model**, WikiText-2 perplexity at 2 bits per-channel (mean of 3 seeds; bold = best in the row). The first four columns are single-pass; the last three add the trajectory refit. E6 vs E14 isolates the initialization at fixed weighting, E6 vs E13 the weighting at fixed initialization. The fourth refitted combination — their weighting with our initialization — is absent by choice: single-pass it is already the worst configuration in the table.

	single pass				+ trajectory refit		
	our weight		their weight		our weight		their weight
Model	our init (E6)	their init (E14)	our init (E13)	their init (E10)	our init **(E8)**	their init (E16)	their init (E11)
Llama-2-7B	11.954	12.275	17.077	14.061	**10.547**	11.204	14.237
Llama-2-13B	13.092	10.003	378	11.541	**8.754**	13.907	133
Llama-2-70B	6.131	5.975	6.703	6.002	5.807	**5.723**	5.852
Llama-3-8B	41.185	**33.532**	103	53.045	34.929	33.994	54.851
Mistral-7B	15.830	13.968	28.080	16.238	**13.543**	13.602	15.887
Sheared-2.7B	16.583	15.608	30.756	18.971	14.289	**14.227**	21.220

Table 21: Our quantile initialization versus the equispaced one, **both with the trajectory refit applied**, per model and width (per-channel, mean of 3 seeds; bold = better of the pair). Weighting, solver, engine and number of passes are identical; only the k-means starting point differs.

	4 bits		3 bits		2 bits	
Model	ours	equisp.	ours	equisp.	ours	equisp.
GPTQ						
Llama-2-7B	5.689	**5.660**	6.178	**6.105**	**10.547**	11.204
Llama-2-13B	5.084	**5.013**	5.455	**5.358**	**8.754**	13.907
Llama-2-70B	3.487	**3.450**	3.821	**3.755**	5.807	**5.723**
Llama-3-8B	6.754	**6.645**	8.161	**7.792**	34.929	**33.994**
Mistral-7B	5.485	**5.414**	5.968	**5.812**	**13.543**	13.602
Sheared-2.7B	7.141	**7.127**	7.676	**7.671**	14.289	**14.227**
QuIP						
Llama-2-7B	5.625	**5.612**	6.041	**6.022**	**9.534**	9.635
Llama-2-13B	5.005	**5.000**	5.299	**5.280**	7.561	**7.558**
Llama-2-70B	**3.425**	3.434	3.706	**3.689**	**5.192**	5.205
Llama-3-8B	**6.530**	6.555	7.509	**7.454**	15.907	**15.856**
Mistral-7B	5.371	**5.367**	5.702	**5.664**	**8.729**	8.819
Sheared-2.7B	7.068	**7.068**	7.507	**7.487**	**11.725**	11.756

D.8 LeanQuant's weighting as published

The LeanQuant-style weighting of Appendix D.7 follows Eq. (6) of Zhang & Shrivastava (2025). Their Algorithm 1 evaluates the weight on the Cholesky factor of the damped H^{-1} instead, $[\mathrm{chol}(H^{-1})]_{jj}^{-p}$ with $p = 4$ for all models, which equals d_j^2 with d_j the pivot of (5) (Appendix A.2): a different quantity from $(1/[H^{-1}]_{jj})^3$, not the same weight at another exponent. We run it with LeanQuant's equispaced initialization — LeanQuant's grid as published — single-pass (E10$^{\mathrm{p4}}$, Q10$^{\mathrm{p4}}$) and with the refit (E11$^{\mathrm{p4}}$, Q11$^{\mathrm{p4}}$), and, to separate weighting from initialization as in Appendix D.7, with our quantile initialization (E13$^{\mathrm{p4}}$, which is not LeanQuant's method). Everything else is as in the rest of the grid. Table 23 gives the means over models, Table 24 the paired statistics, and Table 9 the per-model values. Six Llama-2-70B runs did not complete because the GPU, shared with another job, ran out of memory (E10$^{\mathrm{p4}}$ at 3 and 4 bits, E11$^{\mathrm{p4}}$ at 3 bits, Q10$^{\mathrm{p4}}$ at every width); those cells are means of two seeds and the pairs involving them are dropped, hence the rows with 17 pairs.

Table 22: The same comparison as Table 20 with QuIP's LDLQ as the engine, per model, 2 bits per-channel (mean of 3 seeds; bold = best in the row).

	single pass		+ trajectory refit		
Model	our weight our init (Q6)	their weight their init (Q10)	our weight **our init (Q8)**	our weight their init (Q13)	their weight their init (Q11)
Llama-2-7B	10.078	9.963	**9.534**	9.635	41.711
Llama-2-13B	7.942	7.965	7.561	**7.558**	7.677
Llama-2-70B	5.381	5.404	**5.192**	5.205	8.309
Llama-3-8B	17.400	17.169	15.907	**15.856**	16.990
Mistral-7B	9.366	9.424	**8.729**	8.819	22.658
Sheared-2.7B	12.370	12.218	**11.725**	11.756	47.358

Table 23: **Our grid versus LeanQuant's grid as published**, with and without the trajectory refit (per-channel, WikiText-2 perplexity, mean over six models and three seeds; bold = best in the row). Same layout as Table 19, whose LeanQuant-style columns use the weighting of their Eq. (6). ‡: the mean includes a Llama-2-70B cell over two seeds.

	our grid		LeanQuant as published	
bits	1 pass	**+2P**	1 pass	+2P
GPTQ				
4	5.627	5.607	**5.580**‡	5.581
3	6.261	6.210	**6.164**‡	6.173‡
2	17.462	**14.645**	16.664	100.831
QuIP				
4	5.505	**5.504**	5.504‡	5.505
3	5.972	5.961	**5.928**‡	5.931
2	10.423	**9.775**	10.355‡	24.216

At 2 bits 2PTC beats LeanQuant as published. 2PTC is below LeanQuant's grid on every model with both engines and in all 35 (model, seed) pairs, by 13.0% with GPTQ (95% CI $[-16.4, -9.5]$) and 5.3% with QuIP ($[-6.2, -4.5]$).

The published form is the stronger baseline. With GPTQ, replacing the LeanQuant-style weighting by the published one lowers perplexity at every width (E10 $\rightarrow$ E10$^{\mathrm{p4}}$: -0.34%, -0.95% and -9.1% at 4, 3 and 2 bits, 47 of 52 pairs); with QuIP the two forms are within 0.12% of each other. The conclusions of Appendix D.7 hold against it: at fixed initialization our weighting is better (E14 below E10$^{\mathrm{p4}}$ in all 18 model–width cells and 51 of 52 pairs; E6 below E13$^{\mathrm{p4}}$ in all 16 reported cells), and their initialization is better at fixed weighting (E10$^{\mathrm{p4}}$ below E13$^{\mathrm{p4}}$ in all 16 reported cells).

At 3 and 4 bits their grid is as good as ours or better. Against LeanQuant as published, 2PTC is worse by 0.54% with GPTQ at 4 bits and by 0.54% with QuIP at 3 bits (CIs above zero), and tied in the other two cells. This is the regime in which the equispaced initialization also improves our own configuration (E16 against E8, §5.2).

The refit does not improve their grid. At 3 and 4 bits it leaves LeanQuant's grid unchanged (mean changes within 0.11%, CIs containing zero). At 2 bits it degrades it, as it degrades the LeanQuant-style grid (Appendix D.7): with GPTQ, Llama-2-13B goes from 10.58 to 511 and the median pair worsens by 3.1%; with QuIP, perplexity rises on four of six models and the median pair doubles ($+103\%$), with Llama-2-7B, Mistral-7B and Sheared-2.7B above 24 — the three models on which the LeanQuant-style grid fails under the refit (Table 22).

Their weighting with our initialization is numerically fragile. E13$^{\mathrm{p4}}$ is worse than both E10$^{\mathrm{p4}}$ and E6 in every reported cell. On Llama-2-7B it produced no finite perplexity at 3 bits (all three seeds) and at 4 bits (one seed), and those cells are left empty in Table 9. In further executions of the failing seeds most runs diverged again and some completed (perplexity 6.81 and 6.86 at 3 bits, 6.04 at 4 bits), consistent with the run-to-run GPU non-determinism of Appendix A.1 acting on a

Table 24: **LeanQuant as published at seed level.** Left: 2PTC against LeanQuant's grid as published ($E10^{p4} \to E8$, $Q10^{p4} \to Q8$); right: the refit applied to that grid ($E10^{p4} \to E11^{p4}$, $Q10^{p4} \to Q11^{p4}$). Same statistics as Table 4; for the refit the median is also given, because single blow-ups dominate its mean at 2 bits. Pairs involving a run that did not complete are dropped.

	LeanQuant as published → 2PTC			LeanQuant as published → + refit			
bits	wins	mean [95% CI]	p	wins	mean [95% CI]	median	p
QEP + GPTQ							
4	3/17	+0.54% [+0.33, +0.74]	1.000	6/17	+0.04% [-0.07, +0.14]	+0.05%	0.694
3	6/17	+0.72% [-0.18, +1.65]	0.811	7/17	+0.11% [-0.17, +0.40]	+0.21%	0.771
2	18/18	-12.96% [-16.36, -9.49]	$<10^{-4}$	7/18	+795.29% [+4.89, +1765.20]	+3.06%	0.973
QEP + QuIP							
4	7/17	+0.02% [-0.11, +0.13]	0.756	7/17	+0.02% [-0.05, +0.08]	+0.01%	0.741
3	2/17	+0.54% [+0.40, +0.68]	1.000	9/17	+0.04% [-0.07, +0.17]	-0.00%	0.771
2	17/17	-5.35% [-6.24, -4.49]	$<10^{-4}$	5/17	+135.62% [+65.77, +212.82]	+103.29%	0.999

fragile configuration; we therefore report no value rather than a mean over the runs that happened to converge.

D.9 Zero-shot results in full

Figure 7 and Tables 25–27 break down by task the averages reported in §5.1. At 2 bits, the uniform lane is close to the 25% chance level on ARC-Easy (24.9–31.4%) on every model except Llama-2-70B with QuIP (44.0%), consistently with the perplexities between 61.7 and 2.8×10^5 those configurations produce. At 4 bits the learned-codebook lanes, with or without the refit, are within two points of FP16 on every task (Table 28), and the differences between them carry little information.

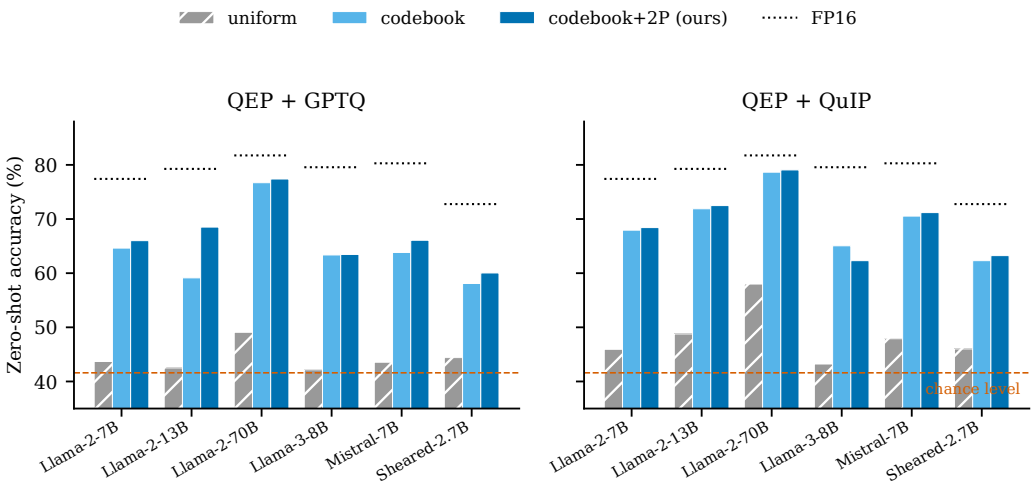


Figure 7: Zero-shot accuracy at 2 bits (average of ARC-Easy, PIQA and StoryCloze; per-channel, mean of 3 seeds), both engine lanes. Dotted lines are the FP16 references; the dashed line is chance level.

Table 25: Per-task zero-shot accuracy (%) at **2 bits**, per-channel, mean of 3 seeds, both engine lanes. u = uniform grid, cb = learned codebook, 2P = codebook + trajectory refit. Chance is 25% for ARC-Easy and 50% for PIQA and StoryCloze.

	ARC-Easy			PIQA			StoryCloze		
Model	u	cb	2P	u	cb	2P	u	cb	2P
QEP + GPTQ									
Llama-2-7B	26.6	57.0	58.6	53.0	68.6	69.6	51.5	68.3	69.8
Llama-2-13B	25.3	46.7	62.6	52.4	65.2	71.3	49.8	65.5	71.7
Llama-2-70B	30.6	76.0	76.9	57.0	77.4	77.8	59.6	76.8	77.5
Llama-3-8B	25.4	54.2	53.5	52.8	67.3	67.7	48.3	68.6	69.1
Mistral-7B	26.6	54.3	57.8	53.4	67.9	69.2	50.6	69.3	71.3
Sheared-2.7B	26.5	45.7	49.9	54.0	64.4	65.4	52.7	64.1	64.8
QEP + QuIP									
Llama-2-7B	28.6	62.0	62.9	54.5	70.7	71.0	54.7	71.1	71.4
Llama-2-13B	31.4	68.2	69.1	57.6	73.2	73.6	57.3	74.3	74.7
Llama-2-70B	44.0	78.8	79.4	64.3	78.7	78.8	65.7	78.4	79.1
Llama-3-8B	24.9	55.6	49.8	52.6	69.1	66.6	52.2	70.5	70.5
Mistral-7B	30.3	65.5	66.4	56.3	72.7	73.4	57.1	73.5	73.7
Sheared-2.7B	28.6	52.0	53.4	55.1	67.4	68.2	54.4	67.6	68.2

Table 26: Per-task zero-shot accuracy (%) at **3 bits**, per-channel, both lanes. Same column key as Table 25.

	ARC-Easy			PIQA			StoryCloze		
Model	u	cb	2P	u	cb	2P	u	cb	2P
QEP + GPTQ									
Llama-2-7B	67.2	73.8	73.4	74.0	76.6	76.6	75.1	76.1	76.2
Llama-2-13B	74.3	75.7	76.5	76.8	77.0	77.6	77.2	77.4	78.3
Llama-2-70B	80.2	82.4	82.4	80.4	81.3	81.4	79.2	80.2	80.4
Llama-3-8B	40.2	75.0	75.9	61.9	77.5	77.5	70.7	76.9	76.7
Mistral-7B	70.6	78.5	78.2	76.7	79.0	78.8	75.6	78.1	77.7
Sheared-2.7B	60.1	63.6	64.5	72.4	74.4	74.6	71.3	73.0	73.6
QEP + QuIP									
Llama-2-7B	69.8	73.7	74.1	76.1	77.1	77.1	75.0	76.8	76.4
Llama-2-13B	75.1	77.5	77.4	77.1	78.1	77.9	77.2	78.5	78.4
Llama-2-70B	80.5	82.2	82.2	80.8	82.0	81.7	79.5	80.4	80.5
Llama-3-8B	66.0	76.6	77.4	73.5	77.7	77.5	75.6	76.6	76.7
Mistral-7B	75.1	78.6	78.1	78.4	79.6	79.3	76.8	77.8	78.5
Sheared-2.7B	62.2	65.9	66.1	73.5	74.5	74.8	72.2	73.4	73.5

Table 27: Per-task zero-shot accuracy (%) at **4 bits**, per-channel, both lanes. Same column key as Table 25.

	ARC-Easy			PIQA			StoryCloze		
Model	u	cb	2P	u	cb	2P	u	cb	2P
QEP + GPTQ									
Llama-2-7B	75.2	75.6	75.4	77.2	77.5	77.4	77.4	78.0	77.9
Llama-2-13B	78.1	77.5	78.2	78.6	77.6	78.2	79.0	78.7	78.7
Llama-2-70B	82.3	82.6	82.4	82.2	81.8	81.7	80.5	80.8	80.7
Llama-3-8B	72.7	80.1	80.2	76.1	78.8	79.0	76.9	78.0	77.8
Mistral-7B	78.9	80.3	80.1	80.2	80.0	79.9	79.2	79.5	79.1
Sheared-2.7B	65.9	67.0	67.3	75.6	75.9	75.9	74.1	74.8	75.1
QEP + QuIP									
Llama-2-7B	75.2	75.7	75.5	77.8	77.5	77.8	77.5	77.3	77.5
Llama-2-13B	78.8	79.1	79.1	78.4	78.7	78.9	79.0	79.0	78.8
Llama-2-70B	82.6	82.7	82.9	81.9	82.0	81.7	80.8	80.6	80.8
Llama-3-8B	77.8	80.2	80.0	78.9	78.9	79.1	77.7	77.9	78.2
Mistral-7B	80.2	79.7	80.2	80.2	80.5	80.4	79.0	79.1	78.8
Sheared-2.7B	67.1	67.5	67.6	75.8	76.0	75.9	74.6	74.6	74.5

Table 28: FP16 baselines used throughout (WikiText-2 perplexity; zero-shot % accuracy per task).

Model	PPL	ARC-Easy	PIQA	StoryCloze
Llama-2-7B	5.472	76.3	78.1	77.8
Llama-2-13B	4.884	79.4	79.1	79.2
Llama-2-70B	3.321	82.7	82.2	80.3
Llama-3-8B	6.136	80.3	79.7	78.7
Mistral-7B	5.252	80.9	80.6	79.4
Sheared-2.7B	6.906	67.3	76.2	74.8