BARQ: BALANCED CODEBOOK REFINEMENT FOR LOW-BIT LLM QUANTIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

As large language models (LLMs) grow in parameter count, model storage and parameter memory traffic have become major bottlenecks to efficient deployment. Codebook-based weight quantization reduces these costs, but imbalanced nearest-codeword assignments during fitting can leave some codewords insufficiently updated, limiting effective codebook utilization. To address this limitation, we propose Balanced Assignment Refinement for Quantization (BARQ), which improves quantization quality through balanced fitting of existing codebooks. Specifically, we first compute joint soft assignments between weight blocks and codewords through entropically regularized optimal transport with uniform marginals and curvature-weighted reconstruction costs, ensuring equal positive fitting mass for every codeword in the exact solution. We then refine the codewords through an assignment-weighted barycentric update, which we prove minimizes the fitting objective for fixed assignments. For finite Sinkhorn iterations, the implemented update retains this optimality provided all codeword masses exceed the denominator floor. Finally, we discard the soft assignments and use the refined codebook for standard hard nearest-codeword encoding, with our analysis establishing sufficient conditions for reducing hard-quantization distortion and evaluation loss. Across multiple LLMs, BARQ achieves lower perplexity and higher mean zero-shot accuracy than the evaluated baselines at comparable bit budgets.

1 INTRODUCTION

Large language models (LLMs) (Grattafiori et al., 2024; Yang et al., 2025) continue to advance in language understanding, generation, and reasoning, but their growing parameter scale makes weight storage and memory traffic major bottlenecks to efficient inference and deployment. Weight-only quantization (Frantar et al., 2023; Lin et al., 2024; Dettmers et al., 2024; Kim et al., 2024; Chee et al., 2023) reduces model storage and data-transfer costs through low-bit parameter representations and has become a common approach to efficient LLM deployment. Preserving performance at low bit widths is essential for deployment.

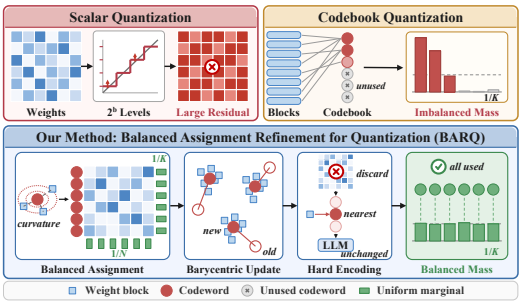

Figure 1: **Comparison of Codebook Fitting.** Prior methods concentrate fitting mass on a few codewords, leaving the rest never updated. In contrast, BARQ fits under uniform marginals, giving every codeword positive mass without changing the inference path.

Conventional scalar quantization maps individual weights to a limited set of quantization levels. At extremely low bit widths, this representation can struggle to preserve weight information, leading to substantial reconstruction error and model-quality degradation (van Baalen et al., 2024; Egiazarian et al., 2024). Codebook-based quantization (van Baalen et al., 2024; Egiazarian et al., 2024; Liu et al., 2024a; Tseng et al., 2024) jointly represents short weight blocks with shared codewords, offering a more flexible representation for extremely low-bit compression. However, realizing this representational advantage also depends on the effective use of codewords during fitting. Conventional codebook fitting (Lloyd, 1982; Ge et al., 2014; van Baalen et al., 2024) typically alternates between nearest-codeword assignment and codeword updates based on the assigned blocks.

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.

Independent nearest-codeword assignment (Bradley et al., 2000; Malinen & Fränti, 2014; Zheng & Vedaldi, 2023) places no constraint on the total fitting mass received by each codeword, potentially leaving some codewords with few or no supporting blocks and limiting effective codebook utilization. Further improving low-bit codebook quantization therefore calls for better codeword fitting and an analysis linking this refinement to final hard-quantization distortion and model performance.

To address this limitation, we propose Balanced Assignment Refinement for Quantization (BARQ), a fitting-stage refinement procedure for existing codebooks. Specifically, BARQ uses curvature-weighted reconstruction costs to compute joint soft assignments between weight blocks and codewords through entropically regularized optimal transport (Cuturi, 2013; Peyré & Cuturi, 2019) with uniform marginals, giving every codeword the same positive fitting mass in the exact solution. It then refines the codewords through assignment-weighted barycentric updates, which minimize the fitting objective for fixed assignments. For finite Sinkhorn iterations, the implemented update retains this optimality provided every codeword's fitting mass is at least the denominator floor. Finally, BARQ discards the soft assignments and uses the refined codebook for standard hard nearest-codeword encoding, preserving the codebook size, index width, and inference path. Our analysis proves that an exact assignment–update pair does not increase the balanced fitting objective and further establishes sufficient conditions for reducing final hard-quantization distortion and evaluation loss. The main contributions are summarized as follows:

- We propose BARQ, which combines curvature-weighted balanced entropic assignment with barycentric updates to refine existing codebooks during fitting while preserving the standard hard encoding rule and inference path.
- We establish a theoretical connection between balanced fitting and final quantization quality by proving that exact refinement does not increase the fitting objective, characterizing codeword optimality under finite Sinkhorn assignments, and deriving sufficient conditions for reduced hard-quantization distortion and evaluation loss.
- We evaluate BARQ across six LLMs. In paired comparisons with GPT-VQ at matched bit budgets, the mean per-configuration relative changes yield a 7.0% reduction in perplexity and a 3.0% improvement in five-task mean accuracy. Further analyses examine assignment concentration, hard codeword utilization, and curvature-weighted reconstruction error.

2 Related Work

Weight and Codebook Quantization. Scalar quantization improves low-bit LLM fidelity through error compensation (Frantar et al., 2023), activation-aware scaling (Lin et al., 2024; Xiao et al., 2023), quantization-parameter optimization (Badri & Shaji, 2023; Shao et al., 2024), sparse–dense decomposition (Kim et al., 2024), and weight rotations (Ashkboos et al., 2024; Liu et al., 2024b). Codebook-based methods jointly represent weight blocks using shared codewords. GPT-VQ (van Baalen et al., 2024) combines learned codebooks with error compensation, AQLM (Egiazarian et al., 2024) adopts additive codebooks, and QuIP# (Tseng et al., 2024) combines incoherence processing with lattice codebooks. VPTQ (Liu et al., 2024a) targets extreme low-bit vector quantization, while PV-Tuning (Malinovskii et al., 2024) improves the fine-tuning of compressed models. However, fitting learned codebooks through independent nearest-codeword assignment can yield imbalanced fitting mass, leaving some codewords insufficiently updated and limiting effective codebook utilization.

Balanced Assignment and Entropic Optimal Transport. Sinkhorn–Knopp scaling (Sinkhorn & Knopp, 1967; Knight, 2008) is a classical method for matrix balancing. Entropic regularization (Cuturi, 2013; Peyré & Cuturi, 2019) enables efficient approximate optimal transport via Sinkhorn iterations, motivating faster solvers (Altschuler et al., 2017; Lin et al., 2019) and transport-based objectives across machine learning (Frogner et al., 2015; Genevay et al., 2018; Mena et al., 2018). Balanced or uniform-marginal assignments have been used to avoid degenerate solutions and improve discrete representation utilization in settings such as representation learning and vector quantization (Caron et al., 2020; Asano et al., 2020). However, balanced codebook utilization alone does not guarantee lower hard-quantization distortion or evaluation loss in LLM weight quantization.

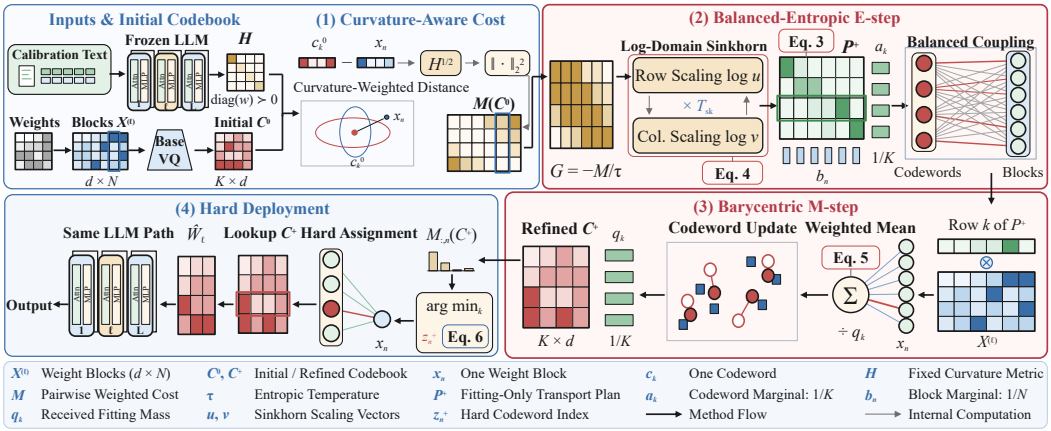

Figure 2: **Overview of the BARQ pipeline (VQ: vector quantization; MLP: multilayer perceptron). Curvature-Aware Cost:** compute the distances between weight blocks and initial codewords, and set the groupwise entropic temperature. **Balanced-Entropic E-step:** use log-domain Sinkhorn scaling to obtain soft assignments with uniform codeword and block marginals. **Barycentric M-step:** update each codeword to the assignment-weighted barycenter of the blocks. **Hard Deployment:** discard the transport state and encode the weights with the refined codebook using the original nearest-codeword rule. Only the refined codebook and hard indices are stored.

3 METHOD

This section introduces BARQ for post-training vector quantization of LLM weights. The presentation first reviews the standard codebook fitting formulation and the limitation of independent nearest-codeword assignments. It then describes how BARQ refines an existing codebook through balanced, curvature-aware assignments while preserving the original hard quantization rule for deployment. Detailed implementation and additional algorithmic details are provided in Appendix A.

3.1 PRELIMINARY: CODEBOOK FITTING

Following learned vector quantizers for LLM weights (van Baalen et al., 2024; Egiazarian et al., 2024; Liu et al., 2024a), we consider one group of a linear layer ℓ. The group contains N weight blocks $X^{(\ell)} = [x_1, \dots, x_N] \in \mathbb{R}^{d\times N}$, represented by a codebook $C = \{c_k\}_{k=1}^{K}$ with K codewords. Calibration data provide positive curvature weights $w \in \mathbb{R}^{d}_{++}$, which define the metric $H = \mathrm{diag}(w) \succ 0$. Given a codebook, hard quantization assigns each block to the codeword with the smallest curvature-weighted distance. We denote this distance by $M_{k,n}(C) = \|c_k - x_n\|_H^2 = (c_k - x_n)^\top H (c_k - x_n)$ and the corresponding assignment by $z_n(C) = \arg\min_k M_{k,n}(C)$. The reconstruction error is

$$D_{\mathrm{hard}}(C) = \sum_{n=1}^{N} b_n \min_k M_{k,n}(C), \tag{1}$$

where $b_n > 0$ specifies the contribution of block n and $\sum_n b_n = 1$.

To refine the codebook, we propose the following variant of the expectation–maximization (EM) algorithm. Instead of assigning each block to a single codeword during fitting, we use a soft assignment matrix $P \in \mathbb{R}_{+}^{K\times N}$, where $P_{k,n}$ is the mass assigned from block n to codeword k. We require the assignments to belong to $\Pi(a,b) = \{P \geq 0 : P\mathbf{1}_N = a,\ P^\top \mathbf{1}_K = b\}$. In our method, $a_k = 1/K$ and $b_n = 1/N$, so each block contributes the same mass and each codeword receives the same total fitting mass.

We jointly optimize codewords C and assignments $P \in \Pi(a,b)$ using the objective

$$\Phi_\tau(C,P) = R(C,P) + \tau\, \mathrm{KL}\big(P \,\big\|\, ab^\top\big), \tag{2}$$

where $R(C,P) = \sum_{k,n} P_{k,n} M_{k,n}(C)$ is the weighted reconstruction cost and $\mathrm{KL}(P\|ab^\top) = \sum_{k,n} P_{k,n} \log \frac{P_{k,n}}{a_k b_n}$ is the Kullback–Leibler (KL) regularizer. The temperature is $\tau = s\varepsilon > 0$, where s accounts for the cost scale of the group and ε controls the regularization strength. The marginal constraints balance the fitting mass, while the regularizer controls the softness of the assignments.

For the analysis, we also write $J_\tau(C) = \min_{P\in\Pi(a,b)} \Phi_\tau(C,P)$ for the objective after optimizing the assignments, and $D_{\mathrm{bal}}(C) = \min_{P\in\Pi(a,b)} R(C,P)$ for the corresponding balanced reconstruction cost without entropy regularization.

3.2 THE BARQ ALGORITHM

Following the objective above, we refine an initial codebook C^0 by first computing balanced soft assignments and then updating the codewords. We perform one assignment–update pair and use the codebook C^+ for hard quantization. Figure 2 shows the four stages; Algorithm 1 gives the steps.

Algorithm 1 BARQ fitting and hard deployment for one group.

Require: Blocks $\{x_n\}_{n=1}^N$; curvature weights $w \in \mathbb{R}^d_{++}$; initial codebook C^0; Sinkhorn steps T_{sk}.
Ensure: Refined codebook C^+ and hard indices $\{z_n^+\}_{n=1}^N$.
$H \leftarrow \mathrm{diag}(w)$; $a_k \leftarrow 1/K$; $b_n \leftarrow 1/N$.
$M_{k,n} \leftarrow (c_k^0 - x_n)^\top H (c_k^0 - x_n)$.
$s \leftarrow \max\{\mathrm{median}(M), 10^{-8}\}$; $\tau \leftarrow s\varepsilon$; $G \leftarrow -M/\tau$.
Initialize $\log u \leftarrow \mathbf{0}$ and $\log v \leftarrow \mathbf{0}$.
for $r = 1, \dots, T_{\mathrm{sk}}$ **do**
  $\log u_k \leftarrow \log a_k - \mathrm{LSE}_n(G_{k,n} + \log v_n)$.
  $\log v_n \leftarrow \log b_n - \mathrm{LSE}_k(G_{k,n} + \log u_k)$.
end for
$P^+_{k,n} \leftarrow \exp(\log u_k + G_{k,n} + \log v_n)$.
$c_k^+ \leftarrow \dfrac{\sum_n P^+_{k,n} x_n}{\max\{\sum_n P^+_{k,n}, 10^{-12}\}}$ for all k.
Discard P^+ and set $z_n^+ \leftarrow \arg\min_k (c_k^+ - x_n)^\top H (c_k^+ - x_n)$.
return C^+ and $\{z_n^+\}_{n=1}^N$.

(1) Curvature-Aware Cost. Given the initial codebook, we first compute the cost of assigning each weight block to each codeword. The entries of the cost matrix are $M_{k,n}(C^0) = \|H^{1/2}(c_k^0 - x_n)\|_2^2$. To account for differences in cost scale across groups, we set $s = \max\{\mathrm{median}(M(C^0)), 10^{-8}\}$. The temperature and the log kernel used in the assignment step are $\tau = s\varepsilon$, $G = -M(C^0)/\tau$. This makes the regularization strength relative to the initial distances within each group. We keep H, s, and τ fixed throughout the E-step and M-step, so both steps optimize the same fitting objective.

(2) Balanced-Entropic E-step. Keeping the codewords fixed, we find the assignment matrix that minimizes Equation (2) under the prescribed marginal constraints:

$$P^+ = P_\tau(C^0) = \arg\min_{P\in\Pi(a,b)} \Phi_\tau(C^0, P). \tag{3}$$

Unlike independent nearest-codeword assignments, this step assigns the blocks jointly so that every codeword receives its prescribed mass. We solve the assignment problem using Sinkhorn scaling. To avoid explicitly forming small exponential kernel entries, we carry out the updates in the log domain:

$$\log u_k \leftarrow \log a_k - \mathrm{LSE}_n(G_{k,n} + \log v_n), \qquad \log v_n \leftarrow \log b_n - \mathrm{LSE}_k(G_{k,n} + \log u_k). \tag{4}$$

where LSE denotes log-sum-exp over the indicated index. The two updates alternately enforce the codeword and block marginals. At convergence, the assignment matrix is recovered as $P^+_{k,n} = \exp(\log u_k + G_{k,n} + \log v_n)$. Theorem 1 establishes this scaling form and the existence of a unique positive solution.

(3) Barycentric M-step. Keeping the assignments found in the E-step fixed, we find the codeword values that minimize the reconstruction cost. The entropy term does not depend on the codewords, so the objective for codeword k is $\sum_n P^+_{k,n}(c_k - x_n)^\top H(c_k - x_n)$. This objective is a quadratic form in c_k. Since the same positive-definite metric H applies to every block, the optimal value is computed in closed form as

$$c_k^+ = \frac{\sum_n P^+_{k,n} x_n}{q_k(P^+)} = \frac{\sum_n P^+_{k,n} x_n}{a_k}, \tag{5}$$

where $q_k(P^+) = \sum_n P^+_{k,n} = a_k$ is the total mass assigned to codeword k. Thus, each updated codeword is the assignment-weighted barycenter of the blocks. The uniform marginal gives $q_k(P^+) = 1/K > 0$, so every codeword has a well-defined update. Theorem 2 shows that this update is the exact fixed-plan minimizer and establishes the fitting-objective descent of the exact pair (P^+, C^+).

(4) Hard Deployment. After updating the codebook, we quantize the original weight blocks using the same nearest-codeword rule as the underlying vector quantizer:

$$z_n^+ = z_n(C^+) = \arg\min_k M_{k,n}(C^+). \tag{6}$$

Each block is reconstructed as $\widehat{x}_n = c^+_{z^+_n}$. The balanced assignments are used only to fit the codebook; they do not constrain the number of blocks assigned to each codeword after hard quantization. We discard P^+ and the Sinkhorn state and retain only C^+ and the hard indices $\{z^+_n\}_{n=1}^N$. Consequently, the codebook size, index width, lookup mechanism, and LLM inference path remain unchanged. In practice, we run T_{sk} Sinkhorn iterations and compute each codeword update using its actual row mass, with a small numerical floor in the denominator. The exact marginal equalities and descent guarantees refer to the converged E-step.

4 Theoretical Analysis

This section analyzes when balanced codebook fitting improves the hard quantizer used at deployment. We first establish existence and positivity of the balanced-entropic assignment and characterize the barycentric codeword update. We then derive fitting-objective descent, give a sufficient condition for improvement after hard encoding, and connect hard distortion to downstream model metrics. Appendix D develops the theoretical details; Appendix E contains the full proofs.

4.1 Balanced-Entropic E-step

Stage 2 minimizes Equation (2) over $\Pi(a, b)$. The following result ensures that the assignment is well defined and that every codeword has positive support for the subsequent update.

Theorem 1 (Well-posed balanced-entropic E-step)**.** *Assume finite costs, $\tau > 0$, and strictly positive probability marginals a and b. The minimizer $P_\tau(C) = \arg\min_{P\in\Pi(a,b)} \Phi_\tau(C, P)$ exists, is unique, and is entrywise positive. It has the form*

$$P_\tau(C) = \mathrm{diag}(u)\exp[-M(C)/\tau]\,\mathrm{diag}(v), \tag{7}$$

where $u \in \mathbb{R}^K_{++}$ and $v \in \mathbb{R}^N_{++}$ are unique up to reciprocal scalar rescaling.

In particular, $q_k(P_\tau(C)) = a_k > 0$ for every codeword, and the uniform marginal gives exactly $1/K$ fitting mass to each. The entropic term makes the coupling unique and entrywise positive, while Equation (7) gives the scaling representation used by the log-domain Sinkhorn updates (Cuturi, 2013; Peyré & Cuturi, 2019).

4.2 Barycentric M-step and Profiled Descent

For fixed assignments, Stage 3 minimizes reconstruction cost by moving each codeword to the weighted center of its assigned blocks.

Definition 1 (Barycentric codeword)**.** For $P \geq 0$, the barycentric codeword of a row with $q_k(P) > 0$ is $c^P_k = q_k(P)^{-1}\sum_n P_{k,n}x_n$. When all row sums are positive, write $C^P = \{c^P_k\}_{k=1}^K$.

Theorem 2 (Barycentric M-step and one-pair descent)**.** *Let $P \geq 0$ have positive row sums. Then C^P uniquely minimizes $R(C, P)$, and the excess cost of any C over C^P is exactly $\sum_k q_k(P)\|c_k - c^P_k\|^2_H$.*

Fix $H \succ 0$, a, b, and $\tau > 0$. For finite costs, the exact pair $P^+ = P_\tau(C^0)$, $C^+ = C^{P^+}$ *satisfies*

$$J_\tau(C^+) \leq J_\tau(C^0) - \sum_k a_k\|c^+_k - c^0_k\|^2_H. \tag{8}$$

Consequently, $J_\tau(C^+) < J_\tau(C^0)$ whenever $C^+ \neq C^0$. For every codebook C, hard distortion is bounded by balanced reconstruction cost and the regularized fitting objective:

$$D_{\mathrm{hard}}(C) \leq D_{\mathrm{bal}}(C) \leq J_\tau(C).$$

The fixed-plan identity is the weighted second Lloyd condition: codeword movement accounts exactly for the reconstruction cost removed by the M-step. Equation (8) therefore guarantees descent of the fitting objective.

4.3 Transfer to Hard Distortion

For the exact pair generated from C^0, let $\Delta_0 = \sum_k a_k\|c^+_k - c^0_k\|^2_H$ be the barycentric gain and let $\Gamma_0 = R(C^0, P^+) - D_{\mathrm{hard}}(C^0)$ be the assignment penalty. These quantities compare the reconstruction gain from moving the codewords with the cost of changing their fitting assignments.

Theorem 3 (Transfer to hard distortion)**.** *For the exact pair generated from C^0, suppose $m_Q = \Delta_0 - \Gamma_0 > 0$. The hard distortion then satisfies*

$$D_{\mathrm{hard}}(C^+) \le D_{\mathrm{hard}}(C^0) - m_Q. \tag{9}$$

The condition is sufficient: hard reassignment after the M-step can remove additional cost. Appendix E.4 gives the exact decomposition and two upper bounds on Γ_0, separating the cost of balancing from the entropic contribution.

4.4 From Hard Distortion to Model Metrics

For a linear layer W partitioned into groups g, use the uniform block weights $b_n = 1/N_g$ and write $D(C) = \sum_g N_g D^{(g)}_{\mathrm{hard}}(C)$ for its aggregate hard distortion. The groupwise certificates give $D(C^+) - D(C^0) \le -m_{\mathrm{tot}}$, where $m_{\mathrm{tot}} = \sum_g N_g m^{(g)}_Q$. Let $Q_C(W)$ be the hard-decoded weight matrix, $\Delta W_C = Q_C(W) - W$, and $\Sigma = \mathbb{E}[\xi\xi^\top]$ the second moment of the layer input ξ.

For the full model, stack the perturbations of all quantized layers into δ_C and collect the group metrics into the correspondingly ordered block-diagonal matrix $\widetilde{H}$, so that $D(C) = \delta_C^\top \widetilde{H} \delta_C$ now denotes total distortion. Write $\mathcal{L}(C)$ and $\mathcal{L}(W)$ for the average per-token negative log-likelihoods of the quantized and original models. The approximation term $\eta_C = |\mathcal{L}(C) - \mathcal{L}(W) - \frac{1}{2}D(C)|$ measures the discrepancy between the actual loss change and its quadratic approximation.

Theorem 4 (Layer-output, loss, and perplexity transfer)**.** *For every codebook, $\mathbb{E}\|\Delta W_C \xi\|_2^2 = \mathrm{tr}(\Delta W_C \Sigma \Delta W_C^\top)$. For one layer, assume that (i) $\Sigma = \mathrm{diag}(\sigma)$ with $\sigma \ge 0$; (ii) the blocks partition W, and each block of group g consists of d consecutive entries of one row in the same input-coordinate set J_g; and (iii) $H_g = \kappa\,\mathrm{diag}(\sigma_{J_g})$ for a common $\kappa > 0$. Then*

$$D(C) = \kappa\,\mathbb{E}\|\Delta W_C \xi\|_2^2, \tag{10}$$

so any strict reduction in layer distortion reduces expected layer-output error.

For the full model, suppose $D(C^+) - D(C^0) \le -m_{\mathrm{tot}}$. Then

$$\mathcal{L}(C^+) - \mathcal{L}(C^0) \le -\tfrac{1}{2} m_{\mathrm{tot}} + \eta_{C^+} + \eta_{C^0}. \tag{11}$$

In particular, $m_{\mathrm{tot}} > 2(\eta_{C^+} + \eta_{C^0})$ implies strict decreases in both negative log-likelihood and perplexity, where $\mathrm{PPL}(C) = \exp \mathcal{L}(C)$. Equation (60) *gives the perplexity ratio bound.*

The layer identity assumes alignment between the refinement metric and the input second moment, as discussed in Appendix A. For the full model, loss improvement follows when the distortion margin exceeds the approximation terms characterized in Lemma 1.

5 Experiments

This section examines three questions. How does BARQ compare with GPT-VQ across LLMs? How does it perform against external low-bit baselines at comparable bit budgets? And how do the observed differences change with codebook size?

5.1 Experimental Setup

Models and Quantization Settings. We evaluate six LLMs: Llama3.2-1B and Llama3.1-8B from the Llama 3 family (Grattafiori et al., 2024), and Qwen3-1.7B, Qwen3-8B, Qwen3-14B, and Qwen3-32B from Qwen3 (Yang et al., 2025). For calibration, we use 128 sequences of length 2048 from WikiText-2 (Merity et al., 2017). We consider weight-only quantization and the block dimension is $d \in \{1, 2, 4\}$, while the group size is selected from $\{32, 64, 128\}$. We additionally evaluate a large-group setting with $d = 4$ and $\mathrm{gs} = 1024$.

Bit Budget. We account for total bits per value (bpv) using the GPT-VQ (van Baalen et al., 2024) metric: $\mathrm{bpv} = \frac{\log_2 K}{d} + \frac{K\,d\,b_c}{l} + \frac{16}{\mathrm{gs}}$, where $K = 2^{Bd}$ is the codebook size, b_c the codebook bit-width, l the number of weights sharing one codebook, and gs the scale group size. The index term is $B = 2$ bits per value throughout, and the scale term adds 0.5, 0.25 and 0.125 at $\mathrm{gs} = 32$, 64 and

Table 1: **Paired comparison with GPT-VQ across 60 configurations.** BARQ lowers PPL in 54/60 configurations and raises Avg. in 53/60; both improve in 47/60. Results cover six models in two panels. PPL is WikiText-2 perplexity; Avg. is the five-task zero-shot mean (%). For each metric, the GPT-VQ baseline and BARQ (Ours) are shown side by side and the better value is bold (superscript: Ours − GPT-VQ). Full per-task accuracies are in Appendix B.

Setting			PPL ↓		Avg. (%) ↑		Setting			PPL ↓		Avg. (%) ↑	
d	gs	bpv	GPT-VQ	**Ours**	GPT-VQ	**Ours**	d	gs	bpv	GPT-VQ	**Ours**	GPT-VQ	**Ours**
			Llama3.2-1B							**Llama3.1-8B**			
1	32	2.500	28.49	**27.70**$^{-0.79}$	46.24	**46.57**$^{+0.33}$	1	32	2.500	12.57	**12.17**$^{-0.40}$	**61.22**	59.60$^{-1.62}$
	64	2.250	**44.24**	46.36$^{+2.12}$	42.48	**43.49**$^{+1.01}$		64	2.250	14.73	**14.06**$^{-0.67}$	54.65	**55.01**$^{+0.36}$
	128	2.125	95.41	**92.04**$^{-3.37}$	**39.66**	39.64$^{-0.02}$		128	2.125	15.91	**15.59**$^{-0.32}$	47.74	**51.03**$^{+3.29}$
2	32	2.500	14.67	**13.25**$^{-1.42}$	57.85	**59.59**$^{+1.74}$	2	32	2.500	8.89	**8.32**$^{-0.57}$	71.42	**71.74**$^{+0.32}$
	64	2.250	20.04	**16.57**$^{-3.47}$	51.08	**54.32**$^{+3.24}$		64	2.250	9.63	**9.13**$^{-0.50}$	66.20	**68.01**$^{+1.81}$
	128	2.125	26.95	**22.00**$^{-4.95}$	48.19	**49.57**$^{+1.38}$		128	2.125	10.92	**10.42**$^{-0.50}$	60.59	**65.39**$^{+4.80}$
4	32	2.500	14.95	**13.19**$^{-1.76}$	58.15	**60.01**$^{+1.86}$	4	32	2.500	6.62	**6.15**$^{-0.47}$	72.11	**72.74**$^{+0.63}$
	64	2.250	15.40	**13.23**$^{-2.17}$	57.67	**59.51**$^{+1.84}$		64	2.250	6.83	**6.16**$^{-0.67}$	71.08	**72.95**$^{+1.87}$
	128	2.125	16.10	**13.21**$^{-2.89}$	56.38	**59.65**$^{+3.27}$		128	2.125	11.00	**8.30**$^{-2.70}$	69.47	**72.07**$^{+2.60}$
	1024	2.500	19.09	**13.84**$^{-5.25}$	52.23	**57.97**$^{+5.74}$		1024	2.500	10.69	**8.38**$^{-2.31}$	64.86	**70.52**$^{+5.66}$
			Qwen3-1.7B							**Qwen3-8B**			
1	32	2.500	**31.97**	33.41$^{+1.44}$	44.03	**44.54**$^{+0.51}$	1	32	2.500	**12.34**	12.54$^{+0.20}$	63.40	**63.54**$^{+0.14}$
	64	2.250	52.57	**50.23**$^{-2.34}$	40.36	**41.30**$^{+0.94}$		64	2.250	13.80	**13.64**$^{-0.16}$	**61.87**	60.74$^{-1.13}$
	128	2.125	68.21	**63.03**$^{-5.18}$	38.08	**39.67**$^{+1.59}$		128	2.125	**14.70**	14.89$^{+0.19}$	57.18	**58.57**$^{+1.39}$
2	32	2.500	17.62	**16.96**$^{-0.66}$	59.46	**60.97**$^{+1.51}$	2	32	2.500	9.93	**9.70**$^{-0.23}$	70.46	**71.70**$^{+1.24}$
	64	2.250	27.26	**23.02**$^{-4.24}$	47.14	**52.80**$^{+5.66}$		64	2.250	10.96	**10.73**$^{-0.23}$	66.36	**69.59**$^{+3.23}$
	128	2.125	40.49	**26.49**$^{-14.00}$	42.46	**46.68**$^{+4.22}$		128	2.125	12.62	**11.35**$^{-1.27}$	63.12	**66.25**$^{+3.13}$
4	32	2.500	20.37	**16.69**$^{-3.68}$	54.61	**61.49**$^{+6.88}$	4	32	2.500	10.16	**9.75**$^{-0.41}$	70.31	**71.71**$^{+1.40}$
	64	2.250	19.24	**17.21**$^{-2.03}$	57.20	**60.88**$^{+3.68}$		64	2.250	10.13	**9.75**$^{-0.38}$	70.46	**71.71**$^{+1.25}$
	128	2.125	20.54	**16.59**$^{-3.95}$	55.75	**59.68**$^{+3.93}$		128	2.125	10.28	**9.72**$^{-0.56}$	70.68	**71.46**$^{+0.78}$
	1024	2.500	24.57	**17.48**$^{-7.09}$	50.92	**59.82**$^{+8.90}$		1024	2.500	10.26	**9.73**$^{-0.53}$	67.61	**71.13**$^{+3.52}$
			Qwen3-14B							**Qwen3-32B**			
1	32	2.500	10.80	**10.73**$^{-0.07}$	**69.45**	67.40$^{-2.05}$	1	32	2.500	9.10	**8.95**$^{-0.15}$	71.17	**72.08**$^{+0.91}$
	64	2.250	11.29	**11.14**$^{-0.15}$	**66.78**	65.90$^{-0.88}$		64	2.250	9.54	**9.40**$^{-0.14}$	**70.76**	70.41$^{-0.35}$
	128	2.125	11.80	**11.79**$^{-0.01}$	**64.55**	63.23$^{-1.32}$		128	2.125	**9.86**	9.88$^{+0.02}$	69.98	**70.87**$^{+0.89}$
2	32	2.500	8.79	**8.65**$^{-0.14}$	74.70	**75.34**$^{+0.64}$	2	32	2.500	7.76	**7.62**$^{-0.14}$	75.91	**76.39**$^{+0.48}$
	64	2.250	9.61	**9.33**$^{-0.28}$	72.41	**72.88**$^{+0.47}$		64	2.250	8.37	**8.09**$^{-0.28}$	73.78	**74.77**$^{+0.99}$
	128	2.125	10.31	**9.92**$^{-0.39}$	69.42	**69.69**$^{+0.27}$		128	2.125	8.91	**8.55**$^{-0.36}$	72.44	**73.96**$^{+1.52}$
4	32	2.500	**8.78**	8.95$^{+0.17}$	74.72	**74.73**$^{+0.01}$	4	32	2.500	7.73	**7.61**$^{-0.12}$	76.31	**76.36**$^{+0.05}$
	64	2.250	8.81	**8.67**$^{-0.14}$	74.57	**74.92**$^{+0.35}$		64	2.250	7.85	**7.61**$^{-0.24}$	75.81	**76.47**$^{+0.66}$
	128	2.125	8.93	**8.66**$^{-0.27}$	74.59	**75.14**$^{+0.55}$		128	2.125	7.84	**7.61**$^{-0.23}$	75.47	**76.41**$^{+0.94}$
	1024	2.500	9.11	**8.73**$^{-0.38}$	73.36	**74.78**$^{+1.42}$		1024	2.500	8.15	**7.69**$^{-0.46}$	74.56	**76.37**$^{+1.81}$

128; Appendix A.1 works through the three terms. BARQ changes neither K, d, l nor the index width, so each paired row of Table 1 compares the two methods at one budget.

Compared Methods and Evaluation Metrics. We compare against GPT-VQ (van Baalen et al., 2024), GPTQ (Frantar et al., 2023), HQQ (Badri & Shaji, 2023), AQLM (Egiazarian et al., 2024) with and without block fine-tuning (BFT), and QuIP (Chee et al., 2023). We report WikiText-2 perplexity and zero-shot accuracy on ARC-Easy, ARC-Challenge (Clark et al., 2018), HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2020), and WinoGrande (Sakaguchi et al., 2021). All evaluations use the lm-evaluation-harness (Gao et al., 2024).

5.2 Comparison with GPT-VQ

Table 1 reports the paired GPT-VQ and BARQ outputs on all six models. Each row shares calibration, weights, nominal codebook size, and reported bits-per-value metadata. On average, BARQ reduces perplexity by 7.0% and improves five-task mean accuracy by 3.0%. The observed mean difference is larger in the larger-codebook strata: the average relative perplexity reduction is 1.2%, 8.2%, and 10.4% at $K \in \{4, 16, 256\}$, while the corresponding accuracy increase is 0.7%, 3.6%, and 4.2%. Individual regressions remain concentrated at $K = 4$, with one perplexity regression at $K = 256$. Tables 4 and 5 in Appendix B give the per-task results.

5.3 Comparison Against Baselines

Table 2 compares BARQ with GPTQ, HQQ, QuIP, VPTQ, and AQLM with and without block fine-tuning. The BARQ and GPT-VQ rows use the lowest-bpv complete paired setting per model. VPTQ uses a 3-bit index budget, excluding codebooks and metadata. Across the dis-

Table 2: **External baseline comparison at the displayed budgets.** VPTQ reports 3.000 index bpv, excluding codebooks and metadata. PPL is WikiText-2 perplexity; Avg. is mean five-task accuracy (%). A dash denotes quantization collapse in PPL and an unavailable result in Avg. Bold/underline mark best/second best; Δ gives BARQ − strongest baseline. Table 6 gives per-task results.

Method	bpv	PPL ↓	Avg. (%) ↑	Method	bpv	PPL ↓	Avg. (%) ↑
Llama3.2-1B				**Llama3-8B**			
GPTQ	2.500	209.53	36.38	GPTQ	2.500	32.59	36.88
HQQ	2.500	4917.51	37.32	HQQ	2.500	204.69	43.80
QuIP	2.000	377.93	–	QuIP	2.000	38.06	37.94
VPTQ	3.000	15.91	56.24	VPTQ	3.000	9.11	69.63
AQLM (no BFT)	2.067	20.30	51.69	AQLM (no BFT)	2.067	11.05	62.20
AQLM (BFT)	2.067	17.39	52.62	AQLM (BFT)	2.067	10.18	64.99
GPT-VQ	2.125	16.10	56.38	GPT-VQ	2.125	11.00	69.47
BARQ (Ours)	2.125	**13.21**	**59.65**	**BARQ (Ours)**	2.125	**8.30**	**72.07**
Δ *vs. best baseline*		−2.70	+3.27	Δ *vs. best baseline*		−0.81	+2.44
Qwen3-1.7B				**Qwen3-8B**			
GPTQ	2.500	666.14	35.62	GPTQ	2.500	17.94	46.00
HQQ	2.500	1018.43	37.15	HQQ	2.500	46.57	44.89
QuIP	2.000	333.99	–	QuIP	2.000	25.51	41.77
VPTQ	3.000	–	36.32	VPTQ	3.000	10.65	70.83
AQLM (no BFT)	2.067	21.68	48.75	AQLM (no BFT)	2.067	10.73	65.80
AQLM (BFT)	2.067	18.30	53.50	AQLM (BFT)	2.067	10.39	66.19
GPT-VQ	2.125	20.54	55.75	GPT-VQ	2.125	10.28	70.68
BARQ (Ours)	2.125	**16.59**	**59.68**	**BARQ (Ours)**	2.125	**9.72**	**71.46**
Δ *vs. best baseline*		−1.71	+3.93	Δ *vs. best baseline*		−0.56	+0.63
Qwen3-14B				**Qwen3-32B**			
GPTQ	2.500	12.90	55.49	GPTQ	2.500	14.26	54.98
HQQ	2.500	18.40	60.01	HQQ	2.500	15.11	60.83
QuIP	2.000	14.83	49.13	QuIP	2.000	14.01	41.91
VPTQ	3.000	9.32	72.99	VPTQ	3.000	8.50	72.34
AQLM (no BFT)	2.067	9.42	68.65	AQLM (no BFT)	2.067	8.83	73.84
AQLM (BFT)	2.067	9.09	70.03	AQLM (BFT)	2.067	8.39	74.38
GPT-VQ	2.125	8.93	74.59	GPT-VQ	2.125	7.84	75.47
BARQ (Ours)	2.125	**8.66**	**75.14**	**BARQ (Ours)**	2.125	**7.61**	**76.41**
Δ *vs. best baseline*		−0.27	+0.55	Δ *vs. best baseline*		−0.23	+0.94

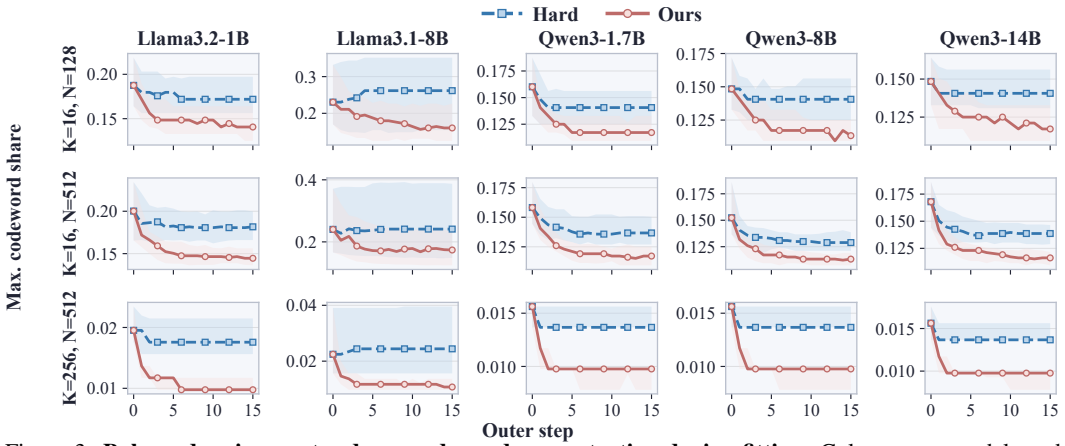

Figure 3: **Balanced assignment reduces codeword concentration during fitting.** Columns are models and rows are the three settings $(K, N) \in \{(16, 128), (16, 512), (256, 512)\}$. The x-axis is the outer refinement step and the y-axis is the maximum codeword share, the fraction of blocks held by the most-used codeword. Each curve is the median over the matched shallow and deep attention tensors and the sampled weight groups of that cell, and the band is the interquartile range. Blue dashed: hard assignment; red solid: BARQ.

played rows, BARQ has the lowest perplexity and highest five-task mean on all six models. Table 6 gives the full per-task results.

5.4 Codebook Utilization During Fitting

Figure 3 shows how the maximum codeword share evolves during codebook fitting. Across all five models and three (K, N) settings, BARQ rapidly reduces the share of the most-used codeword and maintains a lower concentration than hard assignment throughout refinement.

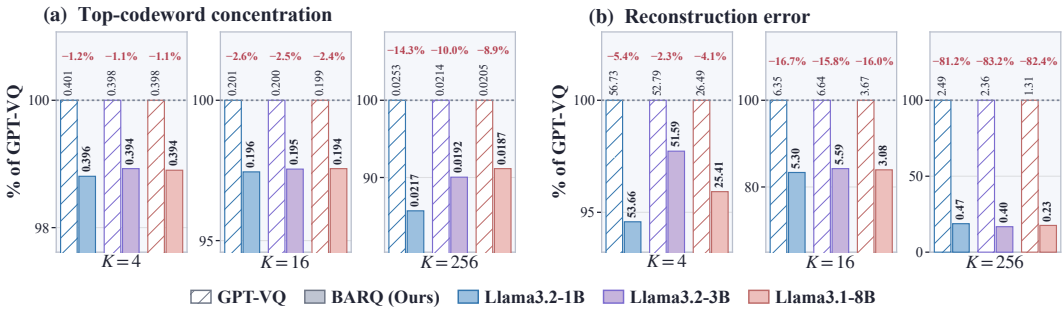

Figure 4: **Flatter codeword concentration and lower curvature-weighted reconstruction error across model scales.** Each codebook size has its own panel and each bar is scaled to its paired GPT-VQ value, so the vertical axes read in percent of GPT-VQ and are zoomed to their panel instead of starting at zero; the number on a bar is its absolute value and the red tag the exact reduction. Color marks the model, hatched bars are GPT-VQ and solid bars are BARQ. **(a)** Mean maximum codeword share $p_{\max}$, lower being flatter at encoding. **(b)** Curvature-weighted reconstruction error $\mathrm{avg}_g \, \|W_g - \widehat{W}_g\|_H^2$, bar labels in units of 10^{-6}.

Codeword Concentration and Reconstruction Error. Two post-refinement measurements, recorded on Llama3.2-1B, Llama3.2-3B and Llama3.1-8B (Grattafiori et al., 2024) rather than on all six evaluated models, separate the two effects. The first is the maximum codeword share $p_{\max} = \max_k p_k$, where p_k is the fraction of blocks routed to codeword k at encoding; a lower value means a flatter realized distribution. It is lower for the paired BARQ output at every scale and codebook size in Figure 4**(a)**, and the layerwise direction holds throughout $K = 16$ and almost throughout $K = 256$, with one exception at the 8B scale. The second is the curvature-weighted reconstruction error, which in Figure 4**(b)** changes far more than the count of distinct codewords: on Llama3.2-1B its weighted aggregate falls from 6.35×10^{-6} to 5.30×10^{-6} at $K = 16$ and from 2.49×10^{-6} to 4.67×10^{-7} at $K = 256$. Both changes are consistent with codeword relocation rather than with utilization alone.

After Hard Encoding. Figure 3 and Figure 4 capture different points in the procedure. The first follows maximum codeword share through refinement; the second evaluates the resulting weights after hard nearest-codeword encoding, the representation used at inference. The concentration difference is therefore visible both during fitting and after final encoding. The paired settings keep nominal K and reported bpv fixed (Table 1), so the final gap reflects a different distribution of blocks over the available codewords. Figure 4**(b)** also records the curvature-weighted cost of that encoded representation. At $K = 16$ and $K = 256$, the BARQ bars are lower for all three plotted models, matching the codebook-size strata with larger average perplexity reductions in Table 1.

Across the three plotted models, the mean top-codeword share falls by about 1% at $K = 4$, 2–3% at $K = 16$, and 9–14% at $K = 256$ relative to paired GPT-VQ. The corresponding curvature-weighted error reductions are about 2–5%, 16%, and 81–83%, respectively. These post-encoding changes are largest where the six-model matched comparison also records its greatest mean perplexity reduction.

6 Conclusion

This work presents BARQ, a balanced codebook fitting method for post-training vector quantization of LLMs. By replacing independent hard assignments with balanced entropic assignments during refinement and updating codewords through barycentric fitting, BARQ improves codebook utilization while preserving the standard hard nearest-codeword quantization rule at deployment. The theoretical analysis establishes well-posed balanced fitting, objective descent under the codeword update, and sufficient conditions under which these improvements transfer to lower hard-quantization distortion and better model metrics. Empirically, BARQ generally improves quantization quality across different LLMs and low-bit settings, with more balanced codeword usage and lower reconstruction error. The paired comparisons and post-encoding diagnostics place this effect at both the model-output and weight-reconstruction levels. At $K = 4$, both methods nearly saturate the codebook and the average perplexity gain is smaller. Larger codebooks show stronger average perplexity reductions and larger post-encoding decreases in curvature-weighted reconstruction error across the three plotted models. These measurements connect codebook fitting to the quality of the hard quantizer used at inference.

Reproducibility Statement

The main text describes the refinement procedure and experimental setup. Appendix A gives the implementation details, Appendix B describes the evaluation and table construction, and Appendix E contains the proofs of the theoretical results.

AI Use Statement

Generative AI tools were used to polish the grammar and wording of the manuscript. The authors reviewed the final text and take responsibility for its content.

References

Jason Altschuler, Jonathan Niles-Weed, and Philippe Rigollet. Near-linear time approximation algorithms for optimal transport via Sinkhorn iteration. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Yuki M. Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. In *International Conference on Learning Representations (ICLR)*, 2020.

Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. QuaRot: Outlier-free 4-bit inference in rotated LLMs. *arXiv preprint arXiv:2404.00456*, 2024.

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

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

Paul S. Bradley, Kristin P. Bennett, and Ayhan Demiriz. Constrained k-means clustering. Technical Report MSR-TR-2000-65, Microsoft Research, May 2000.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

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

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.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2013.

Tim Dettmers, Ruslan Svirschevski, Vage Egiazarian, Denis Kuznedelev, Elias Frantar, Saleh Ashkboos, Alexander Borzunov, Torsten Hoefler, and Dan Alistarh. SpQR: A sparse-quantized representation for near-lossless LLM weight compression. In *International Conference on Learning Representations*, 2024.

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

Charlie Frogner, Chiyuan Zhang, Hossein Mobahi, Mauricio Araya-Polo, and Tomaso Poggio. Learning with a Wasserstein loss. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, et al. A framework for few-shot language model evaluation. `https://github.com/EleutherAI/lm-evaluation-harness`, 2024.

Tiezheng Ge, Kaiming He, Qifa Ke, and Jian Sun. Optimized product quantization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(4):744–755, 2014.

Aude Genevay, Gabriel Peyré, and Marco Cuturi. Learning generative models with sinkhorn divergences. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2018.

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

Robert M. Gray. Vector quantization. *IEEE ASSP Magazine*, 1(2):4–29, 1984.

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

Hervé Jégou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):117–128, 2011.

Sehoon Kim, Coleman Richard Charles Hooper, Amir Gholami, Zhen Dong, Xiuyu Li, Sheng Shen, Michael W. Mahoney, and Kurt Keutzer. SqueezeLLM: Dense-and-sparse quantization. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 23901–23923. PMLR, 2024.

Philip A. Knight. The Sinkhorn–Knopp algorithm: Convergence and applications. *SIAM Journal on Matrix Analysis and Applications*, 30(1):261–275, 2008.

Yann LeCun, John S. Denker, and Sara A. Solla. Optimal brain damage. In *Advances in Neural Information Processing Systems (NeurIPS)*, 1990.

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*, volume 6, 2024.

Tianyi Lin, Nhat Ho, and Michael Jordan. On efficient optimal transport: An analysis of greedy and accelerated mirror descent algorithms. In *International Conference on Machine Learning (ICML)*, 2019.

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. *arXiv preprint arXiv:2409.17066*, 2024a.

Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. SpinQuant: LLM quantization with learned rotations. *arXiv preprint arXiv:2405.16406*, 2024b.

Stuart P. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28 (2):129–137, 1982.

Mikko I. Malinen and Pasi Fränti. Balanced k-means for clustering. In *Structural, Syntactic, and Statistical Pattern Recognition*, volume 8621 of *Lecture Notes in Computer Science*, pp. 32–41. Springer, 2014. doi: 10.1007/978-3-662-44415-3_4.

Vladimir Malinovskii, Denis Mazur, Ivan Ilin, Denis Kuznedelev, Konstantin Burlachenko, Kai Yi, Dan Alistarh, and Peter Richtarik. PV-Tuning: Beyond straight-through estimation for extreme LLM compression. *arXiv preprint arXiv:2405.14852*, 2024.

Gonzalo Mena, David Belanger, Scott Linderman, and Jasper Snoek. Learning latent permutations with Gumbel–Sinkhorn networks. In *International Conference on Learning Representations (ICLR)*, 2018.

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

Gabriel Peyré and Marco Cuturi. Computational optimal transport. *Foundations and Trends in Machine Learning*, 11(5–6):355–607, 2019.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An adversarial Winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

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

Richard Sinkhorn and Paul Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348, 1967.

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.

Mart van Baalen, Andrey Kuzmin, Ivan Koryakovskiy, Markus Nagel, Peter Couperus, Cedric Bastoul, Eric Mahurin, Tijmen Blankevoort, and Paul Whatmough. GPTVQ: The blessing of dimensionality for LLM quantization. *arXiv preprint arXiv:2402.15319*, 2024.

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

An Yang, Anfeng Li, Baosong Yang, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

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

Chuanxia Zheng and Andrea Vedaldi. Online clustered codebook. *arXiv preprint arXiv:2307.15139*, 2023.

APPENDIX

The appendix is organized as follows. Appendix A defines the notation and records the implementation choices, optimality conditions, and stage metrics for BARQ. Appendix B describes the calibration and evaluation protocol and explains the construction of the comparison tables. Appendix C presents additional results on codebook size, assignment behavior, and reconstruction over refinement. Appendix D gives a second-order interpretation of the approximation term in the loss analysis. Appendix E provides auxiliary facts and full proofs of the theoretical results. Appendix F discusses limitations. Appendix G addresses questions about balanced fitting, hard deployment, and the interpretation of the reported comparisons.

A Implementation Details

Table 3 lists the symbols used throughout, and Algorithm 1 is the implemented refinement pair for one group discussed in Section 3.2.

Table 3: **Notation used throughout the paper.** The left block lists quantities inherited from the second-order vector-quantization setting, and the right block lists the quantities introduced by the fitting-stage refinement.

Symbol	*Meaning*	*Symbol*	*Meaning*
N	number of weight blocks in one group	P	soft assignment mass, $P \geq 0$
d	block dimension (weights per block)	$q_k(P)$	fitting mass of codeword k, $\sum_n P_{k,n}$
x_n	the n-th weight block, $x_n \in \mathbb{R}^d$	a_k, b_n	prescribed row and column marginals of P
K	nominal codebook size, $K = 2^{Bd}$	$\Pi(a, b)$	couplings with marginals a and b
B	index bits per value	ε	dimensionless entropic regularization
c_k	the k-th codeword; $C = \{c_k\}_{k=1}^K$	s	groupwise cost scale, fixed per update pair
H	metric of one refinement pair, $H \succ 0$	τ	regularization in cost units, $\tau = \varepsilon s$
$M_{k,n}(C)$	quadratic cost of representing x_n by c_k	T_{sk}	number of Sinkhorn scaling steps
z_n	hard-assignment index of block n		

A.1 Implementation Choices

Layer-Wise Grouping and Block Construction. For each linear layer, the weight matrix is partitioned into independent groups. Within a group, the weights are reshaped into N blocks $x_n \in \mathbb{R}^d$. The block dimension is $d \in \{1, 2, 4\}$ in the reported grid. For each block dimension, the nominal codebook has $K = 2^{Bd}$ entries, with $B = 2$ index bits per scalar weight in the experiments. The tables report recorded bpv, excluding the separate codebook term. Appendix A.1 gives the accounting term by term.

Bit-Budget Accounting. The total budget uses the bits-per-value metric of GPT-VQ (van Baalen et al., 2024), which charges the indices and the codebook that decodes them,

$$\mathrm{bpv}_{\text{GPT-VQ}} = \frac{\log_2 K}{d} + \frac{K\, d\, b_c}{l}, \tag{12}$$

with K centroids, block dimension d, codebook bit-width b_c, and l the number of weights sharing one codebook. We instantiate it as follows.

Index term. A block of d weights stores one index into a codebook of $K = 2^{Bd}$ entries, so the index term is $\log_2(K)/d = B$ bits per value, independent of d. With $B = 2$ every configuration in the grid pays exactly 2 bits per value for indices: $K = 4$ at $d = 1$, $K = 16$ at $d = 2$, and $K = 256$ at $d = 4$ all give $\log_2(K)/d = 2$.

Group-scale term. Both BARQ and the GPT-VQ baseline additionally store one half-precision scale for each group of consecutive weights, which costs $16/g$ bits per value for a group size g: 0.5 at $g = 32$, 0.25 at $g = 64$, and 0.125 at $g = 128$. The three budgets reported in the tables,

$$2.500 = 2 + 0.5, \qquad 2.250 = 2 + 0.25, \qquad 2.125 = 2 + 0.125, \tag{13}$$

are the sum of the index term and this scale term.

Codebook term. The remaining term of Equation (12) is $K\, d\, b_c/l$. It is the same for the two paired methods, because a refinement step rewrites the entries of a codebook of unchanged size K that

is shared by the same l weights; it therefore cancels in every paired comparison of Table 1. Its absolute size depends on how widely one codebook is shared: at $d = 4$, $K = 256$ and $b_c = 16$ it is below 0.01 bits per value once a codebook covers more than 1.6×10^6 weights, and it grows to 0.125 bits per value when a codebook covers 1.3×10^5 weights.

Refinement-Cost Construction. The implementation obtains a positive factor vector d_n from the diagonal of its inverse-curvature Cholesky factor for block n. For the Sinkhorn refinement, it averages these factors over the block index and defines

$$\bar{d}_j = \frac{1}{N} \sum_{n=1}^{N} d_{n,j}, \qquad w_j = \bar{d}_j^{-1} > 0. \tag{14}$$

Thus the E-step cost can be written

$$M_{k,n}(C) = \sum_{j=1}^{d} w_j \big(c_{k,j} - x_{n,j}\big)^2, \qquad w_j > 0. \tag{15}$$

Equivalently, Equation (15) uses $H = \mathrm{diag}(w) \succ 0$, independent of n, exactly as assumed in Section 3.1. Hard initialization, hard lookup, and the optional lookup-table refinement each use their own curvature-derived weights. The exact barycentric and profiled-objective theorems apply to Equation (15). Before the Sinkhorn update, the implementation sets $s = \max\{\mathrm{median}(M), 10^{-8}\}$ and uses M/s. This scaling preserves the feasible set and sets the entropy coefficient in the original cost units to $\tau = \varepsilon s$. Theorem 2 then gives the descent of one exact transport–barycenter pair with this fixed s.

Log-Domain Sinkhorn E-step. For a fixed codebook, the transport step solves the entropy-regularized transport problem in its standard computational form (Cuturi, 2013),

$$\min_{P \in \Pi(a,b)} \ \frac{1}{s} R(C, P) - \varepsilon \mathcal{H}(P), \qquad \mathcal{H}(P) = -\sum_{k,n} P_{k,n}(\log P_{k,n} - 1), \tag{16}$$

with uniform codeword and block marginals. On $\Pi(a, b)$, $\mathcal{H}(P) = 1 + \mathcal{E}(P)$ and $\mathrm{KL}(P \| ab^\top) = \mathcal{E}(a) + \mathcal{E}(b) - \mathcal{E}(P)$, where $\mathcal{E}$ denotes entropy. Consequently,

$$\frac{1}{s}\Phi_\tau(C, P) = \Big[\frac{1}{s} R(C, P) - \varepsilon \mathcal{H}(P)\Big] + \varepsilon\big(1 + \mathcal{E}(a) + \mathcal{E}(b)\big), \tag{17}$$

so the computational and KL objectives share the same minimizer; their objective differences differ only by the scale factor s. We compute the plan in the log domain. Let $G = -M/(s\varepsilon)$, and let (α, β) denote the log scaling variables. The alternating updates are

$$\alpha_k \leftarrow \log a_k - \log \sum_{n=1}^{N} \exp(G_{k,n} + \beta_n), \tag{18}$$

$$\beta_n \leftarrow \log b_n - \log \sum_{k=1}^{K} \exp(G_{k,n} + \alpha_k). \tag{19}$$

The coupling used by the M-step is then

$$P_{k,n} = \exp(\alpha_k + G_{k,n} + \beta_n). \tag{20}$$

The reported primary setting uses $\varepsilon = 10^{-3}$, and the secondary reported setting uses $\varepsilon = 5 \times 10^{-3}$. In the exact feasible problem, every positive row marginal prevents zero fitting mass at any positive temperature. At exact convergence, the row marginal is $1/K$ under the uniform setting. With finitely many alternating scalings, the final v update enforces the block marginal b in exact arithmetic, and the codeword marginal carries a measurable residual.

M-step and Numerical Denominator. Given P, each codeword is updated by the barycentric projection

$$c_k \leftarrow \frac{\sum_{n=1}^{N} P_{k,n} x_n}{\sum_{n=1}^{N} P_{k,n}}. \tag{21}$$

For the exact uniform-marginal plan, the denominator is $1/K$ for every row. The finite-precision implementation instead divides by $\max\{\sum_n P_{k,n}, 10^{-12}\}$, so the update remains numerically defined even before exact marginal convergence. When every actual row sum is at least 10^{-12}, this is the exact fixed-plan barycenter and the variance identity in Theorem 2 applies. For a row below the clamp, Algorithm 1 uses the displayed clamped update. GPT-VQ and BARQ start from the same per-group hard k-means initialization and then apply their own refinement.

Chunking and Memory. Groups are processed independently within each layer. When a layer has many groups, the implementation processes groups in chunks so that the temporary cost and coupling tensors fit within the available memory. Chunking preserves Equations (15)–(21) because each group has its own transport problem.

A.2 Classical Optimality Conditions and Stage Metrics

These classical conditions connect the BARQ update to standard codebook generation.

A codebook quantizer is determined by three components: an encoder that maps each weight block to an index, a decoder that maps each index back to a codeword, and a distortion criterion that scores the reconstruction (Ge et al., 2014; Jégou et al., 2011). BARQ modifies the assignment step within each codebook-refinement update and preserves the hard encoder. The analysis assumes a fixed groupwise quadratic metric. Section A explains how the Hessian-derived weights used in the current implementation differ across the stages.

Quantization distortion is the common objective minimized by vector quantization methods, whose codebook constraints define the resulting quantizer family (Ge et al., 2014). Second-order weight quantization methods (Frantar et al., 2023; van Baalen et al., 2024) further define a local quadratic metric using Hessian information estimated from calibration data. In the implementation, H is a positive diagonal matrix constructed by averaging the inverse-Hessian-derived refinement weights across blocks. The earlier hard fitting and the later lookup-table refinement use their own stage-specific metrics.

A distortion-minimizing quantizer satisfies two classical necessary conditions. For a fixed codebook, a distortion-minimizing encoder must map each block to its nearest codeword, which is the first Lloyd condition (Lloyd, 1982); the resulting quantizer is a Voronoi quantizer (Gray, 1984; Lloyd, 1982). For a fixed encoder and an unconstrained codebook, the distortion-minimizing codeword is the mean of the blocks assigned to it, which is the second Lloyd condition (Ge et al., 2014; Lloyd, 1982). Both statements are classically given for the unweighted Euclidean distortion under hard assignment. Under Equation (1), nearest is measured by the H-weighted cost. Because the same positive-definite H applies to every block in the group, it can be removed from the first-order condition for the codeword update. The optimal codeword therefore remains the arithmetic mean of its assigned blocks. Classical codebook generation alternates between nearest-codeword assignment and mean-based codeword updates until reaching a Lloyd fixed point. Learned-codebook quantizers (van Baalen et al., 2024) commonly follow the same procedure.

For $q_k(P) > 0$, the fixed-plan objective for codeword k has Hessian $2q_k(P)H \succ 0$, so the second Lloyd condition produces the unique assignment-weighted barycenter. Prescribed positive row marginals apply this update to every stored codeword.

B Evaluation and Table Construction Details

Calibration and Metrics. All paired comparisons use the same calibration protocol: 128 WikiText-2 sequences with sequence length 2048. The reported language-model metric is WikiText-2 perplexity. The zero-shot accuracy suite contains ARC-Easy, ARC-Challenge, HellaSwag, PIQA, and WinoGrande, and the Avg. column is the arithmetic mean of those five task accuracies.

Paired Comparison Tables. The body Table 1 reports a compact view (PPL and five-task mean for both methods) so that every configuration is visible on one row. The complete per-task breakdown is given here in Tables 4 (1B/8B) and 5 (Qwen3-14B/32B), and the full external-baseline numbers in Table 6. Every table is generated directly from the recorded results, with no interpolation. Each GPT-VQ/Ours pair shares the model, calibration set, group size, block dimension, nominal codebook size, and the reported bpv metadata. BARQ writes continuous barycentric cen-

Table 4: **Per-task breakdown localizes the mean BARQ gains and the individual-task exceptions on the four 1B/8B models.** Compact view: Table 1. Layout follows Table 1: for each metric the GPT-VQ baseline and BARQ (Ours) are shown side by side and the better value is bold (superscript: Ours − GPT-VQ). PPL is WikiText-2 perplexity; per-task columns are zero-shot accuracy (%); Avg. is their mean.

Setting			PPL ↓		Zero-shot accuracy (%) ↑											
					ARC-e		ARC-c		HellaSwag		PIQA		WinoGrande		Avg.	
d	gs	bpv	GPT-VQ	Ours	GPT-VQ	Ours	GPT-VQ	Ours	GPT-VQ	Ours	GPT-VQ	Ours	GPT-VQ	Ours	GPT-VQ	Ours
Llama3.2-1B																
1	32	2.500	28.49	**27.70**$^{-0.79}$	**46.30**	45.12$^{-1.18}$	26.96	**27.39**$^{+0.43}$	40.94	**42.27**$^{+1.33}$	62.46	**64.15**$^{+1.69}$	**54.54**	53.91$^{-0.63}$	46.24	**46.57**$^{+0.33}$
	64	2.250	**44.24**	46.36$^{+2.12}$	39.48	**42.26**$^{+2.78}$	**25.43**	25.34$^{-0.09}$	35.95	**37.31**$^{+1.36}$	**58.81**	58.71$^{-0.10}$	52.72	**53.83**$^{+1.11}$	42.48	**43.49**$^{+1.01}$
	128	2.125	95.41	**92.04**$^{-3.37}$	34.64	**35.77**$^{+1.13}$	**23.38**	22.53$^{-0.85}$	32.39	**32.52**$^{+0.13}$	**56.26**	55.93$^{-0.33}$	**51.62**	51.46$^{-0.16}$	**39.66**	39.64$^{-0.02}$
2	32	2.500	14.67	**13.25**$^{-1.42}$	60.73	**63.72**$^{+2.99}$	37.12	**37.80**$^{+0.68}$	58.27	**61.28**$^{+3.01}$	72.20	**74.32**$^{+2.12}$	**60.93**	60.85$^{-0.08}$	57.85	**59.59**$^{+1.74}$
	64	2.250	20.04	**16.57**$^{-3.47}$	49.92	**56.61**$^{+6.69}$	31.66	**32.94**$^{+1.28}$	48.37	**53.64**$^{+5.27}$	66.65	**69.86**$^{+3.21}$	**58.80**	58.56$^{-0.24}$	51.08	**54.32**$^{+3.24}$
	128	2.125	26.95	**22.00**$^{-4.95}$	48.70	**50.67**$^{+1.97}$	**28.50**	27.99$^{-0.51}$	42.59	**46.29**$^{+3.70}$	64.80	**65.23**$^{+0.43}$	56.35	**57.70**$^{+1.35}$	48.19	**49.57**$^{+1.38}$
4	32	2.500	14.95	**13.19**$^{-1.76}$	61.28	**63.97**$^{+2.69}$	36.26	**37.63**$^{+1.37}$	58.36	**61.45**$^{+3.09}$	72.80	**74.70**$^{+1.90}$	62.04	**62.27**$^{+0.23}$	58.15	**60.01**$^{+1.86}$
	64	2.250	15.40	**13.23**$^{-2.17}$	60.40	**63.51**$^{+3.11}$	35.49	**36.95**$^{+1.46}$	58.43	**61.56**$^{+3.13}$	72.85	**74.59**$^{+1.74}$	**61.17**	60.93$^{-0.24}$	57.67	**59.51**$^{+1.84}$
	128	2.125	16.10	**13.21**$^{-2.89}$	59.76	**63.93**$^{+4.17}$	35.32	**37.29**$^{+1.97}$	56.53	**61.54**$^{+5.01}$	71.33	**73.99**$^{+2.66}$	58.96	**61.48**$^{+2.52}$	56.38	**59.65**$^{+3.27}$
	1024	2.500	19.09	**13.84**$^{-5.25}$	54.29	**59.51**$^{+5.22}$	31.91	**36.69**$^{+4.78}$	50.51	**59.33**$^{+8.82}$	68.50	**72.85**$^{+4.35}$	55.96	**61.48**$^{+5.52}$	52.23	**57.97**$^{+5.74}$
Llama3.1-8B																
1	32	2.500	12.57	**12.17**$^{-0.40}$	**66.88**	64.52$^{-2.36}$	**41.64**	37.29$^{-4.35}$	**59.91**	59.57$^{-0.34}$	**72.25**	72.14$^{-0.11}$	**65.43**	64.48$^{-0.95}$	**61.22**	59.60$^{-1.62}$
	64	2.250	14.73	**14.06**$^{-0.67}$	58.16	**59.22**$^{+1.06}$	33.02	**34.30**$^{+1.28}$	51.33	**51.95**$^{+0.62}$	67.74	**68.12**$^{+0.38}$	**62.98**	61.48$^{-1.50}$	54.65	**55.01**$^{+0.36}$
	128	2.125	15.91	**15.59**$^{-0.32}$	47.81	**54.29**$^{+6.48}$	28.07	**29.78**$^{+1.71}$	42.68	**46.16**$^{+3.48}$	62.84	**65.18**$^{+2.34}$	57.30	**59.75**$^{+2.45}$	47.74	**51.03**$^{+3.29}$
2	32	2.500	8.89	**8.32**$^{-0.57}$	77.82	**78.37**$^{+0.55}$	54.52	**54.95**$^{+0.43}$	74.76	**75.31**$^{+0.55}$	78.02	**78.07**$^{+0.05}$	71.98	71.98$^{\pm 0.00}$	71.42	**71.74**$^{+0.32}$
	64	2.250	9.63	**9.13**$^{-0.50}$	71.04	**72.22**$^{+1.18}$	45.73	**49.66**$^{+3.93}$	68.89	**70.61**$^{+1.72}$	75.46	**76.93**$^{+1.47}$	69.85	**70.64**$^{+0.79}$	66.20	**68.01**$^{+1.81}$
	128	2.125	10.92	**10.42**$^{-0.50}$	64.06	**72.52**$^{+8.46}$	37.12	**45.31**$^{+8.19}$	62.72	**65.79**$^{+3.07}$	71.98	**75.35**$^{+3.37}$	67.09	**67.96**$^{+0.87}$	60.59	**65.39**$^{+4.80}$
4	32	2.500	6.62	**6.15**$^{-0.47}$	**77.48**	76.85$^{-0.63}$	51.45	**53.50**$^{+2.05}$	78.24	**79.15**$^{+0.91}$	79.87	**80.90**$^{+1.03}$	**73.48**	73.32$^{-0.16}$	72.11	**72.74**$^{+0.63}$
	64	2.250	6.83	**6.16**$^{-0.67}$	75.72	**78.41**$^{+2.69}$	50.60	**53.41**$^{+2.81}$	78.38	**79.38**$^{+1.00}$	79.27	**80.30**$^{+1.03}$	71.43	**73.24**$^{+1.81}$	71.08	**72.95**$^{+1.87}$
	128	2.125	11.00	**8.30**$^{-2.70}$	74.87	**78.70**$^{+3.83}$	50.17	**56.14**$^{+5.97}$	74.05	**75.58**$^{+1.53}$	76.22	**78.73**$^{+2.51}$	**72.06**	71.19$^{-0.87}$	69.47	**72.07**$^{+2.60}$
	1024	2.500	10.69	**8.38**$^{-2.31}$	66.46	**75.80**$^{+9.34}$	43.17	**53.41**$^{+10.24}$	71.82	**75.70**$^{+3.88}$	73.78	**77.15**$^{+3.37}$	69.06	**70.56**$^{+1.50}$	64.86	**70.52**$^{+5.66}$
Qwen3-1.7B																
1	32	2.500	**31.97**	33.41$^{+1.44}$	**41.54**	40.32$^{-1.22}$	26.79	**28.16**$^{+1.37}$	38.26	**39.79**$^{+1.53}$	**60.83**	59.90$^{-0.93}$	52.72	**54.54**$^{+1.82}$	44.03	**44.54**$^{+0.51}$
	64	2.250	52.57	**50.23**$^{-2.34}$	34.13	**35.90**$^{+1.77}$	24.32	**24.40**$^{+0.08}$	34.55	**35.58**$^{+1.03}$	57.56	**58.11**$^{+0.55}$	51.22	**52.49**$^{+1.27}$	40.36	**41.30**$^{+0.94}$
	128	2.125	68.21	**63.03**$^{-5.18}$	**32.95**	32.87$^{-0.08}$	**25.00**	24.15$^{-0.85}$	32.11	**34.08**$^{+1.97}$	51.96	**56.04**$^{+4.08}$	48.38	**51.22**$^{+2.84}$	38.08	**39.67**$^{+1.59}$
2	32	2.500	17.62	**16.96**$^{-0.66}$	67.55	**69.44**$^{+1.89}$	42.49	**43.26**$^{+0.77}$	58.44	**60.45**$^{+2.01}$	69.64	**72.03**$^{+2.39}$	59.19	**59.67**$^{+0.48}$	59.46	**60.97**$^{+1.51}$
	64	2.250	27.26	**23.02**$^{-4.24}$	42.55	**52.57**$^{+10.02}$	28.16	**36.09**$^{+7.93}$	46.65	**51.27**$^{+4.62}$	63.33	**68.34**$^{+5.01}$	55.01	**55.72**$^{+0.71}$	47.14	**52.80**$^{+5.66}$
	128	2.125	40.49	**26.49**$^{-14.00}$	38.13	**42.55**$^{+4.42}$	25.77	**29.44**$^{+3.67}$	39.60	**43.43**$^{+3.83}$	57.89	**64.64**$^{+6.75}$	50.91	**53.35**$^{+2.44}$	42.46	**46.68**$^{+4.22}$
4	32	2.500	20.37	**16.69**$^{-3.68}$	49.49	**70.08**$^{+20.59}$	36.52	**43.60**$^{+7.08}$	58.78	**60.50**$^{+1.72}$	69.21	**72.09**$^{+2.88}$	59.04	**61.17**$^{+2.13}$	54.61	**61.49**$^{+6.88}$
	64	2.250	19.24	**17.21**$^{-2.03}$	58.88	**68.48**$^{+9.60}$	39.93	**43.09**$^{+3.16}$	57.69	**60.21**$^{+2.52}$	70.29	**71.71**$^{+1.42}$	59.19	**60.93**$^{+1.74}$	57.20	**60.88**$^{+3.68}$
	128	2.125	20.54	**16.59**$^{-3.95}$	60.40	**64.31**$^{+3.91}$	37.37	**41.64**$^{+4.27}$	56.52	**60.22**$^{+3.70}$	67.30	**71.38**$^{+4.08}$	57.14	**60.85**$^{+3.71}$	55.75	**59.68**$^{+3.93}$
	1024	2.500	24.57	**17.48**$^{-7.09}$	48.27	**65.78**$^{+17.51}$	31.57	**42.06**$^{+10.49}$	51.02	**59.45**$^{+8.43}$	67.25	**71.16**$^{+3.91}$	56.51	**60.62**$^{+4.11}$	50.92	**59.82**$^{+8.90}$
Qwen3-8B																
1	32	2.500	**12.34**	12.54$^{+0.20}$	72.94	**73.27**$^{+0.33}$	45.56	**46.16**$^{+0.60}$	**60.96**	60.77$^{-0.19}$	**72.91**	71.82$^{-1.09}$	64.64	**65.67**$^{+1.03}$	63.40	**63.54**$^{+0.14}$
	64	2.250	13.80	**13.64**$^{-0.16}$	**73.11**	70.33$^{-2.78}$	**44.71**	42.58$^{-2.13}$	57.37	**58.48**$^{+1.11}$	**70.78**	70.67$^{-0.11}$	**63.38**	61.64$^{-1.74}$	**61.87**	60.74$^{-1.13}$
	128	2.125	**14.70**	14.89$^{+0.19}$	62.88	**66.12**$^{+3.24}$	36.86	**37.54**$^{+0.68}$	**55.32**	55.14$^{-0.18}$	68.82	**69.80**$^{+0.98}$	62.04	**64.25**$^{+2.21}$	57.18	**58.57**$^{+1.39}$
2	32	2.500	9.93	**9.70**$^{-0.23}$	77.82	**80.68**$^{+2.86}$	55.03	**56.91**$^{+1.88}$	74.37	**75.06**$^{+0.69}$	76.99	**77.75**$^{+0.76}$	68.11	68.11$^{\pm 0.00}$	70.46	**71.70**$^{+1.24}$
	64	2.250	10.96	**10.73**$^{-0.23}$	74.33	**79.21**$^{+4.88}$	47.53	**53.92**$^{+6.39}$	68.43	**70.29**$^{+1.86}$	74.65	**75.68**$^{+1.03}$	66.85	**68.82**$^{+1.97}$	66.36	**69.59**$^{+3.23}$
	128	2.125	12.62	**11.35**$^{-1.27}$	71.17	**75.80**$^{+4.63}$	44.71	**48.63**$^{+3.92}$	63.01	**66.38**$^{+3.37}$	71.76	**74.59**$^{+2.83}$	64.96	**65.82**$^{+0.86}$	63.12	**66.25**$^{+3.13}$
4	32	2.500	10.16	**9.75**$^{-0.41}$	78.58	**81.06**$^{+2.48}$	54.18	**57.34**$^{+3.16}$	74.29	**74.94**$^{+0.65}$	76.77	**77.48**$^{+0.71}$	67.72	67.72$^{\pm 0.00}$	70.31	**71.71**$^{+1.40}$
	64	2.250	10.13	**9.75**$^{-0.38}$	79.04	**80.64**$^{+1.60}$	53.84	**56.91**$^{+3.07}$	74.02	**74.86**$^{+0.84}$	76.66	**78.02**$^{+1.36}$	**68.75**	68.11$^{-0.64}$	70.46	**71.71**$^{+1.25}$
	128	2.125	10.28	**9.72**$^{-0.56}$	80.18	**80.39**$^{+0.21}$	55.55	**56.31**$^{+0.76}$	73.84	**74.99**$^{+1.15}$	76.17	**77.75**$^{+1.58}$	67.64	**67.88**$^{+0.24}$	70.68	**71.46**$^{+0.78}$
	1024	2.500	10.26	**9.73**$^{-0.53}$	73.06	**80.13**$^{+7.07}$	50.68	**56.57**$^{+5.89}$	71.72	**74.97**$^{+3.25}$	76.66	**77.04**$^{+0.38}$	65.90	**66.93**$^{+1.03}$	67.61	**71.13**$^{+3.52}$

ters into the codebook, whereas the baseline may additionally refine the codebook with a separate lookup-table step after GPTQ. Table 1 therefore reports the paired end-to-end outputs under shared calibration, nominal codebook size, and recorded bits per value.

External-Baseline Table. Table 2 places BARQ next to reported external baselines at the displayed low-bit-width operating points. For the paired GPT-VQ and BARQ rows it takes, for each model, the complete paired setting with the lowest reported budget, preferring $d = 4$ when several settings report the same value.

Missing Cells. In Tables 2 and 6, a dash in PPL denotes quantization collapse. Other dashes mark unavailable measurements. Every table average is computed from the displayed task values.

C Additional Results

This appendix collects three diagnostic figures (Figures 5–7) and their analysis. Figure 6 combines assignment entropy and reconstruction distortion in panels (a) and (b).

C.1 Discussion and Analysis

The exact theory and the mechanism probes address different quantities. The theory controls soft fitting mass and a regularized refinement objective, whereas the probes measure the hard encoder

Table 5: **At Qwen3-14B/32B, the strongest gains occur for $K = 16$ and $K = 256$, while the per-task breakdown retains the $K = 4$ regressions.** Compact view: the Qwen3-14B/32B panels of Table 1. Columns, bolding and superscripts match Table 4.

Setting			PPL ↓		Zero-shot accuracy (%) ↑											
					ARC-e		ARC-c		HellaSwag		PIQA		WinoGrande		Avg.	
d	gs	bpv	GPT-VQ	Ours	GPT-VQ	Ours	GPT-VQ	Ours	GPT-VQ	Ours	GPT-VQ	Ours	GPT-VQ	Ours	GPT-VQ	Ours
Qwen3-14B																
1	32	2.500	10.80	**10.73**$^{-0.07}$	**77.40**	73.65$^{-3.75}$	**54.61**	48.04$^{-6.57}$	68.91	**69.51**$^{+0.60}$	75.14	**75.30**$^{+0.16}$	**71.19**	70.48$^{-0.71}$	**69.45**	67.40$^{-2.05}$
	64	2.250	11.29	**11.14**$^{-0.15}$	**75.51**	70.79$^{-4.72}$	**48.29**	47.87$^{-0.42}$	65.90	**66.74**$^{+0.84}$	**75.14**	73.99$^{-1.15}$	69.06	**70.09**$^{+1.03}$	**66.78**	65.90$^{-0.88}$
	128	2.125	11.80	**11.79**$^{-0.01}$	**70.20**	65.99$^{-4.21}$	**46.33**	45.56$^{-0.77}$	63.98	**64.26**$^{+0.28}$	**73.56**	72.14$^{-1.42}$	**68.67**	68.19$^{-0.48}$	**64.55**	63.23$^{-1.32}$
2	32	2.500	8.79	**8.65**$^{-0.14}$	82.95	**83.08**$^{+0.13}$	59.98	**61.52**$^{+1.54}$	77.94	**78.99**$^{+1.05}$	79.43	**79.92**$^{+0.49}$	73.16	73.16$^{\pm 0.00}$	74.70	**75.34**$^{+0.64}$
	64	2.250	9.61	**9.33**$^{-0.28}$	79.71	**81.65**$^{+1.94}$	**57.08**	56.91$^{-0.17}$	74.15	**75.31**$^{+1.16}$	77.48	**78.56**$^{+1.08}$	**73.64**	71.98$^{-1.66}$	72.41	**72.88**$^{+0.47}$
	128	2.125	10.31	**9.92**$^{-0.39}$	**77.02**	75.80$^{-1.22}$	**53.16**	52.65$^{-0.51}$	70.04	**71.38**$^{+1.34}$	76.50	**77.04**$^{+0.54}$	70.40	**71.59**$^{+1.19}$	69.42	**69.69**$^{+0.27}$
4	32	2.500	**8.78**	8.95$^{+0.17}$	**83.08**	82.79$^{-0.29}$	**60.07**	59.98$^{-0.09}$	78.14	**78.70**$^{+0.56}$	79.38	**79.98**$^{+0.60}$	**72.93**	72.22$^{-0.71}$	74.72	**74.73**$^{+0.01}$
	64	2.250	8.81	**8.67**$^{-0.14}$	**82.87**	82.66$^{-0.21}$	**60.41**	60.32$^{-0.09}$	78.20	**78.88**$^{+0.68}$	79.65	**79.87**$^{+0.22}$	71.74	**72.85**$^{+1.11}$	74.57	**74.92**$^{+0.35}$
	128	2.125	8.93	**8.66**$^{-0.27}$	**83.25**	83.12$^{-0.13}$	59.73	**61.09**$^{+1.36}$	78.26	**78.72**$^{+0.46}$	80.03	**80.25**$^{+0.22}$	71.67	**72.53**$^{+0.86}$	74.59	**75.14**$^{+0.55}$
	1024	2.500	9.11	**8.73**$^{-0.38}$	81.19	**82.15**$^{+0.96}$	59.47	**60.49**$^{+1.02}$	77.10	**78.37**$^{+1.27}$	79.22	**80.20**$^{+0.98}$	69.85	**72.69**$^{+2.84}$	73.36	**74.78**$^{+1.42}$
Qwen3-32B																
1	32	2.500	9.10	**8.95**$^{-0.15}$	**77.86**	77.31$^{-0.55}$	**54.27**	54.18$^{-0.09}$	75.69	**76.11**$^{+0.42}$	77.09	**79.05**$^{+1.96}$	70.96	**73.72**$^{+2.76}$	71.17	**72.08**$^{+0.91}$
	64	2.250	9.54	**9.40**$^{-0.14}$	**78.62**	75.88$^{-2.74}$	**53.58**	52.90$^{-0.68}$	73.44	**74.59**$^{+1.15}$	**77.69**	77.09$^{-0.60}$	70.48	**71.59**$^{+1.11}$	**70.76**	70.41$^{-0.35}$
	128	2.125	**9.86**	9.88$^{+0.02}$	75.80	**78.54**$^{+2.74}$	54.01	**54.18**$^{+0.17}$	**73.38**	73.36$^{-0.02}$	76.06	**76.82**$^{+0.76}$	70.64	**71.43**$^{+0.79}$	69.98	**70.87**$^{+0.89}$
2	32	2.500	7.76	**7.62**$^{-0.14}$	83.38	**83.63**$^{+0.25}$	60.58	**61.52**$^{+0.94}$	82.02	**82.66**$^{+0.64}$	81.61	**81.77**$^{+0.16}$	71.98	**72.38**$^{+0.40}$	75.91	**76.39**$^{+0.48}$
	64	2.250	8.37	**8.09**$^{-0.28}$	79.59	**80.30**$^{+0.71}$	58.70	**62.03**$^{+3.33}$	79.60	**80.87**$^{+1.27}$	**79.92**	79.54$^{-0.38}$	71.11	71.11$^{\pm 0.00}$	73.78	**74.77**$^{+0.99}$
	128	2.125	8.91	**8.55**$^{-0.36}$	79.63	**80.93**$^{+1.30}$	55.97	**58.87**$^{+2.90}$	76.94	**78.27**$^{+1.33}$	78.13	**78.94**$^{+0.81}$	71.51	**72.77**$^{+1.26}$	72.44	**73.96**$^{+1.52}$
4	32	2.500	7.73	**7.61**$^{-0.12}$	**84.09**	83.21$^{-0.88}$	**61.09**	60.92$^{-0.17}$	82.20	**82.64**$^{+0.44}$	81.18	**81.77**$^{+0.59}$	73.01	**73.24**$^{+0.23}$	76.31	**76.36**$^{+0.05}$
	64	2.250	7.85	**7.61**$^{-0.24}$	**83.71**	83.21$^{-0.50}$	60.32	**61.18**$^{+0.86}$	81.92	**82.63**$^{+0.71}$	80.63	**81.94**$^{+1.31}$	72.45	**73.40**$^{+0.95}$	75.81	**76.47**$^{+0.66}$
	128	2.125	7.84	**7.61**$^{-0.23}$	82.45	**83.21**$^{+0.76}$	60.84	**61.35**$^{+0.51}$	81.70	**82.75**$^{+1.05}$	80.79	**81.72**$^{+0.93}$	71.59	**73.01**$^{+1.42}$	75.47	**76.41**$^{+0.94}$
	1024	2.500	8.15	**7.69**$^{-0.46}$	80.22	**83.80**$^{+3.58}$	**61.43**	60.92$^{-0.51}$	80.58	**82.49**$^{+1.91}$	79.38	**81.39**$^{+2.01}$	71.19	**73.24**$^{+2.05}$	74.56	**76.37**$^{+1.81}$

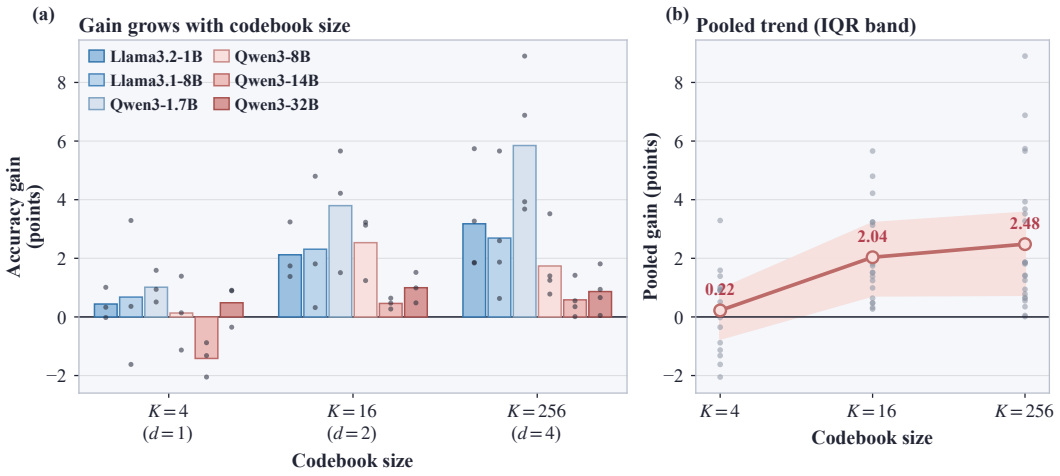

Figure 5: **The observed accuracy gain is larger in the larger-codebook strata.** **(a)** Mean gain per model at each codebook size, with the individual configurations overlaid as points. **(b)** Pooled trend across all models, with the interquartile band. The pooled mean grows with codebook size, while the points retain the regressions at $K = 4$.

after refinement. Keeping these objects separate lets the observed codebook-size pattern inform the mechanism without turning it into a causal claim.

Accuracy Differences across Codebook Sizes. Figure 5 reports the observed accuracy difference between paired rows at each codebook size, per model in panel **(a)** and as a pooled trend in panel **(b)**. The pooled mean difference is smallest at $K = 4$ and larger at $K = 16$ and $K = 256$, while several $K = 4$ configurations regress. This pattern is consistent with the hypothesis that larger codebooks leave more room for balanced refinement, but it neither identifies K as a causal factor nor establishes improved utilization as the underlying mechanism.

Evenness and Local Fit during Refinement. Figure 6 follows the same fitting trajectories with two further quantities. Normalized hard-assignment entropy has a higher step-15 median for BARQ in all 15 model-by-setting cells, and normalized reconstruction distortion has a lower step-15 median in 12 of those 15 cells. Both quantities describe the fitting stage of a single group and say nothing directly about perplexity.

Table 6: **Full per-task external-baseline comparison.** Compact view: Table 2. PPL is WikiText-2 perplexity; per-task columns are zero-shot accuracy (%); Avg. is their mean. VPTQ uses 3.000 index bpv, excluding codebooks and metadata. A dash in PPL denotes quantization collapse; other dashes denote unavailable results. Bold marks the best; underlining marks the runner-up. The Δ row gives BARQ minus the strongest baseline (green: better; red: worse).

Method	bpv	PPL ↓	Zero-shot accuracy (%) ↑					
			ARC-e	ARC-c	HellaSw.	PIQA	WinoG.	**Avg.**
∞ Llama3.2-1B								
GPTQ	2.500	209.53	27.95	26.19	28.36	51.90	47.51	36.38
HQQ	2.500	4917.51	27.36	26.96	28.80	53.92	49.57	37.32
QuIP	2.000	377.93	–	–	–	–	–	–
VPTQ	3.000	15.91	57.91	34.22	56.45	72.63	59.98	56.24
AQLM (no BFT)	2.067	20.30	56.19	30.29	47.35	67.08	57.54	51.69
AQLM (BFT)	2.067	17.39	56.31	31.06	49.68	68.44	57.62	52.62
GPT-VQ	2.125	16.10	59.76	35.32	56.53	71.33	58.96	56.38
BARQ (Ours)	2.125	**13.21**	**63.93**	**37.29**	**61.54**	**73.99**	**61.48**	**59.65**
Δ *vs. best baseline*		−2.70	+4.17	+1.97	+5.01	+1.36	+1.50	+3.27
∞ Llama3-8B								
GPTQ	2.500	32.59	30.81	22.70	28.62	53.43	48.86	36.88
HQQ	2.500	204.69	35.65	25.94	43.46	57.13	56.83	43.80
QuIP	2.000	38.06	29.38	22.01	33.91	52.88	51.54	37.94
VPTQ	3.000	9.11	76.30	52.56	72.53	77.37	69.38	69.63
AQLM (no BFT)	2.067	11.05	66.33	41.72	63.65	72.74	66.54	62.20
AQLM (BFT)	2.067	10.18	71.34	45.22	66.30	75.08	67.01	64.99
GPT-VQ	2.125	11.00	74.87	50.17	74.05	76.22	**72.06**	69.47
BARQ (Ours)	2.125	**8.30**	**78.70**	**56.14**	**75.58**	**78.73**	71.19	**72.07**
Δ *vs. best baseline*		−0.81	+2.40	+3.58	+1.53	+1.36	−0.87	+2.44
Qwen3-1.7B								
GPTQ	2.500	666.14	25.63	24.06	26.99	50.98	50.43	35.62
HQQ	2.500	1018.43	30.26	23.89	27.80	52.12	51.70	37.15
QuIP	2.000	333.99	–	–	–	–	–	–
VPTQ	3.000	–	25.17	26.71	26.70	51.90	51.14	36.32
AQLM (no BFT)	2.067	21.68	46.89	29.95	46.47	63.44	56.99	48.75
AQLM (BFT)	2.067	18.30	56.14	34.98	49.92	67.74	58.72	53.50
GPT-VQ	2.125	20.54	60.40	37.37	56.52	67.30	57.14	55.75
BARQ (Ours)	2.125	**16.59**	**64.31**	**41.64**	**60.22**	**71.38**	**60.85**	**59.68**
Δ *vs. best baseline*		−1.71	+3.91	+4.27	+3.70	+3.64	+2.13	+3.93
Qwen3-8B								
GPTQ	2.500	17.94	38.05	28.24	48.60	61.97	53.12	46.00
HQQ	2.500	46.57	42.13	26.11	43.34	62.46	50.43	44.89
QuIP	2.000	25.51	34.93	25.00	38.33	59.52	51.07	41.77
VPTQ	3.000	10.65	**80.81**	55.29	71.47	76.71	**69.85**	70.83
AQLM (no BFT)	2.067	10.73	75.93	49.40	63.89	73.50	66.30	65.80
AQLM (BFT)	2.067	10.39	76.05	48.55	65.65	73.99	66.69	66.19
GPT-VQ	2.125	10.28	80.18	55.55	73.84	76.17	67.64	70.68
BARQ (Ours)	2.125	**9.72**	80.39	**56.31**	**74.99**	**77.75**	67.88	**71.46**
Δ *vs. best baseline*		−0.56	−0.42	+0.76	+1.15	+1.04	−1.97	+0.63
Qwen3-14B								
GPTQ	2.500	12.90	54.92	34.90	60.84	70.46	56.35	55.49
HQQ	2.500	18.40	65.57	41.81	61.78	72.47	58.41	60.01
QuIP	2.000	14.83	43.86	29.35	54.01	64.74	53.67	49.13
VPTQ	3.000	9.32	80.98	57.94	76.01	78.45	71.59	72.99
AQLM (no BFT)	2.067	9.42	74.79	50.43	70.52	75.68	71.82	68.65
AQLM (BFT)	2.067	9.09	77.99	53.50	71.42	76.22	71.03	70.03
GPT-VQ	2.125	8.93	**83.25**	59.73	78.26	80.03	71.67	74.59
BARQ (Ours)	2.125	**8.66**	83.12	**61.09**	**78.72**	**80.25**	**72.53**	**75.14**
Δ *vs. best baseline*		−0.27	−0.13	+1.36	+0.46	+0.22	+0.71	+0.55
Qwen3-32B								
GPTQ	2.500	14.26	49.96	35.49	63.45	69.31	56.67	54.98
HQQ	2.500	15.11	62.04	43.43	69.03	71.82	57.85	60.83
QuIP	2.000	14.01	25.08	22.70	56.55	49.51	55.72	41.91
VPTQ	3.000	8.50	78.20	55.63	79.50	77.97	70.40	72.34
AQLM (no BFT)	2.067	8.83	82.07	60.49	77.35	77.80	71.51	73.84
AQLM (BFT)	2.067	8.39	82.41	60.15	77.95	79.49	71.90	74.38
GPT-VQ	2.125	7.84	82.45	60.84	81.70	80.79	71.59	75.47
BARQ (Ours)	2.125	**7.61**	**83.21**	**61.35**	**82.75**	**81.72**	**73.01**	**76.41**
Δ *vs. best baseline*		−0.23	+0.76	+0.51	+1.05	+0.93	+1.11	+0.94

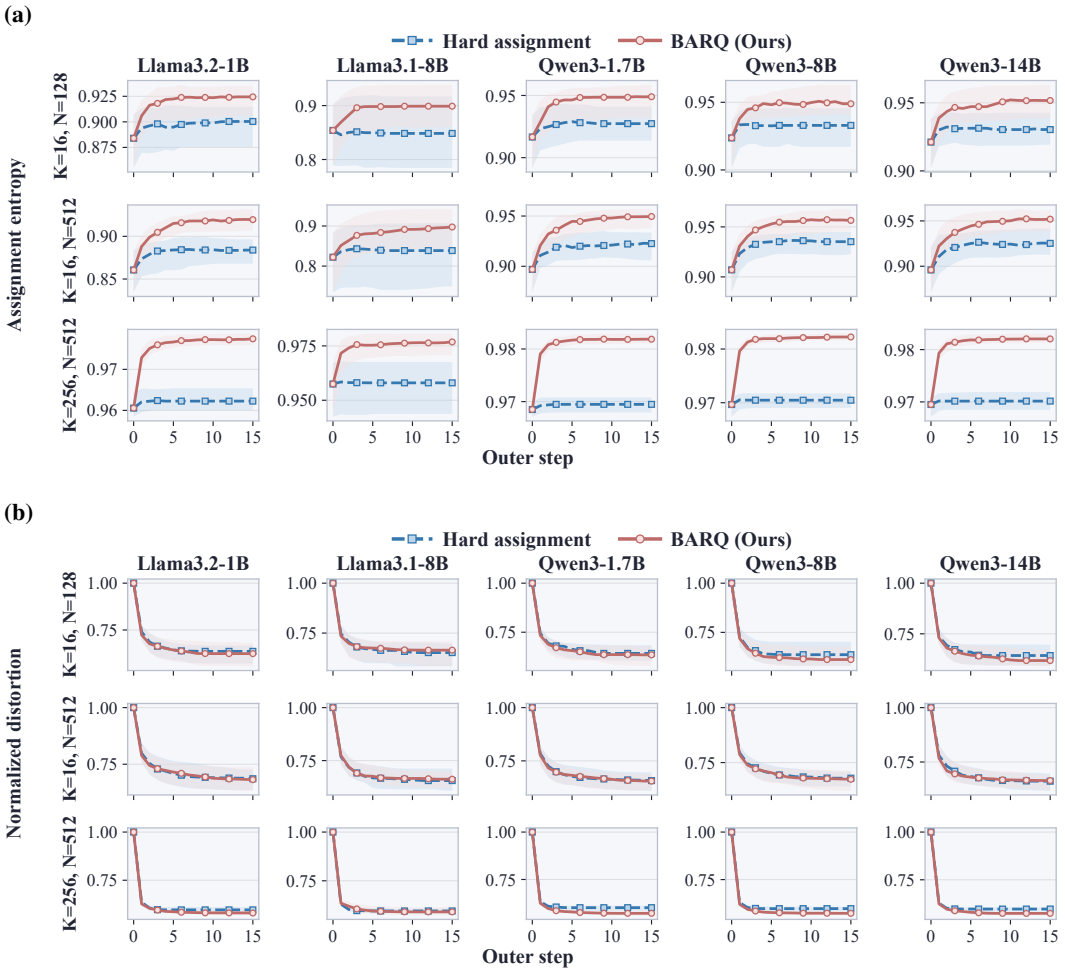

Figure 6: **Assignment evenness and local reconstruction over the same fitting trajectories. (a)** Normalized hard-assignment entropy, where BARQ has the higher step-15 median in all 15 model-by-setting cells. **(b)** Euclidean reconstruction distortion normalized by each group's step-zero value, where BARQ has the lower step-15 median in 12 of the 15 cells. In both panels, columns are models, rows are the three (K, N) settings, the x-axis is the outer refinement step, and the dashed grey curve is the hard-assignment baseline while the solid curve is BARQ.

Realized Codebook Use. The first post-refinement measurement is the number of codewords that receive at least one block, normalized by $\min(K, N)$, where N is the number of blocks in the group. Figure 7 reports this quantity by layer for the same three Llama 3 scales used in Section 5.4. At $K = 4$ both methods nearly saturate the codebook under nearest-codeword assignment, so this stratum separates the two methods very little. At $K = 16$ the paired BARQ output has the higher value at every measured layer, and at $K = 256$ it is higher in 74 of the 75 matched layers. The matched weighted aggregate is higher at all three model scales. The paired outputs therefore reach a different realized encoder distribution at the two larger codebook sizes, although realized use is not by itself the mechanism.

D Second-Order Interpretation of the Approximation Term

The loss guarantee in Theorem 4 depends on the approximation term η_C. The following expansion separates its first-order, metric-mismatch, and higher-order contributions.

Lemma 1 (Local loss expansion)**.** *Suppose $\mathcal{L}$ is three times continuously differentiable on a neighbourhood of the segment between W and $Q_C(W)$, and $|D^3\mathcal{L}(\cdot)[\delta_C, \delta_C, \delta_C]| \le L_3\|\delta_C\|_2^3$ on that segment. Let g and $\mathfrak{H}$ be the gradient and Hessian of $\mathcal{L}$ at W. Then*

$$\mathcal{L}(C) = \mathcal{L}(W) + g^\top \delta_C + \tfrac{1}{2}\delta_C^\top \mathfrak{H}\delta_C + r_C, \qquad |r_C| \le \tfrac{L_3}{6}\|\delta_C\|_2^3, \tag{22}$$

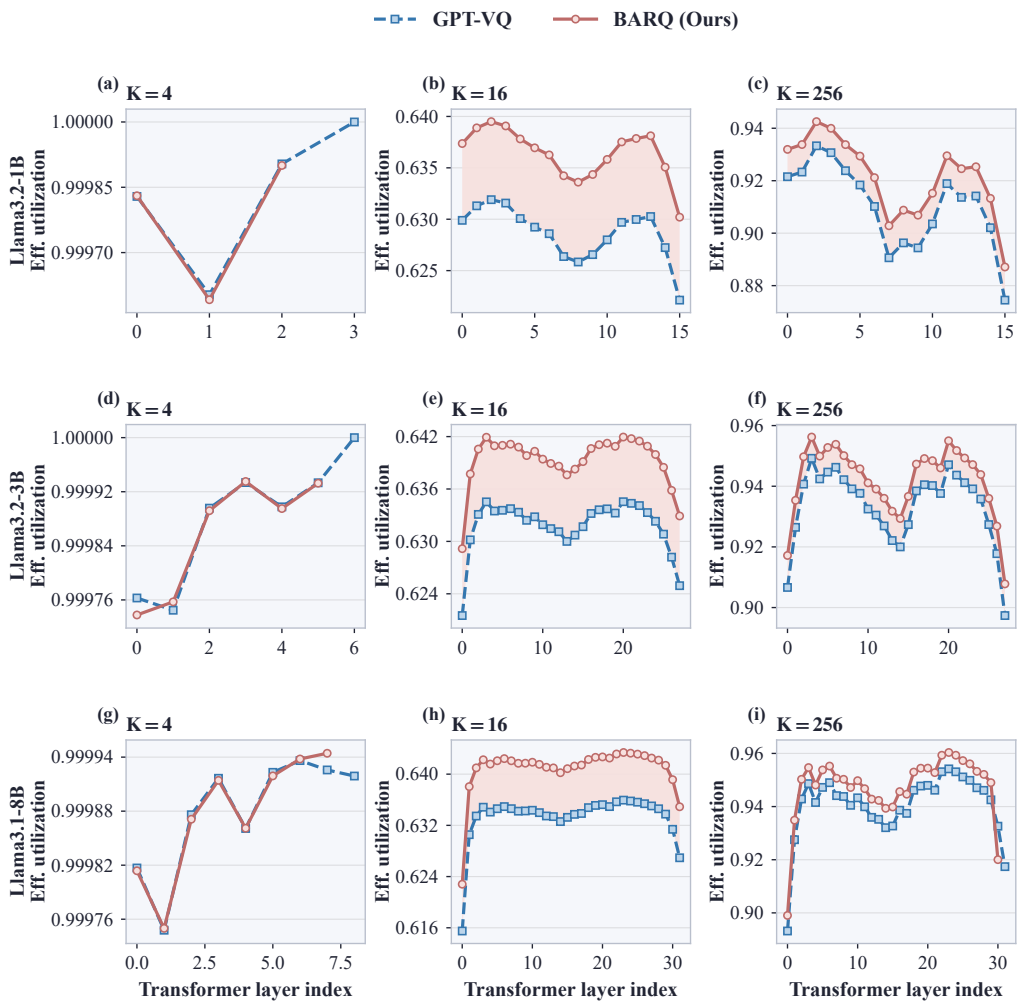

Figure 7: **Hard codeword use is higher at the larger codebook sizes across layers and model scales.** Each panel shows effective utilization, the number of used codewords divided by $\min(K, N)$ with N the number of blocks per group, for three model scales and three codebook sizes. At $K = 4$ both methods nearly saturate the codebook under nearest-codeword assignment, because every group holds many more blocks than codewords. At $K = 16$ BARQ is higher at every measured layer. At $K = 256$ it is higher in 74 of 75 matched layers; one further 8B baseline layer has no BARQ counterpart. The matched weighted aggregate is higher at all three model scales.

and consequently

$$\eta_C \le |g^\top \delta_C| + \tfrac{1}{2}\big|\delta_C^\top(\mathfrak{H} - \widetilde{H})\delta_C\big| + \tfrac{L_3}{6}\|\delta_C\|_2^3. \tag{23}$$

The proof is given in Appendix E.6. Equation (23) makes explicit that small quantization perturbations control the cubic remainder, but do not by themselves remove the gradient or metric-mismatch terms. This second-order interpretation follows the perspective of weight-pruning analysis (LeCun et al., 1990; Hassibi & Stork, 1992) and calibration-based quantization objectives (Frantar et al., 2023; van Baalen et al., 2024).

E Proofs for Theoretical Analysis

We prove Theorems 1–4 in their main-text order, followed by Lemma 1. Throughout the refinement analysis, $H = H^\top \succ 0$, the costs are finite, a and b are strictly positive probability vectors, and H, a, b, and $\tau > 0$ remain fixed within an exact pair. We write $\mathcal{E}(p) = -\sum_i p_i \log p_i$, with $0 \log 0 = 0$.

E.1 Auxiliary Facts

We recall the standard facts used below. The coupling-entropy and hard-decoding identities are recorded explicitly because they determine the assignment-penalty bounds.

Fact 1 (Gibbs' inequality)**.** *For probability vectors p and r with $r_i > 0$, $\mathrm{KL}(p\|r) \geq 0$, with equality if and only if $p = r$.*

Fact 2 (Uniform vector)**.** *For a probability vector p on K entries and $\mathbf{u} = K^{-1}\mathbf{1}_K$,*

$$\mathcal{E}(p) = \log K - \mathrm{KL}(p\|\mathbf{u}) \leq \log K. \tag{24}$$

Also, $\max_k p_k \geq 1/K$. Equality in either bound holds exactly when $p = \mathbf{u}$.

Fact 3 (Entropy of a coupling)**.** *For $P \in \Pi(a, b)$,*

$$\mathcal{E}(P) \geq \max\{\mathcal{E}(a), \mathcal{E}(b)\}, \qquad \mathrm{KL}(P\|ab^\top) = \mathcal{E}(a) + \mathcal{E}(b) - \mathcal{E}(P), \tag{25}$$

and hence

$$0 \leq \mathrm{KL}(P\|ab^\top) \leq \min\{\mathcal{E}(a), \mathcal{E}(b)\}. \tag{26}$$

Proof. Let $p_k^{(n)} = P_{k,n}/b_n$ and $r_n^{(k)} = P_{k,n}/a_k$ be the column and row conditional distributions. The marginal constraints give

$$\begin{aligned}
\mathcal{E}(P) &= \mathcal{E}(b) + \sum_n b_n \mathcal{E}(p^{(n)}) = \mathcal{E}(a) + \sum_k a_k \mathcal{E}(r^{(k)}) \geq \max\{\mathcal{E}(a), \mathcal{E}(b)\}, \\
\mathrm{KL}(P\|ab^\top) &= \sum_{k,n} P_{k,n} \log P_{k,n} - \sum_k a_k \log a_k - \sum_n b_n \log b_n \\
&= \mathcal{E}(a) + \mathcal{E}(b) - \mathcal{E}(P) \leq \min\{\mathcal{E}(a), \mathcal{E}(b)\}.
\end{aligned}$$

The lower bound follows from Gibbs' inequality. □

Fact 4 (Transport polytope and entropic objective)**.** *The set $\Pi(a, b)$ is nonempty, convex, and compact. For finite costs and $\tau > 0$, $P \mapsto \Phi_\tau(C, P)$ is continuous and strictly convex on $\Pi(a, b)$.*

Indeed, $ab^\top$ is feasible, the marginal constraints define a closed and bounded convex set, and $p \mapsto p \log p$ is continuous on $[0, \infty)$ and strictly convex.

Fact 5 (Hard decoding never exceeds a column-feasible assignment)**.** *For $P \geq 0$ with $P^\top \mathbf{1}_K = b$,*

$$D_{\mathrm{hard}}(C) \leq R(C, P). \tag{27}$$

Equality holds exactly when $P_{k,n} > 0$ implies $M_{k,n}(C) = \min_j M_{j,n}(C)$.

Proof. Writing $m_n = \min_j M_{j,n}(C)$, we have

$$\begin{aligned}
R(C, P) - D_{\mathrm{hard}}(C) &= \sum_{k,n} P_{k,n} M_{k,n}(C) - \sum_n b_n m_n \\
&= \sum_{k,n} P_{k,n}\big(M_{k,n}(C) - m_n\big) \geq 0.
\end{aligned}$$

The equality condition follows from the nonnegativity of each summand. □

Fact 6 (Stationarity under linear constraints)**.** *If F is differentiable on $\mathbb{R}_{++}^{K\times N}$ and a positive $P^\star \in \Pi(a, b)$ minimizes F, there are vectors λ and μ such that*

$$\frac{\partial F}{\partial P_{k,n}}(P^\star) + \lambda_k + \mu_n = 0. \tag{28}$$

This is the first-order condition on the affine marginal constraints: $\nabla F(P^\star)$ is orthogonal to matrices with zero row and column sums.

Fact 7 (Taylor expansion with third-order remainder)**.** *If $\mathcal{L}$ is C^3 near $\{W + t\delta : 0 \leq t \leq 1\}$ and $|D^3\mathcal{L}(W + t\delta)[\delta, \delta, \delta]| \leq L_3\|\delta\|_2^3$ on this segment, then*

$$\mathcal{L}(W + \delta) = \mathcal{L}(W) + \nabla\mathcal{L}(W)^\top\delta + \tfrac{1}{2}\delta^\top\nabla^2\mathcal{L}(W)\delta + r, \qquad |r| \leq \tfrac{L_3}{6}\|\delta\|_2^3. \tag{29}$$

Here weights are viewed as vectors. The integral remainder is used in Appendix E.6.

E.2 Proof of Theorem 1

Proof. Fix C and abbreviate $M = M(C)$ and $\Phi(P) = \Phi_\tau(C, P)$. Compactness, continuity, and strict convexity from Fact 4 ensure that

$$P^\star = P_\tau(C) = \arg\min_{P \in \Pi(a,b)} \Phi(P) \tag{30}$$

exists and is unique. It remains to prove positivity and identify the scaling representation.

Suppose $Z = \{(k, n) : P^\star_{k,n} = 0\}$ is nonempty. For $0 < t < 1$, the coupling $P_t = (1-t)P^\star + tab^\top$ is feasible. With $\psi_{k,n}(p) = p \log(p/(a_k b_n))$, the objective change is

$$\begin{aligned}\Phi(P_t) - \Phi(P^\star) &= t\langle M, ab^\top - P^\star\rangle + \tau \sum_{(k,n)\in Z} t a_k b_n \log t \\ &\quad + \tau \sum_{(k,n)\notin Z} \big[\psi_{k,n}(P_{t,k,n}) - \psi_{k,n}(P^\star_{k,n})\big] \\ &= \tau t \log t \sum_{(k,n)\in Z} a_k b_n + O(t) < 0\end{aligned} \tag{31}$$

for sufficiently small $t > 0$. The $O(t)$ term uses differentiability at the finitely many positive entries, whereas the coefficient of $t \log t$ is strictly positive. This contradicts optimality, so $P^\star > 0$.

Stationarity under the marginal constraints now gives multipliers λ and μ with

$$0 = M_{k,n} + \tau\left(\log \frac{P^\star_{k,n}}{a_k b_n} + 1\right) + \lambda_k + \mu_n, \tag{32}$$

$$\begin{aligned}P^\star_{k,n} &= a_k b_n \exp\left(-1 - \frac{\lambda_k + \mu_n}{\tau}\right) \exp(-M_{k,n}/\tau) \\ &= u_k \exp(-M_{k,n}/\tau) v_n,\end{aligned} \tag{33}$$

where $u_k = a_k \exp(-1 - \lambda_k/\tau)$ and $v_n = b_n \exp(-\mu_n/\tau)$ are positive. This proves Equation (7). Any two representations of the same coupling satisfy $u'_k/u_k = v_n/v'_n$ for every (k, n) because the kernel is positive; thus $(u', v') = (\alpha u, \alpha^{-1} v)$ for some $\alpha > 0$.

To relate the representation to Algorithm 1, let $\mathcal{K} = \exp(-M/\tau)$. Exponentiating its log-domain updates gives

$$u^{(r)}_k = \frac{a_k}{(\mathcal{K} v^{(r-1)})_k}, \qquad v^{(r)}_n = \frac{b_n}{(\mathcal{K}^\top u^{(r)})_n}. \tag{34}$$

The two half-steps enforce the row and column marginals, respectively. At a positive fixed point, the scaled coupling $\bar{P} = \operatorname{diag}(u)\mathcal{K}\operatorname{diag}(v)$ belongs to $\Pi(a, b)$, and $\nabla\Phi(\bar{P})_{k,n} = \rho_k + \sigma_n$, where $\rho_k = \tau(\log(u_k/a_k) + 1)$ and $\sigma_n = \tau \log(v_n/b_n)$. For any $P \in \Pi(a, b)$, convexity yields

$$\begin{aligned}\Phi(P) - \Phi(\bar{P}) &\geq \langle \nabla\Phi(\bar{P}), P - \bar{P}\rangle \\ &= \sum_k \rho_k \big[(P - \bar{P})\mathbf{1}_N\big]_k + \sum_n \sigma_n \big[(P - \bar{P})^\top \mathbf{1}_K\big]_n \\ &= 0.\end{aligned}$$

Hence $\bar{P} = P_\tau(C)$ by uniqueness. This identifies the positive fixed-point coupling; a finite iterate need not satisfy both marginals. □

E.3 Proof of Theorem 2

Proof. Let $q_k = q_k(P) > 0$ and write

$$\bar{c}_k = c^P_k = \frac{\sum_n P_{k,n} x_n}{q_k(P)}, \qquad C^P = \{\bar{c}_k\}_{k=1}^K. \tag{35}$$

The centering identity $\sum_n P_{k,n}(\bar{c}_k - x_n) = 0$ eliminates the cross term in the quadratic expansion:

$$\begin{aligned} R(C,P) - R(C^P,P) &= \sum_{k,n} P_{k,n}\big(\|c_k - x_n\|_H^2 - \|\bar{c}_k - x_n\|_H^2\big) \\ &= \sum_k q_k \|c_k - \bar{c}_k\|_H^2 + 2\sum_k \left\langle c_k - \bar{c}_k, \sum_n P_{k,n}(\bar{c}_k - x_n) \right\rangle_H \\ &= \sum_k q_k(P)\|c_k - c_k^P\|_H^2. \end{aligned} \tag{36}$$

Since $q_k > 0$ and $H \succ 0$, the right-hand side vanishes exactly when $C = C^P$. Thus C^P is the unique fixed-plan minimizer, as in the weighted second Lloyd condition (Lloyd, 1982).

For the exact pair, $q_k(P^+) = a_k$ and $C^{P^+} = C^+$. Let $\Delta_0 = \sum_k a_k \|c_k^+ - c_k^0\|_H^2$. Applying Equation (36) while keeping the coupling fixed gives the profiled descent chain

$$\begin{aligned} J_\tau(C^+) &= \min_{P \in \Pi(a,b)} \Phi_\tau(C^+, P) \\ &\le \Phi_\tau(C^+, P^+) \\ &= R(C^+, P^+) + \tau\,\mathrm{KL}(P^+ \| ab^\top) \\ &= R(C^0, P^+) - \Delta_0 + \tau\,\mathrm{KL}(P^+ \| ab^\top) \\ &= J_\tau(C^0) - \Delta_0. \end{aligned} \tag{37}$$

Here the last equality uses exact E-step optimality. If any codeword moves, then $\Delta_0 > 0$ and hence $J_\tau(C^+) < J_\tau(C^0)$.

Finally, Fact 5 and nonnegativity of the KL term give, for every C,

$$\begin{aligned} D_{\mathrm{hard}}(C) &\le \min_{P \in \Pi(a,b)} R(C,P) = D_{\mathrm{bal}}(C) \\ &\le \min_{P \in \Pi(a,b)} \big[R(C,P) + \tau\,\mathrm{KL}(P \| ab^\top)\big] = J_\tau(C). \end{aligned} \tag{38}$$

Combining the two bounds also gives $D_{\mathrm{hard}}(C^+) \le J_\tau(C^0) - \Delta_0$. □

E.4 Proof of Theorem 3

Proof. Let (P^+, C^+) be the exact pair generated from C^0. Recall the gain and penalty

$$\Delta_0 = \sum_k a_k \|c_k^+ - c_k^0\|_H^2, \qquad \Gamma_0 = R(C^0, P^+) - D_{\mathrm{hard}}(C^0). \tag{39}$$

The additional cost removed by hard reassignment at C^+ is

$$S_1 = R(C^+, P^+) - D_{\mathrm{hard}}(C^+) \ge 0, \tag{40}$$

where nonnegativity follows from Fact 5. Using the barycentric identity, we obtain

$$\begin{aligned} D_{\mathrm{hard}}(C^+) - D_{\mathrm{hard}}(C^0) &= R(C^+, P^+) - S_1 - D_{\mathrm{hard}}(C^0) \\ &= R(C^0, P^+) - \Delta_0 - S_1 - D_{\mathrm{hard}}(C^0) \\ &= \Gamma_0 - \Delta_0 - S_1 \qquad (41) \\ &\le \Gamma_0 - \Delta_0. \qquad (42) \end{aligned}$$

Thus $\Delta_0 > \Gamma_0$ gives Equation (9) with $m_Q = \Delta_0 - \Gamma_0$. Equivalently, the condition is $R(C^+, P^+) < D_{\mathrm{hard}}(C^0)$. It is sufficient rather than necessary because the bound discards S_1.

Bounding the Assignment Penalty. To separate the balancing and entropic contributions, set

$$\Gamma_{\mathrm{bal}}(C^0) = D_{\mathrm{bal}}(C^0) - D_{\mathrm{hard}}(C^0), \tag{43}$$

and let $h_{a,b} = \min\{\mathcal{E}(a), \mathcal{E}(b)\}$. For every $P \in \Pi(a,b)$, exact E-step optimality gives

$$\begin{aligned} R(C^0, P^+) &\le R(C^0, P^+) + \tau\,\mathrm{KL}(P^+ \| ab^\top) \\ &= \Phi_\tau(C^0, P^+) \\ &\le \Phi_\tau(C^0, P) = R(C^0, P) + \tau\,\mathrm{KL}(P \| ab^\top). \end{aligned} \tag{44}$$

Choose an unregularized balanced minimizer P^{bal} and apply Fact 3. Then

$$\begin{aligned} \Gamma_0 &\le R(C^0, P^{\mathrm{bal}}) - D_{\mathrm{hard}}(C^0) + \tau\,\mathrm{KL}(P^{\mathrm{bal}} \| ab^\top) \\ &\le \Gamma_{\mathrm{bal}}(C^0) + \tau h_{a,b}. \end{aligned} \tag{45}$$

It remains to bound $\Gamma_{\mathrm{bal}}(C^0)$ using the old hard assignment.

Write $M = M(C^0)$, $z_n = z_n(C^0)$, and $m_n = M_{z_n,n} = \min_k M_{k,n}$. The hard coupling $P^{\mathrm{h}}_{k,n} = b_n \mathbf{1}\{k = z_n\}$ has column marginal b, row marginal q^{h}, and cost $R(C^0, P^{\mathrm{h}}) = D_{\mathrm{hard}}(C^0)$. Set

$$\Omega_0 = \max_n \big[\max_k M_{k,n} - \min_k M_{k,n}\big]. \tag{46}$$

Let $S = \{k : q^{\mathrm{h}}_k > a_k\}$ and $T = \{k : q^{\mathrm{h}}_k < a_k\}$ be the surplus and deficit rows, with $e_k = q^{\mathrm{h}}_k - a_k$ for $k \in S$ and $f_j = a_j - q^{\mathrm{h}}_j$ for $j \in T$. Their common total mass, the total variation (TV) distance, is

$$\theta = \sum_{k \in S} e_k = \sum_{j \in T} f_j = \tfrac{1}{2}\|q^{\mathrm{h}} - a\|_1 = \mathrm{TV}(q^{\mathrm{h}}, a). \tag{47}$$

If $\theta = 0$, P^{h} is already balanced and $\Gamma_{\mathrm{bal}}(C^0) = 0$. Assume below that $\theta > 0$.

A Feasible Reference Coupling. For $k \in S$, $j \in T$, and $N_k = \{n : z_n = k\}$, define $Y_{k,j} = e_k f_j/\theta$ and $\mu_{k,n} = b_n e_k / q^{\mathrm{h}}_k$ for $n \in N_k$. These quantities satisfy

$$\begin{aligned} &\sum_{j \in T} Y_{k,j} = e_k, \qquad \sum_{k \in S} Y_{k,j} = f_j, \\ &\sum_{n \in N_k} \mu_{k,n} = \frac{e_k}{q^{\mathrm{h}}_k} \sum_{n \in N_k} b_n = e_k, \qquad 0 < \mu_{k,n} < b_n. \end{aligned} \tag{48}$$

Remove $\mu_{k,n}$ from the hard entry (k,n) and send it to deficit row j in proportion $Y_{k,j}/e_k$. More precisely, the nonzero entries of the removed and added matrices are $\Theta^-_{k,n} = \mu_{k,n}$ and $\Theta^+_{j,n} = \mu_{k,n} Y_{k,j}/e_k$ for $k \in S$, $j \in T$, and $n \in N_k$. Define

$$P^{\mathrm{ref}} = P^{\mathrm{h}} - \Theta^- + \Theta^+. \tag{49}$$

The bound $\mu_{k,n} < b_n$ gives nonnegativity, and the identities above give

$$\begin{aligned} &\sum_j P^{\mathrm{ref}}_{j,n} = b_n - \mu_{k,n} + \mu_{k,n} \sum_{j \in T} \frac{Y_{k,j}}{e_k} = b_n, \qquad n \in N_k,\ k \in S, \\ &\sum_n P^{\mathrm{ref}}_{k,n} = \begin{cases} q^{\mathrm{h}}_k - e_k = a_k, & k \in S, \\ q^{\mathrm{h}}_k + \sum_{j \in S} Y_{j,k} = a_k, & k \in T, \\ q^{\mathrm{h}}_k = a_k, & k \notin S \cup T. \end{cases} \end{aligned} \tag{50}$$

All other columns are unchanged, so $P^{\mathrm{ref}} \in \Pi(a,b)$. Only the transferred mass incurs additional cost. Since $0 \le M_{j,n} - m_n \le \Omega_0$, we have

$$\begin{aligned} R(C^0, P^{\mathrm{ref}}) - D_{\mathrm{hard}}(C^0) &= \sum_{k \in S} \sum_{n \in N_k} \sum_{j \in T} \mu_{k,n} \frac{Y_{k,j}}{e_k} (M_{j,n} - m_n) \\ &\le \Omega_0 \sum_{k \in S} \sum_{n \in N_k} \mu_{k,n} \sum_{j \in T} \frac{Y_{k,j}}{e_k} \\ &= \Omega_0 \sum_{k \in S} e_k = \theta \Omega_0. \end{aligned} \tag{51}$$

Consequently,

$$\Gamma_{\mathrm{bal}}(C^0) \le R(C^0, P^{\mathrm{ref}}) - D_{\mathrm{hard}}(C^0) \le \theta\Omega_0,$$
$$\Gamma_0 \le \Gamma_{\mathrm{bal}}(C^0) + \tau h_{a,b} \le \mathrm{TV}(q^{\mathrm{h}}, a)\,\Omega_0 + \tau h_{a,b}. \tag{52}$$

The same bound holds for $\theta = 0$ by choosing $P^{\mathrm{ref}} = P^{\mathrm{h}}$.

Either upper bound in Equation (52) yields a sufficient condition for the main-text certificate. One such condition is

$$\Delta_0 > \mathrm{TV}(q^{\mathrm{h}}, a)\,\Omega_0 + \tau \min\{\mathcal{E}(a), \mathcal{E}(b)\} \tag{53}$$

implies $\Delta_0 > \Gamma_0$ and hence strict hard-distortion descent. Under uniform marginals, $h_{a,b} = \log\min\{K, N\}$ and $\mathrm{TV}(q^{\mathrm{h}}, a) = \frac{1}{2}\|q^{\mathrm{h}} - K^{-1}\mathbf{1}\|_1$. Thus balancing and entropy contribute separately to the threshold; failure of this sufficient condition does not rule out an improvement. □

E.5 Proof of Theorem 4

Proof. We first establish the layer-output identity, then compare the two model losses. The latter comparison uses only the distortion margin and the definition of η_C, independently of the metric-alignment hypotheses.

Layer-Output Error. For one layer, let $\Delta = \Delta W_C = Q_C(W) - W$. Since $\Sigma = \mathbb{E}[\xi\xi^\top]$,

$$\mathbb{E}\|\Delta\xi\|_2^2 = \mathbb{E}\,\mathrm{tr}(\Delta\xi\xi^\top\Delta^\top)$$
$$= \mathrm{tr}\big(\Delta\mathbb{E}[\xi\xi^\top]\Delta^\top\big) = \mathrm{tr}(\Delta\Sigma\Delta^\top). \tag{54}$$

Under $\Sigma = \mathrm{diag}(\sigma)$, this is $\sum_{i,j}\sigma_j\Delta_{ij}^2$. For a block n of group g lying in row $i(n)$, the perturbation on J_g is $c_{z_n}^{(g)} - x_n$, where z_n minimizes its H_g-weighted cost. The aggregate distortion is

$$D(C) = \sum_g N_g D_{\mathrm{hard}}^{(g)}(C) = \sum_g \sum_{n\in g} \min_k \|c_k^{(g)} - x_n\|_{H_g}^2. \tag{55}$$

Using the aligned metrics and the block partition gives

$$D(C) = \sum_g \sum_{n\in g} \|c_{z_n}^{(g)} - x_n\|_{H_g}^2$$
$$= \kappa \sum_g \sum_{n\in g} \sum_{j\in J_g} \sigma_j \Delta_{i(n),j}^2$$
$$= \kappa \sum_{i,j} \sigma_j \Delta_{ij}^2$$
$$= \kappa\,\mathrm{tr}(\Delta\Sigma\Delta^\top) = \kappa\,\mathbb{E}\|\Delta W_C\xi\|_2^2. \tag{56}$$

The block partition justifies the third equality. Since $\kappa > 0$ is independent of C, a strict distortion reduction is equivalent to a strict reduction in expected layer-output error.

Loss and Perplexity. Now sum distortion over all quantized layers. The groupwise certificates aggregate as

$$D(C^+) - D(C^0) = \sum_g N_g\big[D_{\mathrm{hard}}^{(g)}(C^+) - D_{\mathrm{hard}}^{(g)}(C^0)\big] \le -\sum_g N_g m_Q^{(g)} = -m_{\mathrm{tot}}. \tag{57}$$

By definition,

$$\eta_C = \big|\mathcal{L}(C) - \mathcal{L}(W) - \tfrac{1}{2}D(C)\big|, \tag{58}$$

so $\mathcal{L}(C)$ lies between $\mathcal{L}(W) + \frac{1}{2}D(C) - \eta_C$ and $\mathcal{L}(W) + \frac{1}{2}D(C) + \eta_C$. Applying the upper bound to C^+ and the lower bound to C^0 yields

$$\mathcal{L}(C^+) - \mathcal{L}(C^0) \le \big[\mathcal{L}(W) + \tfrac{1}{2}D(C^+) + \eta_{C^+}\big] - \big[\mathcal{L}(W) + \tfrac{1}{2}D(C^0) - \eta_{C^0}\big]$$
$$= \tfrac{1}{2}\big[D(C^+) - D(C^0)\big] + \eta_{C^+} + \eta_{C^0}$$
$$\le -\tfrac{1}{2}m_{\mathrm{tot}} + \eta_{C^+} + \eta_{C^0}. \tag{59}$$

If $m_{\mathrm{tot}} > 2(\eta_{C^+} + \eta_{C^0})$, the right-hand side is negative. Under this condition, exponentiation gives

$$\begin{aligned}\frac{\mathrm{PPL}(C^+)}{\mathrm{PPL}(C^0)} &= \exp\big(\mathcal{L}(C^+) - \mathcal{L}(C^0)\big) \\ &\le \exp\big(-\tfrac{1}{2}m_{\mathrm{tot}} + \eta_{C^+} + \eta_{C^0}\big) < 1. \end{aligned} \tag{60}$$

This proves the loss and perplexity claims. □

E.6 Proof of Lemma 1

Proof. View the loss as a function of the vectorized weights and let $h(t) = \mathcal{L}(W + t\delta_C)$ for $0 \le t \le 1$. The chain rule gives $h'(0) = g^\top \delta_C$ and $h''(0) = \delta_C^\top \mathfrak{H} \delta_C$. Taylor's formula gives

$$\begin{aligned}\mathcal{L}(C) - \mathcal{L}(W) &= h(1) - h(0) \\ &= h'(0) + \tfrac{1}{2}h''(0) + \tfrac{1}{2}\int_0^1 (1-t)^2 h'''(t)\,dt \\ &= g^\top \delta_C + \tfrac{1}{2}\delta_C^\top \mathfrak{H} \delta_C + r_C, \end{aligned} \tag{61}$$

$$\begin{aligned}|r_C| &\le \tfrac{1}{2}\int_0^1 (1-t)^2 \big|D^3\mathcal{L}(W + t\delta_C)[\delta_C, \delta_C, \delta_C]\big|\,dt \\ &\le \tfrac{L_3}{2}\|\delta_C\|_2^3 \int_0^1 (1-t)^2\,dt = \tfrac{L_3}{6}\|\delta_C\|_2^3. \end{aligned} \tag{62}$$

This proves Equation (22). Subtracting the quadratic distortion $D(C) = \delta_C^\top \widetilde{H} \delta_C$ and applying the remainder bound gives

$$\begin{aligned}\eta_C &= \big|g^\top \delta_C + \tfrac{1}{2}\delta_C^\top(\mathfrak{H} - \widetilde{H})\delta_C + r_C\big| \\ &\le |g^\top \delta_C| + \tfrac{1}{2}\big|\delta_C^\top(\mathfrak{H} - \widetilde{H})\delta_C\big| + |r_C| \\ &\le |g^\top \delta_C| + \tfrac{1}{2}\big|\delta_C^\top(\mathfrak{H} - \widetilde{H})\delta_C\big| + \tfrac{L_3}{6}\|\delta_C\|_2^3. \end{aligned} \tag{63}$$

This is Equation (23). □

F Limitations

The study focuses on codebook-based weight quantization at low bit widths. Quantization behavior may also depend on model architecture, calibration data, and deployment conditions. Exploring the method across a broader range of models and settings may further clarify how codebook refinement relates to downstream performance.

G More Discussions

Q1. Does balanced fitting force uniform codeword use at deployment?

No. The uniform row marginal fixes the fitting mass of each codeword in the exact soft assignment (Theorem 1). Deployment discards that assignment and applies the ordinary nearest-codeword encoder (Equation 1). Hard use is therefore measured separately. Figure 7 shows higher realized use for BARQ at every measured layer for $K = 16$ and in 74 of 75 matched layers for $K = 256$; it is an observed result, not a consequence of the marginal constraint alone.

Q2. What remains true when Sinkhorn scaling stops after finitely many iterations?

The exact E-step has a unique positive coupling with the prescribed marginals. A finite iterate can retain a row-marginal residual, as described in Appendix A. The implemented M-step divides by the actual row mass with a 10^{-12} floor. Whenever each mass exceeds that floor, its codeword update is still the exact barycenter of the computed assignment (Equation 21); the exact-coupling guarantee and the fixed-assignment update are distinct statements.

Q3. Why can a better fitting objective improve the final hard quantizer?

Fitting-objective descent alone does not establish hard-distortion descent. Theorem 3 separates the barycentric gain Δ_0 from the cost Γ_0 of using the balanced assignment at the old codebook: $\Delta_0 > \Gamma_0$ is sufficient for a lower hard distortion after reassignment. The measured perplexity and zero-shot differences in Table 1 are empirical comparisons of the deployed quantizers; the theorem states the condition under which the local fitting gain transfers.

Q4. How should the reported bit budgets be compared?

The displayed bpv values count the index and group-scale terms. The full expression also contains codebook storage Kdb_c/l (Equation 12), whose size depends on codebook sharing. The term is identical within each paired GPT-VQ/BARQ setting because the two methods use the same codebook size and sharing configuration. The external-baseline table instead places published methods at nearby reported operating points; its bpv values should be read as approximate budget matches rather than exact compressed-file sizes.

Q5. Is higher codeword utilization sufficient to explain the performance gains?

The measurements separate use from reconstruction quality. At $K = 4$, both methods nearly saturate the codebook and several paired configurations still regress; at $K = 16$ and $K = 256$, BARQ usually uses more codewords and has larger average perplexity gains (Figure 5; Table 1). Figure 4 also measures lower curvature-weighted reconstruction error in the plotted larger-codebook settings. These observations support reading utilization alongside the encoded reconstruction error, not as a stand-alone performance metric.

Q6. Does the GPT-VQ comparison isolate only the assignment rule?

The paired rows share model weights, calibration data, nominal codebook size, and reported bpv. They begin from the same hard k-means codebook, but the GPT-VQ pipeline may include a later lookup-table refinement while BARQ uses a continuous barycentric update (Appendix B). Table 1 therefore compares the resulting end-to-end quantizers under matched settings; it does not attribute the entire difference to one assignment step.