RECAP: REFINE QUANTIZED LLM CHECKPOINTS, DON'T REPLACE THEM

Anonymous authors
Paper under double-blind review

ABSTRACT

When a deployed quantized large language model requires higher accuracy, transmitting a complete replacement checkpoint repeats information already available at the client. We formulate successive model refinement as encoding an incremental correction with the cached checkpoint as shared side information. **The central challenge is to recover model quality with an incremental correction substantially smaller than a replacement checkpoint.** We introduce RECAP, which combines two correction representations with different communication costs. An activation-subspace code transmits coefficients in a basis reconstructed at both endpoints from the cached model and public calibration data, with water-filling allocating bits by activation-weighted coefficient energy; a complement code transmits low-rank factors to correct the remaining residual. Our analysis establishes the shared basis's minimax optimality, derives optimal coefficients and approximate rate allocation, and quantifies the complement's benefit. We ask whether a correction can deliver at least the quality of the next-higher-bit replacement checkpoint while transmitting fewer bytes. From four 1.58-bit checkpoints of 7B models, RECAP reaches the perplexity of same-family 2-bit replacements with 2–19% of their bytes; at a fixed budget, it also achieves lower perplexity than conventional reconstruction baselines at both 7B and 13B scale. These findings show that even a checkpoint with poor standalone performance can remain valuable side information for communication-efficient model refinement. Code is available here.

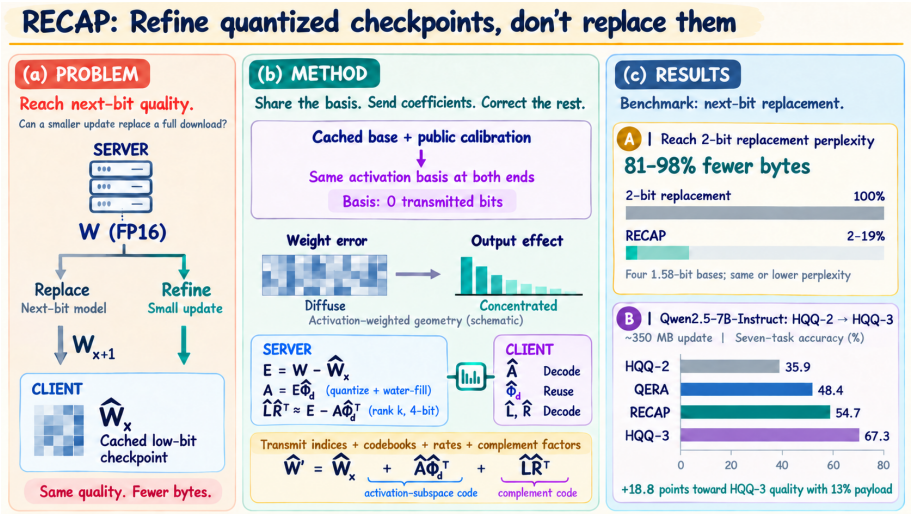


Figure 1: **RECAP refines rather than replaces PTQ checkpoints.** (a) **Reuse the deployed model:** send a compact residual update instead of a higher-bit replacement. (b) **Combine shared and transmitted directions:** send coefficients in a locally reconstructed activation basis plus a low-rank complement for the remaining residual; the basis costs no transmitted bits. (c) **More quality per byte:** reach 2-bit replacement perplexity with **81–98% fewer bytes** and outperform QERA in accuracy at matched payload, moving toward 3-bit quality.

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.

1 INTRODUCTION

Serving LLMs such as Llama 2, Qwen, and Qwen2.5 Touvron et al. (2023); Bai et al. (2023); Qwen Team (2024) away from the data center makes memory, bandwidth, and energy binding constraints. Post-training quantization (PTQ) methods, including GPTQ, AWQ, SmoothQuant, HQQ, and SINQ Frantar et al. (2023); Lin et al. (2024); Xiao et al. (2023); Badri & Shaji (2023); Müller et al. (2026), make such deployment practical by compressing an already-trained checkpoint with little or no calibration data. Each client then caches a quantized checkpoint $\widehat{\mathbf{W}}_x$ at a bit width x chosen for its storage–quality tradeoff. Deployment is not the end of the model's life: its task mix or service target may later demand higher quality. The usual response is to re-quantize and re-transmit a complete checkpoint. Even the 2-bit replacements for the 7B models in our study cost 1.82–1.84 GB per client over the targeted linear layers. On metered or bandwidth-limited links, reducing these bytes helps fit updates within a client's download budget or update window.

We call the alternative *successive model refinement*: given the reference $\mathbf{W}$ at the server and a deployed checkpoint at the client, the server encodes only what the client is missing. This is source coding with the cached checkpoint as decoder-side information; the cost is the complete incremental message in bits, and checkpoint re-transmission is the operational comparison. LoRC in ZeroQuant-V2, LQER, and QERA Yao et al. (2023); Zhang et al. (2024; 2025) provide natural starting points for quantization-error reconstruction, but rank is not a communication cost: these low-rank corrections transmit both factors, whereas a factor that the receiver can reproduce need not be transmitted.

Starting from four 7B 1.58-bit checkpoints, RECAP matches or exceeds the seven-task accuracy of a same-family 2-bit replacement while sending only 37–38% as many bytes (Appendix D.2). For each of these four bases, the first tested RECAP payload that reaches the corresponding replacement's perplexity uses 2–19% of its bytes (Figure 2). **Even a checkpoint with poor standalone quality can therefore retain information that substantially reduces the cost of reaching a higher quality target.** Realizing this gain requires a residual code matched to model execution. In weight space, the quantization residual $\mathbf{E} = \mathbf{W} - \widehat{\mathbf{W}}_x$ has energy spread across many directions in the PTQ codecs we test. Its effect on a linear layer is instead $\mathbf{Ea}$, where $\mathbf{a}$ is the input activation. Directions with greater activation energy receive more weight in the output error, so correcting them can be valuable even when the weight residual itself is diffuse. Both endpoints can reconstruct these activation directions from the cached base and public calibration data (Figure 1; Section 4.1).

RECAP (REsidual Coding in Activation Principal directions) is the codec implied by this geometry. Its activation-subspace code projects each residual onto the shared basis, and the client reconstructs that basis by running a fixed public calibration protocol through its deployed checkpoint. The basis therefore costs zero transmitted bits. The subspace code spends its payload on the quantized payload matrix, codebooks, and rate metadata; the leading coefficient cost scales with output rows times basis dimension rather than with the full weight matrix. Water-filling then allocates the available rate across directions and precisions. The complement code spends the remaining budget on a quantization-aware low-rank correction fitted to the residual left by the decoded subspace code (Section 4). The subspace distortion-floor theorem identifies the optimal correction inside the shared subspace and explains the observed saturation, while the variation across PTQ families shows that nominal precision alone does not determine the refinement frontier.

CONTRIBUTIONS

1. **A source-coding formulation of successive model refinement.** The formulation compares residual coding and checkpoint replacement by the additional communication needed to reach a target quality, given the client's cached checkpoint (Section 3).
2. **RECAP: an activation-subspace code and a complement code.** Our analysis establishes the optimal correction within the shared subspace and decomposes its distortion into unrepresented residual energy and coefficient-quantization error. Under a common metric, sequential fitting attains the unquantized two-code joint distortion floor (Section 4, Theorems 1 and 3).
3. **Refinement at a fraction of replacement cost.** Across four 7B 1.58-bit checkpoints, RECAP matches or exceeds the seven-task accuracy of same-family 2-bit replacements using only 37–38% of their transmitted bytes, and reaches their perplexity with 2–19%. It also achieves best or tied-best perplexity in 22 of 24 matched-payload comparisons across both size tiers (Section 5).

2 Related Work

Refining a model that a client already holds touches three literatures: how the deployed base was produced, how a quantized matrix can be corrected, and how such a correction has been treated as a message. We review each in turn and close with the gap they leave.

Post-training quantization and activation-aware design. GPTQ Frantar et al. (2023) uses second-order calibration to quantize weights. HQQ and SINQ Badri & Shaji (2023); Müller et al. (2026) are calibration-free weight-only methods; LLM.int8() and SpQR Dettmers et al. (2022; 2024) handle outliers explicitly; and QuIP# and AQLM Tseng et al. (2024); Egiazarian et al. (2024) use vector quantization at extreme rates. Of particular relevance, AWQ Lin et al. (2024) protects activation-salient weight channels, whereas SmoothQuant Xiao et al. (2023) transfers activation-quantization difficulty into the weights. Both show that a base codec determines not only residual magnitude but its geometry relative to the activation distribution.

Quantization-error reconstruction. Most directly related, these methods augment a quantized matrix with a compact correction, and they differ chiefly in the statistic that shapes it. LoRC, the low-rank component of ZeroQuant-V2 Yao et al. (2023), factors the weight error itself. By the Eckart–Young theorem Eckart & Young (1936), this is optimal for that error in the Frobenius sense but blind to how the network propagates it. LQER Zhang et al. (2024) scales the error by an activation-derived diagonal before factoring. QERA Zhang et al. (2025) minimizes the layer output error by preconditioning the weight error with the square root of the input autocorrelation before truncation. CALDERA Saha et al. (2024) quantizes the factors themselves and makes the factorization rounding-aware, alternating least-squares steps with quantization of each iterate. SERQ Park et al. (2026) targets saliency-aware weight-and-activation quantization. ProjQ Yu et al. (2026) co-designs quantization and low-rank compensation, shaping activation-space quantization error toward an adapter-compatible subspace. It changes base construction, whereas RECAP refines a fixed checkpoint already cached at the client. Of these, QERA is closest in its activation-weighted objective; RECAP instead asks how to encode such a correction under a complete transmitted-bit budget.

Progressive transmission and delta compression. Sending a coarse representation and later improving it is classical successive refinement Equitz & Cover (1991). Unlike Wyner–Ziv coding Wyner & Ziv (1976), where side information is available only to the decoder, RECAP's cached checkpoint is available to both endpoints. Communication-oriented relatives include BitDelta, Delta-CoMe, DeltaZip, and LoRA Liu et al. (2024); Ping et al. (2024); Yao et al. (2025); Hu et al. (2022), which encode or parameterize checkpoint differences using sign, low-rank, sparse, or quantized representations. These methods target the delta between a model and a fine-tuned variant of it, whose structure differs from a quantization residual, so they answer a different question than the one posed here.

Together, these threads motivate successive model refinement: encoding a quantization correction against a cached checkpoint under a complete transmitted-bit budget.

3 Problem

We now pose the codec design problem that Section 2 leaves open. A public PTQ codec (Q_x, Q_x^{-1}), with nominal precision x bits per weight, maps a full-precision matrix $\mathbf{W} \in \mathbb{R}^{m\times n}$ to low-bit integer codes and scales, and the deployed base is the deterministic dequantization $\widehat{\mathbf{W}}_x$. Fractional values such as $x = 1.58$ denote a nominal coding rate; scales and other metadata are counted separately in the complete payload. What quantization discarded is the residual

$$\mathbf{E} \;=\; \mathbf{W} - \widehat{\mathbf{W}}_x, \tag{1}$$

so a refinement carries an estimate $\widehat{\mathbf{E}}$ of it and the client forms $\widehat{\mathbf{W}}' = \widehat{\mathbf{W}}_x + \widehat{\mathbf{E}}$, leaving $\mathbf{W} - \widehat{\mathbf{W}}' = \mathbf{E} - \widehat{\mathbf{E}}$. Reconstruction is therefore estimation of $\mathbf{E}$ under a rate constraint: $\widehat{\mathbf{E}} = \mathbf{E}$ restores $\mathbf{W}$ exactly, while a dense full-precision representation has the same nominal size as $\mathbf{W}$. A compact codec trades the leftover $\mathbf{E} - \widehat{\mathbf{E}}$ against the bits that describe $\widehat{\mathbf{E}}$.

A *server* holds the full-precision checkpoint $\mathcal{W}$, the collection of all target matrices, and a fleet of *clients* each cache its quantized counterpart $\widehat{\mathcal{W}}_x$. Refinement touches only the linear projections of

each transformer block; embeddings, the LM head, normalization, the architecture, and the tokenizer are fixed side context, identical at both ends and never transmitted. For one matrix an encoder emits a message component and a decoder reconstructs from that component and the cached base, with the public calibration data and protocol treated as fixed side context,

$$\mathcal{M} \;=\; f_{\mathrm{enc}}\big(\mathbf{W},\, \widehat{\mathbf{W}}_x;\, \boldsymbol{\eta}\big), \qquad \widehat{\mathbf{W}}' \;=\; g_{\mathrm{dec}}\big(\mathcal{M},\, \widehat{\mathbf{W}}_x\big) \;=\; \widehat{\mathbf{W}}_x + \widehat{\mathbf{E}}, \tag{2}$$

where $\boldsymbol{\eta}$ collects codec parameters and the exchange is one-shot, deterministic, and free of client-side optimization. The full message M concatenates the per-matrix components, and $|\mathsf{M}|$ counts every newly transmitted bit needed to decode it: quantized values, scales, codebooks, and per-matrix metadata, excluding the cached base and the side context.

Quality is judged on the assembled checkpoint by a task loss $\mathcal{L}(\cdot)$, WikiText-2 perplexity in our experiments with downstream accuracy as a guard, through the gap to full precision $\Delta(\widehat{\mathcal{W}}') = \mathcal{L}(\widehat{\mathcal{W}}') - \mathcal{L}(\mathcal{W})$. Given a total transmission budget B_{total}, the codec design problem is

$$\boxed{\min_{f_{\mathrm{enc}},\, g_{\mathrm{dec}}} \quad \Delta\big(\widehat{\mathcal{W}}'\big) \quad \text{s.t.} \quad |\mathsf{M}| \le B_{\mathrm{total}},} \tag{3}$$

4 METHOD

RECAP instantiates these encoder–decoder maps for each target matrix: the encoder transmits coded residual coefficients and complement factors, and the decoder reconstructs a correction to the cached weights. Section 4.1 develops a tractable surrogate for designing these messages.

4.1 RECAP'S ACTIVATION-SUBSPACE CODE

Directly optimizing f_{enc} and g_{dec} in (3) is costly: the task loss couples corrections across layers through the full nonlinear model. We therefore replace this global objective with a tractable layer-wise surrogate, using per-matrix bit budgets that together satisfy B_{total}. For each matrix, we design the correction $\widehat{\mathbf{E}}$ to minimize the output distortion caused by the remaining error $\mathbf{R} = \mathbf{E} - \widehat{\mathbf{E}}$:

$$D_{\mathbf{H}}(\mathbf{R}) := \frac{1}{T}\sum_{t=1}^{T} \|\mathbf{R}\mathbf{a}_t\|_2^2 = \mathrm{tr}(\mathbf{R}\mathbf{H}\mathbf{R}^\top), \qquad \mathbf{H} := \frac{1}{T}\sum_{t=1}^{T} \mathbf{a}_t\mathbf{a}_t^\top. \tag{4}$$

Here $\mathbf{a}_t \in \mathbb{R}^n$ are inputs collected from the cached checkpoint, and $\mathbf{H}$ is their uncentered second moment. This calibration objective Frantar et al. (2023); Zhang et al. (2025) guides codec design; Section 5 evaluates the resulting model-level task loss.

To avoid transmitting a basis, RECAP's activation-subspace code represents the correction as $\mathbf{A}\mathbf{V}^\top$, with the $n \times d$ transform chosen from $\mathbf{H}$. Both endpoints reproduce $\mathbf{H}$ under the shared calibration protocol of Section 3, whereas only the server knows $\mathbf{E}$. Since $\mathbf{H}$ weights input directions in the distortion, we retain its leading eigenvectors. Writing $\mathbf{H} = \mathbf{U}\,\mathrm{diag}(\lambda_1, \ldots, \lambda_n)\mathbf{U}^\top$, with $\lambda_1 \ge \cdots \ge \lambda_n \ge 0$, we use $\boldsymbol{\Phi}_d = [\mathbf{u}_1, \ldots, \mathbf{u}_d] \in \mathbb{R}^{n\times d}$. The following result separates choosing the basis from fitting its coefficients (Appendix A.1).

Theorem 1 (Basis and coefficient optimality)**.** *Fix $\mathbf{H} \succeq 0$ and $1 \le d \le n$, with $\lambda_{n+1} := 0$. (i) Let $\mathcal{R}_{\mathbf{H}}(\mathbf{V}) := \sup_{\|\mathbf{E}\|_F \le 1} \min_{\mathbf{A}} D_{\mathbf{H}}(\mathbf{E} - \mathbf{A}\mathbf{V}^\top)$ denote the worst-case output distortion after optimal coefficient fitting. Among transforms chosen using the sole information from $\mathbf{H}$, the leading basis satisfies*

$$\boldsymbol{\Phi}_d \in \arg\min_{\mathbf{V}\in\mathbb{R}^{n\times d}} \mathcal{R}_{\mathbf{H}}(\mathbf{V}), \qquad \mathcal{R}_{\mathbf{H}}(\boldsymbol{\Phi}_d) = \lambda_{d+1}. \tag{5}$$

(ii) For any $\mathbf{E}$, an optimal coefficient matrix in this basis is

$$\mathbf{A}^\star = \mathbf{E}\boldsymbol{\Phi}_d \in \arg\min_{\mathbf{A}} D_{\mathbf{H}}(\mathbf{E} - \mathbf{A}\boldsymbol{\Phi}_d^\top). \tag{6}$$

With the basis fixed, the subspace payload need only describe the payload matrix $\mathbf{A}^\star$. For decoded coefficients $\widehat{\mathbf{A}}$, diagonality of $\mathbf{U}^\top\mathbf{H}\mathbf{U}$ eliminates cross-terms and gives the exact decomposition

$$D_{\mathbf{H}}(\mathbf{E} - \widehat{\mathbf{A}}\boldsymbol{\Phi}_d^\top) = \underbrace{\sum_{i>d} \lambda_i \|\mathbf{E}\mathbf{u}_i\|_2^2}_{D_{\mathrm{omit}}:\ \text{omitted directions}} + \underbrace{\sum_{j\le d} \lambda_j \|\mathbf{A}^\star_{:,j} - \widehat{\mathbf{A}}_{:,j}\|_2^2}_{\text{coefficient coding error}}. \tag{7}$$

For a fixed d, only the second term can be reduced by coefficient bits. The subspace code quantizes each active column with a 2^{b_j}-level scalar K-means codebook. Its squared-error objective matches the column's contribution up to the constant weight λ_j.

To allocate bits before fitting the codebooks, we approximate this coding error using the high-resolution scalar-quantization law: cell widths scale as 2^{-b_j}, so squared errors scale as 2^{-2b_j} Bennett (1948); Panter & Dite (1951). Using a common shape constant $c > 0$ after normalizing each column by its root-mean-square magnitude gives $\lambda_j \|\mathbf{A}^\star_{:,j} - \widehat{\mathbf{A}}_{:,j}\|_2^2 \approx c S_j 2^{-2b_j}$, where

$$S_j := \lambda_j \|\mathbf{A}^\star_{:,j}\|_2^2 = \lambda_j \sum_{r=1}^{m} (A^\star_{rj})^2. \tag{8}$$

The server computes S_j from the activation eigenvalue and the unquantized coefficient column. Under the common-shape, high-resolution assumptions, Equation (7) gives the approximate allocation model $D_{\text{omit}} + \sum_j c S_j 2^{-2b_j}$ (Appendix A.2).

Theorem 2 (Optimal allocation under the distortion model)**.** *Let $c > 0$, $S_j \geq 0$, and let $B_{\text{idx}} \geq 0$ be the coefficient-index budget. Minimizing $\sum_{j=1}^{d} c S_j 2^{-2b_j}$ over real $b_j \geq 0$, subject to $m \sum_j b_j \leq B_{\text{idx}}$, gives*

$$b_j^*(\theta) = \max\left(0, \frac{1}{2}\log_2 \frac{S_j}{\theta}\right). \tag{9}$$

For positive budget and at least one positive signal, $\theta > 0$ is chosen so that $m \sum_j b_j^*(\theta) = B_{\text{idx}}$, with $b_j^* = 0$ for $S_j = 0$.

This reverse water-filling rule assigns more bits to larger activation-weighted coefficient energies and equalizes $S_j 2^{-2b_j}$ across active columns. The common constant c does not affect the allocation. The theorem solves the continuous approximation (Appendix A.3).

The transmitted codec must also account for integer rates and overhead. For nominal average precision b, the subspace code rounds and clips the rates to $\{0, \ldots, 2b\}$, then repairs them greedily to fit the complete B_{AS}-bit subspace allowance, nominally mdb bits. The charged cost is

$$\rho_{\text{AS}} = \rho_{\text{hdr}} + 4d + \sum_{j: b_j > 0} (m b_j + 16 \cdot 2^{b_j}) \quad \text{bits}, \tag{10}$$

counting the header, four-bit rate entries, coefficient indices, and FP16 codebooks. A zero-rate column is decoded as zero and needs no codebook. With the subspace code alone, the client reconstructs $\widehat{\mathbf{W}}' = \widehat{\mathbf{W}}_x + \widehat{\mathbf{A}}\widehat{\mathbf{\Phi}}_d^\top$. Calibration and codec settings follow the experimental protocol (Appendix C); we also evaluate endpoint agreement (Appendix D.5).

4.2 The complement code: extending the shared subspace

Equation (7) also explains why the subspace code can saturate: increasing coefficient precision cannot remove D_{omit}. Increasing d reduces this floor but leaves fewer bits per coefficient at a fixed payload. RECAP therefore spends part of the budget on a transmitted rank-k complement fitted to the residual remaining after decoding the subspace code:

$$\widehat{\mathbf{W}}' = \widehat{\mathbf{W}}_x + \widehat{\mathbf{A}}\widehat{\mathbf{\Phi}}_d^\top + \widehat{\mathbf{L}}\widehat{\mathbf{R}}^\top, \qquad \boxed{\rho_{\text{RECAP}}(d,k) \approx mdb + (m+n)kb_{\text{f}} \quad \text{bits}}, \tag{11}$$

Here b_{f} is the complement-factor precision; both factors, their FP16 row scales, and metadata are charged in the reported payload. The residual-dependent complement can expand the correction beyond the shared subspace while preserving one-shot decoding.

The server first reconstructs $\widehat{\mathbf{A}}$ from the coefficient codes, so complement fitting sees the residual left by the decoded subspace code, including its coefficient-rounding error. It fits this residual under the full-precision model's activation statistic $\mathbf{H}_{\text{fp}}$, which the client never needs. QERA's activation-weighted truncated SVD, from Zhang et al. (2025), initializes the factors; CALDERA-style rounding-aware alternating refits, from Saha et al. (2024), account for their low-bit representation (Appendix C.2).

To characterize the complement's expansion of the shared subspace, consider unquantized coefficients and factors evaluated under the same activation statistic $\mathbf{H}$. Let $\mathbf{E}_\perp = \mathbf{E}(\mathbf{I}_n - \mathbf{\Phi}_d\mathbf{\Phi}_d^\top)$ be the residual outside the shared subspace and $\sigma_1 \geq \sigma_2 \geq \cdots \geq 0$ the singular values of $\mathbf{E}_\perp \mathbf{H}^{1/2}$.

Theorem 3 (Optimal complement and joint distortion floor)**.** *Let $\mathbf{\Phi}_d$ hold the top-d eigenvectors of $\mathbf{H}$. Fix d and k. For every $\mathbf{A} \in \mathbb{R}^{m\times d}$ and every $\mathbf{LR}^\top$ of rank at most k,*

$$\boxed{D_{\mathbf{H}}\big(\mathbf{E} - \mathbf{A}\mathbf{\Phi}_d^\top - \mathbf{LR}^\top\big) \;\geq\; \sum_{i>k} \sigma_i^2,} \tag{12}$$

with equality attained at $\mathbf{A} = \mathbf{A}^\star$ and a complement satisfying $\mathbf{R}^\top\mathbf{\Phi}_d = \mathbf{0}$.

For this common-metric, unquantized problem, fitting the subspace coefficients first and the complement on the remainder attains the joint optimum without overlap between the two corrections (Appendix A.4). Each transmitted direction costs $(m+n)b_{\mathrm{f}}$ leading bits against the shared direction's mb, exposing the finite-rate tradeoff between inexpensive shared directions and more flexible transmitted ones (Appendix D.6). Algorithm 1 summarizes the complete two-code procedure (Appendix B).

5 Experiments

Setup. We evaluate Llama-2 (7B and 13B) and Qwen-2.5-Instruct (7B and 14B) using SINQ, HQQ, and GPTQ checkpoints (Appendix C). We compare refinement of 1.58-bit bases with same-family 2-bit re-transmission, and RECAP with LoRC, LQER, QERA, and CALDERA on 2- and 3-bit bases at matched payloads. We report WikiText-2 test perplexity and average accuracy across seven downstream tasks: ARC-Easy and ARC-Challenge Clark et al. (2018), PIQA Bisk et al. (2020), HellaSwag Zellers et al. (2019), WinoGrande Sakaguchi et al. (2020), BoolQ Clark et al. (2019), and OpenBookQA Mihaylov et al. (2018). We count complete transmitted payloads in decimal MB. Base rows transmit no payload; bold marks the best result and, where shown, underline marks the second best.

Refinement versus re-transmission. We test whether a compact residual can substitute for a complete replacement by comparing refinement of 1.58-bit bases with same-family 2-bit re-transmission. Figure 2 shows that across all four 7B bases, RECAP reaches replacement perplexity using 2–19% of its bytes. With 37–38%, it matches or exceeds replacement accuracy (Appendix D.2). It also reaches replacement perplexity on three of the four larger bases within the tested payload range (Table 6). Every observed WikiText-2 replacement crossing also occurs on held-out C4 (Table 10), extending the communication advantage beyond the calibration corpus. These results separate standalone quality from value as side information: even a poor checkpoint can substantially reduce the communication needed to reach replacement-level performance.

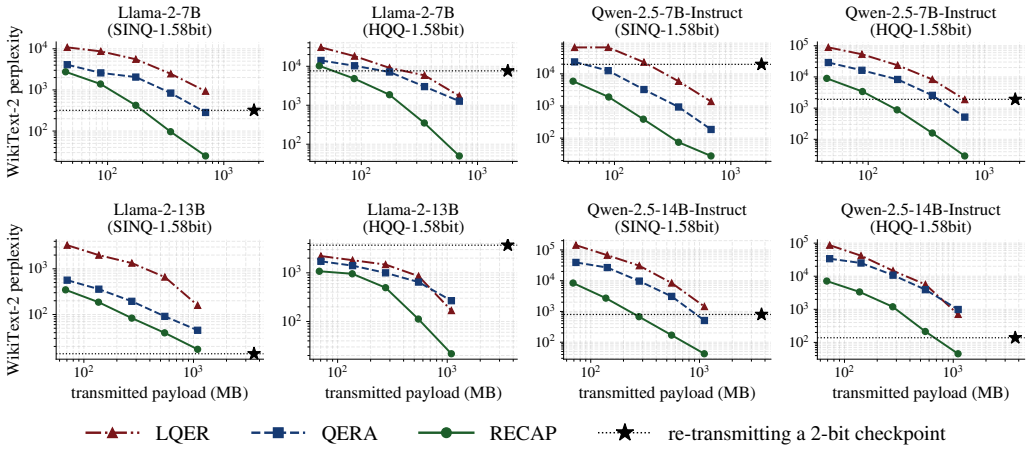


Figure 2: We compare RECAP, QERA and LQER in WikiText-2 perplexity versus payload when refining 1.58-bit checkpoints on 7B (top) and 13B-class (bottom) models. **RECAP can match or surpass same-family 2-bit re-transmission with far fewer bytes.**

Matched-payload reconstruction. At equal transmitted bytes, we compare RECAP with four reconstruction baselines using task accuracy (Table 1) and perplexity (Table 4 in Appendix D.1). RE-

CAP achieves the best or tied-best perplexity in 22 of 24 model–base settings, including all twelve 2-bit settings, and the best or tied-best accuracy in 18 of 24. Its largest gains occur on 2-bit SINQ and HQQ bases across both model families and size tiers; improvements over 3-bit bases are more modest. Together, these results show that more effective use of the communication budget improves task performance as well as language-model perplexity.

Table 1: We compare seven-task accuracy (↑, %) when refining 2- and 3-bit checkpoints at matched payloads. **RECAP is best or tied for best in 18 of 24 settings, with its largest gains on the 2-bit SINQ and HQQ bases.**

Tier	Method	Llama-2-7B 348 MB			Qwen-2.5-7B-Instruct 356 MB		
		SINQ	HQQ	GPTQ	SINQ	HQQ	GPTQ
2-bit	base	41.4	35.4	38.9	36.3	35.9	36.7
	LoRC	48.0	34.2	**40.7**	41.1	41.9	39.2
	LQER	50.3	50.4	40.4	48.0	50.8	39.8
	QERA	51.1	48.6	39.2	47.1	48.4	38.2
	CALDERA (scalar)	48.2	37.2	37.8	46.8	47.4	37.6
	RECAP (ours)	**54.0**	**53.8**	40.6	**53.7**	**54.7**	**40.4**
3-bit	base	64.1	60.4	62.2	64.8	67.3	65.2
	LoRC	64.3	61.6	**63.9**	66.6	68.5	67.5
	LQER	**64.8**	64.8	**63.9**	**68.6**	68.6	67.6
	QERA	**64.8**	64.7	62.5	68.1	68.7	66.3
	CALDERA (scalar)	64.3	63.9	62.7	67.3	68.0	67.3
	RECAP (ours)	**64.8**	**65.2**	**63.9**	68.1	**68.9**	**68.2**
Tier	Method	Llama-2-13B 545 MB			Qwen-2.5-14B-Instruct 554 MB		
		SINQ	HQQ	GPTQ	SINQ	HQQ	GPTQ
2-bit	base	43.8	35.7	44.2	37.9	38.9	36.9
	LoRC	48.3	38.3	47.7	43.3	42.2	40.5
	LQER	57.3	58.7	**47.9**	53.1	54.6	40.1
	QERA	57.0	56.6	45.2	50.0	53.1	38.8
	CALDERA (scalar)	54.9	47.5	44.2	49.7	49.8	38.8
	RECAP (ours)	**59.0**	**59.4**	47.4	**55.8**	**56.5**	**41.3**
3-bit	base	67.9	66.9	67.6	70.6	70.6	70.2
	LoRC	68.0	67.1	67.5	71.4	70.5	71.8
	LQER	68.3	68.4	67.8	**72.4**	72.1	**72.1**
	QERA	68.2	68.4	67.6	71.9	71.8	71.0
	CALDERA (scalar)	68.2	67.9	**67.9**	72.0	70.9	71.0
	RECAP (ours)	**68.4**	**68.5**	67.8	72.0	**72.3**	71.2

What each component contributes. To explain these gains, we compare the two codes separately and jointly, vary their budget split, and remove factor refitting at matched payload on the 7B models (348 MB for Llama and 356 MB for Qwen; Table 2). Combining the codes with refitting outperforms either alone on all four 1.58-bit bases on both WikiText-2 and C4; the default split is best in seven of the eight comparisons. Refitting improves every matched-split comparison. At 2 bits, the default combination leads on SINQ, while the complement alone leads on GPTQ (Appendix D.6). The results favor combining inexpensive shared directions with a flexible transmitted correction and fitting it around factor quantization, with the useful split depending on the base.

Activation-basis choice. The next three studies isolate the subspace code's basis, coverage, and coding. To test the benefit of activation alignment, we compare leading activation eigenvectors with random, coordinate, and residual-aware alternatives on 7B SINQ-2 and GPTQ-2 bases at fixed dimension and coding (Appendix C.2). In Figure 3, the activation eigenbasis gives the lowest perplexity in every setting, capturing more activation-weighted error than random subspaces and preserving more of that benefit after coding. Residual-aware selection adds indices without improving perplexity. Thus, activation alignment provides effective shared directions, without transmitting a residual-dependent basis choice.

Table 2: We compare RECAP's components and splits by perplexity on WikiText-2 and C4. **Both codes together beat either alone; the default split wins seven of eight comparisons.**

(a) WikiText-2	Llama-2-7B		Qwen-2.5-7B-Instruct	
	SINQ-1.58	HQQ-1.58	SINQ-1.58	HQQ-1.58
base, sends nothing	18552	25109	478807	1194833
subspace code alone	1314.2	2376.4	2243.8	1416.6
complement code alone	199.2	1217.4	110.5	190.7
RECAP ($d = 16$)	133.4	907.1	89.4	**152.6**
RECAP ($d = 64$)	116.6	449.1	83.7	156.1
RECAP ($d = 64$, without factor refitting)	266.0	567.0	164.9	369.1
RECAP ($d = 256$, default)	**96.5**	**350.8**	**74.2**	158.8
(b) C4	Llama-2-7B		Qwen-2.5-7B-Instruct	
	SINQ-1.58	HQQ-1.58	SINQ-1.58	HQQ-1.58
subspace code alone	653.6	1701.3	2714.6	1492.7
complement code alone	400.4	1643.6	480.6	815.4
RECAP ($d = 16$)	278.5	1121.0	352.8	570.3
RECAP ($d = 64$)	263.6	690.4	334.7	543.6
RECAP ($d = 64$, without factor refitting)	448.7	857.2	662.3	1171.6
RECAP ($d = 256$, default)	**209.4**	**545.7**	**286.6**	**526.3**

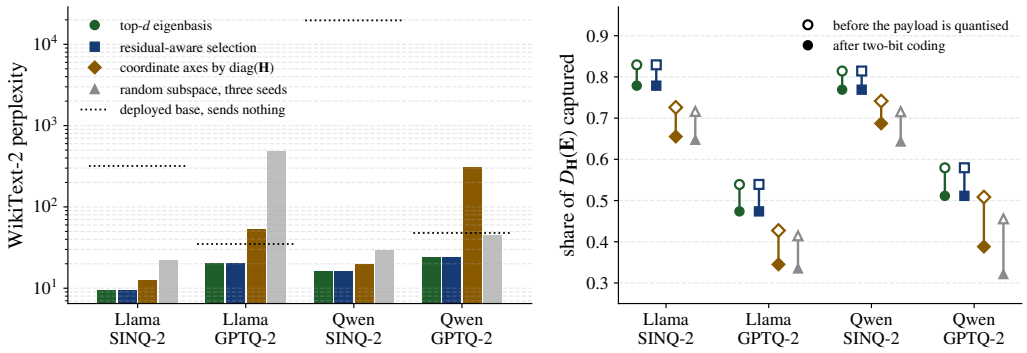


Figure 3: We compare bases by perplexity (left) and captured activation-weighted error before and after coding (right). **The activation eigenbasis is best without extra selection bits.**

Coverage versus precision. We examine why higher coefficient precision can worsen refinement at a fixed payload. On the four 7B SINQ-2 and GPTQ-2 bases, we vary the subspace code's dimension and coefficient precision at approximately fixed payload, repeating at half the budget (Appendix C.2). Figure 4 shows how higher precision reduces coefficient error but leaves more residual outside the shrinking subspace. Total distortion is lowest at two-bit coefficients on SINQ and one-bit coefficients on GPTQ at both budgets. Consistent with (7), broader, coarsely coded coverage can therefore be more effective than a narrow, precise correction.

Low-bit payload coding. We test which coefficient coder enables effective low-bit coverage. Across twelve 7B model–base settings, we match subspace-coder payloads by adjusting dimension (Table 3); RECAP is the combined-code reference. Two-bit per-row rounding can worsen perplexity, whereas learned codebooks with water-filling at an average of two bits never worsen the base across 48 base–payload points and outperform higher-precision coders at the largest payload. At fixed basis, dimension, and payload, codebooks provide most of the two-bit gain, with further gains from water-filling (Appendix D.3). Effective low-bit coverage thus combines distribution-aware codebooks with allocation by activation-weighted column energy.

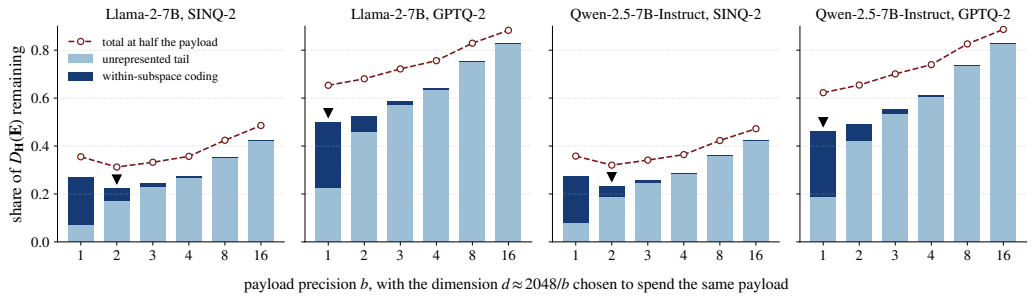


Figure 4: We split subspace-code distortion into omitted-direction and coefficient-coding errors at fixed payload. **Broader low-bit coverage can beat a narrower, higher-precision correction.**

Table 3: We compare matched-payload coders by perplexity (left) and wins and regressions (right). **Low-bit codebooks improve payload efficiency, and RECAP wins most settings.**

	Qwen-2.5-7B-Instruct, SINQ-2				12 model–base cells	
payload coder	payload (MB)				largest-payload wins/12	worse than base /48
	47	92	181	359		
Subspace code alone						
unquantized (b=16)	50.6	43.1	35.9	29.3	0	2
per-row rounding (b=4)	35.8	29.2	23.6	18.8	0	2
per-row rounding (b=2)	97.8	158.2	203.2	271.5	0	11
learned codebooks + water-filling (avg. b=2)	35.6	28.3	20.9	16.0	4	**0**
RECAP (ours)	**23.0**	**18.7**	**15.8**	**13.6**	**8**	**0**

The deployed base shapes the refinement frontier. To understand why similar-quality checkpoints respond differently, we compare Llama-2-13B's SINQ-2 and GPTQ-2 bases: they begin at similar perplexity and accuracy, yet the same payload produces much larger gains on SINQ (Tables 1 and 4). In the paired 7B comparisons, GPTQ also leaves less activation-weighted residual error in the shared subspace than SINQ (Appendix D.6). Checkpoint choice should therefore account for residual alignment, not only standalone quality.

6 Conclusion

RECAP shows that a quantized checkpoint can remain valuable as side information even when its standalone performance is poor, enabling replacement-level performance with a fraction of the communication. Its advantage comes from matching residual coding to activation geometry: shared directions provide inexpensive coverage, while a transmitted complement captures what they miss. Under a fixed payload, representing more useful directions at low precision can also outperform coding fewer directions precisely. These findings motivate designing quantization and refinement jointly: a checkpoint should be judged not only by its standalone quality, but also by how efficiently it can be improved. Successive refinement thus offers a communication-efficient alternative to replacing deployed checkpoints.

Limitations

RECAP targets one-shot refinement of fixed PTQ checkpoints using a shared public calibration protocol. Shared-basis decoding requires agreed numerical conventions, and communication savings entail local computation and storage costs that depend on the serving representation. Our evaluation focuses on dense language models and selected text benchmarks. Broader architectures and tasks, repeated refinement, and end-to-end deployment across heterogeneous devices remain future work.

AI USE STATEMENT

Generative AI tools were used to improve the readability of the manuscript and assist with limited code editing. All AI-assisted text and code were carefully reviewed and verified by the authors. Generative AI tools were not used to develop the research idea, design the methodology, interpret the experimental results, or derive the main scientific claims. The authors take full responsibility for the content and conclusions of this work.

REFERENCES

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

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, et al. Qwen technical report. arXiv preprint arXiv:2309.16609, 2023.

William R. Bennett. Spectra of quantized signals. *The Bell System Technical Journal*, 27(3):446–472, 1948. doi: 10.1002/j.1538-7305.1948.tb01340.x.

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 (AAAI)*, 2020.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019.

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.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley, 2nd edition, 2006.

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

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

Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

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.

William H. R. Equitz and Thomas M. Cover. Successive refinement of information. *IEEE Transactions on Information Theory*, 37(2):269–275, 1991.

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

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, July 2024. URL https://zenodo.org/records/12608602.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

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 *MLSys*, 2024.

James Liu, Guangxuan Xiao, Kai Li, Jason D. Lee, Song Han, Tri Dao, and Tianle Cai. BitDelta: Your fine-tune may only be worth one bit. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

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

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

Lorenz K. Müller, Philippe Bich, Jiawei Zhuang, Ahmet Çelik, Luca Benfenati, and Lukas Cavigelli. SINQ: Sinkhorn-normalized quantization for calibration-free low-precision LLM weights. In *International Conference on Machine Learning (ICML)*, 2026.

Philip F. Panter and William Dite. Quantization distortion in pulse-count modulation with nonuniform spacing of levels. *Proceedings of the IRE*, 39(1):44–48, 1951. doi: 10.1109/JRPROC.1951.230419.

Yeonsik Park, Hyeonseong Kim, and Seungkyu Choi. SERQ: Saliency-aware low-rank error reconstruction for LLM quantization. In *International Conference on Learning Representations (ICLR)*, 2026.

Bowen Ping, Shuo Wang, Hanqing Wang, Xu Han, Yuzhuang Xu, Yukun Yan, Yun Chen, Baobao Chang, Zhiyuan Liu, and Maosong Sun. Delta-CoMe: Training-free delta-compression with mixed-precision for large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

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

Rajarshi Saha, Naomi Sagan, Varun Srivastava, Andrea J. Goldsmith, and Mert Pilanci. Compressing large language models using low rank and low precision decomposition. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

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

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

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

Aaron D. Wyner and Jacob Ziv. The rate-distortion function for source coding with side information at the decoder. *IEEE Transactions on Information Theory*, 22(1):1–10, 1976.

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.

Xiaozhe Yao, Qinghao Hu, and Ana Klimovic. DeltaZip: Efficient serving of multiple full-model-tuned LLMs. In *Proceedings of the Twentieth European Conference on Computer Systems (EuroSys)*, 2025.

Zhewei Yao, Xiaoxia Wu, Cheng Li, Stephen Youn, and Yuxiong He. ZeroQuant-V2: Exploring post-training quantization in LLMs from comprehensive study to low rank compensation. *arXiv preprint arXiv:2303.08302*, 2023.

Wenya Yu, Chao Zhang, Li Wang, Samson Lasaulce, and Merouane Debbah. ProjQ: Project-and-quantize for adapter-aware LLM compression. In *International Conference on Machine Learning (ICML)*, 2026.

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.

Cheng Zhang, Jianyi Cheng, George A. Constantinides, and Yiren Zhao. LQER: Low-rank quantization error reconstruction for LLMs. In *International Conference on Machine Learning (ICML)*, 2024.

Cheng Zhang, Jeffrey T. H. Wong, Can Xiao, George A. Constantinides, and Yiren Zhao. QERA: An analytical framework for quantization error reconstruction. In *International Conference on Learning Representations (ICLR)*, 2025.

CONTENTS

1 Introduction **2**

2 Related Work **3**

3 Problem **3**

4 Method **4**
4.1 RECAP's activation-subspace code 4
4.2 The complement code: extending the shared subspace 5

5 Experiments **6**

6 Conclusion **9**

A Proofs **14**
A.1 Proof and scope of Theorem 1 14
A.2 From scalar quantization error to the allocation signal 15
A.3 Proof of Theorem 2 16
A.4 Proof of Theorem 3 17

B RECAP Algorithm **18**

C Experimental Setup **19**
C.1 Models and quantized checkpoints 19
C.2 RECAP configuration 19
C.3 Baseline configurations 20
C.4 Calibration and evaluation 20
C.5 Payload accounting 20

D Ablation Studies and Additional Results **20**
D.1 Matched-payload perplexity comparison 21
D.2 Subspace code alone versus RECAP 21
D.3 Coefficient coding and rate allocation 22
D.4 Calibration and cross-corpus transfer 23
D.5 Basis agreement between endpoints 24
D.6 RECAP's components and budget split 25

A PROOFS

Theorem 1 concerns a basis selected before observing the residual and the optimal unquantized coefficients within it. Theorem 2 concerns optimal continuous rates after adopting a scalar quantization approximation. We prove the first result, derive that approximation, and then prove the allocation rule. We then prove the joint distortion floor of Theorem 3.

A.1 PROOF AND SCOPE OF THEOREM 1

The transform is chosen with $\mathbf{H}$ fixed and before observing $\mathbf{E}$. The encoder subsequently observes $\mathbf{E}$ and optimally fits $\mathbf{A}$. The worst-case distortion in Theorem 1(i) is

$$\mathcal{R}_{\mathbf{H}}(\mathbf{V}) := \sup_{\|\mathbf{E}\|_F \le 1} \min_{\mathbf{A}\in\mathbb{R}^{m\times d}} D_{\mathbf{H}}(\mathbf{E} - \mathbf{A}\mathbf{V}^\top). \tag{13}$$

We show that

$$\min_{\mathbf{V}\in\mathbb{R}^{n\times d}} \mathcal{R}_{\mathbf{H}}(\mathbf{V}) = \mathcal{R}_{\mathbf{H}}(\mathbf{\Phi}_d) = \lambda_{d+1}, \tag{14}$$

where $\lambda_{n+1} := 0$. The comparison permits nonorthogonal and rank-deficient transforms; the information restriction is the order of the decisions. The second claim is the coefficient optimum in (6).

Proof. Expansion in the activation eigenbasis gives

$$D_{\mathbf{H}}(\mathbf{E} - \mathbf{A}\mathbf{\Phi}_d^\top) = \sum_{i>d} \lambda_i \|\mathbf{E}\mathbf{u}_i\|_2^2 + \sum_{j\le d} \lambda_j \|\mathbf{E}\mathbf{u}_j - \mathbf{A}_{:,j}\|_2^2. \tag{15}$$

The second term is nonnegative and vanishes at $\mathbf{A}^\star = \mathbf{E}\mathbf{\Phi}_d$, proving coefficient optimality for every $\mathbf{E}$. Since $\lambda_i \le \lambda_{d+1}$ for $i > d$, the resulting error satisfies

$$\min_{\mathbf{A}} D_{\mathbf{H}}(\mathbf{E} - \mathbf{A}\mathbf{\Phi}_d^\top) = \sum_{i>d} \lambda_i \|\mathbf{E}\mathbf{u}_i\|_2^2 \le \lambda_{d+1}\|\mathbf{E}\|_F^2. \tag{16}$$

Thus the worst-case error over $\|\mathbf{E}\|_F \le 1$ is at most λ_{d+1}. For $d < n$, the residual $\mathbf{E} = \mathbf{e}_1\mathbf{u}_{d+1}^\top$ attains this value, where $\mathbf{e}_1$ is a unit coordinate vector in $\mathbb{R}^m$.

For the matching lower bound when $d < n$, take any fixed transform $\mathbf{V}$ and let $\mathbf{U}_{d+1} = [\mathbf{u}_1, \dots, \mathbf{u}_{d+1}]$. The $d \times (d+1)$ matrix $\mathbf{V}^\top\mathbf{H}\mathbf{U}_{d+1}$ has a nonzero null vector. Hence a unit $\mathbf{q} \in \operatorname{span}(\mathbf{U}_{d+1})$ exists with $\mathbf{V}^\top\mathbf{H}\mathbf{q} = 0$. For $\mathbf{E} = \mathbf{e}_1\mathbf{q}^\top$,

$$\begin{aligned} D_{\mathbf{H}}(\mathbf{E} - \mathbf{A}\mathbf{V}^\top) &= \mathbf{q}^\top\mathbf{H}\mathbf{q} + \operatorname{tr}(\mathbf{A}\mathbf{V}^\top\mathbf{H}\mathbf{V}\mathbf{A}^\top) \\ &\ge \mathbf{q}^\top\mathbf{H}\mathbf{q} \ge \lambda_{d+1}. \end{aligned} \tag{17}$$

The cross-term vanishes because $\mathbf{E}\mathbf{H}\mathbf{V} = 0$, and $\mathbf{A} = 0$ achieves the first inequality. Every fixed transform therefore has worst-case distortion at least λ_{d+1}, matching (16). For $d = n$, the full basis represents $\mathbf{E}$ exactly, giving zero distortion as specified by $\lambda_{n+1} = 0$. □

By homogeneity, replacing the unit ball by $\|\mathbf{E}\|_F \le r$ changes the minimax value to $r^2\lambda_{d+1}$ for any $r > 0$, without changing the optimal basis. The radius is not needed to construct or decode the basis. Without normalization or an energy bound, worst-case distortion is unbounded whenever $d < \operatorname{rank}(\mathbf{H})$. The result includes singular $\mathbf{H}$: retaining every direction with positive eigenvalue gives zero calibration distortion. Tied eigenvalues can yield multiple optimal bases; coefficients in zero-eigenvalue directions need not be unique.

This guarantee uses an uncertainty set allowing all residual orientations. It is a conservative design criterion based only on $\mathbf{H}$, not an assertion that this basis is best for each individual residual or for a narrower uncertainty set imposed by a specific quantizer. A residual-dependent basis can do better for a particular $\mathbf{E}$, but the client would need additional information to recover it. The theorem concerns unquantized coefficients; finite-bit coding is handled by the separate allocation model. Substituting $\mathbf{A} = \widehat{\mathbf{A}}$ in (15) proves the exact decomposition (7), independently of that model.

For unquantized coefficients, Theorem 1(ii) and (7) give the exact subspace floor. Writing $\mathbf{c}_i = \mathbf{E}\mathbf{u}_i$,

$$\boxed{\min_{\mathbf{A}} D_{\mathbf{H}}\big(\mathbf{E} - \mathbf{A}\mathbf{\Phi}_d^\top\big) \;=\; \sum_{i=d+1}^{n} \lambda_i \,\|\mathbf{c}_i\|_2^2.} \tag{18}$$

Remark (general orthonormal bases). For an arbitrary orthonormal $\mathbf{\Phi}_d$ whose span is *not* $\mathbf{H}$-invariant, the cross-term between the in-subspace and omitted components need not vanish. One $D_{\mathbf{H}}$-optimal coefficient is the $\mathbf{H}$-weighted fit

$$\mathbf{A}^\star = \mathbf{E}\mathbf{H}\mathbf{\Phi}_d(\mathbf{\Phi}_d^\top \mathbf{H}\mathbf{\Phi}_d)^\dagger. \tag{19}$$

The resulting residual can have strictly smaller $D_{\mathbf{H}}$ than the Euclidean residual $\mathbf{E}(\mathbf{I}_n - \mathbf{\Phi}_d\mathbf{\Phi}_d^\top)$. The general floor is

$$\min_{\widehat{\mathbf{A}}} D_{\mathbf{H}}\big(\mathbf{E} - \widehat{\mathbf{A}}\mathbf{\Phi}_d^\top\big) = \mathrm{tr}(\mathbf{E}\mathbf{H}\mathbf{E}^\top) - \mathrm{tr}\big(\mathbf{E}\mathbf{H}\mathbf{\Phi}_d(\mathbf{\Phi}_d^\top \mathbf{H}\mathbf{\Phi}_d)^\dagger \mathbf{\Phi}_d^\top \mathbf{H}\mathbf{E}^\top\big),$$

which reduces to (18) for the top-d eigenbasis. This distinguishes fitting coefficients in a fixed basis from choosing the transform under the minimax criterion of Theorem 1(i).

Proposition 1 (Within-subspace quantization noise)**.** *Let $\mathbf{\Phi}_d$ hold the top-d eigenvectors of $\mathbf{H}$, set $\mathbf{A}^\star = \mathbf{E}\mathbf{\Phi}_d$, and let $\mathbf{N} = \widehat{\mathbf{A}} - \mathbf{A}^\star$ be the payload quantization error, and let $\mathbf{n}_j$ be its j-th column. Then*

$$D_{\mathbf{H}}\big(\mathbf{E} - \widehat{\mathbf{A}}\mathbf{\Phi}_d^\top\big) \;=\; \underbrace{\sum_{i=d+1}^{n} \lambda_i \|\mathbf{c}_i\|_2^2}_{\textit{subspace floor}} \;+\; \underbrace{\sum_{j=1}^{d} \lambda_j \|\mathbf{n}_j\|_2^2}_{\textit{quantizer noise}}. \tag{20}$$

The identity holds for any decoded coefficients, independently of the quantizer or its rate.

Proof. Write $\widehat{\mathbf{A}} = \mathbf{A}^\star + \mathbf{N}$. Since $\mathbf{A}^\star = \mathbf{E}\mathbf{\Phi}_d$ attains the floor, $\mathbf{E} - \widehat{\mathbf{A}}\mathbf{\Phi}_d^\top = \mathbf{E}_\perp - \mathbf{N}\mathbf{\Phi}_d^\top$, whose two parts are supported on orthogonal, $\mathbf{H}$-invariant subspaces, so the cross-term vanishes and the distortions add, as in (15). The first is $D_{\mathbf{H}}(\mathbf{E}_\perp) = \sum_{i>d} \lambda_i \|\mathbf{c}_i\|_2^2$; the second is $D_{\mathbf{H}}(\mathbf{N}\mathbf{\Phi}_d^\top) = \mathrm{tr}(\mathbf{N}\mathbf{\Phi}_d^\top \mathbf{H}\mathbf{\Phi}_d\mathbf{N}^\top) = \sum_{j\le d} \lambda_j \|\mathbf{n}_j\|_2^2$ because $\mathbf{\Phi}_d^\top \mathbf{H}\mathbf{\Phi}_d = \mathrm{diag}(\lambda_1, \dots, \lambda_d)$. □

A.2 From scalar quantization error to the allocation signal

Fix an active payload column with nonzero energy and define its squared root-mean-square scale

$$s_j^2 := \frac{1}{m}\|\mathbf{A}^\star_{:,j}\|_2^2. \tag{21}$$

This is a second moment, not a centered variance; no zero-mean assumption is needed. To model a scalar codebook, treat the column entries as samples from a smooth density f_j. The following is the high-resolution approximation for fixed-rate nonuniform scalar quantization, rather than an exact law for a finite list of coefficients Bennett (1948); Panter & Dite (1951).

Let $K = 2^{b_j}$ and let $\nu_j(y)$ be the normalized density of reconstruction levels, with $\int \nu_j(y)\, dy = 1$. At high resolution, a cell near y has width approximately $1/(K\nu_j(y))$. Approximating the source density as locally constant and placing the reconstruction level near the cell center gives

$$\mathbb{E}(Y - Q(Y))^2 \approx \frac{1}{12K^2} \int \frac{f_j(y)}{\nu_j(y)^2}\, dy. \tag{22}$$

This step assumes sufficiently small cells in the relevant density region and negligible unmodeled overload error. Minimizing the integral subject to $\int \nu_j = 1$ gives $\nu_j(y) = f_j(y)^{1/3} / \int f_j(t)^{1/3}\, dt$ and therefore

$$\mathbb{E}(Y - Q(Y))^2 \approx \frac{2^{-2b_j}}{12} \left(\int f_j(y)^{1/3}\, dy \right)^3. \tag{23}$$

The integral must be finite. K-means fits a finite-data squared-error codebook, whose learned levels approximate this nonuniform allocation when the high-resolution model is appropriate; no global optimum of a particular K-means run is assumed.

Write $f_j(y) = s_j^{-1} g_j(y/s_j)$, where g_j is the density after RMS normalization. Changing variables in (23) isolates the scale:

$$\mathbb{E}(Y - Q(Y))^2 \approx \kappa_j s_j^2 2^{-2b_j}, \qquad \kappa_j := \frac{1}{12} \left(\int g_j(z)^{1/3}\, dz \right)^3. \tag{24}$$

Summing over the m entries and weighting by λ_j yields

$$\lambda_j \|\mathbf{A}^\star_{:,j} - \widehat{\mathbf{A}}_{:,j}\|_2^2 \approx \kappa_j \lambda_j m s_j^2 2^{-2b_j} = \kappa_j S_j 2^{-2b_j}, \qquad S_j = \lambda_j \|\mathbf{A}^\star_{:,j}\|_2^2. \tag{25}$$

Thus S_j combines activation sensitivity with the actual coefficient energy. It is computed at the server after forming $\mathbf{A}^\star = \mathbf{E}\mathbf{\Phi}_d$, by summing its squared entries columnwise and multiplying by the corresponding eigenvalues. No new calibration statistic is required.

The subspace code replaces the potentially different shape constants κ_j by a common $c > 0$. This is an additional modeling approximation, appropriate to the extent that RMS-normalized coefficient distributions have similar shapes. It gives

$$D_{\mathbf{H}}(\mathbf{E} - \widehat{\mathbf{A}}\mathbf{\Phi}_d^\top) \approx \sum_{i>d} \lambda_i \|\mathbf{E}\mathbf{u}_i\|_2^2 + \sum_{j=1}^{d} c S_j 2^{-2b_j}. \tag{26}$$

The first term is constant during allocation. Since a common positive c only scales the rate-dependent objective, its numerical value need not be estimated to determine the rates. If shape differences were modeled explicitly, the allocation signal would instead be $\kappa_j S_j$.

At the one- to four-bit rates studied here, this approximation serves as an allocation model, not an exact error prediction or a universal upper bound. In particular, a pruned zero-rate column has exact distortion S_j, whereas the extrapolated model gives cS_j. Quantizer fitting, FP16 centroid rounding, and finite sample size can also affect the actual error. Equation (7) remains exact for the decoded coefficients regardless of these approximations.

A.3 Proof of Theorem 2

Proof. For $S_j > 0$, the second derivative of $cS_j 2^{-2b_j}$ is $(2\ln 2)^2 cS_j 2^{-2b_j} > 0$; zero-signal terms are constant. Thus the allocation problem in Theorem 2 is convex. For positive budget and at least one positive signal, an optimum uses the entire index budget, because increasing a positive-signal rate strictly decreases the objective.

Let $\mu \geq 0$ multiply the budget constraint and $\alpha_j \geq 0$ the constraint $-b_j \leq 0$. The Lagrangian is

$$\mathcal{J}(\mathbf{b}, \mu, \boldsymbol{\alpha}) = \sum_j cS_j 2^{-2b_j} + \mu \left(m \sum_j b_j - B_{\mathrm{idx}} \right) - \sum_j \alpha_j b_j. \tag{27}$$

For an active rate $b_j > 0$, complementary slackness gives $\alpha_j = 0$, and stationarity gives

$$-2\ln(2) cS_j 2^{-2b_j} + \mu m = 0, \qquad S_j 2^{-2b_j} = \theta := \frac{\mu m}{2c \ln 2}. \tag{28}$$

Consequently $b_j = \frac{1}{2}\log_2(S_j/\theta)$. For $b_j = 0$, stationarity and $\alpha_j \geq 0$ require $S_j \leq \theta$. Together these conditions give (9), setting the rate to zero when $S_j = 0$. The budget equation determines θ; its left-hand side is continuous and strictly decreasing over the positive-budget range. These KKT conditions are sufficient by convexity. Zero budget forces all rates to zero, and if all signals are zero then zero rates are one optimum. □

The solution is classical reverse water-filling for the stated distortion model Cover & Thomas (2006). If a continuous upper cap $b_{\max}$ is imposed, the same KKT argument gives

$$b_j^*(\theta) = \min\left\{ b_{\max}, \max\left(0, \frac{1}{2}\log_2 \frac{S_j}{\theta}\right)\right\}, \tag{29}$$

with the water level adjusted to the available budget up to the total capacity of the positive-signal columns. A zero-signal column can always be assigned zero bits. At full capacity all positive-signal columns are at the cap, and unused budget has no further benefit in this model.

Theorem 2 addresses real-valued rates and coefficient indices only. The implemented codec additionally uses integer rates, FP16 codebooks, and rate metadata. In particular, the codebook cost depends on the selected rates, so the complete constraint is not just $m\sum_j b_j \leq B_{\mathrm{idx}}$. The subspace code uses the continuous solution to initialize its rate decisions, then clips, rounds, and repairs them against the full payload in (10). The theorem does not claim that this discrete repair globally optimizes the complete codec.

A.4 Proof of Theorem 3

Throughout, $\mathbf{P} = \mathbf{\Phi}_d\mathbf{\Phi}_d^\top$ is the projector onto the shared subspace, $\mathbf{H}^{1/2} = \sum_i \lambda_i^{1/2}\mathbf{u}_i\mathbf{u}_i^\top$, and $\mathbf{H}^{-1/2} = \sum_{\lambda_i>0} \lambda_i^{-1/2}\mathbf{u}_i\mathbf{u}_i^\top$ is the pseudo-inverse square root, so that $\mathbf{\Pi} = \mathbf{H}^{-1/2}\mathbf{H}^{1/2}$ is the orthogonal projector onto $\mathrm{range}(\mathbf{H})$. All of them commute with $\mathbf{P}$, because they share the eigenvectors $\mathbf{u}_i$. Let $\mathbf{E}_\perp\mathbf{H}^{1/2} = \sum_i \sigma_i\mathbf{p}_i\mathbf{q}_i^\top$ be a singular value decomposition, with $\mathbf{M}_k = \sum_{i\le k}\sigma_i\mathbf{p}_i\mathbf{q}_i^\top$ its rank-k truncation; columns with $\sigma_i = 0$ may be dropped.

Proof. Step 1: The distortion splits over the two eigenspaces. For any $\mathbf{X} \in \mathbb{R}^{m\times n}$, $D_\mathbf{H}(\mathbf{X}) = \|\mathbf{X}\mathbf{H}^{1/2}\|_F^2 = \|\mathbf{X}\mathbf{H}^{1/2}\mathbf{P}\|_F^2 + \|\mathbf{X}\mathbf{H}^{1/2}(\mathbf{I}_n - \mathbf{P})\|_F^2$, because $\mathbf{P}$ and $\mathbf{I}_n - \mathbf{P}$ are complementary orthogonal projectors; by commutation the two terms are $D_\mathbf{H}(\mathbf{X}\mathbf{P})$ and $D_\mathbf{H}(\mathbf{X}(\mathbf{I}_n - \mathbf{P}))$. This is the vanishing cross-term underlying (15). Apply it to $\mathbf{X} = \mathbf{E} - \mathbf{A}\mathbf{\Phi}_d^\top - \mathbf{L}\mathbf{R}^\top$. Since $\mathbf{\Phi}_d^\top(\mathbf{I}_n - \mathbf{P}) = \mathbf{0}$, the coefficients drop out of the second term:

$$D_\mathbf{H}(\mathbf{X}) \;=\; D_\mathbf{H}\big((\mathbf{E} - \mathbf{L}\mathbf{R}^\top)\mathbf{P} - \mathbf{A}\mathbf{\Phi}_d^\top\big) \;+\; \big\|\mathbf{E}_\perp\mathbf{H}^{1/2} - \mathbf{L}\mathbf{R}^\top(\mathbf{I}_n - \mathbf{P})\mathbf{H}^{1/2}\big\|_F^2. \tag{30}$$

Step 2: The coefficients absorb the in-subspace term. The first term of (30) is non-negative and vanishes at $\mathbf{A} = (\mathbf{E} - \mathbf{L}\mathbf{R}^\top)\mathbf{\Phi}_d$, by Theorem 1(ii) applied to the residual $\mathbf{E} - \mathbf{L}\mathbf{R}^\top$. Hence $D_\mathbf{H}(\mathbf{X}) \ge \|\mathbf{E}_\perp\mathbf{H}^{1/2} - \mathbf{Z}\|_F^2$ with $\mathbf{Z} = \mathbf{L}\mathbf{R}^\top(\mathbf{I}_n - \mathbf{P})\mathbf{H}^{1/2}$, a matrix of rank at most k.

Step 3: Eckart–Young in the whitened metric. Over all matrices of rank at most k, $\|\mathbf{E}_\perp\mathbf{H}^{1/2} - \mathbf{Z}\|_F^2$ is minimised by $\mathbf{Z} = \mathbf{M}_k$, with value $\sum_{i>k}\sigma_i^2$ Eckart & Young (1936). This is the inequality in (12); the original subspace floor is $\sum_i \sigma_i^2 = \|\mathbf{E}_\perp\mathbf{H}^{1/2}\|_F^2 = D_\mathbf{H}(\mathbf{E}_\perp) = \sum_{i>d}\lambda_i\|\mathbf{c}_i\|_2^2$, as in (18).

Step 4: Attainment. The right singular vectors $\mathbf{q}_i$ with $\sigma_i > 0$ span the row space of $\mathbf{E}_\perp\mathbf{H}^{1/2} = \mathbf{E}\mathbf{H}^{1/2}(\mathbf{I}_n - \mathbf{P})$, which lies in $\mathrm{range}(\mathbf{I}_n - \mathbf{P}) \cap \mathrm{range}(\mathbf{H})$; there both $\mathbf{I}_n - \mathbf{P}$ and $\mathbf{\Pi}$ act as the identity, and $\mathbf{\Phi}_d^\top\mathbf{q}_i = \mathbf{0}$. Take $\mathbf{L} = [\sigma_1\mathbf{p}_1, \dots, \sigma_k\mathbf{p}_k]$ and $\mathbf{R} = \mathbf{H}^{-1/2}[\mathbf{q}_1, \dots, \mathbf{q}_k]$, so that $\mathbf{L}\mathbf{R}^\top = \mathbf{M}_k\mathbf{H}^{-1/2}$. Then $\mathbf{Z} = \mathbf{M}_k\mathbf{H}^{-1/2}(\mathbf{I}_n - \mathbf{P})\mathbf{H}^{1/2} = \mathbf{M}_k(\mathbf{I}_n - \mathbf{P})\mathbf{\Pi} = \mathbf{M}_k$, so the second term of (30) equals $\sum_{i>k}\sigma_i^2$. Moreover $\mathbf{R}^\top\mathbf{\Phi}_d = [\mathbf{q}_1, \dots, \mathbf{q}_k]^\top\mathbf{H}^{-1/2}\mathbf{\Phi}_d = [\mathbf{q}_1, \dots, \mathbf{q}_k]^\top\mathbf{\Phi}_d\mathbf{\Lambda}_d^{-1/2} = \mathbf{0}$, with $\mathbf{\Lambda}_d^{-1/2}$ under the pseudo-inverse convention, so $\mathbf{L}\mathbf{R}^\top\mathbf{\Phi}_d = \mathbf{0}$ and the minimising coefficients of Step 2 reduce to $\mathbf{A} = \mathbf{E}\mathbf{\Phi}_d = \mathbf{A}^\star$, at which the first term vanishes. The bound is attained. □

Consequences. (i) *Sequential optimality.* Given $\mathbf{A} = \mathbf{A}^\star$, the complement faces $\min D_\mathbf{H}(\mathbf{E}_\perp - \mathbf{L}\mathbf{R}^\top) = \min\|\mathbf{E}_\perp\mathbf{H}^{1/2} - \mathbf{L}\mathbf{R}^\top\mathbf{H}^{1/2}\|_F^2$ over rank at most k, which Step 3 bounds below by $\sum_{i>k}\sigma_i^2$ and $\mathbf{M}_k\mathbf{H}^{-1/2}$ attains, because $\mathbf{M}_k\mathbf{\Pi} = \mathbf{M}_k$. Fitting the coefficients first and the complement on the leftover therefore reaches the joint optimum, and the complement never needs to revisit the coefficients. (ii) *No direction is paid for twice.* Step 4 gives $\mathbf{R}^\top\mathbf{\Phi}_d = \mathbf{0}$: the minimising complement is supported on $\mathrm{range}(\mathbf{I}_n - \mathbf{P})$, the directions the shared basis cannot express. (iii) *A transmitted rank is worth at least a shared direction.* Extending the subspace code to the next k eigenvectors, with optimal coefficients, adds the rank-k correction $\mathbf{N} = \sum_{i=d+1}^{d+k}\mathbf{c}_i\mathbf{u}_i^\top$ and reaches $D_\mathbf{H}(\mathbf{E}_\perp - \mathbf{N}) = \sum_{i>d+k}\lambda_i\|\mathbf{c}_i\|_2^2$. Since $\mathbf{N}\mathbf{H}^{1/2}$ has rank at most k, Step 3 gives $\sum_{i>d+k}\lambda_i\|\mathbf{c}_i\|_2^2 \ge \sum_{i>k}\sigma_i^2$, that is, $\sum_{i\le k}\sigma_i^2 \ge \sum_{i=d+1}^{d+k}\lambda_i\|\mathbf{c}_i\|_2^2$: the complement's floor with k ranks is at most the subspace floor with $d + k$ directions. The complement adapts its directions to the residual, which is why they must be transmitted, at $(b_\mathrm{f}/b)(1 + n/m)$ times the price of a shared direction: four for a square projection, and between about 2.4 and 16 across the projections we refine. The relevant distortion reductions are the squared singular values σ_i^2 and the shared-direction energies $\lambda_i\|\mathbf{c}_i\|_2^2$. Comparing their reduction per leading payload bit explains why both types of direction can be useful; the quarter-budget configuration in Appendix C.2 is selected empirically. (iv) *The two ablation arms.* With $k = 0$, (12) is (18). With $d = 0$, $\mathbf{E}_\perp = \mathbf{E}$ and the floor is the tail of the singular values of $\mathbf{E}\mathbf{H}^{1/2}$, attained by the activation-weighted truncated singular value decomposition, the closed form QERA uses Zhang et al. (2025), from which the complement code alone starts. For every d and k the joint floor lies below both, because multiplying $\mathbf{E}\mathbf{H}^{1/2}$ by the projector $\mathbf{I}_n - \mathbf{P}$ cannot increase any of its singular values.

Finite-rate realization and calibration. Theorem 3 characterizes the unquantized optimum under a common activation statistic. In the codec, complement fitting takes the decoded remainder $\mathbf{E}_\perp -$

$\mathbf{N}\mathbf{\Phi}_d^\top$, where $\mathbf{N}$ is the coefficient noise of Proposition 1. It can therefore correct both omitted directions and coefficient-coding error. The alternating refits in Appendix C.2 account for factor rounding, and Appendix D.6 measures the resulting quantized correction.

RECAP derives the shared basis from the deployed checkpoint's statistic $\mathbf{H}$ and fits the complement under the server-only statistic $\mathbf{H}_{\mathrm{fp}}$. If the shared subspace is also invariant under $\mathbf{H}_{\mathrm{fp}}$, the unquantized argument applies in that metric with its corresponding singular values. Otherwise, the general weighted coefficient fit in (19) replaces $\mathbf{E}\mathbf{\Phi}_d$ in that metric. The common-metric joint floor is thus a structural benchmark for the two codes, while their implemented fit uses the separate statistics described in Section 4.2.

B RECAP ALGORITHM

Algorithm 1 refines one weight matrix. Both endpoints hold the cached checkpoint $\widehat{\mathcal{W}}_x$ and public protocol $\mathcal{P}$; only the server has the full-precision reference. Appendix C.2 specifies the input dimensions, precisions, and payload allowances.

Both endpoints collect layer inputs from public tokens passed through the cached checkpoint, form their second moment $\mathbf{H}$, and construct $\mathbf{\Phi}_d$ from its leading d eigenvectors under $\mathcal{P}$'s solver and eigenpair conventions. The shared basis therefore requires no transmission.

The server projects the quantization residual $\mathbf{E}$ into this basis and codes the resulting coefficients with water-filled bits, including codebook and metadata costs within B_{AS}. It decodes this payload before fitting a quantized low-rank complement to the remainder, using the server-only full-precision statistic $\mathbf{H}_{\mathrm{fp}}$ and rounding-aware refits. The complement thus targets both omitted directions and coefficient-coding error.

The message contains coefficient indices, codebooks, and rates, plus coded complement factors, scales, and metadata. The client decodes both corrections and adds them to its cached weights as in (11). Full-precision weights, activation statistics, and the shared basis are not transmitted.

Algorithm 1 RECAP: server-side encoder and client-side decoder for one matrix

Require: server target $\mathbf{W} \in \mathbb{R}^{m\times n}$; cached checkpoint $\widehat{\mathcal{W}}_x$ shared by both ends and containing the deployed counterpart $\widehat{\mathbf{W}}_x \in \mathbb{R}^{m\times n}$; subspace dimension d and rate b, full subspace allowance B_{AS}, complement rank k and factor precision b_{f}; server-only activation statistic $\mathbf{H}_{\mathrm{fp}}$ calibrated on the full-precision checkpoint; fixed protocol $\mathcal{P}$ (tokens, accumulation, and eigenpair conventions).

Setup (once per deployment, at both ends)

$\{\mathbf{a}_t \in \mathbb{R}^n\}_{t=1}^T \leftarrow \mathrm{CalibrateInputs}(\widehat{\mathcal{W}}_x, \widehat{\mathbf{W}}_x; \mathcal{P})$ ▷ inputs to the target map in the cached checkpoint

$\mathbf{H} \leftarrow T^{-1}\sum_t \mathbf{a}_t\mathbf{a}_t^\top$

$(\mathbf{U}, \boldsymbol{\lambda}) \leftarrow \mathrm{Eigh}_{\mathcal{P}}(\mathbf{H}), \quad \lambda_1 \geq \cdots \geq \lambda_n$ ▷ solver, signs, and ties fixed by $\mathcal{P}$

$\mathbf{\Phi}_d \leftarrow \mathbf{U}_{:,\,1:d}$ ▷ identical at both ends; zero transmitted bits

Server: encode

$\mathbf{E} \leftarrow \mathbf{W} - \widehat{\mathbf{W}}_x; \quad \mathbf{A}^\star \leftarrow \mathbf{E}\mathbf{\Phi}_d$

$S_j \leftarrow \lambda_j \|\mathbf{A}^\star_{:,j}\|_2^2, \quad j = 1, \ldots, d$

$\{b_j\} \leftarrow \mathrm{IntegerWaterFill}(\{S_j\}_{j\leq d}, B_{\mathrm{AS}}; \{0, \ldots, 2b\})$ ▷ reserve metadata/codebook bits; repair to cost $\leq B_{\mathrm{AS}}$

$(\mathbf{z}_j, \mathcal{C}_j) \leftarrow \mathrm{KMeans}(\mathbf{A}^\star_{:,j},\, 2^{b_j})$ for each j with $b_j > 0$

store $\mathcal{C}_j$ in FP16; reconstruct $\widehat{\mathbf{A}}_{:,j} \leftarrow \mathcal{C}_j[\mathbf{z}_j]$, or $\mathbf{0}$ if $b_j = 0$

$(\widehat{\mathbf{L}}, \widehat{\mathbf{R}}) \leftarrow \mathrm{FitComplement}_{b_{\mathrm{f}}}(\mathbf{E} - \widehat{\mathbf{A}}\mathbf{\Phi}_d^\top,\, \mathbf{H}_{\mathrm{fp}},\, k)$ ▷ weighted rank-k initialization; rounding-aware refits

transmit $\mathcal{M} = (d, \{b_j\}, \{\mathbf{z}_j\}, \{\mathcal{C}_j\}, \widehat{\mathbf{L}}, \widehat{\mathbf{R}})$ ▷ coefficient codes; complement factors and scales

Client: decode

$\widehat{\mathbf{A}}_{:,j} \leftarrow \mathcal{C}_j[\mathbf{z}_j]$, or $\mathbf{0}$ if $b_j = 0$

$\widehat{\mathbf{W}}' \leftarrow \widehat{\mathbf{W}}_x + \widehat{\mathbf{A}}\mathbf{\Phi}_d^\top + \widehat{\mathbf{L}}\widehat{\mathbf{R}}^\top$

C EXPERIMENTAL SETUP

This appendix collects the configurations and measurement protocols used in Section 5 and Appendix D.

C.1 MODELS AND QUANTIZED CHECKPOINTS

We evaluate Llama-2-7B and Llama-2-13B Touvron et al. (2023), together with Qwen-2.5-7B-Instruct and Qwen-2.5-14B-Instruct Bai et al. (2023); Qwen Team (2024). The matched-payload comparisons use 2- and 3-bit SINQ, HQQ, and GPTQ checkpoints Müller et al. (2026); Badri & Shaji (2023); Frantar et al. (2023), with group size 64 for SINQ and HQQ and 128 for GPTQ. The replacement study starts from 1.58-bit SINQ and HQQ checkpoints and compares refinement with a same-family 2-bit replacement. Refinement covers the seven linear projections in each transformer block; embeddings, the language-model head, normalization, architecture, and tokenizer are shared at both endpoints.

The larger models have 12.69B and 13.21B refined parameters, respectively. Their GPTQ bases are constructed at group size 128 using 256 WikiText-2 calibration samples.

C.2 RECAP CONFIGURATION

The subspace code uses water-filled 2-bit coefficients. The default configuration uses one budget-split rule across bases: the subspace code takes a quarter of the payload, with its dimension rounded to a power of two, so $d = 32$, 64, 128 and 256 at the four payloads of the grid and 512 at the largest payload of Table 5, and the entire remaining budget goes to the complement code, whose factors use $b_{\mathrm{f}} = 4$ bits and are fitted by rounding-aware alternation rather than rounded once after fitting. Payloads are matched within 1% to the subspace code alone at 2 bits and $d = 128$ to 2048, the reference ladder for payload matching. Uniform complement ranks cannot exceed the smaller dimension of the narrowest projection: 4096 for Llama-2-7B and 512 for the key and value projections of Qwen-2.5-7B-Instruct.

At the operating points in Tables 4 and 1, the payloads are 348 MB for Llama-2-7B and 356 MB for Qwen-2.5-7B-Instruct. For Llama-2-13B and Qwen-2.5-14B-Instruct, they are 545 MB and 554 MB, with subspace dimension $d = 256$ and complement ranks 203 and 187, respectively. Across the larger models' five-payload 1.58-bit sweep, the subspace dimensions are $32/64/128/256/512$, with complement ranks $20/46/99/203/412$ for Llama-2-13B and $18/42/91/187/380$ for Qwen-2.5-14B-Instruct. Appendix D.6 reports the component and budget-split comparisons behind this configuration.

The largest tested 7B messages contain 675–696 MB; the rounded column headings in Table 5 summarize these model-dependent payloads.

Complement fitting. The server decodes the subspace payload and fits the remaining residual under $\mathbf{H}_{\mathrm{fp}}$, as described in Section 4.2. RECAP balances the factors from the activation-weighted truncated SVD, then alternates rounding-aware refits around their 4-bit representations, keeping the best iterate. The factors use FP16 scales per row, whereas our CALDERA baseline uses one scale per matrix. Both factors, their scales, and metadata are charged in the complete payload.

Subspace-code design studies. Unless stated otherwise, the subspace code alone uses $d = 1024$ with water-filled 2-bit coefficients, transmitting 348 MB for Llama-2-7B and 356 MB for Qwen-2.5-7B-Instruct. Its basis and allocation statistic are calibrated on the deployed checkpoint. These settings isolate the shared basis from the complement code and differ from RECAP's $d = 256$ split at the same operating payload.

Activation-basis comparison. For Figure 3, we fix the checkpoint, activation statistic, $d = 1024$, and water-filled 2-bit coefficient coding on the four 7B SINQ-2 and GPTQ-2 settings. We compare the leading activation eigenvectors with random orthonormal subspaces over three seeds, the coordinate axes with largest diagonal entries of $\mathbf{H}$, and the d eigenvectors with largest $\lambda_i \|\mathbf{E}\mathbf{u}_i\|_2^2$ among the leading $P = \min(n, 4d)$ eigenvectors. Each basis uses its activation-weighted optimum,

$\mathbf{A}^\star = \mathbf{E}\mathbf{H}\mathbf{\Phi}(\mathbf{\Phi}^\top\mathbf{H}\mathbf{\Phi})^\dagger$. We use exact eigendecomposition and charge residual-aware selection $d\lceil\log_2 P\rceil$ index bits per matrix, totaling 0.3 MB. We report perplexity and the distortion fraction captured before and after coding, weighting matrices by their original distortion; across random seeds, we report median perplexity and mean distortion capture.

Coverage–precision comparison. For Figure 4, we vary the subspace code's dimension d and coefficient precision b on the four 7B SINQ-2 and GPTQ-2 settings, keeping db approximately constant and complete payloads, including codebooks, within 3%. We use water-filled per-column codebooks from one to eight bits, also testing per-row rounding at eight bits and using it at sixteen bits. We split distortion into omitted-direction and coefficient-coding errors at both the main operating payload and half that budget.

C.3 Baseline configurations

LoRC, LQER, and QERA use 8-bit factors, while CALDERA uses its scalar 4-bit path Yao et al. (2023); Zhang et al. (2024; 2025); Saha et al. (2024). Their ranks are selected to match RECAP's complete transmitted payload. Across the larger models' five-payload sweep, the 8-bit two-factor baselines use ranks $16/33/68/138/277$ for Llama-2-13B and $15/31/63/127/256$ for Qwen-2.5-14B-Instruct; CALDERA uses twice these ranks with 4-bit factors. Ranks are rounded upward when integer granularity prevents an exact payload match.

C.4 Calibration and evaluation

RECAP's subspace code constructs its activation statistic and allocation signal from $16{,}384$ WikiText-2 training tokens Merity et al. (2017) passed through the deployed checkpoint. Its complement code and the reconstruction baselines use calibration on the full-precision model. The core ablations use Llama-2-7B and Qwen-2.5-7B-Instruct with SINQ-2 and GPTQ-2 bases at 348 MB and 356 MB, respectively; individual studies state their departures from these settings. Appendix D.4 varies the calibration budget and corpus.

We evaluate materialized refined checkpoints in FP16 by default, using bfloat16 for the overflow cases identified in Appendix D.2. We report WikiText-2 test perplexity with a 2048-token sliding window and the seven-task average of standard or normalized accuracy on ARC-Easy, ARC-Challenge, PIQA, HellaSwag, WinoGrande, BoolQ, and OpenBookQA Clark et al. (2018); Bisk et al. (2020); Zellers et al. (2019); Sakaguchi et al. (2020); Clark et al. (2019); Mihaylov et al. (2018), evaluated with the LM Evaluation Harness Gao et al. (2024). Held-out C4 additionally tests the replacement crossings and the effect of the calibration corpus.

C.5 Payload accounting

All payloads are reported in decimal MB and include indices, codebooks, rate vectors, scales, and any basis-selection indices. RECAP's shared activation basis is reconstructed at both endpoints rather than transmitted; the complement factors are transmitted. Re-transmission is counted over the same target matrices as refinement.

The subspace allowance B_{AS} covers the complete cost in (10), not just coefficient indices. Because metadata and codebooks share this allowance, the realized average index rate can be below the nominal b.

Local reconstruction and inference. Zero transmitted basis bits still require local calibration and basis construction. Keeping the correction factorized stores the $n \times d$ basis locally and adds $O((m+n)(d+k))$ arithmetic per token for each target matrix. Alternatively, merging both codes into dense weights once costs $O(mn(d+k))$ arithmetic and requires dense model storage; subsequent inference uses the materialized weights. These operation counts exclude calibration and basis construction.

D Ablation Studies and Additional Results

We first report the full matched-payload perplexity comparison, then study the subspace code alone, coding, calibration, endpoint agreement, and RECAP's components and budget split. Appendix C

gives the shared setup and default configurations. The coding study in Appendix D.3 uses the subspace code alone; the calibration and endpoint studies also evaluate RECAP.

D.1 MATCHED-PAYLOAD PERPLEXITY COMPARISON

Table 4 gives the full matched-payload perplexity comparison discussed in Section 5. Base rows transmit no payload; bold entries mark the lowest perplexity in each setting and underlined entries the second lowest.

Table 4: We compare WikiText-2 perplexity (↓) when refining 2- and 3-bit checkpoints across model sizes at the largest matched payloads. **RECAP is best or tied in** 22 **of** 24 **settings, including all twelve** 2**-bit settings.**

Tier	Method	Llama-2-7B 348 MB			Qwen-2.5-7B-Instruct 356 MB		
		SINQ	HQQ	GPTQ	SINQ	HQQ	GPTQ
2-bit	base	319.03	7628.5	34.98	19702.5	1905.1	47.81
	LoRC	23.17	6837.6	21.86	336.50	128.34	25.21
	LQER	13.77	12.38	20.87	20.87	18.46	25.52
	QERA	10.86	11.89	24.81	17.00	17.46	30.51
	CALDERA (scalar)	14.64	85.24	44.97	52.71	25.38	36.36
	RECAP (ours)	**9.30**	**9.48**	**18.50**	**13.63**	**13.47**	**21.03**
3-bit	base	6.13	7.05	6.27	9.85	9.05	8.86
	LoRC	6.06	6.77	6.07	9.61	8.76	8.45
	LQER	6.07	5.91	**6.06**	8.31	8.30	**8.39**
	QERA	5.86	5.84	6.18	8.25	8.21	8.68
	CALDERA (scalar)	5.95	6.13	6.20	8.29	8.36	8.63
	RECAP (ours)	**5.80**	**5.79**	6.08	**8.09**	**8.08**	8.45
Tier	Method	Llama-2-13B 545 MB			Qwen-2.5-14B-Instruct 554 MB		
		SINQ	HQQ	GPTQ	SINQ	HQQ	GPTQ
2-bit	base	13.83	3694.5	12.94	797.3	138.3	32.42
	LoRC	10.17	2168.5	10.68	97.91	55.50	20.52
	LQER	8.55	7.96	10.45	16.46	15.36	20.57
	QERA	7.96	8.20	11.70	14.00	13.81	22.72
	CALDERA (scalar)	8.56	17.12	12.75	21.55	18.68	30.04
	RECAP (ours)	**7.31**	**7.34**	**10.27**	**11.60**	**11.40**	**18.37**
3-bit	base	5.33	5.60	5.36	7.34	7.44	7.21
	LoRC	5.27	5.55	5.30	7.10	7.19	7.01
	LQER	5.21	5.19	5.29	6.78	6.77	**7.00**
	QERA	5.18	5.15	5.31	6.75	6.69	7.11
	CALDERA (scalar)	5.22	5.40	5.31	6.85	6.89	7.16
	RECAP (ours)	**5.15**	**5.11**	**5.28**	**6.64**	**6.54**	**7.00**

D.2 SUBSPACE CODE ALONE VERSUS RECAP

We examine the complement code's contribution on 1.58-bit SINQ and HQQ checkpoints across model sizes, comparing RECAP with the subspace code alone at matched payloads.

Both use water-filled 2-bit coefficients: the subspace code alone receives the entire budget, whereas RECAP assigns it a quarter and spends the rest on the complement code (Appendix C.2). Tables 5 and 6 give the sub-2-bit payload sweeps, including LQER, QERA, and 2-bit re-transmission for comparison; bold marks the best coded result in each base–payload column. C4 perplexity and 7B task accuracy are additionally evaluated at the largest payload. In Table 5, † and ‡ identify bfloat16 perplexity evaluation for the marked base case and an individual payload, respectively, when FP16 overflows.

Below 2 bits, RECAP beats all three alternatives throughout the WikiText-2 payload sweep and reaches replacement perplexity on all four 7B bases and three of the four larger bases; the stronger

Table 5: We compare WikiText-2 perplexity versus payload for LQER, QERA, the subspace code alone, and RECAP on 1.58-bit 7B checkpoints against same-family 2-bit replacements, and report largest-payload C4 perplexity where evaluated. **RECAP leads at every WikiText-2 payload and reaches the replacement's perplexity on every base with 2–19% of its bytes.**

deployed base	codec	Base PPL	WikiText-2, payload (MB)					C4	re-tx 2-bit
			44	88	176	352	≈ 690	≈ 690	≈ 1800
Llama SINQ-1.58	LQER	18552	10821.8	8666.4‡	5565.9	2464.4	930.9	–	319.0
	QERA		4099.6‡	2595.3	2053.1	837.6	283.2	–	
	subspace code alone		22840.6	10918.4	6352.5	1314.2	112.3	79.1	
	RECAP		**2757.6**	**1393.0**	**422.3**	**96.5**	**24.8**	**50.3**	
Llama HQQ-1.58†	LQER	25109	30550.5	18204.1	9099.4	5855.3	1724.8	–	7628.5
	QERA		14111.6	10302.1	7138.9	3008.1	1273.4	–	
	subspace code alone		37669.0	16196.9	6659.1	2376.4	130.4	104.5	
	RECAP		**10277.0**	**4832.4**	**1861.6**	**350.8**	**50.0**	**79.7**	
Qwen SINQ-1.58	LQER	478807	66241.7	66516.0	22957.8	5850.0	1376.5	–	19702.5
	QERA		23481.5	12549.7	3244.1	928.5	185.6	–	
	subspace code alone		34674.3	27350.7	11775.6	2243.8	194.6	259.7	
	RECAP		**5898.9**	**1897.0**	**387.4**	**74.2**	**27.8**	**73.1**	
Qwen HQQ-1.58	LQER	1194833	88577.4	53252.2	23586.9	8309.9	1873.0	–	1905.1
	QERA		28922.9	16328.1	8295.5	2569.3	515.2	–	
	subspace code alone		29460.8	15272.1	7831.4	1416.6	113.7	171.6	
	RECAP		**8975.0**	**3396.3**	**878.6**	**158.8**	**29.3**	**72.9**	

Table 6: We compare WikiText-2 perplexity versus payload for LQER, QERA, the subspace code alone, and RECAP on 1.58-bit 13B-class checkpoints against same-family 2-bit replacements, and report largest-payload C4 perplexity where evaluated. **RECAP leads at every WikiText-2 payload and reaches the replacement's perplexity on three of the four bases with 2–30% of its bytes.**

deployed base	codec	Base PPL	WikiText-2, payload (MB)					C4	re-tx 2-bit
			68	136	273	545	≈ 1090	≈ 1090	≈ 3600
Llama SINQ-1.58	LQER	10474	3289.8	1990.8	1333.4	659.9	158.5	–	13.83
	QERA		564.1	358.3	194.4	91.0	45.2	–	
	subspace code alone		1492.5	1018.6	495.2	142.0	30.9	32.9	
	RECAP		**346.5**	**186.2**	**83.3**	**39.7**	**17.3**	**30.1**	
Llama HQQ-1.58	LQER	15061	2210.2	1816.0	1476.7	854.3	167.8	–	3694.5
	QERA		1724.8	1403.3	998.0	645.2	267.3	–	
	subspace code alone		3242.9	2733.5	1607.3	344.7	39.5	**32.3**	
	RECAP		**1076.3**	**955.2**	**495.0**	**112.2**	**21.7**	36.0	
Qwen SINQ-1.58	LQER	4123381	142630.6	67204.3	30649.7	8389.9	1443.8	–	797.3
	QERA		39207.6	26714.0	9702.6	3081.3	508.1	–	
	subspace code alone		31481.7	21450.8	8947.6	2134.5	319.7	426.1	
	RECAP		**8391.7**	**2709.4**	**676.1**	**169.8**	**42.1**	**128.6**	
Qwen HQQ-1.58	LQER	1514293	86665.3	41655.8	14637.2	5638.6	702.5	–	138.3
	QERA		33827.7	25068.4	10766.2	3920.9	987.7	–	
	subspace code alone		32994.6	19556.2	7366.0	1551.1	199.3	247.3	
	RECAP		**7149.1**	**3350.1**	**1193.1**	**214.0**	**45.4**	**119.3**	

Llama-2-13B SINQ replacement remains below the best tested refinement. The advantage over the subspace code largely persists on C4, with a reversal on larger-model Llama HQQ. At the largest 7B sub-2-bit payload, RECAP's seven-task accuracies are 41.5, 38.8, 41.2, and 40.8%, versus 41.4, 35.4, 36.3, and 35.9% for 2-bit re-transmission, in the base order of Table 5. It matches or exceeds replacement accuracy using 675–696 MB versus 1821–1835 MB, or 37–38% of its bytes.

These results favor spending part of the payload on the residual left by the shared subspace at very low precision, rather than devoting every byte to enlarging that subspace. The complement code's clearest benefit is more payload-efficient perplexity recovery, which can bring replacement quality within a smaller communication budget.

D.3 Coefficient coding and rate allocation

We separate the benefits of learned coefficient codebooks from those of rate allocation, and test whether these design choices support task accuracy as well as perplexity.

Complementing the matched-payload coder comparison in Section 5 (Table 3), Table 7 fixes basis, dimension, and payload within each block, comparing per-row rounding, equal-rate per-column K-means, and water-filled codebooks at two and four bits. Allocation uses activation sensitivity λ_j, coefficient energy $\|\mathbf{A}^\star_{:,j}\|_2^2$, or their product S_j from (8). Table 8 measures seven-task accuracy after replacing either the eigenbasis with a random subspace drawn with one seed or the learned codebook with per-row rounding, at the subspace code's default operating point. RECAP provides the matched-payload reference from Table 1, using $d = 256$ plus the complement instead of the subspace-only $d = 1024$.

Table 7: We compare WikiText-2 perplexity (↓) of payload coders and rate-allocation signals for the subspace code at fixed basis, dimension and payload. **At two bits learned codebooks supply most of the gain over per-row rounding and water-filling on S_j adds a further improvement in every cell, while at four bits the coders are much closer.**

	Llama-2-7B		Qwen-2.5-7B-Instruct	
	SINQ-2	GPTQ-2	SINQ-2	GPTQ-2
two-bit coefficient budget, $d = 1024$				
per-row rounding	16.47	64.34	271.5	66.30
learned codebooks, equal column rates	9.77	20.47	18.75	25.05
water-filling: activation sensitivity only (λ_j)	9.52	**20.15**	15.99	24.15
water-filling: coefficient energy only	10.50	21.41	19.94	25.94
water-filling: combined signal (S_j, default)	**9.51**	20.23	**15.98**	**24.00**
four-bit coefficient budget, $d = 512$				
per-row rounding	10.52	27.52	**18.79**	29.76
learned codebooks, equal column rates	10.53	25.00	19.59	29.27
water-filling: combined signal (S_j, default)	**10.49**	**24.77**	18.99	**28.92**

Table 8: We compare seven-task accuracy (↑, %) of subspace-code ablations and RECAP. **Replacing the eigenbasis or learned codebooks reduces accuracy in every setting.**

	Llama-2-7B		Qwen-2.5-7B-Instruct	
	SINQ-2	GPTQ-2	SINQ-2	GPTQ-2
subspace code alone	**54.9**	**41.5**	<u>52.1</u>	<u>39.3</u>
random subspace in place of the eigenbasis, one seed	44.3	34.7	44.2	37.1
per-row round-to-nearest in place of the codebook	45.4	38.3	41.0	36.7
RECAP (ours)	<u>54.0</u>	<u>40.6</u>	**53.7**	**40.4**

At fixed payload, the complement reallocates capacity from additional shared-basis directions to a correction fitted to the remaining residual. This allocation lowers WikiText-2 perplexity relative to the subspace code alone in all four corresponding 2-bit settings (Table 13) and all 40 tested operating points across eight 1.58-bit bases (Tables 5 and 6). The complement therefore improves the quality attainable per transmitted byte, particularly where shared-subspace coding alone leaves substantial residual error.

The controlled comparison attributes most of the two-bit improvement to learned codebooks, with a further gain from allocation on S_j; four-bit differences are smaller and mixed. Activation sensitivity alone nearly matches the combined signal and occasionally improves on it, whereas coefficient energy alone is worse. Both downstream replacements reduce accuracy in every tested setting.

Effective low-bit coding depends on matching the coefficient distribution and allocating bits according to activation sensitivity, not precision alone. The downstream comparison also shows that the shared basis and learned codebook contribute to recovered task performance.

D.4 Calibration and cross-corpus transfer

We study how public calibration choices affect refinement quality and whether the resulting communication advantage transfers to another corpus.

Table 9 varies calibration for the subspace code alone on four 7B SINQ-2 and GPTQ-2 bases, fixing the checkpoint, dimension, coding configuration, and payload. We vary the WikiText-2 training-token budget and evaluate on WikiText-2, then compare WikiText-2 and C4 calibration at the default budget on both held-out corpora. Table 10 instead holds the WikiText-2-calibrated RECAP refinements fixed on 1.58-bit bases across both model-size tiers. It reports the first tested payload that reaches each same-family 2-bit replacement's perplexity on each corpus, as a fraction of replacement bytes; *none* means no tested payload crosses.

Table 9: We compare perplexity (↓) of the subspace code at its operating point as the calibration budget and corpus behind its basis and effective signal change. **More calibration tokens help in every cell, and calibrating on the evaluation corpus helps most on that corpus, above all on the GPTQ bases.**

	Llama-2-7B		Qwen-2.5-7B-Instruct	
	SINQ-2	GPTQ-2	SINQ-2	GPTQ-2
calibration budget, evaluated on WikiText-2				
$T = 2{,}048$	10.14	28.62	17.99	32.41
$T = 8{,}192$	9.71	21.59	16.25	25.89
$T = 16{,}384$, as used throughout	9.51	20.23	15.98	24.00
$T = 32{,}768$	**9.37**	**18.95**	**15.51**	**23.53**
calibration corpus, evaluated on WikiText-2				
WikiText-2 calibration	**9.51**	**20.23**	**15.98**	**24.00**
C4 calibration	9.79	25.84	16.16	27.47
calibration corpus, evaluated on C4				
WikiText-2 calibration	12.99	40.96	24.47	77.95
C4 calibration	**12.56**	**35.05**	**23.77**	**60.80**

Table 10: We report, on WikiText-2 and on held-out C4, the smallest tested payload, as a share of the 2-bit replacement's bytes, at which RECAP reaches the replacement's perplexity on that corpus. **Most crossing payloads are unchanged on C4, so the communication savings persist beyond the calibration corpus.**

		replacement PPL		RECAP crossing	
tier	deployed base	WikiText-2	C4	WikiText-2	C4
7B	Llama SINQ-1.58	319.0	495.7	19%	19%
	Llama HQQ-1.58	7628.5	5809.4	5%	5%
	Qwen SINQ-1.58	19702.5	24603.3	2%	2%
	Qwen HQQ-1.58	1905.1	2773.3	10%	10%
13B-class	Llama SINQ-1.58	13.83	17.43	none	none
	Llama HQQ-1.58	3694.5	2797.7	2%	2%
	Qwen SINQ-1.58	797.3	1020.9	7%	15%
	Qwen HQQ-1.58	138.3	196.3	30%	30%

More calibration tokens improve perplexity in every tested setting, and each calibration corpus performs best on its corresponding evaluation corpus. GPTQ is more sensitive than SINQ to both calibration choices. Every base with a measured WikiText-2 replacement crossing also crosses on C4; only larger-model Qwen SINQ requires a later payload.

Calibration therefore offers a quality–computation tradeoff at fixed communication cost, while the crossing comparison demonstrates transfer beyond the calibration corpus. The public protocol should account for the intended workload, since calibration shapes the basis and allocation signal, and the precise replacement saving depends on the evaluation distribution.

D.5 Basis agreement between endpoints

We test whether independently reconstructed bases decode a fixed server payload consistently, and how changes in basis reconstruction affect refinement.

On Llama-2-7B SINQ-2 and Qwen-2.5-7B-Instruct GPTQ-2, we encode once per server configuration and decode the fixed payload under client-side changes. For the subspace code alone, we compare independent, matched GPU reconstruction with a CPU eigensolver on the same statistic, varying precision and the sign convention (Table 11). The RECAP audit adds GPU calibration in batches of eight sequences, CPU calibration, and 4-bit storage of the server's basis (Table 12); CPU calibration changes hardware and arithmetic together.

Table 11: We report WikiText-2 perplexity (↓) of the subspace code alone, $d = 1024$ with water-filled 2-bit payload entries, when the client decodes the server's payload with a basis solved by the server's GPU solver, by a CPU solver, or by a CPU solver without the sign convention. **Only the server's own solver reproduces its model: another solver costs up to 40 perplexity in single precision, and dropping the sign convention breaks decoding.**

	Server reference	Same GPU solver	CPU eigensolver	CPU eigensolver without sign convention
basis computation: single precision				
Llama-2-7B, SINQ-2	9.51	9.51	11.90	786.1
Qwen-2.5-7B-Instruct, GPTQ-2	24.00	24.00	63.96	3084.4
basis computation: double precision				
Llama-2-7B, SINQ-2	9.48	9.48	9.83	791.8
Qwen-2.5-7B-Instruct, GPTQ-2	24.27	24.27	26.70	4670.4

Table 12: We report WikiText-2 perplexity (↓) of RECAP at the 7B operating points of Table 4 when the client decodes the server's payload with a basis from a batched calibration pass, a CPU calibration pass or a CPU eigensolver, or holds the server's basis at 4 bits. **Kernel changes and a 4-bit basis cost at most 0.24; a different calibration device or eigensolver costs up to 2.1, far less than a different eigensolver costs the subspace code alone.**

	Server reference	Batched GPU calibration	CPU calibration	CPU eigensolver	4-bit basis
basis computation: single precision					
Llama-2-7B, SINQ-2	9.30	9.30	9.47	9.52	9.31
Qwen-2.5-7B-Instruct, GPTQ-2	21.04	21.05	23.12	22.67	21.28
basis computation: double precision					
Llama-2-7B, SINQ-2	9.30	9.30	9.44	9.44	9.32
Qwen-2.5-7B-Instruct, GPTQ-2	20.61	20.61	22.20	21.14	20.71

Matched calibration and solver settings reproduce the server model, whereas removing the sign convention breaks decoding. Solver or calibration-device changes have larger effects than batching or basis compression. Double precision reduces disagreement between solvers, which can resolve nearly equal eigenvalues differently. In the same-statistic audit, the remaining double-precision discrepancy comes from the randomized solver for the widest `down_proj` matrices, whose random numbers differ across devices. Each client is compared with its corresponding server configuration, since precision also changes the server reference.

Reproducible decoding requires agreed calibration, solver, precision, sign, and randomized-solver conventions. In the common eigensolver test, RECAP's combined design—a smaller shared subspace and most payload bytes in transmitted complement factors—shows lower sensitivity than the subspace code alone.

D.6 RECAP'S COMPONENTS AND BUDGET SPLIT

We isolate the contributions of RECAP's two codes and test how budget allocation and rounding-aware factor refitting affect their combination. Table 2 gives the full 1.58-bit component, split, and refitting comparison on WikiText-2 and C4 in the main text; here we give the 2-bit comparison and the shared protocol.

On the two 7B models, we compare each code alone with combinations at $d = 16, 64, 256$, adjusting the complement rank to keep total payload fixed (Tables 2 and 13). The complement-only

arm retains RECAP's full-precision-model calibration, 4-bit factor quantizer, and fitting procedure (Appendix C.2). At the same $d = 64$ split, we also replace rounding-aware refitting with fitting the factors once and then quantizing them. The quarter-budget rule, selected after the component comparison, is also compared with fixed $d = 64$ across the 7B payload grids wherever the two differ.

Table 13: We compare WikiText-2 perplexity (↓) of RECAP's components and splits on the 2-bit bases, at 348 MB for Llama-2-7B and 356 MB for Qwen-2.5-7B-Instruct. **The default setting is best on both SINQ-2 cells and the complement code alone on both GPTQ-2 cells, whose residual the shared subspace reaches less.**

	Llama-2-7B		Qwen-2.5-7B-Instruct	
	SINQ-2	GPTQ-2	SINQ-2	GPTQ-2
subspace code alone	9.51	20.23	15.98	24.00
complement code alone	9.53	**18.07**	<u>13.82</u>	**19.75**
RECAP ($d = 16$)	<u>9.47</u>	18.46	13.85	<u>20.14</u>
RECAP ($d = 64$)	<u>9.47</u>	<u>18.20</u>	13.84	20.57
RECAP ($d = 64$, without factor refitting)	9.77	22.84	14.80	29.27
RECAP ($d = 256$, default)	**9.30**	18.50	**13.63**	21.03

With refitting, combining the codes outperforms either alone on all four 1.58-bit bases, on both WikiText-2 and C4. At 2 bits, the default combination is best on SINQ, whereas the complement code alone is best on GPTQ. Refitting improves every matched-split comparison. The quarter-budget rule lowers WikiText-2 perplexity at 13 of 16 compared points on the 1.58-bit bases and 22 of 36 on the 2- and 3-bit grid; every 3-bit difference is within 0.5%.

The results support combining inexpensive shared directions with a flexible transmitted correction, with their useful balance depending on the base's residual geometry. On the four 7B SINQ-2 and GPTQ-2 settings, residual isotropy is 75–92% of a same-shaped Gaussian matrix's, whereas the deployed-checkpoint activation statistic has effective rank only 9–15% of its dimension. At $d = 256$, the shared subspace captures 64–65% of activation-weighted residual error for SINQ, versus 25–26% for GPTQ. Rounding-aware refitting helps preserve the correction's benefit after factor quantization.