PRISMQUANT: OPTIMAL NULL-SPACE ROTATIONS FOR GROUPED QUANTIZERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Smaller activation outliers do not necessarily imply better low-bit quantization: their alignment with the quantizer matters. We introduce **PrismQuant**, a quantizer-aware rotation framework that aligns the leading activation eigenspace with the constant group subspace of asymmetric grouped INT4. The affine offsets represent the energy in this subspace without widening the range within the group. We formulate rotation design as a Ky Fan trace maximization and derive a closed-form solution that is *provably optimal for this alignment objective*. Compact Householder transformations and their compact-WY representation enable gradient-free construction and efficient application at both foldable and online sites. A predictive range law further connects unaligned activation energy and group size to quantization-relevant variation. Experiments on Llama, Qwen, and Mistral span dense models up to 70B parameters and a 30B mixture-of-experts model. Under W4A4KV4, PrismQuant sets the state of the art on Llama-3.2-3B among the compared methods in both perplexity and accuracy. On Llama-3.1-70B, it attains **3.85** perplexity and **72.46%** average zero-shot accuracy, only 0.22 percentage points below the full precision. In the deployment study on Llama-3.1-8B, our optimized implementation achieves $1.51\times$ prefill and $1.22\times$ CUDA Graph decode speedups over matched FP16 baselines, with **56.34%** lower decode peak memory and only **2.35%** additional Graph decode latency over Hadamard.

1 INTRODUCTION

4 bit weight-and-activation quantization offers a practical route to reducing the memory and arithmetic costs of large language model inference, but preserving accuracy remains challenging (Ashkboos et al., 2024; Sun et al., 2024b). A central obstacle is activation anisotropy: a few channels or low-dimensional directions can dominate the quantization range, leaving insufficient resolution for the remaining signal (Dettmers et al., 2022; Xiao et al., 2023; Sun et al., 2024a). Rotations and other equivalent transforms mitigate this problem by reshaping activation distributions while preserving the full-precision function (Lin et al., 2024a; Hu et al., 2025; Liu et al., 2025). Yet large magnitude alone does not determine quantization difficulty: what matters is how the signal interacts with the quantizer's representation. This motivates a complementary design question: *which activation directions does the quantizer already represent efficiently, and how should a rotation exploit them?*

A grouped quantizer scales each group by its own extremes. A component that is uniform within one group and absent from the others, therefore, raises no coordinate above that group's own scale, and in an asymmetric format its level is carried by the affine offset outright. Across a d-dimensional activation with group size g, these group-constant directions span a structured subspace of dimension d/g that the quantizer tolerates best. The quantizer's geometry, not only its precision, is thus something a transform can exploit. We introduce **PrismQuant**, a quantizer-aware rotation framework that aligns dominant activation eigendirections with this group-constant subspace through a closed-form rotation computed from calibration statistics alone. As Figure 1 illustrates, this alignment leaves substantially less within-group variation for the same INT4 quantizer to resolve.

We formalize this principle in Section 3. Let Σ denote the activation second moment and $\mathcal{S}$ the group-constant subspace. Rotation design becomes the problem of maximizing the expected energy projected onto $\mathcal{S}$ over orthogonal transforms. By the Ky Fan maximum principle (Fan, 1949; 1950), the full-capacity optimum maps the leading d/g eigendirections of Σ into $\mathcal{S}$, yielding a solution that

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

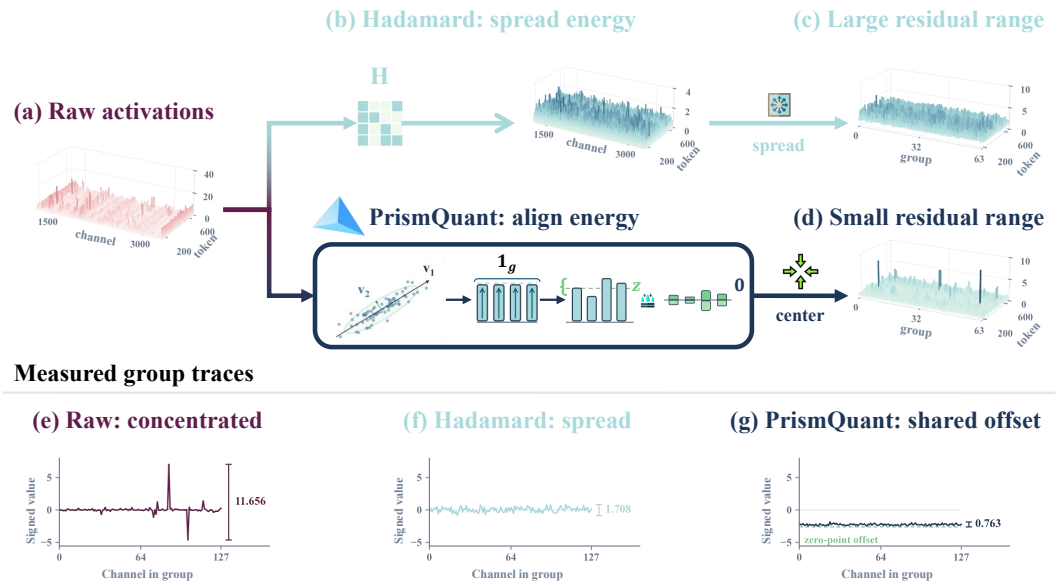

Figure 1: **Aligning activations with quantizer geometry.** Llama-3.2-3B layer 27 down-projection inputs, g=128. **(a–d)** PrismQuant reduces the mean within-group range from 1.76 to 0.84 relative to Hadamard under matched asymmetric INT4 quantization. **(e–g)** Token/group traces illustrate the mechanism; the dashed line marks the stored affine offset.

is *provably optimal for the alignment objective*. In practice, we estimate the leading eigenspace from all calibration tokens and realize the transform using Householder reflections in compact WY form (Schreiber & Van Loan, 1989). A rank parameter k controls the alignment–cost trade-off, without gradient-based training. The transform folds into adjacent weights where possible; at non-foldable sites such as the down-projection input, its compact representation avoids a dense online rotation.

The same geometry reveals a second role for the size of the group. Smaller groups provide not only finer scale resolution but also more affine offsets, enlarging the subspace available for alignment. In Section C, we derive a two-factor range law that separates residual unaligned energy from the dependence of extreme-values on the size of the group under a Gaussian residual approximation. This distinction matters: capturing more energy need not improve quantization if it requires coarser groups, since the captured energy depends only on the number of slots. Our metadata-matched ablations demonstrate this trade-off directly: among the tested configurations, allocating the same activation bit budget to finer groups is more effective than adding affine directions within larger groups. Alignment capacity and quantization granularity must therefore be considered together.

Our experiments connect this geometry to local quantization error and end-to-end model quality across the Llama, Qwen, and Mistral families, from 0.6B to 70B parameters. Across all 28 down-projection inputs of Llama-3.2-3B, PrismQuant reduces mean within-group range and activation NMSE by approximately 25% and 40% relative to Hadamard under matched quantization settings (Figure 2); end to end, its strongest configuration reaches the lowest 8.58 WikiText-2 perplexity and the highest eight-task zero-shot mean among compared methods (**61.23%**,Table 1). Controlled ablations in Section 4.3 probe the design inward, at the alignment rank, the metadata budget, and the calibration of sparsely routed experts; outward, at how the subspace is estimated; and at the quantizer itself, where the gain survives a symmetric format and moving the aligned level out of the offset's reach costs only a quarter of it (Appendix A.4). Estimating the subspace from all calibration tokens rather than from a few extreme ones recovers about a quarter more of the gain, and the gain is insensitive to the calibration set, the eigensolver, and the signed permutation (Appendix A.4). The same construction extends to mixtures of experts: with one rotation per expert and the router untouched, Qwen3-30B-A3B recovers 85% of the accuracy that Hadamard loses (Table 2). The rotation is also cheap to run: a two-kernel Tensor Core implementation of the rank-k correction, integrated into a packed-INT4 pipeline with CUDA Graph replay, adds 44 MB and 2.4% decode latency over Hadamard on Llama-3.1-8B while preserving the backend's 1.5× prefill throughput, 1.2× decode speed, and 56% lower peak memory than FP16 (Appendix D).

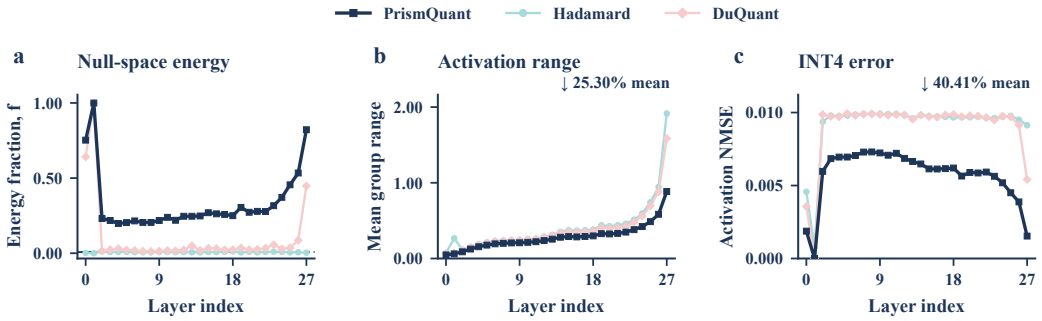

Figure 2: **Quantizer-aware alignment predicts INT4 error.** All 28 Llama-3.2-3B down-projection inputs, g=128, evaluated on identical tokens. **(a)** Energy captured by the group-constant subspace. **(b,c)** Within-group range and activation NMSE under matched asymmetric INT4; PrismQuant reduces them by 25% and 40% on average relative to Hadamard.

In summary, our contributions are:

- **Quantizer-induced subspace alignment.** We formalize the group-constant geometry of grouped quantization and cast rotation design to maximize the activation energy captured by this subspace. Because the target subspace is fixed by the quantizer rather than estimated from outlier statistics, alignment becomes a single well-posed spectral problem with a closed-form optimum.
- **Spectral optimality with compact execution.** We characterize the alignment optimum through the Ky Fan principle and develop a training-free Householder realization with controllable rank, supporting both weight folding and online activation transforms. The online part reduces to a block Hadamard plus a rank-k correction, a structure that maps directly onto Tensor Core kernels and deploys in a packed-INT4 pipeline at negligible overhead.
- **Range law and end-to-end validation.** We prove that residual energy bounds the aggregate squared range, derive a two-factor law for the quantization step whose only approximation is a single measured crest-factor ratio, and test the mechanism with pre-registered ablations. In W4A4KV4 comparisons, PrismQuant achieves the strongest results among the methods compared in various models and delivers real speedups and memory savings on commodity GPUs.

2 Related Work

Activation outliers and smoothing. LLM activations are highly anisotropic, with a small number of channels, tokens, or low-dimensional directions carrying extreme values (Bondarenko et al., 2021; Dettmers et al., 2022; Sun et al., 2024a; Wang et al., 2025). SmoothQuant (Xiao et al., 2023) migrates the activation difficulty into weights through per-channel scaling, OmniQuant (Shao et al., 2024) jointly optimizes scaling and clipping, Outlier Suppression+ (Wei et al., 2023) adds a per-channel shift that absorbs asymmetric outliers in a bias, and AWQ (Lin et al., 2024b) exploits activation statistics for weight-only quantization. These channel-wise transformations rebalance ranges without mixing information across channels. Token-level isolation such as PrefixQuant (Chen et al., 2026) is complementary: with outlier tokens held in full precision, most of PrismQuant's gain remains (Appendix A.4). PrismQuant instead asks which directions the downstream quantizer efficiently represents and rotates dominant energy toward them.

Rotation-based quantization. QuaRot (Ashkboos et al., 2024) uses fixed Hadamard transformations, SpinQuant (Liu et al., 2025) learns orthogonal rotations, and DuQuant (Lin et al., 2024a) combines channel reordering with structured block rotations. OSTQuant (Hu et al., 2025), FlatQuant (Sun et al., 2024b), DFRot (Xiang & Zhang, 2024), KurTail (Akhondzadeh et al., 2025), and DartQuant (Shao et al., 2026) further optimize scaling, rotation geometry, or activation distributions. These methods primarily optimize activation geometry.

A second line locates outlier directions from token statistics and treats them specially: ResQ (Saxena et al., 2024) keeps the high-variance directions in higher precision, and OffQ (Wang et al., 2026) selects the largest-norm token per sequence and rotates its principal direction into one channel per group, where the asymmetric zero-point absorbs it. Both lines take the activation as the object to be

reshaped and the quantizer as given. PrismQuant reverses the roles. The asymmetric group quantizer already has a free subspace, spanned by the per-group constant directions; we treat that subspace as the target and ask which rotation carries the most activation energy into it. The answer is a spectral problem with a closed-form optimum over the full second moment (Section 3), in which outliers enter only as energy the rotation captures rather than as directions to be found first. It is realized by a rank-k compact-WY factor that folds into weights or runs online at the same cost (Appendix D), and the energy that remains unaligned is what our range law charges for, as a function of group size and metadata budget (Section C, Figure 4).

Grouped asymmetric quantization. Group-wise scales and zero-points are widely used in low-bit weights, activations, and KV caches (Yuan et al., 2023; Lin et al., 2024b; Zhao et al., 2024; Lin et al., 2025). KIVI (Liu et al., 2024), for example, shows that quantization granularity should reflect the statistical structure of keys and values. PrismQuant highlights a complementary geometric consequence: in an asymmetric group, the affine offset induces a group-constant direction that can be deliberately targeted by an equivalent transform. Group size therefore controls not only metadata cost and local resolution, but also the dimension of the subspace available for alignment (Table 6).

3 METHOD

A grouped asymmetric quantizer represents one direction in every group for free: the constant vector spanned by its offset. PrismQuant rests on a single observation: an equivalent rotation can steer the dominant activation energy into exactly these directions, so that what would otherwise set the quantization range is absorbed by metadata the format already pays for. Rather than flattening activations independently of the format, we therefore target the quantizer's own group-constant, range-neutral subspace. We derive the optimal spectral alignment, realize it through compact Householder transforms, and then describe Transformer integration and the metadata trade-off that follows. Proofs, numerical qualifications, and extended diagnostics are collected in Section B.

3.1 QUANTIZER-INDUCED RANGE-NULL SUBSPACE

Let $X \in \mathbb{R}^{N\times d}$ hold token activations as rows. At a rotation site every token is transformed by the same orthogonal R, so $X' = XR^\top$, or $y = Rx$ for a single column-vector activation. Each transformed token is split into $M = d/g$ contiguous groups of g features, assuming g divides d. Groups are per token: $d = 8192$ and $g = 128$ give 64 scales and 64 offsets for every token. Asymmetric INT4 encodes a nonconstant group $y^{(j)}$ as

$$q^{(j)} = \mathrm{clip}_{[0,15]}\left(\mathrm{round}\left(\frac{y^{(j)} - z_j\mathbf{1}_g}{s_j}\right)\right), \qquad \hat{y}^{(j)} = s_j q^{(j)} + z_j\mathbf{1}_g, \tag{1}$$

with ideal parameters $z_j = \min y^{(j)}$ and $s_j = \mathrm{range}(y^{(j)})/15$, where $\mathrm{range}(v) = \max_i v_i - \min_i v_i$. The offset z_j fixes the grid's origin and the scale s_j its resolution; both are stored in fp16, with a positive fallback scale for degenerate groups. The analysis below treats the metadata as exact (Section B.1). The range is blind to shared levels: if $y^{(j)} = c_j\mathbf{1}_g + e^{(j)}$, its ideal scale depends only on $\mathrm{range}(e^{(j)})$, while its offset becomes $c_j + \min e^{(j)}$. Let $u_j = \mathbf{1}_{\mathcal{I}_j}/\sqrt{g}$ be the normalized indicator of group j. These M orthonormal vectors span the *group-constant subspace*

$$\mathcal{S} = \mathrm{span}\{u_1, \ldots, u_M\}, \qquad P_{\mathcal{S}} = UU^\top, \quad U = [u_1, \ldots, u_M], \tag{2}$$

whose projector replaces every group by its mean. Consequently,

$$\mathrm{range}(y^{(j)}) = \mathrm{range}\Big([(I - P_{\mathcal{S}})y]^{(j)}\Big). \tag{3}$$

We call $\mathcal{S}$ a *range-null subspace*: its components do not contribute to the within-group range, rather than disappearing from the reconstructed activation. Its $M = d/g$ degrees of freedom are represented through the $16M$ bits of offsets already stored by the format. This decomposition makes explicit a property of the existing quantizer, it does not require explicit mean subtraction.

The shared level and the stored offset need not be identical: the latter also contains the minimum of the remaining variation. Thus the existing min–max quantizer already represents the aligned component without an extra coefficient per token. A large activation magnitude is compatible with a narrow group range when neighboring coordinates share that level.

Flattening alone does not ensure that dominant energy enters $\mathcal{S}$. A Hadamard transform can reach group-constant directions, but it does not select them according to the activation spectrum. For example, if the image of an outlier channel is a signed pattern with zero mean in every group, its projection onto $\mathcal{S}$ is zero and its full signed variation remains for the INT4 grid. Other Hadamard columns can be group-constant, so zero capture is not a universal property. For an isotropically oriented direction, the expected energy fraction in $\mathcal{S}$ is $M/d = 1/g$, providing the reference level in Figure 2. PrismQuant instead explicitly targets these directions using calibration data.

3.2 Optimal Alignment

We obtain dominant directions from the leading eigenvectors of the uncentered second moment $\Sigma = \mathbb{E}[xx^\top]$, estimated offline from N_{cal} calibration tokens as $\widehat{\Sigma} = N_{\text{cal}}^{-1} X_{\text{cal}}^\top X_{\text{cal}}$, where $X_{\text{cal}} \in \mathbb{R}^{N_{\text{cal}} \times d}$ contains calibration activations as rows. We do not center: a persistent nonzero component also carries energy that the quantizer must represent, and alignment can turn it into a group-common component. For the analysis, let (λ_i, v_i) denote the eigenpairs of Σ, ordered so that $\lambda_1 \geq \cdots \geq \lambda_d$, and write $V_k = [v_1, \ldots, v_k]$. Every token then decomposes as

$$x = V_k a + r, \qquad a = V_k^\top x, \qquad r = (I - V_k V_k^\top)x, \tag{4}$$

where the directions are fixed at calibration and the coefficients a vary per token. Choose $k \leq M$ distinct targets $U_k = [u_{j_1}, \ldots, u_{j_k}]$ and require $RV_k = U_k$, absorbing fixed signs into the eigenvector convention. Then

$$Rx = \underbrace{U_k a}_{\text{group-common component}} + \underbrace{Rr}_{\text{remaining variation}}. \tag{5}$$

Under this alignment, the i-th leading component contributes a token-dependent shared level $a_i/\sqrt{g}$ to group j_i. The existing affine offset absorbs this shared level, leaving the within-group range governed entirely by the residual Rr. Any group-constant component of the residual likewise leaves the range unchanged. This structure is realized directly by rotating and quantizing Rx, without explicitly computing or separately storing the coefficients a at inference time. Section B.1 illustrates this mechanism numerically.

Objective and its optimum. The energy delivered to the selected targets is

$$\mathcal{J}_k(R) = \mathbb{E}\|U_k^\top Rx\|_2^2 = \operatorname{Tr}\big(U_k^\top R \Sigma R^\top U_k\big), \qquad \max_{R^\top R = I} \mathcal{J}_k(R). \tag{6}$$

Because R preserves total energy, maximizing $\mathcal{J}_k$ is equivalent to minimizing the energy outside the selected targets, $\operatorname{Tr}(\Sigma) - \mathcal{J}_k(R)$.

Proposition 1 (Optimal spectral alignment)**.** *The maximum of Equation* (6) *is* $\sum_{i=1}^k \lambda_i$*. It is attained by any orthogonal* R *mapping a leading* k*-dimensional eigenspace of* Σ *onto* $\operatorname{span}(U_k)$.

The result follows from the Ky Fan maximum principle (Fan, 1949; 1950): $\operatorname{Tr}(Q^\top \Sigma Q)$ over $d \times k$ matrices with orthonormal columns is maximized by a leading eigenspace, and $Q = R^\top U_k$ ranges over exactly that set. The selected alignment is therefore provably optimal for this objective. At $k = M$, it minimizes total energy outside $\mathcal{S}$; for $k < M$, the guarantee applies to the selected k-dimensional target.

The optimal subspace, rather than a unique orthogonal matrix, is the object of this construction. Eigenvalue ties may admit several equally valid leading eigenspaces. In practice, alignment of an orthonormal estimate $\widehat{V}_k$ captures $\operatorname{Tr}(\widehat{V}_k^\top \Sigma \widehat{V}_k)$ in the selected targets. This separates the quality of the calibrated directions from their structured realization: estimating the empirical moment and constructing its rotation are distinct steps, and neither requires optimizing a task loss.

3.3 Structured Construction

We realize the alignment without storing a dense transform:

$$R = H_g D \Pi G, \qquad G = I - WY^\top, \quad W, Y \in \mathbb{R}^{d \times k}. \tag{7}$$

G maps leading eigenvectors to coordinate anchors, Π arranges coordinates across groups to balance residual energy while keeping these anchors fixed, D applies fixed signs, and H_g converts each anchor into a group constant direction. Here H_g denotes the block-diagonal transform with a normalized size-g Walsh–Hadamard matrix in each block(Fino & Algazi, 1976; Ashrafi, 2017).

Alignment by Householder reflections. The construction first maps spectral directions to coordinate anchors. Set $G_0 = I$ and $t_i = 1 + (i-1)g$ for $i = 1, \ldots, k$. At step i, $b_i = G_{i-1} v_i$ is orthogonal to the previously fixed anchors. For $\delta_i = b_i - e_{t_i} \neq 0$, define

$$h_i = \frac{\delta_i}{\|\delta_i\|_2}, \qquad \mathcal{H}_i = I - 2 h_i h_i^\top, \qquad G_i = \mathcal{H}_i G_{i-1}. \tag{8}$$

The reflection sends b_i to e_{t_i} while preserving earlier anchors; after at most k reflections, $G = G_k$ aligns every selected direction. The numerical skip rule and induction proof are given in Section B.3.

The compact WY representation (Schreiber & Van Loan, 1989) collects the reflectors into two thin factors. Using the column-vector convention in Equation (7), batched application is

$$XG^\top = X - (XY)W^\top, \qquad XY \in \mathbb{R}^{N \times k}. \tag{9}$$

The activation dimension remains d: G is orthogonal and full rank, preserving all information and the Euclidean norm of each activation. Only the correction $I - G = WY^\top$ has rank at most k. Applying this correction through the two thin factors requires $\mathcal{O}(dk)$ operations per token and $\mathcal{O}(dk)$ storage for W and Y, compared with $\mathcal{O}(d^2)$ computation and storage for a dense transform.

Anchor placement and residual balancing. The k aligned anchors occupy the first coordinates of the first k groups. When $k < M$, the first coordinates of the remaining groups are filled by the lowest-energy available coordinates. We estimate coordinate energies after G from calibration activations, corresponding to the diagonal of $G\Sigma G^\top$. All other coordinates are sorted by decreasing energy and assigned greedily to the group with the lowest accumulated residual energy, subject to its $g - 1$ non-anchor slots; ties are resolved deterministically. The low-energy fillers keep reduced-rank ablations from unintentionally assigning additional high-energy coordinates to unused constant slots. These choices preserve the selected eigenspace alignment.

Block mixing and calibration. Within each group, a normalized Walsh–Hadamard matrix maps e_1 to $\mathbf{1}_g/\sqrt{g}$, while its remaining columns are orthogonal to that constant direction. Therefore, the complete construction satisfies $Rv_i = \pm u_i$ for the selected directions, converting the aligned anchors into shared levels and mixing the residual within each group. This final stage costs $\mathcal{O}(d \log g)$ per token, bringing the total transform cost to $\mathcal{O}(dk + d \log g)$. To obtain the spectral directions used in this construction, we employ a randomized eigensolver (Halko et al., 2011) at wide activation sites, avoiding explicit formation of the full second moment. Our implementation uses three passes with orthogonalization and Rayleigh–Ritz extraction, while the smaller value moments within each attention head are formed explicitly and diagonalized directly. The coordinate energies for the permutation are estimated in a separate calibration step, after which all transform factors and seeded signs are fixed for evaluation without gradient-based training. We record Ritz and anchor residuals separately to distinguish eigenspace approximation from alignment accuracy. Section B.4 details the pass accounting, the sketch parameters and the numerical checks.

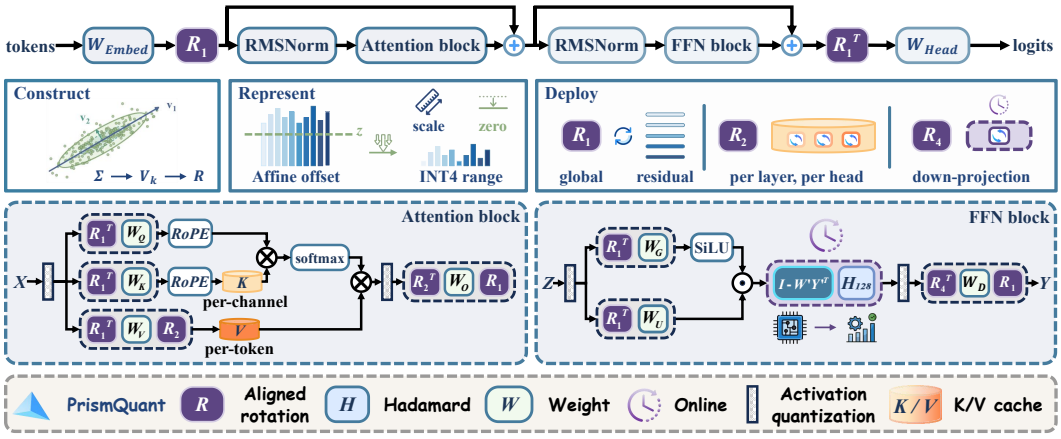

Figure 3: **Overview of PrismQuant.** Calibration takes the leading eigenvectors V_k of the second moment Σ and builds a rotation that turns them into per-group shared levels, which the affine offset z represents outside the INT4 range. R_1 is folded into the embedding, head, and all residual-facing weights; R_2 is applied after W_V and its transpose folded into W_O so the V cache is quantized in the aligned basis; at the down-projection input only the rank-k correction $I - W'Y'^\top$ and the block Hadamard H_{128} run online (clock), with $R_4^\top$ folded into W_D. The schematic uses right-acting row-vector rotations.

3.4 Deployment in the Transformer

For a consuming linear map A, $Ax = (AR^\top)(Rx)$, and $AR^\top$ is precomputed offline. Whether the forward operation Rx disappears depends on the upstream computation, as distinguished in Figure 3.

Residual-stream inputs (R_1). A single rotation R_1 re-expresses the residual stream in one shared basis for the whole network. Because every layer reads from and writes to the same stream, the change of basis is absorbed entirely into the weights: once the RMSNorm gains are folded into the adjacent reading weights, each residual-reading matrix becomes $A_{\text{read}}R_1^\top$, each residual-writing matrix $R_1 A_{\text{write}}$, and the embedding and output head transform accordingly, so this site needs no online rotation. Since R_1 is shared, we calibrate it on a second moment pooled over all layers from the attention-input RMSNorm outputs, weighting every token equally; the resulting rotation is optimal for this pooled objective, even though individual layers would each prefer a different one. Per-layer diagnostics report how much alignment each site retains under the shared rotation.

Down-projection input (R_4). For $h = \text{SiLU}(A_{\text{gate}}x) \odot (A_{\text{up}}x)$, a rotation cannot be moved through the element-wise gate, so R_4 is calibrated on each layer's down-projection input, applied online after the gate, and its transpose folded into $A_{\text{down}}R_4^\top$. Its signed permutation still can be folded: with $T = D\Pi$, setting $A_{\text{gate}} \leftarrow \Pi A_{\text{gate}}$ and $A_{\text{up}} \leftarrow D\Pi A_{\text{up}}$ yields $h_T = Th$ directly, where only the up branch receives signs because SiLU does not commute with sign flips. What remains online is:

$$R_4 h = H_g \widetilde{G} h_T, \qquad \widetilde{G} = TGT^\top, \tag{10}$$

a block Hadamard and a rank-k correction, which a two-kernel Tensor Core implementation executes without any online gather; see Section D and Figure 7 for details.

Values and the KV cache (R_2). Values receive a calibrated head-dimensional rotation R_2 after the value projection, with inverse compensation folded into the corresponding output-projection head blocks, including GQA expansion. At each layer, the value second moment is pooled across tokens and KV heads, and one rotation is shared across those heads. For the 128-dimensional heads used here, a single head is one quantization group, so R_2 aligns one leading direction ($k_{R_2} = 1$). The reference evaluation applies this forward rotation through a value-projection hook.

Keys receive no additional quantization rotation after RoPE. They are quantized per channel across completed groups of 32 tokens; values are quantized per token across each head's feature dimension. The KIVI-style policy retains the current residual key chunk (up to 32 tokens) and the most recent 32 value tokens at full precision (Liu et al., 2024). Thus, keys and values use different grouping axes, while value alignment exploits the per-head affine offset. The nominal rank k controls R_1 and R_4, capped by each site's number of group-constant directions; R_2 uses its head-specific rank above.

3.5 Range and Metadata

The objective controls residual energy, whereas the quantizer responds to range. The two are linked by the deterministic inequality $\sum_j \text{range}(y^{(j)})^2 \le 2\|(I - P_{\mathcal{S}})y\|_2^2$, which bounds the aggregate squared range but does not fix the mean step. Assuming the mixed residual keeps its normalized shape as energy is removed, its range scales with the amplitude $\sqrt{1 - f_k}$, with $f_k = \sum_{i \le k} \lambda_i / \text{Tr}(\Sigma)$ the aligned energy fraction, which motivates the baseline-relative predictor

$$\widehat{s}(g, k) = s_{\text{H}}(g)\sqrt{1 - f_k}, \tag{11}$$

where $s_{\text{H}}(g)$ is the measured mean step of the paired Hadamard reference and no coefficient is fitted. It is an approximation, not a bound: an exact identity (Section C.1) isolates its error in one shape factor, the ratio of residual crest factors after and before alignment, measured near one at the q/k/v input and 0.87–0.92 at the down-projection input.

The same g governs a second resource, **the metadata budget**. With one fp16 scale and offset per group, codes and metadata occupy $4 + 32/g$ bits per value; halving g doubles both the number of local scales and the dimension of $\mathcal{S}$, so finer groups buy resolution and alignment capacity together at a metadata cost, whereas raising k leaves the format budget untouched and costs factor storage **and online work** instead. Extended-affine controls, which add $m - 1$ fixed within-group directions at $4 + 16(m+1)/g$ bits, separate the two: at a shared 4.25-bit budget, one direction per group of 128 beats three per group of 256 **on both Llama models** (Table 6a). Section C derives the law, gives the metadata accounting, and reports the range diagnostics (Figure 4).

Table 1: **W4A4KV4 quantization across the Llama family.** WikiText-2 perplexity (↓) and zero-shot accuracy (%, ↑); Avg. averages the eight displayed tasks. Published rows and the 3B/8B bf16 references follow Wang et al. (2026) (3B) and He et al. (2025) (8B/70B); the 70B reference, Hadamard, and PrismQuant rows are measured in our pipeline. Bold marks the best quantized result per column within each model; **PrismQuant** rows are shaded light for $k = 8$ and darker for $k = \max$.

		Zero-shot accuracy ↑								
Method	**PPL ↓**	ARC-c	ARC-e	BoolQ	HellaS	OBQA	PIQA	SIQA	WinoG	**Avg. ↑**
Llama-3.2-3B (bf16)	7.80	46.20	71.70	73.10	73.70	43.40	77.40	47.20	69.10	62.73
QuaRot (NeurIPS '24)	10.10	38.60	59.00	65.90	66.50	35.80	74.40	43.10	65.20	56.06
SpinQuant (ICLR '25)	9.20	38.90	64.80	68.00	69.10	39.40	74.90	45.10	62.90	57.89
KurTail (EMNLP-F '25)	9.00	42.20	66.70	69.80	68.80	39.80	**75.60**	44.80	64.60	59.04
ResQ (ICML '25)	8.80	43.10	65.60	68.80	70.50	38.40	75.10	45.60	64.80	58.99
OffQ (arXiv '26)	8.78	**44.80**	**70.41**	71.62	71.49	39.60	75.52	45.65	67.32	60.80
Hadamard	9.04	43.00	67.63	66.97	71.18	40.00	74.27	45.60	65.67	59.29
PrismQuant ($k = 8$)	8.60	43.43	69.91	74.07	71.59	40.20	74.97	45.50	66.77	60.80
PrismQuant ($k = \max$)	**8.58**	43.69	70.24	**74.16**	**71.60**	**41.00**	75.08	**45.91**	**68.19**	**61.23**
Llama-3.1-8B (bf16)	6.23	53.50	81.19	82.08	78.90	44.80	81.23	47.19	73.56	67.81
QuaRot (NeurIPS '24)	7.82	46.08	72.31	77.34	74.46	42.40	76.82	44.37	68.51	62.79
SpinQuant (ICLR '25)	7.51	47.87	76.05	76.76	74.63	43.20	79.27	45.70	67.17	63.83
OSTQuant (ICLR '25)	7.40	47.44	75.34	78.84	75.48	41.80	78.24	47.13	68.51	64.10
BASE-Q (arXiv '25)	7.17	49.15	77.19	78.56	76.41	42.40	79.16	45.85	69.22	64.74
Hadamard	7.21	50.34	78.24	80.03	76.13	42.60	**79.33**	46.72	69.53	65.37
PrismQuant ($k = 8$)	6.99	48.89	**79.00**	80.12	**76.95**	**44.20**	78.13	**47.24**	68.35	65.36
PrismQuant ($k = \max$)	**6.92**	**50.77**	77.86	**80.55**	76.72	43.00	78.84	45.50	**70.24**	**65.43**
Llama-3.1-70B (bf16)	2.81	61.09	81.69	87.16	85.66	47.80	84.22	51.79	82.00	72.68
QuaRot (NeurIPS '24)	5.31	57.51	80.35	83.27	81.45	44.80	81.66	45.60	75.77	68.80
SpinQuant (ICLR '25)	4.74	59.13	82.28	84.37	82.24	46.00	82.21	47.24	76.40	69.98
BASE-Q (arXiv '25)	4.17	60.49	**84.13**	83.12	83.10	47.40	83.24	47.85	77.43	70.85
Hadamard	4.22	60.90	83.30	85.00	83.80	47.20	83.20	48.60	78.90	71.36
PrismQuant ($k = 8$)	**3.85**	**61.86**	83.59	**86.73**	**84.64**	**47.60**	**83.79**	**51.02**	**80.43**	**72.46**
PrismQuant ($k = \max$)	3.89	61.52	83.63	86.42	84.59	46.60	83.57	49.08	80.35	71.97

4 EXPERIMENTS

We evaluate W4A4KV4 post-training quantization on Llama-3.2-3B, Llama-3.1-8B, and Llama-3.1-70B, the mixture-of-experts Qwen3-30B-A3B-Base, and, in the appendix, the Qwen3 dense family from 0.6B to 8B and Mistral-7B-v0.3 (Grattafiori et al., 2024; Yang et al., 2025). Activations use the asymmetric group quantizer of Section 3 with $g = 128$ (4.25 bits per value); weights use GPTQ INT4 and the KV cache follows KIVI (Liu et al., 2024). Rotations are applied at the standard sites: R_1 on the residual stream, R_2 after the value projection with its transpose folded into the output projection, and R_4 online before the down projection; GPTQ runs with the rotations in place. The Hadamard baseline places a dense Hadamard at every site; PrismQuant keeps everything else fixed and replaces the Hadamards with the null-space-aligned rotations of Section 3, with k aligned directions per site and $k = \max$ using every group slot. Rotation statistics and GPTQ share 128 calibration sequences of 2048 tokens. All reported PrismQuant results are averaged over three random seeds. We report WikiText-2 and C4 perplexity, zero-shot accuracy on eight tasks, and, for the Qwen3 family, MMLU, MMLU-Redux, and GSM8K. Algorithm 1 (Appendix A.5) lists the full procedure and Appendix A.1 the complete protocol.

4.1 DENSE MODELS

PrismQuant consistently achieves the lowest perplexity among all displayed quantized methods across the three Llama scales (Table 1). Against the metadata-matched Hadamard baseline, it closes $\mathbf{37}\%$, $\mathbf{30}\%$, and $\mathbf{26}\%$ of the perplexity gap to bf16 on 3B, 8B, and 70B, respectively. It also recovers most of the average accuracy loss on 3B and 70B, gaining $\mathbf{1.94}$ points ($59.29 \rightarrow 61.23$) and $\mathbf{1.10}$ points ($71.36 \rightarrow 72.46$). On 70B, it improves all eight tasks over Hadamard and comes within $\mathbf{0.22}$ points of bf16. Compared with published baselines, PrismQuant surpasses OffQ on 3B by $\mathbf{0.43}$ accuracy points with 0.20 lower PPL, and BASE-Q by 0.69 points on 8B and $\mathbf{1.61}$ points with 0.32 lower PPL on 70B. The preferred rank varies with model scale: $k = \max$ yields the best perplexity and average accuracy on 3B and 8B, while $k = 8$ leads on 70B.

Table 2: **W4A4KV4 quantization of Qwen3-30B-A3B-Base.** WikiText-2 and C4 perplexity (↓) and zero-shot accuracy (%, ↑) on the eight-task suite of Table 1. All rows are evaluated in our pipeline with three seeds; the router is kept in bf16, and every expert receives its own R_4 ($k = 6$, its full slot count), so the two **PrismQuant** rows differ only in the rank of R_1. Bold marks the best quantized result per column.

	PPL ↓		Zero-shot accuracy ↑								
Method	WT2	C4	ARC-c	ARC-e	BoolQ	HellaS	OBQA	PIQA	SIQA	WinoG	**Avg.** ↑
Qwen3-30B-A3B (bf16)	6.11	10.70	57.68	79.55	81.41	81.40	44.00	80.96	52.15	72.53	68.71
Hadamard	6.68	11.47	55.38	79.04	81.41	79.24	44.80	79.33	50.26	70.32	67.47
PrismQuant ($k = 8$)	6.45	11.19	**57.25**	79.29	81.90	**80.08**	**45.40**	**80.30**	51.28	**72.61**	**68.52**
PrismQuant ($k = \max$)	**6.42**	**11.18**	55.63	**79.84**	**83.55**	79.60	44.40	79.60	**51.94**	71.27	68.23

Table 3: **Alignment rank.** Activation-only recovery (%, ↑) of the Hadamard-to-bf16 WikiText-2 perplexity gap by PrismQuant at rank k; the first row lists the gap itself. Weights and KV states remain in bf16; ranks are capped per site by the number of group slots. Bold marks the largest recovery; row shading darkens with k.

Configuration	Llama-3.2-3B	Llama-3.1-8B	Qwen3-8B
Hadamard gap, $P_{\mathrm{Had}} - P_{\mathrm{bf16}}$ (PPL)	**+0.154**	**+0.142**	**+0.442**
PrismQuant ($k = 8$)	38.59	35.53	59.12
PrismQuant ($k = 16$)	38.30	35.74	51.97
PrismQuant ($k = 32$)	41.88	40.23	85.57
PrismQuant ($k = 64$)	40.83	41.84	99.85
PrismQuant ($k = \max$)	**42.67**	**43.12**	**99.95**

4.2 Mixture of Experts

The advantage transfers to sparsely routed models (Table 2). Qwen3-30B-A3B routes each token to 8 of 128 experts per layer; we quantize every expert with its own R_4, keep the router in bf16 with R_1 folded into its input so that routing is unchanged, and calibrate cold experts with the shrinkage estimator of Eq. 12 (Appendix A.1). Hadamard costs 0.57 WikiText-2 PPL, 0.77 C4 PPL, and 1.24 accuracy points against bf16; PrismQuant recovers $\mathbf{46}\%$ and $\mathbf{38}\%$ of the two perplexity gaps at $k = \max$ and $\mathbf{1.05}$ of the 1.24 points at $k = 8$, ahead of Hadamard on all eight tasks and only $\mathbf{0.19}$ below bf16, a gain of 2.4 standard errors. The two PrismQuant rows share their expert rotations and differ only in the rank of R_1; their 0.29-point difference is within one standard error.

4.3 Ablation Studies

Alignment rank. A small rank captures most of the gain on Llama, and the optimum is model dependent. Table 3 reports the fraction of the Hadamard-to-bf16 perplexity gap that PrismQuant closes when only the q/k/v and down-projection inputs are quantized. On Llama-3.2-3B and Llama-3.1-8B, $k = 8$ recovers $\mathbf{38.59}\%$ and $\mathbf{35.53}\%$ of the gap, and $k = \max$ only 42.67% and 43.12%. On Qwen3-8B, whose Hadamard gap is three times larger and whose down-projection inputs are dominated by group-constant directions (Appendix E), recovery rises from 59.12% to $\mathbf{99.95}\%$: alignment removes the whole activation-quantization penalty. Recovery is not monotone at intermediate ranks: the alignment objective is solved exactly at every k, but the perplexity it induces is not. We therefore use $k = 8$ as the operating point and report $k = \max$ as the alignment optimum.

Further ablations. Three more ablations, reported in Appendix A.4, probe the remaining design choices: how a fixed activation bit budget should be split between group size and represented directions, whether the moe result depends on how cold experts are calibrated, and which calibration tokens should define the subspace, including the outlier-token selection used by prior work.

5 Conclusion

PrismQuant starts from the structure of the quantizer: the directions a grouped format tolerates best are fixed before any data is seen, and rotation design reduces to a spectral problem with a closed-form optimum, a rank-k Householder realization, and a range law that ties the residual step to unaligned energy and group size. The construction is correspondingly simple, insensitive to how the residual is permuted, how the calibration set is chosen, or how precisely the eigenspace is solved, and it extends without change from dense models to sparsely routed experts. Beyond this work, an INT4 GEMM that consumes group-wise asymmetric activations natively would turn the transform's measured overhead into a checkpoint-level system, and the same alignment applies to any format that scales by group extrema, including block floating-point and microscaling variants.

AI USE STATEMENT

In this work, we used generative AI tools for manuscript drafting and figure preparation. We have not used generative AI tools for generating novel research ideas, writing experimental code, or generating data samples for calibration and benchmarking, and other tasks with required disclosure are not applicable to this work. Additionally, we used generative AI tools for language editing and polishing.

We have reviewed all AI-assisted work. Specifically, the authors manually verified all drafted text for scientific accuracy, reviewed all generated figures to ensure they correctly represent our methodology, and checked all citations and references to prevent hallucinations. We take full responsibility for the final content of this work, including text, claims, results, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies the quantization of existing language models using publicly available checkpoints and benchmark datasets, without conducting new human-subject studies or collecting personal data. Quantization may preserve or alter the biases and unsafe behaviors of the underlying models; improved quantization accuracy should not be interpreted as evidence of safety or fairness. Downstream use should respect applicable model and dataset licenses and include appropriate application-specific safety assessments.

REPRODUCIBILITY STATEMENT

Section 3 describes the quantizer, alignment objective, structured rotation, and deployment procedure. Section B provides the mathematical assumptions and detailed proofs. Section 4 describes the evaluated models, calibration and evaluation protocols, quantization settings, and metrics. Additional metadata-budget and runtime analyses are provided in Section D. The implementation records experimental configurations and includes numerical checks of rotation equivalence and structured-transform correctness.

REFERENCES

Mohammad Sadegh Akhondzadeh, Aleksandar Bojchevski, Evangelos Eleftheriou, and Martino Dazzi. Kurtail: Kurtosis-based llm quantization. In *EMNLP (Findings)*, pp. 17404–17419, 2025.

Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. Quarot: Outlier-free 4-bit inference in rotated llms. *Advances in Neural Information Processing Systems*, 37:100213–100240, 2024.

Ashkan Ashrafi. Walsh–hadamard transforms: a review. *Advances in imaging and electron physics*, 201:1–55, 2017.

Yelysei Bondarenko, Markus Nagel, and Tijmen Blankevoort. Understanding and overcoming the challenges of efficient transformer quantization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7947–7969, 2021.

Mengzhao Chen, Yi Liu, Jiahao Wang, Yi Bin, Wenqi Shao, and Ping Luo. Prefixquant: Eliminating outliers by prefixed tokens for large language models quantization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Patrik Czakó, Gábor Kertész, and Sándor Szénási. Smoothrot: Combining channel-wise scaling and rotation for quantization-friendly llms. In *2025 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 6461–6466. IEEE, 2025.

Herbert A David and Haikady N Nagaraja. *Order statistics*. John Wiley & Sons, 2004.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. Gpt3. int8 (): 8-bit matrix multiplication for transformers at scale. *Advances in neural information processing systems*, 35: 30318–30332, 2022.

Ky Fan. On a theorem of weyl concerning eigenvalues of linear transformations i. *Proceedings of the National Academy of Sciences*, 35(11):652–655, 1949.

Ky Fan. On a theorem of weyl concerning eigenvalues of linear transformations: Ii. *Proceedings of the National Academy of Sciences*, 36(1):31–35, 1950.

Fino and Algazi. Unified matrix treatment of the fast walsh-hadamard transform. *IEEE Transactions on Computers*, 100(11):1142–1146, 1976.

Aryo Pradipta Gema, Joshua Ong Jun Leang, Giwon Hong, Alessio Devoto, Alberto Carlo Maria Mancino, Rohit Saxena, Xuanli He, Yu Zhao, Xiaotang Du, Mohammad Reza Ghasemi Madani, et al. Are we done with mmlu? In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5069–5096, 2025.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53 (2):217–288, 2011.

Liulu He, Shenli Zheng, Karwei Sun, Yijiang Liu, Yufei Zhao, Chongkang Tan, Huanrui Yang, Yuan Du, and Li Du. Base-q: Bias and asymmetric scaling enhanced rotational quantization for large language models. *arXiv preprint arXiv:2506.15689*, 2025.

Xing Hu, Yuan Cheng, Dawei Yang, Zukang Xu, Zhihang Yuan, Jiangyong Yu, Chen Xu, Zhe Jiang, and Sifan Zhou. Ostquant: Refining large language model quantization with orthogonal and scaling transformations for better distribution fitting. *arXiv preprint arXiv:2501.13987*, 2025.

Haokun Lin, Haobo Xu, Yichen Wu, Jingzhi Cui, Yingtao Zhang, Linzhan Mou, Linqi Song, Zhenan Sun, and Ying Wei. Duquant: Distributing outliers via dual transformation makes stronger quantized llms. *Advances in Neural Information Processing Systems*, 37:87766–87800, 2024a.

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. *Proceedings of machine learning and systems*, 6:87–100, 2024b.

Yujun Lin, Haotian Tang, Shang Yang, Zhekai Zhang, Guangxuan Xiao, Chuang Gan, and Song Han. Qserve: W4a8kv4 quantization and system co-design for efficient llm serving. *Proceedings of Machine Learning and Systems*, 7, 2025.

Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. Spinquant: Llm quantization with learned rotations. In *International Conference on Learning Representations*, volume 2025, pp. 92009–92032, 2025.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. Kivi: A tuning-free asymmetric 2bit quantization for kv cache. *arXiv preprint arXiv:2402.02750*, 2024.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

Utkarsh Saxena, Sayeh Sharify, Kaushik Roy, and Xin Wang. Resq: Mixed-precision quantization of large language models with low-rank residuals. *arXiv preprint arXiv:2412.14363*, 2024.

Robert Schreiber and Charles Van Loan. A storage-efficient wy representation for products of householder transformations. *SIAM Journal on Scientific and Statistical Computing*, 10(1):53–57, 1989.

Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Gao Peng, Yu Qiao, and Ping Luo. Omniquant: Omnidirectionally calibrated quantization for large language models. In *International Conference on Learning Representations*, volume 2024, pp. 45472–45496, 2024.

Yuantian Shao, Yuanteng Chen, Peisong Wang, Jianlin Yu, Jing Lin, Zhihui Wei, Jian Cheng, et al. Dartquant: efficient rotational distribution calibration for llm quantization. *Advances in Neural Information Processing Systems*, 38:143936–143970, 2026.

Mingjie Sun, Xinlei Chen, J Zico Kolter, and Zhuang Liu. Massive activations in large language models. *arXiv preprint arXiv:2402.17762*, 2024a.

Yuxuan Sun, Ruikang Liu, Haoli Bai, Han Bao, Kang Zhao, Yuening Li, Jiaxin Hu, Xianzhi Yu, Lu Hou, Chun Yuan, et al. Flatquant: Flatness matters for llm quantization. *arXiv preprint arXiv:2410.09426*, 2024b.

Haoqi Wang, Tong Zhang, and Mathieu Salzmann. Demystifying singular defects in large language models. *arXiv preprint arXiv:2502.07004*, 2025.

Haoqi Wang, Lorenz K Mueller, Jiawei Zhuang, Mathieu Salzmann, and Lukas Cavigelli. Offq: Taming structured outliers in llm quantization by offsetting. *arXiv preprint arXiv:2606.07116*, 2026.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290, 2024.

Xiuying Wei, Yunchen Zhang, Yuhang Li, Xiangguo Zhang, Ruihao Gong, Jinyang Guo, and Xianglong Liu. Outlier suppression+: Accurate quantization of large language models by equivalent and effective shifting and scaling. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 1648–1665, 2023.

Jingyang Xiang and Sai Qian Zhang. Dfrot: Achieving outlier-free and massive activation-free for rotated llms with refined rotation. *arXiv preprint arXiv:2412.00648*, 2024.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. Smoothquant: Accurate and efficient post-training quantization for large language models. In *International conference on machine learning*, pp. 38087–38099. PMLR, 2023.

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

Zhihang Yuan, Lin Niu, Jiawei Liu, Wenyu Liu, Xinggang Wang, Yuzhang Shang, Guangyu Sun, Qiang Wu, Jiaxiang Wu, and Bingzhe Wu. Rptq: Reorder-based post-training quantization for large language models. *arXiv preprint arXiv:2304.01089*, 2023.

Yilong Zhao, Chien-Yu Lin, Kan Zhu, Zihao Ye, Lequn Chen, Size Zheng, Luis Ceze, Arvind Krishnamurthy, Tianqi Chen, and Baris Kasikci. Atom: Low-bit quantization for efficient and accurate llm serving. *Proceedings of Machine Learning and Systems*, 6:196–209, 2024.

A Experimental Details and Additional Results

A.1 Experimental Protocol

Models. We evaluate Llama-3.2-3B, Llama-3.1-8B, and Llama-3.1-70B (Section 4.1), the mixture-of-experts Qwen3-30B-A3B-Base (Section 4.2), the Qwen3 Base family at 0.6B, 1.7B, 4B, and 8B (Appendix A.2), and Mistral-7B-v0.3 (Appendix A.3). Every model runs in fp32 containers holding the released bf16 weights; the bf16 reference rows use the same containers, so quantized and reference rows share one numerical path.

Quantizers. Activations are dynamically quantized to asymmetric INT4 per-group with $g = 128$ at each linear input; each group stores an fp16 scale and an fp16 zero-point for 4.25 bits per value. Weights are quantized with GPTQ to asymmetric INT4: per-group with $g = 128$ (fp16 scale and INT4 zero, 4.16 bits) for Llama-3.2-3B, the Qwen3 family, Qwen3-30B-A3B, and Mistral-7B, and per-channel (4.00 bits) for Llama-3.1-8B and Llama-3.1-70B. On the 70B the Gram products of GPTQ are accumulated in fp64, because the damped down-projection Hessian of its massive-activation layer is indefinite in fp32 by an amount comparable to the damping. The KV cache follows KIVI (Liu et al., 2024): keys are quantized per channel in groups of 32 along the token axis and values per token in groups of 128 along the head axis, both with fp16 metadata; the most recent 32 tokens stay in full precision. At context 2048 this gives 5.17 bits per key and 4.43 bits per value.

Rotations and calibration. Rotations act at the standard sites: R_1 on the residual stream, folded into the embedding, the language-model head, and every linear that reads from or writes to the residual after RMSNorm fusion; R_2 on the value heads, folded into the value and output projections; and R_4 online on the down-projection input. The Hadamard baseline uses random-sign Hadamard matrices at all sites; widths that are not powers of two (2560 and 9728) factor as 128×20 and 128×76 with Paley blocks. PrismQuant replaces R_1 and R_4 by the rank-k rotations of Algorithm 1, with $k = \max$ equal to the number of group slots $n/128$ at each site, and uses a rank-one R_2. Second moments are uncentered and estimated on 128 sequences of 2048 tokens from the WikiText-2 training split; the same sequences serve as the GPTQ calibration set. Before any quantizer is attached, every rotation reconstructs to a relative round-trip error below 10^{-6} and the folded model reproduces its bf16 reference to $|\Delta\mathrm{NLL}| \leq 10^{-5}$ per token.

Evaluation. WikiText-2 perplexity is computed over 141 contiguous windows of 2048 test tokens with fp32 negative log-likelihoods, and C4 perplexity over 256 windows of 2048 tokens from the first validation shard. Zero-shot accuracy uses a pinned revision of lm-evaluation-harness with `acc_norm` on ARC-e, ARC-c, HellaSwag, OBQA, and PIQA and `acc` on BoolQ, SIQA, and WinoGrande; the binomial standard error of the eight-task mean is about 0.43 points. The Qwen3 family follows the Base-model evaluation of the Qwen3 technical report where the harness supports it: MMLU and MMLU-Redux 5-shot (the latter generative with an 8-token answer), GSM8K 4-shot chain-of-thought with 512 new tokens and flexible answer extraction, and ARC-e zero-shot. End-to-end rows use three seeds; the activation-only ablations on Llama average three paired rotation seeds, which vary the random signs, the randomized eigensolver, and the GPTQ sample.

Baselines. The main tables contain two kinds of rows. Hadamard and PrismQuant are run in our pipeline and share every setting above except the rotation: the Hadamard row is the QuaRot construction (Ashkboos et al., 2024), random-sign Hadamard matrices at R_1, R_2, and R_4, placed under our group-128 asymmetric activation grid, and it can therefore exceed the published QuaRot numbers, which quantize activations per token on a symmetric grid. The gain attributed to PrismQuant is measured against the Hadamard row, and the subspace-estimator comparison of Appendix A.4 places the mechanism of the closest prior method inside the same pipeline.

Mixture of experts. Qwen3-30B-A3B-Base has 48 layers with 128 experts of intermediate width 768, top-8 routing with renormalized weights, and no shared expert. R_1 is folded into the router's input columns, so routing logits are unchanged in exact arithmetic; the router stays in bf16 and reads the unquantized post-norm activation. The expert input is quantized once per token in the R_1 basis and shared by the eight routed experts. Each expert receives its own R_4 at $k = 6$, its full slot count, and its own GPTQ Hessian; experts that see fewer than 512 routed calibration tokens (1,346 of 6,144) fall back to the layer-pooled Hessian rotated into the expert basis. Cold-expert covariances use the shrinkage of Eq. 12. With all rotations folded and no quantizer attached, perplexity moves

by less than 10^{-4} and top-8 routing decisions flip at most $1.26\times$ as often as an identity probe run in fp32 ($1.78\times$ for Hadamard). Effective widths are 4.16 bits for weights, 4.25 for activations, and $5.17/4.43$ for keys/values; fused-kernel timing is not reported for this model, since a routed-expert GEMM with per-expert rotations requires its own kernel.

A.2 QWEN3 DENSE FAMILY

Table 4: **W4A4KV4 quantization across the Qwen3 dense family.** All models are Base checkpoints evaluated in our pipeline with three seeds. WT2 and C4 report perplexity (↓); task scores are percentages (↑). MMLU and MMLU-R (Redux) are 5-shot; GSM8K is 4-shot CoT; ARC-e is zero-shot(Raffel et al., 2020; Cobbe et al., 2021; Wang et al., 2024; Gema et al., 2025). Bold marks the best quantized value per model and column, including rounded ties.

	Perplexity ↓		Task score ↑			
Method	WT2	C4	MMLU	MMLU-R	GSM8K	ARC-e
Qwen3-0.6B (bf16)	12.66	19.38	52.52	55.83	60.96	57.95
Hadamard	16.50	25.36	43.46	43.49	28.96	**56.69**
PrismQuant ($k = 8$)	**15.28**	**23.92**	44.22	**46.23**	**31.54**	53.62
PrismQuant ($k = \max$)	15.38	**23.92**	**44.69**	45.27	28.66	56.36
Qwen3-1.7B (bf16)	9.40	15.02	62.68	66.98	72.78	68.48
Hadamard	11.09	17.94	55.81	59.66	58.07	69.11
PrismQuant ($k = 8$)	10.33	16.53	**58.20**	**62.05**	**62.47**	67.47
PrismQuant ($k = \max$)	**10.31**	**16.49**	57.61	61.74	61.71	**72.52**
Qwen3-4B (bf16)	7.93	13.02	73.11	77.09	85.14	76.01
Hadamard	8.76	14.40	69.23	73.55	74.83	74.75
PrismQuant ($k = 8$)	**8.50**	13.96	**69.97**	74.03	**78.92**	**77.06**
PrismQuant ($k = \max$)	8.52	**13.95**	68.75	**75.33**	78.70	74.75
Qwen3-8B (bf16)	8.80	11.66	76.83	81.35	87.11	80.01
Hadamard	11.39	13.08	73.55	77.17	81.73	76.01
PrismQuant ($k = 8$)	**9.73**	12.31	74.39	78.52	**84.31**	80.68
PrismQuant ($k = \max$)	9.85	**12.28**	**74.58**	**79.51**	83.55	**81.06**

Table 4 runs the unchanged W4A4KV4 pipeline across the Qwen3 Base family, so the PrismQuant–Hadamard margin can be read as a function of model size. The smaller the model, the more W4A4KV4 costs and the more of that cost PrismQuant removes in absolute terms: Hadamard's WikiText-2 degradation falls from 30% at 0.6B to 18% at 1.7B and 10% at 4B, the PrismQuant margin from 1.22 to 0.78 and 0.26 PPL, while the fraction of the degradation removed stays between 31% and 46%. Qwen3-8B breaks the trend: Hadamard degrades it by 29%, as much as the 0.6B, and PrismQuant removes 64% of that ($11.39 \to 9.73$). This is the model whose massive-activation layer places most of its down-projection input on a group-constant direction, the direction R_4 absorbs and a Hadamard spreads; the margin therefore tracks how much of the Hadamard degradation is group-constant, not model size per se. Few-shot and generative metrics move with perplexity: at $k = 8$, PrismQuant is above Hadamard on MMLU and GSM8K at every size, by 0.7–2.4 and 2.6–4.4 points. $k = 8$ and $k = \max$ are within 0.12 PPL of each other at every size and trade places across benchmarks.

A.3 MISTRAL-7B-V0.3

Mistral-7B-v0.3 is a model that W4A4KV4 damages little: in Table 5, Hadamard sits 0.19 PPL above bf16 and loses 1.48 eight-task points. PrismQuant at $k = 8$ lowers WikiText-2 perplexity to 5.62 and raises the eight-task mean by 0.38 points, recovering 26% of that loss; $k = \max$ gives the lowest C4 perplexity (8.59). The accuracy margin is inside one standard error of the eight-task mean and is reported as such. Against published rows, 5.62 is below every displayed entry (PrefixQuant 5.76, SpinQuant 5.80, SmoothRot 6.19, QuaRot 6.41) and also below the GPTQ variants that QuaRot and SmoothRot report for perplexity only (5.79 and 5.81) (Ashkboos et al., 2024; Czakó et al., 2025; Liu et al., 2025; Chen et al., 2026), while SpinQuant retains the higher reported eight-task mean (68.60%).

Table 5: **W4A4KV4 quantization on Mistral-7B-v0.3.** WT2/C4 perplexity (↓) and zero-shot accuracy (%, ↑). Avg. 5 averages ARC-c, ARC-e, HellaS, PIQA, and WinoG; Avg. 8 covers all eight tasks. The bf16 reference is ours; QuaRot and SmoothRot rows are their reported RTN results, and cross-paper protocols differ. A dash denotes an unreported result. Bold marks the best displayed quantized result per column.

	PPL ↓		Zero-shot accuracy ↑								Mean ↑	
Method	WT2	C4	ARC-c	ARC-e	BoolQ	HellaS	OBQA	PIQA	SIQA	WinoG	Avg. 5	Avg. 8
Mistral-7B-v0.3 (bf16)	5.45	8.32	54.35	80.13	83.64	80.67	47.40	81.99	47.75	74.19	74.27	68.77
QuaRot (NeurIPS '24)	6.41	10.01	45.22	72.26	—	75.14	—	79.22	—	68.82	68.13	—
SmoothRot (SMC '25)	6.19	9.78	46.59	73.19	—	75.97	—	79.11	—	68.11	68.59	—
SpinQuant (ICLR '25)	5.80	—	52.50	77.30	80.20	79.20	**58.40**	80.30	**48.90**	72.30	72.32	**68.60**
PrefixQuant (TPAMI '26)	5.76	—	50.00	77.74	—	78.39	—	80.41	—	70.88	71.48	—
Hadamard	5.64	8.65	**52.82**	**79.00**	81.96	**79.84**	45.00	80.96	46.32	72.45	73.01	67.29
PrismQuant ($k = 8$)	**5.62**	8.61	52.73	78.83	**82.32**	79.39	46.40	**81.94**	47.24	**72.53**	**73.08**	67.67
PrismQuant ($k = \max$)	5.63	**8.59**	52.47	78.96	81.87	79.27	45.20	81.01	46.57	72.06	72.75	67.18

Table 6: **Ablations of metadata allocation and expert calibration. (a)** Activation-only WikiText-2 PPL (↓) under matched representation budgets; H/PQ denote Hadamard/PrismQuant at $k = \max$, and green parentheses give PQ − H. **(b)** W4A4KV4 WikiText-2 PPL (↓) for the MoE calibration variants at $k = \max$; ΔPPL is relative to shrinkage, from unrounded values. Bold denotes the lower PPL within each H/PQ pair in (a) and the lowest PPL in (b); mint rows mark the configuration adopted in the main tables.

(a) Activation metadata allocation: WikiText-2 PPL

Configuration	**Llama-3.2-3B**		**Llama-3.1-8B**	
(g, m); bits/value	H	**PQ**	H	**PQ**
$(256, 1)$; 4.13	7.79	**7.74** (−0.05)	6.37	**6.31** (−0.06)
$(256, 2)$; 4.19	7.79	**7.73** (−0.06)	6.38	**6.30** (−0.08)
$(256, 3)$; 4.25	7.79	**7.73** (−0.06)	6.37	**6.30** (−0.07)
$(128, 1)$; 4.25 default	7.77	**7.71** (−0.06)	6.35	**6.29** (−0.06)
$(128, 2)$; 4.38	7.77	**7.70** (−0.07)	6.35	**6.29** (−0.06)
$(64, 1)$; 4.50	7.74	**7.68** (−0.06)	6.34	**6.26** (−0.08)

(b) MoE calibration: Qwen3-30B-A3B, W4A4KV4

Calibration scheme	**PPL ↓**	**ΔPPL**
Per-expert covariance; cold experts use Hadamard	6.43	+0.01
Pooled covariance for all experts	6.45	+0.02
PrismQuant with covariance shrinkage default	**6.42**	0.00

A.4 Additional Experiments

This section reports the three ablations summarized in Section 4.3: the allocation of activation metadata and the calibration of sparsely routed experts (Table 6), and the token population that defines the subspace (Table 7). Panel (a) of Table 6 is an activation-only diagnostic with bf16 weights and KV states; panel (b) is the full W4A4KV4 MoE setting, so absolute perplexities are not comparable across panels.

Metadata allocation. Alignment is worth more than the metadata separating our coarsest and finest quantizers, and scale resolution and represented directions are distinct levers. Table 6a varies the group size g and the number m of represented within-group directions, including the constant one, at a budget of $4 + 16(m + 1)/g$ bits per value; every PrismQuant row uses $k = \max$ and three paired seeds. PrismQuant is lower at every matched configuration; at $(256, 1)$, 4.13 bits, it already matches or beats Hadamard at $(64, 1)$, 4.50 bits (7.74 vs. 7.74; 6.31 vs. 6.34). Extra directions help only when energy is aligned into them: Hadamard is flat across m at both group sizes, whereas PrismQuant gains up to 0.01 PPL. At the shared 4.25-bit budget, the default $(128, 1)$ beats $(256, 3)$ on both models, so halving the group is worth more than two additional directions.

Calibrating sparsely routed experts. The gain does not hinge on how cold experts are calibrated. Over the calibration set, $2{,}002$ of the $6{,}144$ experts receive fewer than $2{,}048$ routed tokens, so we

Table 7: **Which tokens define the subspace.** Activation-only WikiText-2 PPL with bf16 weights and KV states, three paired seeds, $k = \max$. **(a)** Both sites quantized; Δ is the paired difference to the all-token estimator with its 90% Student-t interval. **(b)** Δ when only one site is quantized (BOS included); bold marks intervals that exclude zero. **(c)** Diagnostics at the down-projection input: captured group-constant energy f and per-group range relative to Hadamard. Mint rows mark our estimator.

(a) Both sites quantized: WikiText-2 PPL

Subspace estimator	Llama-3.2-3B		Llama-3.1-8B	
	PPL	Δ [90% CI]	PPL	Δ [90% CI]
All tokens (PrismQuant)	**7.705**	0	**6.285**	0
Top-1 token / seq. (OffQ-style)	7.723	+0.017 [−0.002, +0.036]	6.301	+0.016 [+0.012, +0.020]
BOS excluded	7.800	+0.095 [+0.092, +0.098]	6.342	+0.057 [+0.049, +0.065]
Global top 1% of tokens	7.711	+0.006 [−0.012, +0.024]	6.293	+0.008 [+0.002, +0.014]
BOS excluded	7.806	+0.101 [+0.072, +0.129]	6.329	+0.044 [+0.037, +0.052]
Hadamard (no alignment)	7.771	+0.066 [+0.039, +0.093]	6.346	+0.061 [+0.056, +0.066]

(b) One site quantized: Δ PPL vs. all tokens

Subspace estimator	Llama-3.2-3B		Llama-3.1-8B	
	q/k/v	down	q/k/v	down
Top-1 token / seq. (OffQ-style)	+0.002	**+0.012**	**+0.003**	**+0.017**
Global top 1% of tokens	+0.002	**+0.013**	**+0.003**	**+0.009**
Hadamard (no alignment)	**+0.017**	**+0.037**	**+0.018**	**+0.053**

(c) Down-projection input: captured energy and range

Subspace estimator	Llama-3.2-3B		Llama-3.1-8B	
	$f \uparrow$	range/H $\downarrow$	$f \uparrow$	range/H $\downarrow$
All tokens (PrismQuant)	**0.333**	**0.747**	**0.324**	**0.746**
Top-1 token / seq. (OffQ-style)	0.154	0.845	0.152	0.839
Global top 1% of tokens	0.226	0.812	0.215	0.811
Hadamard (no alignment)	0.007	1.000	0.007	1.000

shrink each expert's second moment toward its layer's pooled estimate,

$$\widetilde{\Sigma}_e = \frac{n_e \Sigma_e + n_0 \Sigma_{\text{pool}}}{n_e + n_0}, \qquad n_0 = 2048, \tag{12}$$

where n_e is the expert's routed-token count; the pooled term dominates exactly for the cold experts. Table 6b compares this with a plain Hadamard R_4 for the cold experts and with one pooled covariance for all experts. Shrinkage is best, but the three schemes differ by at most 0.023 PPL, an order of magnitude less than the 0.26 margin over Hadamard: alignment on the well-populated experts carries the gain.

Which tokens define the subspace. PrismQuant estimates its subspace from the uncentered second moment of all calibration tokens, on the premise that the quantizer's free directions are fixed and the only question is where the activation energy lies. A natural alternative is to let extreme tokens define the subspace, as outlier-driven methods do; OffQ (Wang et al., 2026), for instance, keeps only the maximum-L_∞ token of each calibration sequence. This pre-registered ablation tests that choice inside the PrismQuant pipeline with everything else fixed, so the token population is the only variable. Variant A is our estimator; B keeps the top-1 token per sequence (OffQ-style); C keeps the global top 1% of tokens per layer and site; both B and C are also run with BOS (position 0) removed from the candidate set. All variants share the Householder construction, the slot permutations of A, the block Hadamard, $k = \max$, the three paired rotation seeds, the 64 evaluation chunks, and the activation-only protocol of Table 3. B uses a thin SVD of the selected tokens; where fewer than eight directions are identified (7 layer/site estimates on 3B, 5 on 8B) the remainder is completed by a seeded random orthogonal complement. Table 7 reports the results.

Three findings follow. First, the all-token estimator has the lowest perplexity on both models at every site. With both sites quantized, top-1 selection costs $+0.016$ PPL on Llama-3.1-8B with an interval that excludes zero and $+0.017$ on Llama-3.2-3B, where the interval overlaps zero at both sites but excludes it at the down-projection input alone ($+0.012$ $[+0.003, +0.022]$); the global top-1% variant

Table 8: **Calibration and estimation robustness on Qwen3-4B-Base.** Activation-only WikiText-2 (WT2) and C4 PPL, three paired seeds, $k = \max$; Δ is the paired difference to Hadamard with its 90% interval. **(a)** Calibration set size and corpus. **(b)** Randomized eigensolver passes versus exact fp64 eigendecomposition; f is the captured energy at the down-projection input, Δ is relative to the exact solver. Mint rows mark the default.

(a) Calibration set: size and corpus				
Calibration	WT2 PPL	Δ [90% CI]	C4 PPL	Δ [90% CI]
Hadamard (no calibration)	7.877	0	13.160	0
WT2, 64 seq.	7.690	-0.187 [-0.199, -0.174]	12.851	-0.309 [-0.330, -0.289]
WT2, 128 seq. default	7.694	-0.183 [-0.194, -0.171]	12.844	-0.316 [-0.337, -0.295]
WT2, 256 seq.	7.703	-0.174 [-0.185, -0.163]	12.852	-0.309 [-0.330, -0.287]
C4, 128 seq.	7.703	-0.174 [-0.185, -0.162]	12.852	-0.309 [-0.330, -0.287]
(b) Eigensolver accuracy				
Solver	Ritz res.	f (down)	WT2 PPL	Δ vs. exact [90% CI]
Randomized, 1 pass	2.8	0.009	7.752	$+0.059$ [$+0.051$, $+0.067$]
Randomized, 2 passes	0.43	0.290	7.700	$+0.007$ [$+0.000$, $+0.013$]
Randomized, 3 passes default	0.12	0.325	7.697	$+0.004$ [-0.002, $+0.011$]
Exact (fp64)	$< 10^{-14}$	0.334	**7.693**	0
Llama-3.2-3B: 3 passes 7.705 ($f = 0.314$) vs. exact 7.700 ($f = 0.322$), $\Delta = +0.005$ [-0.000, $+0.010$]				

sits between the two. Panel (b) localizes the effect: at the q/k/v input the estimators are within 0.003 PPL of each other, at the down-projection input they separate. Second, the outlier-token estimator relies on the BOS token. Removing BOS from the candidate set costs a further 0.04–0.08 PPL and lands at or above Hadamard's perplexity on both models, and with BOS included the selected token set is rank deficient at several layers. Third, the diagnostics in panel (c) explain why the perplexity gap is nevertheless modest. The top-1 subspace captures less than half the group-constant energy of ours at the down-projection input ($f = 0.15$ vs. 0.33), reduces the per-group range less (0.84 vs. 0.75 of Hadamard's), and its leading eight directions lie $77°$ from ours in principal angle; yet it still recovers about three quarters of the all-token gain over Hadamard on both models. A few dominant directions carry most of the benefit of alignment, and pooling all tokens supplies the remainder.

Calibration and estimation robustness. The gain depends on neither the calibration set nor the accuracy of the eigensolver, only on the captured energy f. Table 8a varies the calibration set for Qwen3-4B-Base at $k = \max$: shrinking or doubling the 128 WikiText-2 sequences moves the WikiText-2 gain by less than 7%, and calibrating on C4 instead retains 95% of it while matching the WikiText-2 calibration on C4 evaluation. The corpus change reassigns 99% of the residual coordinates and lowers the down-projection f from 0.315 to 0.273, yet perplexity barely moves, so the greedy placement is not load-bearing and modest losses of captured energy are tolerated. Panel (b) varies the randomized eigensolver: a single pass captures almost nothing ($f = 0.009$) and costs 0.059 PPL, two passes recover $f = 0.290$, and the default three passes are within seed noise of an exact fp64 eigendecomposition on both models. Principal angles between the randomized and exact subspaces stay near $90°$ throughout, because the trailing directions of a 76-dimensional leading eigenspace are nearly degenerate; perplexity nevertheless tracks f monotonically, which is the quantity the objective controls.

Does the gain need the offset? At $g = 128$ the gain does not require the affine offset; the offset is what makes the aligned level free, but flatness and localization carry most of the perplexity. Table 9a swaps the quantizer while keeping the rotation bit-identical: under symmetric per-group INT4, where no offset exists, PrismQuant's margin over Hadamard is unchanged on both models (-0.194 vs. -0.179 on Qwen3-4B-Base, -0.060 vs. -0.061 on Llama-3.2-3B). The reason is what the rotation does to the leading directions themselves: a dense Hadamard turns a distributed eigenvector into a Gaussian-like vector whose largest coordinate is several times its r.m.s., whereas PrismQuant makes it exactly uniform within one group and zero elsewhere, so any quantizer that scales by a group extremum sees a smaller range whether or not it can absorb the level. Panel (b) isolates the offset inside the asymmetric format by moving the anchors to a nonconstant Walsh column of the same group, which keeps the aligned level flat and confined to one group but exposes it to the range. The cost is 16–25% of the total gain, and the same move is invisible under symmetric quantization, where the level was never absorbed; the anchor groups' measured range grows by only 11–19%, because the level and the residual extremes do not add linearly. The offset dominates when a group is the whole token: with one offset per token ($g = d$, $k = 1$) it accounts for 37% (Qwen) and

Table 9: **Quantizer format and target column.** Activation-only WikiText-2 PPL, three paired seeds, both sites quantized. **(a)** The same PrismQuant rotation ($k = \max$; $k = 1$ for $g = d$) under four quantizers; Δ is PQ minus Hadamard, and every 90% interval excludes zero. **(b)** Anchors moved from the constant Walsh column to a nonconstant one at $g = 128$, $k = \max$; Δ is relative to the constant column under the asymmetric and symmetric formats, and s is the offset-specific share $\Delta_{\text{asym}}/(\text{PPL}_{\text{Had}} - \text{PPL}_{\text{PQ}})$. Mint rows mark the configuration of the main tables.

(a) Quantizer format

Quantizer	Qwen3-4B-Base			Llama-3.2-3B		
	Had	PQ	Δ	Had	PQ	Δ
Asymmetric, $g = 128$ default	7.877	**7.697**	-0.179	7.767	**7.705**	-0.061
Symmetric, $g = 128$	7.958	**7.764**	-0.194	7.819	**7.759**	-0.060
Asymmetric, $g = d$ ($k = 1$)	8.075	**7.857**	-0.218	7.926	**7.886**	-0.041
Symmetric, $g = d$ ($k = 1$)	8.178	**8.041**	-0.137	8.008	**7.995**	-0.013

(b) Target column at $g = 128$, $k = \max$

Anchor target	Qwen3-4B-Base			Llama-3.2-3B		
	Δ_{asym}	Δ_{sym}	s	Δ_{asym}	Δ_{sym}	s
Constant column default	0	0	—	0	0	—
Sequency-1 column ($\pm$ halves)	$+0.029$	+0.009	0.16	$+0.015$	+0.005	0.25
Alternating column	$+0.023$	+0.001	0.13	$+0.013$	+0.004	0.22

Table 10: **Anchor placement, fillers, and signs on Qwen3-4B-Base.** Activation-only WikiText-2 PPL, three paired seeds, G identical across rows; Δ is relative to the default at the same rank with its 90% interval. Hadamard: 7.877.

Variant	$k = 8$		$k = \max$	
	PPL	Δ [90% CI]	PPL	Δ [90% CI]
Balanced Π, low-energy fillers default	7.707	0	7.697	0
Identity Π	7.710	+0.002 [-0.004, $+0.009$]	7.697	-0.001 [-0.007, $+0.006$]
Random Π	7.708	+0.001 [-0.005, $+0.007$]	7.694	-0.003 [-0.011, $+0.004$]
High-energy fillers	**7.698**	-0.009 [-0.015, -0.003]	7.697	0 (identical)
Fixed signs $D = +1$	7.724	$+0.017$ [$+0.011$, $+0.023$]	7.697	-0.001 [-0.007, $+0.005$]

69% (Llama) of the gain. Finally, quantizing only the 0.15% of positions flagged as BOS or massive (top 0.1% residual-stream L_∞ at the layer with the largest group-constant share) yields 20% and 26% of the full gain on the two models, and quantizing the remaining 99.85% yields the rest, with the two parts adding to the all-token result within 0.012 PPL; the tokens with the largest crest factor benefit most, as flatness predicts, while the offset-specific part of the gain is not concentrated on them. The split also shows how PrismQuant interacts with prefix-based outlier handling (Chen et al., 2026): with BOS and massive positions kept in full precision, an idealized form of that approach, PrismQuant retains 72–74% of its margin over Hadamard, so the two are largely complementary. Together with the subspace-estimator ablation above, this fixes the reading of the objective: the group-constant directions are the ones a per-group quantizer tolerates best, and the offset is the third of three reasons.

Construction ablations. The signed permutation is not load-bearing for perplexity. Table 10 varies it on Qwen3-4B-Base with the Householder factor G fixed. Energy balancing does what it is designed to do, equalizing residual energy across groups (max/median spread 1.0 versus 3.4 for the identity permutation at the down-projection input), yet identity and random permutations reach the same perplexity within seed noise at both ranks: per-group scales already absorb the imbalance that balancing removes. At $k = 8$, filling the unused constant slots with the highest-energy coordinates instead of the lowest-energy ones improves perplexity by 0.009, because those slots are as tolerant as the anchored ones (Appendix A.4); we keep the low-energy fillers in the main tables so that the rank sweep of Table 3 measures eigenspace alignment alone, and note the improvement as available at reduced rank. Random signs matter only at reduced rank: fixing $D = +1$ costs 0.017 at $k = 8$ and nothing at $k = \max$, where every constant slot is eigen-aligned and the sign pattern no longer correlates the fillers across groups.

A.5 Algorithm

Algorithm 1 lists the complete procedure. Stage I computes, for every site s, the null-space-aligned rotation from uncentered second moments: the top-k_s eigendirections, with k_s the site's rank capped by its number of group slots, are mapped onto the first coordinate of k_s distinct groups by a sequence

of Householder reflections held in compact-WY form; a signed permutation then places the remaining coordinates, and the block Hadamard turns each anchored coordinate into that group's constant direction. Stage II folds R_1 and R_2 into the surrounding weights and registers the online R_4 as a rank-k_s correction inside the down-projection input; the post-RoPE query/key rotation of QuaRot, R_3, is the identity here because keys are quantized per channel without rotation (Section 3.4), and it is listed only so that the four-site convention is complete. Stage III quantizes the rotated weights with GPTQ and attaches the activation and KV quantizers; Stage IV is the grouped asymmetric INT4 quantizer, whose fp16 offset represents the group-constant level.

B Theoretical Foundations, Proofs, and Additional Method Details

This appendix collects the proofs and implementation details supporting Section 3; the range law, its metadata accounting, and its empirical diagnostics are in Section C. Exact statements assume real arithmetic unless finite precision is explicitly discussed. Orthogonal maps act on column vectors; row-batched activations are transformed by right multiplication with the transpose.

B.1 Group-Constant Geometry and Metadata Precision

For a group $y \in \mathbb{R}^g$, let $\Delta(y) = \max_i y_i - \min_i y_i$. For every scalar c,

$$\min_i (y_i + c) = \min_i y_i + c, \qquad \Delta(y + c\mathbf{1}_g) = \Delta(y). \tag{13}$$

Consequently, for the ideal affine quantizer with $z(y) = \min_i y_i$ and $s(y) = \Delta(y)/15 > 0$,

$$q(y + c\mathbf{1}_g) = q(y), \qquad \hat{y}(y + c\mathbf{1}_g) = \hat{y}(y) + c\mathbf{1}_g. \tag{14}$$

A constant group is represented by its offset, with $s = 1$ and $q = 0$.

The embedded group indicators have disjoint supports and unit norm, so $U^\top U = I_M$. The restriction of $UU^\top y$ to group j is its arithmetic mean repeated g times. Subtracting this component is a group-common shift, which proves Equation (3). The projection is an analytical decomposition, not an additional mean-subtraction operation in the algorithm. In particular, the stored min-based offset need not equal the group mean: if $y^{(j)} = c_j \mathbf{1}_g + r^{(j)}$, then $z_j = c_j + \min_i r_i^{(j)}$.

The implementation uses $\tilde{s} = \mathrm{fp16}(s)$ and $\tilde{z} = \mathrm{fp16}(z)$ before computing codes. In general, $\mathrm{fp16}(z + c) \neq \mathrm{fp16}(z) + c$, so exact code invariance does not extend to arbitrary shifts with finite-precision metadata. The underlying range identity remains valid in real arithmetic; metadata rounding is a separate numerical error source. Zero or underflowed scales are handled by the implementation's nonzero-scale guard. These qualifications are why the main text does not describe the offsets as an unlimited, exactly lossless storage channel.

A four-feature illustration. Mapping $v_1 = (1, -1, 1, -1)^\top/2$ to $u_1 = (1, 1, 1, 1)^\top/2$ sends the dominant component $(10, -10, 10, -10)^\top$ to $(10, 10, 10, 10)^\top$. Its energy is unchanged, but its contribution to range falls from 20 to zero. With transformed residual $(-0.20, -0.03, 0.03, 0.20)^\top$, the quantizer receives $(9.80, 9.97, 10.03, 10.20)^\top$, with ideal offset 9.80 and step $0.40/15$. The gain comes from rotating a varying pattern into a shared level, not translating an unchanged group. The group mean is 10, illustrating why it differs from the stored min-based offset.

Residual energy bounds quantization range. Let $e = (I - P_{\mathcal{S}})y$. For any nonconstant group $e^{(j)}$, choose distinct indices p and q attaining its maximum and minimum. Then

$$\mathrm{range}(e^{(j)})^2 = (e_p^{(j)} - e_q^{(j)})^2 \leq 2\big((e_p^{(j)})^2 + (e_q^{(j)})^2\big) \leq 2\|e^{(j)}\|_2^2. \tag{15}$$

The constant case is immediate. Summing over groups and using Equation (3) proves Equation (26). The same argument applies to $e_k = (I - U_k U_k^\top)y$, because the removed component is group-constant even when $k < M$. Thus maximizing $\mathcal{J}_k$ minimizes the corresponding residual-energy upper bound, not necessarily the realized range.

Algorithm 1 PrismQuant: structured alignment and grouped INT4

Input: Model $\mathcal{M}$ with L layers; calibration data $\mathcal{D}$; group size $g = d_h = 128$; nominal rank k.
Output: W4A4KV4 model $\widehat{\mathcal{M}}$ and structured rotation factors $\{W_s, Y_s, \Pi_s, D_s\}_{s\in\mathcal{J}}$.

I. Spectral alignment: Householder / compact-WY

$\mathcal{J} \leftarrow \{1\} \cup \{(2,\ell),(4,\ell) : \ell = 1,\dots,L\}$ — residual, value, down-input sites
$\{\Sigma_s\}_{s\in\mathcal{J}} \leftarrow \textsc{CollectMoments}(\mathcal{M},\mathcal{D})$, $\Sigma_s = N_s^{-1} X_s^\top X_s$ — uncentered
$k_1 \leftarrow \min(k, d/g)$, $k_{2,\ell} \leftarrow 1$, $k_{4,\ell} \leftarrow \min(k, d_{\mathrm{ff},\ell}/g)$ — per-site rank k_s, capped by the slot count d_s/g
for all $s \in \mathcal{J}$ **do**
  $[v_{s,1},\dots,v_{s,k_s}] \leftarrow \mathrm{TopEig}_{k_s}(\Sigma_s)$, $W_s, Y_s \leftarrow [\,],[\,]$
  for $i = 1,\dots,k_s$ **do** — anchor $t_i = 1 + (i-1)g$: first coordinate of group i
    $\delta_i \leftarrow v_{s,i} - W_s(Y_s^\top v_{s,i}) - e_{t_i}$
    if $\|\delta_i\|_2 \geq 10^{-7}$ **then** — skip if already anchored
      $h_i \leftarrow \delta_i/\|\delta_i\|_2$
      $W_s \leftarrow [\,W_s - 2h_i(h_i^\top W_s),\ 2h_i\,]$, $Y_s \leftarrow [\,Y_s,\ h_i\,]$
    end if
  end for
  $G_s \equiv I - W_s Y_s^\top$, $\epsilon_s \leftarrow \mathrm{diag}(G_s \Sigma_s G_s^\top)$ — per-coordinate energy after G_s
  $\Pi_s \leftarrow \textsc{AnchorBalance}(\epsilon_s, k_s, g)$, $D_s \leftarrow \mathrm{diag}(\xi_s)$, $\xi_s \in \{-1,+1\}^{d_s}$ — anchors fixed; seeded signs
  $R_s \equiv H_g D_s \Pi_s (I - W_s Y_s^\top)$ — $R_s v_{s,i} = \pm u_i$, $u_i = \mathbf{1}_{\mathcal{I}_i}/\sqrt{g}$
end for

II. Fold weights and register R_1, R_2, R_3, R_4

$\mathcal{M} \leftarrow \textsc{FuseRMSNorm}(\mathcal{M})$, $R_{3,\ell} \leftarrow I$ $(\forall \ell)$ — no post-RoPE query/key rotation; keys quantized per channel
$E \leftarrow R_1 E$, $A_{\mathrm{lm}} \leftarrow A_{\mathrm{lm}} R_1^\top$ — embedding and head
$A_{\mathrm{read}} \leftarrow A_{\mathrm{read}} R_1^\top$, $A_{\mathrm{write}} \leftarrow R_1 A_{\mathrm{write}}$ — every linear reading from or writing to the residual
for $\ell = 1,\dots,L$ **do**
  $A_{\mathrm{o},h}^{(\ell)} \leftarrow A_{\mathrm{o},h}^{(\ell)} R_{2,\ell}^\top$ $(\forall h)$, $A_{\mathrm{down}}^{(\ell)} \leftarrow A_{\mathrm{down}}^{(\ell)} R_{4,\ell}^\top$ — R_2 into the output projection, $R_4^\top$ into the down projection
  $T_\ell \leftarrow D_{4,\ell}\Pi_{4,\ell}$, $A_{\mathrm{gate}}^{(\ell)} \leftarrow \Pi_{4,\ell} A_{\mathrm{gate}}^{(\ell)}$, $A_{\mathrm{up}}^{(\ell)} \leftarrow T_\ell A_{\mathrm{up}}^{(\ell)}$ — signed permutation folded; signs only on the up branch
  $\widetilde{W}_\ell \leftarrow T_\ell W_{4,\ell}$, $\widetilde{Y}_\ell \leftarrow T_\ell Y_{4,\ell}$ — $\widetilde{G}_\ell = T_\ell G_{4,\ell} T_\ell^\top$
end for
$\mathsf{V}_\ell(v) \equiv R_{2,\ell} v$ — applied after the value projection, shared across KV heads
$\mathsf{D}_\ell(h_T) \equiv H_g\big[h_T - \widetilde{W}_\ell(\widetilde{Y}_\ell^\top h_T)\big]$ — online R_4: rank-k_s correction and block Hadamard
$\mathcal{M}_{\mathrm{rot}} \leftarrow \textsc{AttachRotations}(\mathcal{M}, \{\mathsf{V}_\ell, \mathsf{D}_\ell\}_\ell)$

III. Quantize weights offline; activations and KV online

$\widehat{\mathcal{M}} \leftarrow \mathrm{GPTQ}_{4,g,\mathrm{asym}}(\mathcal{M}_{\mathrm{rot}}; \mathcal{D})$ — A/KV quantization off during GPTQ
$\mathsf{A}(y) \equiv \mathcal{Q}_{4,g}(y)$ — each quantized linear input
KV : $\widehat{K}_{\mathrm{old}} \leftarrow \mathcal{Q}_{4,32}^{\mathrm{time}}(K_{\mathrm{old}})$, $\widehat{V}'_{\mathrm{old}} \leftarrow \mathcal{Q}_{4,g}^{\mathrm{head}}(V'_{\mathrm{old}})$ — keys per channel over 32 tokens; values per token over the head
$K_{\text{current chunk}}$, $V'_{\text{last }32}$: full precision
return $\textsc{AttachQuantizers}(\widehat{\mathcal{M}}, \mathsf{A}, \mathsf{KV})$

IV. Grouped asymmetric INT4 with fp16 scale and offset

function $\mathcal{Q}_{4,g}(y)$
  for all token/group vectors $y^{(j)} \in \mathbb{R}^g$ **do**
    $z_j \leftarrow \mathrm{fp16}(\min y^{(j)})$, $s_j \leftarrow \mathrm{fp16}\Big(\frac{\max y^{(j)} - \min y^{(j)}}{15}\Big)$ — a group-constant level enters z_j, not s_j
    $s_j \leftarrow 1$ if $s_j \leq 0$ — positive fallback scale
    $q^{(j)} \leftarrow \mathrm{clip}_{[0,15]}\Big(\mathrm{round}\Big(\frac{y^{(j)} - z_j \mathbf{1}_g}{s_j}\Big)\Big)$
    $\widehat{y}^{(j)} \leftarrow s_j q^{(j)} + z_j \mathbf{1}_g$
  end for
  return $\widehat{y}$ — QDQ form; codes and metadata are (q, s, z)
end function

Table 11: **fp16 offset rounding at fixed codes.** PrismQuant at $g = 128$, $k = \max$, activation-only, three seeds. Offset error is $|z_j - \mathrm{fp16}(z_j)|$ in units of the group's step; the unaligned cost is $2|c|/15$ for the same level c. ΔPPL rows re-store only the offset at the stated precision.

	Qwen3-4B-Base	**Llama-3.2-3B**
$\lvert z\rvert/\mathrm{range}$, median / P99 / max (worst layer)	0.51 / 10.0 / 21.5	0.52 / 5.8 / 10.2
Offset error / step, median / P99 (worst layer)	0.0012 / 0.025	0.0012 / 0.017
Groups with error > 0.5 step, overall	$8 \times 10^{-5}\%$	$5 \times 10^{-5}\%$
worst layer/site	0.003%	0.0008%
Median (offset error)/(unaligned cost)	0.0085	0.0093
ΔPPL, fp32 offset	$+0.0008$	$+0.0007$
ΔPPL, bf16 offset	$+0.0007$	$+0.0007$

Under ideal min–max quantization, nearest-grid rounding incurs coordinate error at most $s_j/2$. Consequently,

$$\|\hat{y} - y\|_2^2 \le \frac{g}{4 \cdot 15^2} \sum_{j=1}^{M} \mathrm{range}(y^{(j)})^2 \le \frac{g}{2 \cdot 15^2} \|(I - P_{\mathcal{S}})y\|_2^2. \tag{16}$$

These are worst-case bounds in ideal arithmetic, not equality statements or population predictions. Finite-precision metadata and subsequent weight quantization require separate treatment.

Offset precision. Range-nullity is exact under ideal affine quantization; the stored fp16 offset carries a relative rounding error of at most 2^{-11}, which grows with the aligned level. Table 11 measures this error at fixed codes: for every group of the activation-only runs of Appendix A.4, the offset is re-stored in fp32 or in bf16 (seven fraction bits) while codes and scales are held bit-identical, and the rounding error of the fp16 offset is compared with the quantization step of its group. Aligned levels reach ten to twenty times the within-group range at the first layers, yet the rounding error has a median of 0.001 steps, a 99th percentile of 0.025 steps, and exceeds half a step in fewer than one group in a million; the only such groups are BOS positions at the first down-projection input. Relative to the step that the same level would cost if it entered the range unaligned, $2|c|/15$, the rounding error is about one percent. Perplexity is insensitive to the offset's precision: fp32 and even bf16 offsets change it by less than 0.001, for PrismQuant and Hadamard alike. The level is therefore not stored losslessly, but its storage error is two orders of magnitude below the alternative and invisible end to end.

B.2 Proof of Optimal Spectral Alignment

Let $B = R^\top U_k$. Orthogonality of R implies $B^\top B = I_k$, and

$$\mathcal{J}_k(R) = \mathrm{Tr}(B^\top \Sigma B) = \sum_{i=1}^{d} \lambda_i \gamma_i, \qquad \gamma_i = \|B^\top v_i\|_2^2. \tag{17}$$

Because $BB^\top$ is a rank-k orthogonal projector, $0 \le \gamma_i \le 1$ and $\sum_i \gamma_i = k$. Since the eigenvalues are nonincreasing, allocating these k units of mass to the largest eigenvalues gives

$$\mathcal{J}_k(R) \le \sum_{i=1}^{k} \lambda_i. \tag{18}$$

Equality holds when $\mathrm{span}(B) = \mathrm{span}(V_k)$, which is achieved by an orthogonal map carrying V_k onto U_k. This proves Proposition 1, a direct application of the Ky Fan maximum principle (Fan, 1949; 1950). Eigenvalue ties can yield multiple maximizers; no uniqueness is asserted.

For $k = M$, energy preservation gives

$$\min_{R^\top R = I} \mathbb{E}\|(I - P_{\mathcal{S}})Rx\|_2^2 = \mathrm{Tr}(\Sigma) - \sum_{i=1}^{M} \lambda_i. \tag{19}$$

For $k < M$, optimality applies to the selected target U_k, not the unrestricted M-dimensional target. If calibration returns an orthonormal approximation $\widehat{V}_k$, exact alignment of that approximation captures $\mathrm{Tr}(\widehat{V}_k^\top \Sigma \widehat{V}_k)$ in the selected target. The discrepancy from the spectral optimum is due to eigenspace estimation, not the Householder representation. Likewise, optimizing the empirical calibration moment is distinct from guaranteeing the population optimum.

B.3 Householder Construction and Compact Representation

Let $t_i = 1 + (i-1)g$ and $e_{t_1}, \ldots, e_{t_k}$ be distinct coordinate anchors; set $G_0 = I$. At step i, write $b_i = G_{i-1}v_i$. If $b_i = e_{t_i}$, skip the reflection. Otherwise, define

$$h_i = \frac{b_i - e_{t_i}}{\|b_i - e_{t_i}\|_2}, \qquad \mathcal{H}_i = I - 2h_i h_i^\top, \qquad G_i = \mathcal{H}_i G_{i-1}. \tag{20}$$

The vectors b_i and e_{t_i} have unit norm, giving $\mathcal{H}_i b_i = e_{t_i}$. For $\ell < i$, orthogonality implies $b_i^\top e_{t_\ell} = v_i^\top v_\ell = 0$. Distinct anchors also satisfy $e_{t_i}^\top e_{t_\ell} = 0$. Hence h_i is orthogonal to every previously aligned anchor, and $\mathcal{H}_i$ preserves them. Induction yields $G_k v_i = e_{t_i}$ for every $i \le k$.

Choose $\Pi e_{t_i} = e_{p_i}$, where p_i is the first coordinate of group j_i. A diagonal sign matrix maps this anchor to $\pm e_{p_i}$, and the normalized block Hadamard maps it to $\pm u_{j_i}$. Therefore $H_g D \Pi G_k v_i = \pm u_{j_i}$, proving that the structured construction attains the selected-subspace optimum for exact eigenvectors. Balancing the other coordinates does not change these anchor constraints.

To obtain compact factors, suppose $G_{i-1} = I - W_{i-1}Y_{i-1}^\top$. Then

$$G_i = I - 2h_i h_i^\top - \mathcal{H}_i W_{i-1} Y_{i-1}^\top = I - W_i Y_i^\top, \tag{21}$$

where

$$W_i = [\mathcal{H}_i W_{i-1},\ 2h_i], \qquad Y_i = [Y_{i-1},\ h_i]. \tag{22}$$

Starting with empty factors proves a representation with at most k columns, consistent with compact Householder representations (Schreiber & Van Loan, 1989). Thus $Gx = x - W(Y^\top x)$ and, for row batches, $XG^\top = X - (XY)W^\top$. The orthogonal matrix G is full rank; it is the correction $I - G$ that has rank at most k.

Coordinate placement and finite-precision checks. The selected anchors occupy the first coordinates of the first k groups. When $k < M$, unused first-coordinate slots receive the lowest-energy available coordinates, using energies estimated after G from the calibration data. Remaining coordinates are sorted by decreasing energy and assigned greedily to the group with the least accumulated residual energy, subject to its $g - 1$ non-anchor slots. Ties are deterministic. This preserves the selected anchor constraints and avoids unintentionally allocating high-energy coordinates to unused constant slots in reduced-rank comparisons. It is a residual-balancing rule, not a separate guarantee of optimal group range.

The implementation skips a reflection when $\|b_i - e_{t_i}\|_2 < 10^{-7}$. The induction above establishes exact alignment for orthonormal inputs in real arithmetic; the threshold, estimated eigenspace, and floating-point products introduce separately recorded numerical residuals. Signs may change u_{j_i} to $-u_{j_i}$ without changing either its span or captured energy.

B.4 Calibration and Numerical Approximation

Wide activation sites use a randomized sketch with oversampling 16 rather than an explicitly formed $d \times d$ second moment. Two second-moment applications with orthogonalization are followed by a third pass for Rayleigh–Ritz extraction and residual checks. Permutation-energy collection is an additional calibration step, not part of the three eigensolver passes. For sketch width ℓ, streamed products cost $\mathcal{O}(N_{\text{cal}} d\ell)$ per pass, in addition to orthogonalization and the smaller projected eigendecomposition. Head-dimensional value moments are small enough to form explicitly and diagonalize directly.

Calibration is site-specific. The R_1 moment pools the outputs of attention-input RMSNorms across layers, with equal token weighting. Each R_4 uses that layer's gated down-projection inputs. Each value moment pools tokens and KV heads within its layer; the resulting R_2 is shared across those heads. A global pooled R_1 objective must not be confused with a separate per-layer optimum inferred from activation diagnostics.

All factors and seeded signs are fixed before evaluation, without gradient-based training. Ritz residuals assess eigenspace estimation, while anchor and rotation checks assess its realization. Calibration settings are fixed by the experimental protocol; rank controls alignment capacity and online cost. These checks do not convert the empirical calibration optimum into a guaranteed optimum for unseen activations.

B.5 Gaussian Range Model and the Two-Factor Predictor

Consider a conditional model for a group after mixing, $y = c\mathbf{1}_g + \sigma Z$, where the coordinates of Z are independent standard normal variables. The common component does not affect range, so

$$\mathbb{E}[\Delta(y) \mid c, \sigma] = \sigma\eta_g, \qquad \eta_g = 2\mathbb{E}\max_{i\le g} Z_i. \tag{23}$$

Symmetry gives the second identity. If Φ and ϕ denote the standard normal distribution and density, the maximum has distribution $\Phi(t)^g$, and therefore

$$\eta_g = 2g\int_{-\infty}^{\infty} t\phi(t)\Phi(t)^{g-1}\,dt. \tag{24}$$

The ideal mean quantization step is $\sigma\eta_g/15$. Assuming comparable coordinate fluctuation scale across group sizes gives Equation (33); the factor η_g is determined by g, without fitting a regression coefficient.

C Range Law and Metadata Budget

C.1 Range Prediction

From residual energy to range. The alignment objective controls residual energy, whereas the quantizer responds to within-group range. For a group $v \in \mathbb{R}^g$, define $e = v - \bar{v}\mathbf{1}_g$, where $\bar{v} = g^{-1}\mathbf{1}_g^\top v$. Subtracting a shared level leaves the range unchanged. If p and q index a maximum and a minimum, respectively, then

$$\mathrm{range}(v)^2 = (e_p - e_q)^2 \le 2(e_p^2 + e_q^2) \le 2\|e\|_2^2, \tag{25}$$

with the constant-group case immediate. Summing over groups yields

$$\sum_{j=1}^{M} \mathrm{range}(y^{(j)})^2 \le 2\|(I - P_{\mathcal{S}})y\|_2^2. \tag{26}$$

This deterministic inequality requires no distributional assumption. It motivates residual energy as a surrogate for controlling range: reducing the right-hand side tightens the bound but does not guarantee a decrease in every realized group range. Predicting the typical range additionally requires a model of the residual distribution.

Energy factor. Let $\mathcal{E} = \mathrm{Tr}(\Sigma) > 0$ and $f_k = \sum_{i\le k}\lambda_i/\mathcal{E}$ be the deliberately aligned energy fraction in the ideal construction. The decomposition in Equation (4) and orthogonality of R give

$$\mathbb{E}\|Rr\|_2^2 = \mathbb{E}\|r\|_2^2 = (1 - f_k)\mathcal{E}. \tag{27}$$

Since $RV_k a = U_k a \in \mathcal{S}$, $(I - P_{\mathcal{S}})Rx = (I - P_{\mathcal{S}})Rr$, and hence

$$\mathbb{E}\sum_{j=1}^{M} \mathrm{range}((Rx)^{(j)})^2 \le 2(1 - f_k)\mathcal{E}. \tag{28}$$

At $k = M$, Rr lies entirely outside $\mathcal{S}$. At $k < M$, unused constant directions may absorb additional residual energy, so f_k need not equal the total fraction captured by $\mathcal{S}$. For an estimated eigenspace, the corresponding captured fraction is $\mathrm{Tr}(\widehat{V}_k^\top \Sigma \widehat{V}_k)/\mathcal{E}$; the eigenvalue sum is exact for the moment whose leading eigenvectors are used.

Under the approximation that mixed residuals retain comparable normalized shapes and their scales change proportionally across matched token–group observations, their amplitude, and hence range, decreases as $\sqrt{1 - f_k}$. Aggregate energy alone does not imply this scaling of mean range. Hadamard mixing motivates the approximation but does not guarantee it for arbitrary activations.

What the inequality does bound. Define the mean ideal INT4 step of a rotation R as $\bar{s}_R = \frac{1}{15M}\mathbb{E}_x \sum_j \mathrm{range}((Rx)^{(j)})$. Cauchy–Schwarz over groups and Jensen's inequality applied to Equation (28) give the distribution-free bound

$$\bar{s}_R \le \frac{1}{15}\sqrt{\frac{2(1-f_k)\mathcal{E}}{M}}. \tag{29}$$

Its prefactor is set by the total energy, not by the measured Hadamard step; replacing it with $s_{\mathrm{H}}(g)$ is the modeling step of the predictor below, not a consequence of the inequality.

Exact decomposition. Let $\sigma_R^2 = \mathbb{E}\|(I - P_{\mathcal{S}})Rx\|_2^2/d$ be the residual energy per coordinate under R, f_R the fraction of energy in $\mathcal{S}$ under R, and $\kappa_R = 15\,\bar{s}_R/\sigma_R$ the mean group range in units of the residual r.m.s., a crest factor. By definition,

$$\frac{\bar{s}_{\mathrm{PQ}}}{\bar{s}_{\mathrm{H}}} = \frac{\kappa_{\mathrm{PQ}}}{\kappa_{\mathrm{H}}}\sqrt{\frac{1-f_{\mathrm{PQ}}}{1-f_{\mathrm{H}}}}, \tag{30}$$

with no assumption. At full capacity and exact alignment $f_{\mathrm{PQ}} = f_k$, and an isotropic Hadamard captures $f_{\mathrm{H}} \approx 1/g$. The predictor is the special case $\kappa_{\mathrm{PQ}} = \kappa_{\mathrm{H}}$, $f_{\mathrm{H}} = 0$: it assumes the residual keeps its crest factor. Equation (30) therefore locates every departure of the predictor in one measurable ratio, reported in Section C.3.

Group-size factor. Let $Z_i \overset{\mathrm{iid}}{\sim} \mathcal{N}(0,1)$ and define

$$\eta_g = \mathbb{E}\left[\max_{i\le g} Z_i - \min_{i\le g} Z_i\right]. \tag{31}$$

A centered Gaussian surrogate $e = \sigma(Z - \bar{Z}\mathbf{1}_g)$ satisfies $\mathbb{E}\,\mathrm{range}(e) = \sigma\eta_g$, because subtracting the sample mean leaves the range unchanged. This formulation respects the zero-sum constraint on group residuals; the centered coordinates themselves are not independent, and $\mathbb{E}\|e\|_2^2 = (g-1)\sigma^2$.

Writing ϕ and Φ for the standard normal density and distribution function, the maximum has distribution $\Phi(t)^g$. Symmetry and the order-statistic density give

$$\begin{aligned}\eta_g &= 2g\int_{-\infty}^{\infty} t\,\phi(t)\Phi(t)^{g-1}\,dt \\ &= 2\int_0^{\infty}\left[1 - \Phi(t)^g - (1-\Phi(t))^g\right]dt.\end{aligned} \tag{32}$$

We compute the finite-g reference by numerical quadrature: $\eta_{64} \approx 4.687467$, $\eta_{128} \approx 5.189195$, and $\eta_{256} \approx 5.653727$. Although $\eta_g \sim 2\sqrt{2\ln g}$ asymptotically (David & Nagaraja, 2004), the reported comparisons use these finite-g values rather than the leading asymptotic expression.

Under Gaussian mixing with comparable scale parameters at two group sizes, the mean ideal steps of the paired Hadamard references satisfy

$$\frac{s_{\mathrm{H}}(g_2)}{s_{\mathrm{H}}(g_1)} \approx \frac{\eta_{g_2}}{\eta_{g_1}}. \tag{33}$$

Here an ideal INT4 step is the group range divided by 15. If the Gaussian scale parameters differ, their ratio also enters Equation (33); comparability of these scales is an assumption rather than a consequence of the order-statistic formula.

Predictor. Under the shape assumption above, and neglecting incidental capture by the reference and unused target slots, we use the baseline-relative predictor

$$\widehat{s}(g,k) = s_{\mathrm{H}}(g)\sqrt{1-f_k}. \tag{34}$$

The measured reference $s_{\mathrm{H}}(g)$ absorbs both the group-size factor and the INT4 denominator 15. It is evaluated at the same reference site and group size using matched evaluation tokens, while f_k is obtained from the calibration spectrum. No multiplicative constant is fitted. Unlike Equations (28) and (29), Equation (34) is not a distribution-free upper bound on the mean step: it is the special

case of Equation (30) with $\kappa_{\mathrm{PQ}} = \kappa_{\mathrm{H}}$ and $f_{\mathrm{H}} = 0$. Unused target slots, nonuniform token or group scales, correlations, heavy tails, and calibration mismatch all enter through that crest-factor ratio, and the residual-shape approximation does not follow from spectral optimality. The pooled regression and held-out MoE expert visualization in Section C.3 assess the empirical trend separately; they neither prove nor parameterize Equation (34). The per-site check there measures the crest-factor ratio directly: it is within 3% of one at the q/k/v input and 0.87–0.92 at the down-projection input.

C.2 Metadata Accounting and Extended-Affine Controls

Two resources, one budget. Finer grouping increases both the number of local scales and the dimension of $\mathcal{S}$: halving g doubles both counts. Scale refinement changes local quantization resolution, while additional constant directions help only to the extent that they capture relevant energy. With g INT4 codes, one FP16 scale, and one FP16 offset per group, the logical storage is

$$b_{\mathrm{eff}}(g, 1) = \frac{4g + 16 + 16}{g} = 4 + \frac{32}{g} \quad \text{bits per value.} \tag{35}$$

This gives 4.5 at $g = 64$, 4.25 at $g = 128$, and 4.125 at $g = 256$. Raising k at fixed g leaves this per-value format budget unchanged and instead increases factor storage and, at online sites, transform work. These counts cover codes and per-group metadata. Shared transform factors, packing or allocation overhead, temporary buffers, and full-precision cache tails are separate deployment costs.

Metadata precision. The range predictor describes ideal affine arithmetic. For a nonconstant group, write $z = \min_i v_i$, $s = \mathrm{range}(v)/15$, and denote the stored FP16 parameters by $\widetilde{z} = z + \delta_z$ and $\widetilde{s} = s + \delta_s$. The encoder uses these rounded parameters, so its integer codes may differ from those obtained with the ideal grid. For finite $\widetilde{z}$ and positive finite $\widetilde{s}$, with the remaining arithmetic evaluated exactly, nearest-point encoding on the rounded grid satisfies

$$|\hat{v}_i - v_i| \le \frac{s}{2} + |\delta_z| + 15|\delta_s|. \tag{36}$$

To see this, each ideal grid point $z + qs$ moves by $\delta_z + q\delta_s$, whose magnitude is at most $|\delta_z| + 15|\delta_s|$ for $q \in \{0, \dots, 15\}$. The nearest point on the rounded grid is no farther than the perturbed ideal nearest point. Additional floating-point arithmetic errors are outside this bound. Thus a shared level remains exactly range-neutral, although a large offset can increase absolute FP16 rounding error. Constant groups have zero ideal range and use a positive fallback scale for encoding; metadata precision still limits reconstruction.

Extended-affine controls. To distinguish representation capacity from scale resolution, the controls retain the offset and add $m - 1$ FP16 coefficients along fixed, orthonormal, nonconstant Walsh directions. Let $B \in \mathbb{R}^{g \times (m-1)}$ satisfy $B^\top B = I$ and $B^\top \mathbf{1}_g = 0$. Representing a component Bc separately leaves a remainder to be encoded on the affine INT4 grid, with reconstruction

$$\hat{v} = \widetilde{s}\, q + \widetilde{z}\, \mathbf{1}_g + B\widetilde{c}, \qquad q \in \{0, \dots, 15\}^g. \tag{37}$$

The explicitly represented subspace $\mathrm{span}(\mathbf{1}_g, B)$ has dimension m. The additional directions generally have nonzero range: they are represented through extra coefficients, rather than becoming range-null directions of the original affine quantizer. The offset sets the origin of the grid for the remainder and need not equal the original group mean.

The fixed Walsh basis is shared, so there is no per-group basis cost. Counting the codes, scale, offset, and additional coefficients gives

$$b_{\mathrm{eff}}(g, m) = \frac{4g + 16 + 16 + 16(m-1)}{g} = 4 + \frac{16(m+1)}{g}. \tag{38}$$

Coefficient precision also matters: if $\delta c = \widetilde{c} - c$, then $\|B\delta c\|_2 = \|\delta c\|_2$. More directions can increase captured energy without guaranteeing a smaller quantization error or better perplexity at fixed storage.

Measured comparisons. At the shared 4.25-bit budget, $(g, m) = (128, 1)$ outperforms $(256, 3)$ in activation-only perplexity on both tested Llama checkpoints. Moreover, PrismQuant at $(256, 1)$

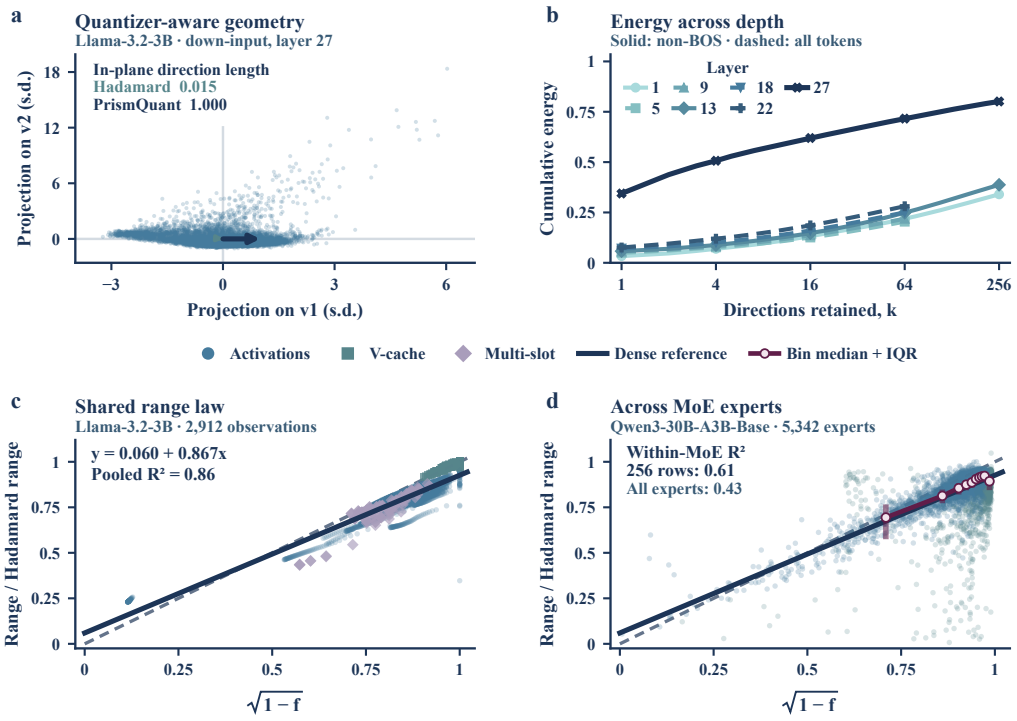

Figure 4: **From null-space geometry to a shared range law. (a)** Standardized token projections at the Llama-3.2-3B layer-27 down input; arrows are the group-constant direction pulled back through each rotation. **(b)** Cumulative energy in the leading directions at seven layers. **(c)** Group range relative to paired Hadamard versus $\sqrt{1-f}$ for 2,520 activation, 280 V-cache, and 112 multi-slot observations under one pooled fit; dashed is the identity. **(d)** The same fitted line on 5,342 Qwen3-30B-A3B experts held out of the fit: bin medians (Q1–Q3 bars) follow the dense reference line, and the scatter is widest for experts with fewer than 2,048 tokens.

outperforms Hadamard at $(128, 1)$ while using fewer logical activation bits (Table 6, Figure 6). Group size therefore determines both local resolution and the available constant subspace, while rank controls how much leading energy is explicitly assigned to that subspace. The matched-budget comparison changes both g and m; it evaluates their trade-off rather than isolating either factor. These observations do not establish a universal optimum over quantizer formats, and neither larger rank nor more represented directions alone guarantees better perplexity at a fixed resource budget. In the same study, the step ordering predicted by Equation (34) agrees with the mean-PPL ordering over the six bit-accounted PrismQuant configurations on each Llama checkpoint (Spearman correlation 1.0). This is a ranking observation for those configurations, not a measure of absolute prediction accuracy.

C.3 Empirical Diagnostics of the Range Law

Geometry and pooled range trend. Figure 4 collects the geometric, spectral, and range visualizations. Panel (a) shows standardized token projections and the pulled-back group-constant directions, while panel (b) shows cumulative leading energy at seven layers. The pooled dense-reference diagnostic in panel (c) contains 2,520 activation observations, 280 V-cache observations, and 112 extended-affine controls, for 2,912 observations in total. Let ρ denote group range relative to its paired Hadamard reference. We retain the source-defined absorbed-energy fractions f, which need not coincide with the deliberately aligned spectral fraction f_k for every configuration. The displayed least-squares line is

$$\rho = 0.0598022 + 0.8665148\sqrt{1-f}, \qquad R^2 = 0.861389. \tag{39}$$

This is one pooled descriptive fit, not a separate regression per family. The dashed identity line shows $\rho = \sqrt{1-f}$ for comparison. The fitted intercept and slope are not used in Equation (34),

Table 12: **Per-site error of the range law at $g = 128$.** Relative error $e = (\widehat{s} - s_{\text{meas}})/s_{\text{meas}}$ of the predicted step, activation-only protocol, three paired seeds; medians and 90th percentiles of $|e|$ over layers, and the median signed error, at the default operating point $k = 8$ and at $k = \max$. Positive signed error means the law over-predicts the step; the last column is the crest-factor ratio of Equation (30), $1/(1 + \text{median}\, e)$.

Model	Site	$k = 8$ median $\lvert e\rvert$	$k = \max$ median $\lvert e\rvert$	$k = \max$ P90 $\lvert e\rvert$	$k = \max$ median e	$k = \max$ $\kappa_{\text{PQ}}/\kappa_{\text{H}}$
Llama-3.2-3B	q/k/v	1.7%	1.7%	8.9%	+1.7%	0.98
Llama-3.2-3B	down	11.1%	8.9%	15.0%	+8.4%	0.92
Llama-3.1-8B	q/k/v	2.0%	1.3%	6.4%	+1.3%	0.99
Llama-3.1-8B	down	12.2%	8.8%	15.8%	+8.6%	0.92
Qwen3-4B-Base	q/k/v	2.5%	3.5%	6.2%	−3.2%	1.03
Qwen3-4B-Base	down	21.2%	18.1%	52.2%	+15.4%	0.87

which has no fitted multiplicative coefficient. Accordingly, the reported R^2 describes the regression fit and is not an accuracy score for the unfitted predictor.

Held-out expert diagnostic. Panel (d) applies the unchanged dense-reference line to 5,342 Qwen3-30B-A3B experts held out of that fit. The per-expert points and binned medians with Q1–Q3 intervals expose variability beyond the average trend, including the wider scatter among experts with fewer than 2,048 calibration tokens. The quartile intervals describe within-bin spread, not confidence intervals for the medians. The holdout concerns the pooled regression and does not imply that the experts were excluded from their own rotation calibration. This is an empirical transfer diagnostic, rather than an exact consequence of the Gaussian range model or a universal per-expert prediction guarantee.

Group-size diagnostic. On the two Llama checkpoints in the activation-only metadata study, measured Hadamard step ratios for $256/128$ and $128/64$ at both q/k/v and down-input sites differ from the finite-g Gaussian ratios by less than 0.2%. The reference ratios use Equation (32). This agreement concerns the group-size factor alone, not the absolute prediction error of the complete energy-and-group-size law.

Per-site prediction error. The complete law is accurate at the q/k/v input and conservative at the down-projection input. Figure 5 and Table 12 compare the predicted step $\widehat{s}(g, k) = s_{\text{H}}(g)\sqrt{1 - f_k}$ with the measured mean step at every layer, site, and rank of the activation-only studies, with 90% intervals over three paired seeds and no fitted coefficient. At $g = 128$ and $k = \max$, the median absolute relative error is 1–4% at q/k/v but 9% on both Llama models and 18% on Qwen3-4B-Base at the down-projection input, and its sign is systematic: the measured step is smaller than predicted. The bias has the direction the assumptions allow. The law assumes that removing aligned energy leaves a residual of the same shape, so that the range shrinks with the amplitude $\sqrt{1 - f_k}$; the leading directions are, however, the heavy-tailed part of the activation, and removing them shrinks the extremes by more than the r.m.s., most strongly on Qwen3-4B-Base, whose leading directions are the most concentrated. Incidental capture by the Hadamard reference would bias the prediction the other way, so it is not the cause. Two properties survive unchanged: the predicted ordering of the six bit-accounted configurations matches the measured perplexity ordering (Section C.2), and the group-size ratios agree to 0.2% (above). In the terms of Equation (30), the measured $\kappa_{\text{PQ}}/\kappa_{\text{H}}$ is 0.98–1.03 at q/k/v and 0.87–0.92 at the down-projection input: alignment removes the heavy-tailed directions and leaves a flatter residual, the same flattening that the anchor-column ablation of Appendix A.4 identifies as the main source of the gain. The predictor is therefore used to order configurations and as an empirically conservative estimate at the down-projection input.

D Efficient Deployment and End-to-End Performance

We evaluate whether the additional structure of PrismQuant can be implemented without sacrificing the practical benefits of low-bit inference. Our implementation combines a compact low-rank rotation, Tensor Core execution, pre-bound kernel launches, and CUDA Graph replay. On Llama-3.1-8B, the resulting pipeline achieves up to $1.51\times$ **prefill throughput** and $1.22\times$ **Graph decoding speedup** over the corresponding FP16 baseline, while reducing Graph decoding peak memory by 56.34%. Relative to the matched QuaRot–Hadamard pipeline, the additional Graph decoding la-

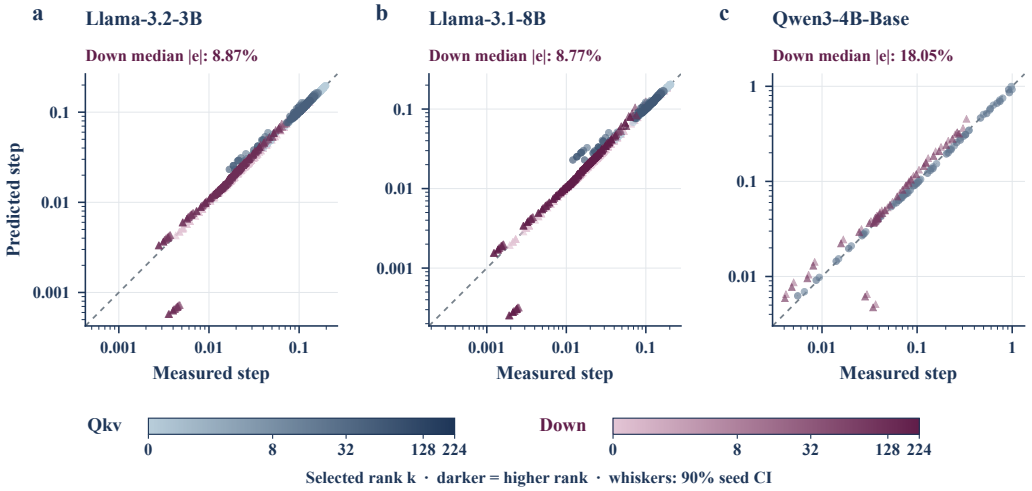

Figure 5: **Predicted versus measured quantization step.** Each point is one layer and site of the activation-only studies at $g = 128$; whiskers are 90% intervals over three paired seeds and the dashed line is the identity. Points above the line are over-predicted. The q/k/v input follows the identity closely at every rank; the down-projection input sits below it, more so on Qwen3-4B-Base; the ratio of measured to predicted step there is the crest-factor ratio of Equation (30).

tency is only 2.35%. The experiments below separate quantization-design choices, local kernel improvements, and measured whole-model performance.

D.1 Design Trade-offs and the Low-Rank Operating Point

Figure 6 collects the two accuracy results that motivate a compact implementation and adds the local cost of the transform. Panels (a) and (b) plot the metadata-allocation study of Table 6a and the rank study of Table 3: at a fixed activation budget of $4 + 16(m+1)/g$ bits per value, alignment improves perplexity without additional metadata, and $k = 8$ already recovers most of the Hadamard-to-bf16 gap on Llama, with larger ranks giving model-dependent rather than monotone gains. Panel (c) measures the transform's share of decoder-layer time at $T = 2048$ in isolation; these local durations are not whole-model overheads, which Section D.3.1 reports separately. We therefore adopt $k = 8$ as the deployment operating point, not as a claim that it maximizes accuracy for every model.

D.2 Compact Kernels and Runtime Integration

Two-kernel rotation. We reuse the existing CUTLASS INT4 matrix multiplication and replace only the online R_4 transform preceding the MLP down-projection input quantizer. For a row-batched activation $X \in \mathbb{R}^{T\times d}$, the exported Householder/compact-WY factors give the online computation

$$U = XA, \qquad Z = XH_{\text{blk}} - UB, \qquad A \in \mathbb{R}^{d\times k}, \quad B \in \mathbb{R}^{k\times d}. \tag{40}$$

Here H_{blk} applies normalized Hadamard transforms to 128-channel blocks, and the offline construction absorbs the relevant signed permutation into the factors and, for a compensated model, the corresponding weights. No dense $d\times d$ rotation is materialized online. The projection and correction each require $O(Tdk)$ arithmetic; k controls the correction rank, not the number of model channels.

Kernel A evaluates $U = XA$ on Tensor Cores, with separate FP32 accumulators for the FP16 high/low factor terms. Kernel B combines the block-Hadamard transform and low-rank correction within each token-tile/channel-block program. On A40, the 128-channel Hadamard is implemented using Tensor Core matrix multiplication, and the correction retains the three leading high/low cross-products. This avoids writing a full-width Hadamard intermediate between these operations; the two kernels communicate through a compact FP32 partial buffer. Our integration writes Z in FP16 and then invokes the shared QuaRot quantizer and INT4 GEMM(Ashkboos et al., 2024). It does not replace their interface with the native group-asymmetric packing epilogue used in the separate transform microbenchmark.

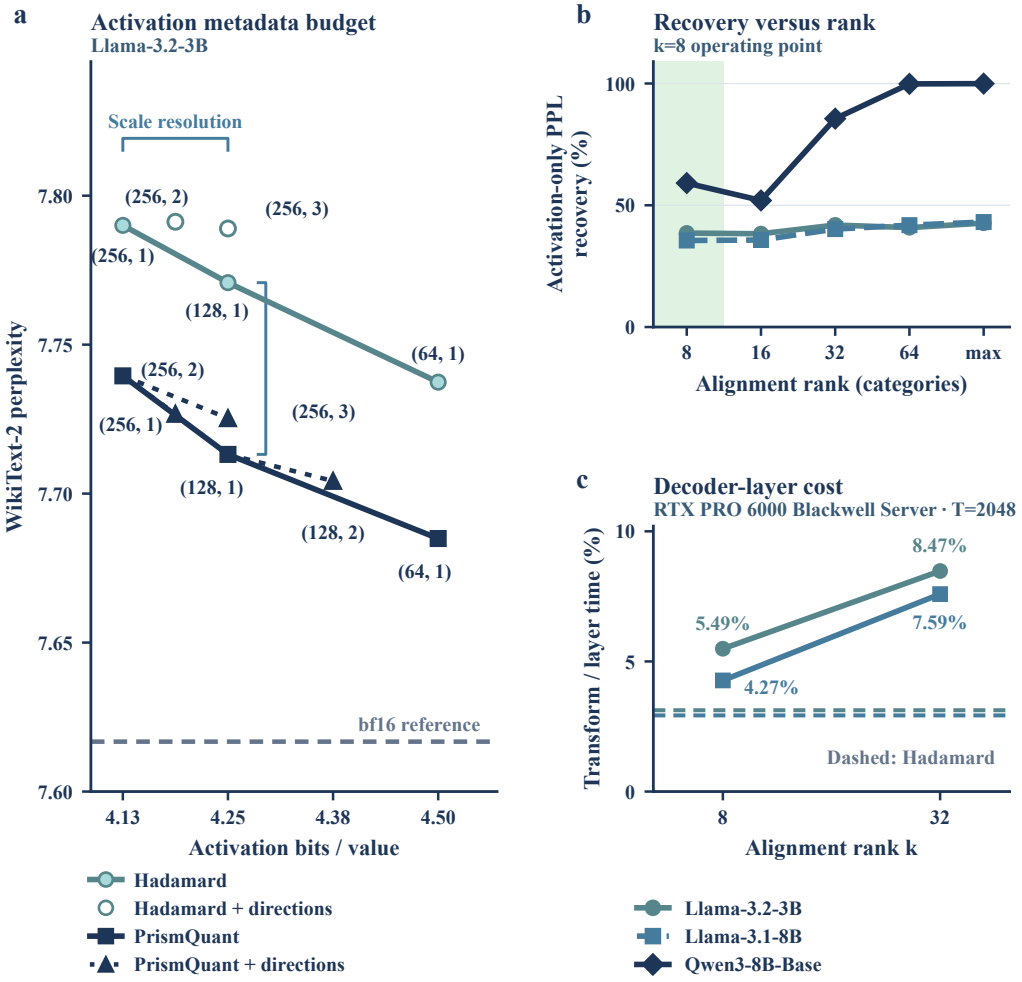

Figure 6: **Activation design and local transform cost.** **(a)** Activation representation budget versus WikiText-2 PPL on Llama-3.2-3B; (g, m) denotes group size and the number of represented directions. **(b)** Activation-only PPL recovery versus rank. The highlighted $k = 8$ category is the deployment operating point; the actual rank is capped separately at each quantization site. Llama results use three seeds and 64 chunks per seed; Qwen results use three seeds and 146 chunks, with bf16 weights and KV states. **(c)** Original local cost measurements at $T = 2048$ on an NVIDIA RTX PRO 6000 Blackwell Server Edition: $100t_{\text{transform}}/(t_{\text{decoder layer}} + t_{\text{transform}})$. These separately benchmarked durations are not whole-model decode overhead; end-to-end deployment is evaluated separately.

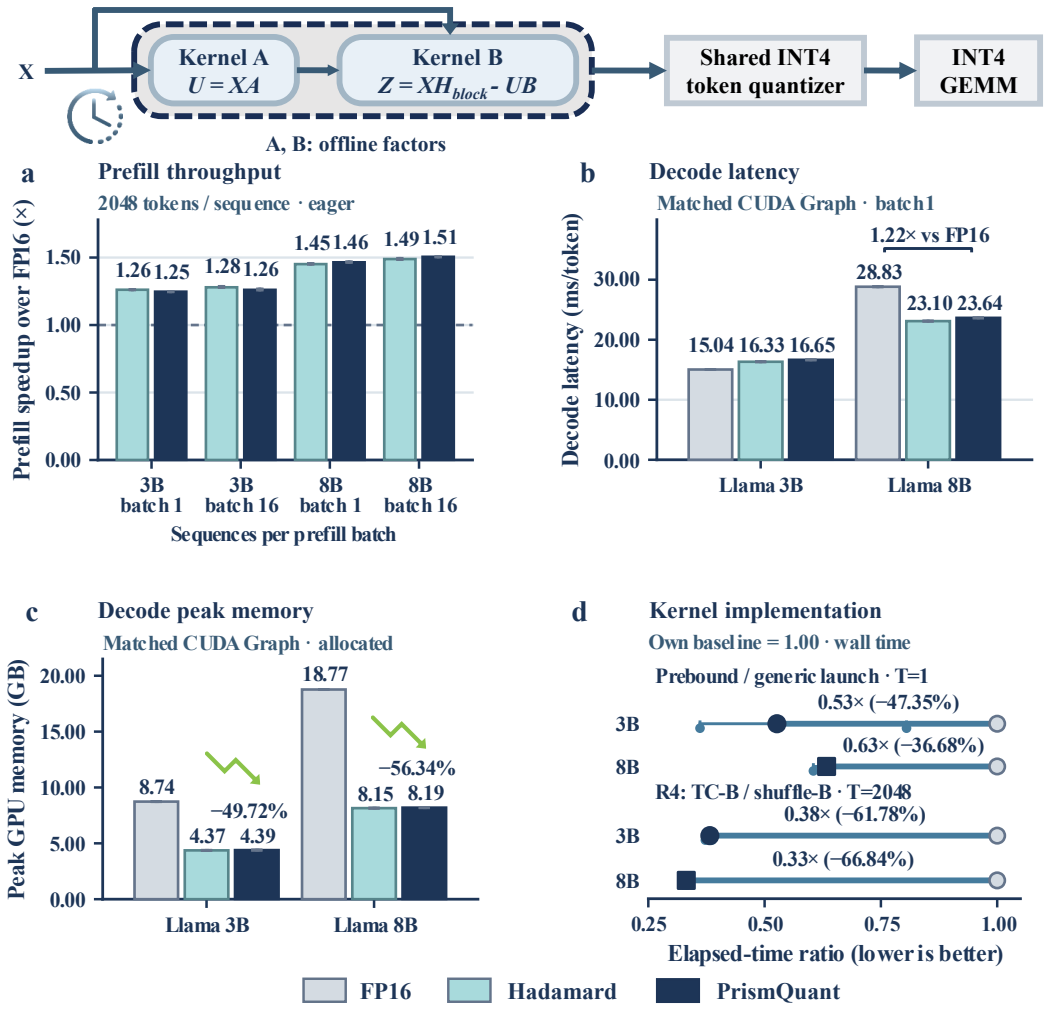

Figure 7: **Practical efficiency of the compact PrismQuant implementation.** The upper strip shows the two-kernel R_4 computation followed by the shared INT4 quantizer and GEMM; box widths do not represent time. **(a)** Eager prefill throughput relative to matched FP16, with 2048 input tokens per sequence. **(b,c)** Matched CUDA Graph decode latency and peak allocated memory on A40, using batch size 1 and a 2048-token prefix. **(d)** Local implementation ablations: pre-bound versus generic launches at $T = 1$, and complete R_4 paths using Tensor Core versus shuffle-based Kernel B at $T = 2048$. Each ablation has its own reference; these are not whole-model speedups. Timing centers are pooled medians; ablation centers are medians of paired-session ratios. Whiskers show session ranges, not confidence intervals. All performance models use random weights.

Reducing dispatch and allocation overhead. Compiled kernel calls are pre-bound to avoid repeated Python/Triton argument binding. Temporary projection buffers and the fixed Hadamard matrix are reused across sequential layers. Common cache metadata is prepared once per decode step, and a current-stream-compatible backend enables complete one-step CUDA Graph capture for all three methods. Replay updates positions and KV metadata while preserving the growing causal context; it does not repeatedly evaluate a fixed cache state. These shared runtime adaptations are applied consistently to the FP16, Hadamard, and PrismQuant paths wherever applicable.

D.3 Benchmark Protocol

We benchmark Llama-3.2-3B and Llama-3.1-8B on one NVIDIA A40 (sm_{86}), using PyTorch 2.4.1, CUDA 12.4, Triton 3.0.0, and the QuaRot-derived integer pipeline. The two quantized methods share packed INT4 weights, quantizer, GEMM, attention, and KV-cache implementations; their common random states are verified by tensor hashes. All performance models use random weights,

Table 13: **End-to-end performance on a single NVIDIA A40.** Prefill reports input tokens/s (↑), and decode reports ms/token (↓). Both decode columns use the matched current-stream backend; Graph speedup and peak memory use CUDA Graph execution. Speedup is relative to the FP16 row of the same model and mode. All rows use random weights; bold marks the best quantized entry per model and metric.

	Prefill ↑		Decode ↓		Graph deployment	
Method	B1	B16	Eager	Graph	Peak GB ↓	Speedup ↑
Llama-3.2-3B (FP16)	10,163.48	13,985.13	21.86	15.04	8.74	1.00×
QuaRot–Hadamard	**12,819.74**	**17,894.66**	**42.83**	**16.33**	**4.37**	**0.92×**
PrismQuant ($k = 8$)	12,674.40	17,630.27	43.08	16.65	4.39	0.90×
Llama-3.1-8B (FP16)	5,241.70	6,418.43	30.22	28.83	18.77	1.00×
QuaRot–Hadamard	7,607.40	9,556.00	52.31	**23.10**	**8.15**	**1.25×**
PrismQuant ($k = 8$)	**7,677.01**	**9,661.75**	**49.99**	23.64	8.19	1.22×

making this a *kernel-swap performance benchmark*, not a model-quality evaluation. The floating-point performance reference is FP16.

Prefill processes 2048 input tokens per sequence at batch sizes 1 and 16, with only the final-position vocabulary logits computed for all methods. Decode uses batch size 1 after a 2048-token prefix, executes 128 causal steps, and excludes the first 8 from steady-state timing. Each condition has 10 warmup runs and 50 measured runs in each of three balanced sessions; tables report pooled timing medians. Continuous eager timing synchronizes around the measured sequence, not between its individual decode steps. Both decode modes in Table 13 use the matched current-stream backend; prefill uses the separately recorded eager prefill cohort. No speedup is computed across unmatched backends or execution modes.

Graph capture and compilation are excluded from steady-state timing, while replay metadata updates are included. The timed Graph layout uses one preallocated 2176-token page; separate page-64 tests verify actual page-boundary behavior. All six matched Graph paths pass the growing-cache replay checks. Peak allocated memory is measured in independent processes and reported in decimal GB, using the maximum of the three session peaks.

D.3.1 End-to-End Gains and Memory Efficiency

Preserving prefill throughput. Figure 7(a) and Table 13 show that the additional rotation structure has little impact on the throughput of the integer pipeline. On 3B, PrismQuant retains 98.87% and 98.52% of Hadamard's prefill throughput at batch sizes 1 and 16. On 8B, it reaches 100.92% and 101.11%, respectively. Thus, all four prefill conditions remain within 1.50% of Hadamard, while PrismQuant achieves 1.25–$1.51\times$ the matched FP16 throughput. The small 8B batch-1 advantage is interpreted as near parity rather than a robust throughput improvement.

Efficient decoding with small incremental cost. In the matched 8B eager comparison, PrismQuant reduces decode latency from 52.31 to 49.99 ms/token, a 4.43% reduction. Graph replay further reduces PrismQuant's latency from 43.08 to 16.65 ms/token on 3B and from 49.99 to 23.64 ms/token on 8B, corresponding to $2.59\times$ and $2.11\times$ improvements over its own matched eager execution. These gains measure the execution-mode change, not a rotation-only speedup over Hadamard. When both methods use Graph replay, PrismQuant incurs only 1.96% and 2.35% additional latency on 3B and 8B. On 8B, it also achieves $1.22\times$ faster Graph decoding than FP16; on 3B, the 16.65 ms/token result remains above the FP16 reference of 15.04 ms/token. Together, these results show that the more structured rotation can retain competitive end-to-end performance at a small incremental cost.

Substantial memory savings. Under the same Graph mode, PrismQuant reduces peak allocated memory from 8.74 to 4.39 GB on 3B and from 18.77 to 8.19 GB on 8B, corresponding to savings of 49.72% and 56.34%. Its resident high/low factors occupy $96d$ bytes per layer: $64d$ for the rank-padded projection factors and $32d$ for the correction factors. Across all layers, these factors require 22.02 MB on 3B and 44.04 MB on 8B, plus one shared 32 KiB Hadamard matrix and shape-dependent scratch storage. This compact representation explains why PrismQuant remains close to

Table 14: **Rotation-plus-quantization frontend speedup over QuaRot.** Each value is the reciprocal of the median of three paired-session PrismQuant/Hadamard elapsed-time ratios. Values above $1.00\times$ indicate faster execution and are bold. These are directly measured local frontend chains, not end-to-end model speedups; T is the number of input activation rows.

Model	$T = 1$	$T = 2048$	$T = 32768$
Llama-3.2-3B	$\mathbf{1.25}\times$	$0.93\times$	$0.85\times$
Llama-3.1-8B	$\mathbf{1.73}\times$	$\mathbf{1.17}\times$	$\mathbf{1.12}\times$

Hadamard's memory footprint despite its richer transform. The reported peaks are inference memory, not serialized weight sizes.

D.4 Kernel-Level Evidence and Measurement Scope

Table 14 isolates the directly timed *rotation-plus-quantization frontend*, with the same QuaRot quantizer after either transform. On 8B, the optimized PrismQuant frontend is faster at all three tested shapes, with paired speedups of 1.12–$1.73\times$. On 3B, it improves the single-token frontend by $1.25\times$, while larger token batches incur extra local cost. These shape-dependent measurements are consistent with the small 3B prefill throughput reduction observed end to end; a local speedup or slowdown should not be equated with the same whole-model change.

The controlled implementation ablations in Figure 7(d) support both design choices. At $T = 1$, prebound launches reduce the local elapsed-time ratio to 0.53 on 3B and 0.63 on 8B relative to generic launches of the same compiled arithmetic. At $T = 2048$, replacing shuffle-based Kernel B with its Tensor Core implementation reduces the complete R_4 path ratio to 0.38 and 0.33, respectively. The latter ratios compare full transform paths, not Kernel B alone; all other tested shapes, including unfavorable cases, are retained in the accompanying experiment records.

Scope of the deployment evidence. These experiments establish compact-transform efficiency within a shared packed-INT4 pipeline. The performance backend uses per-token symmetric activation quantization and KV interfaces distinct from the paper's native group-128 asymmetric accuracy protocol; the two experiments do not constitute a checkpoint-level joint accuracy–speed measurement. The recorded large-accumulator FP16-conversion failures in the shared backend remain an explicit numerical limitation, separate from the passed rotation and growing-cache Graph checks. Within this scope, the results demonstrate that PrismQuant preserves low-bit prefill and memory benefits with only a small additional Graph decoding cost.

E LOCAL ACTIVATION VISUALIZATIONS

To complement the quantitative alignment, range, and quantization-error analyses, we visualize local activation magnitudes from Qwen3-8B-Base. Figures 8, 9, and 10 show end-to-end query-projection inputs, end-to-end down-projection inputs, and paired-local query-projection inputs, respectively. Each figure contains four columns corresponding to decoder Blocks 1, 13, 24, and 36, and four rows corresponding to the unrotated reference, Hadamard, PrismQuant ($k = 8$), and PrismQuant ($k = \max$). The unrotated reference is obtained from an unquantized, norm-fused FP32 forward pass using the original bf16 checkpoint values. The end-to-end views use the existing quantize–dequantize (QDQ) evaluation pipeline.

All three figures use the same local window: sample 0, tokens 0–127, and channels 0–511, covering four complete quantization groups of size $g = 128$. Every point in this window is retained without pooling or striding. The surface height is the absolute activation value $|Y_{t,c}|$, not a group-centered residual or a quantization reconstruction error. Heights are linear; the blue–orange colors use a square-root mapping to distinguish moderate amplitudes without changing their heights. Within each column, all four rows share the same local height and color limits; no method receives independent normalization. The pale plane indicates $z = 0$. These fixed-window visualizations are descriptive examples rather than aggregates over the evaluation set.

End-to-end query-projection inputs. Figure 8 visualizes the residual-stream activations entering `q_proj`, rather than the projected query vectors. In the rotated models, the global R_1 basis is already represented in the residual stream, and the observer records its values immediately before input quantization without applying another rotation. The shared-scale comparison exposes how activation magnitude is distributed across tokens, channels, and depth during the full quantized forward path. Unlike the unrotated reference, the rotated rows also reflect upstream weight and activation quantization effects. This view therefore characterizes the resulting end-to-end activation geometry; the paired-local comparison below isolates the effect of the rotation itself.

End-to-end down-projection inputs. Figure 9 examines the gated MLP activations after the layer-specific online R_4 transform and immediately before the `down_proj` input quantizer. This is the online site at which PrismQuant directly aligns activation directions with the group-constant subspace. The local window permits inspection of four consecutive quantization groups without channel binning. Importantly, the method does not require uniformly smaller absolute peaks: a large component shared within a group can be represented by the affine offset without widening that group's range. The raw surfaces should therefore be read as evidence of energy redistribution, together with the quantitative group-range and quantization-error analyses, rather than as a direct ranking of quantization quality by peak height.

Paired-local query-projection inputs. Figure 10 applies the different transforms to identical captured `q_proj` inputs from the unquantized reference forward pass. Each PrismQuant variant uses its deployed global R_1, not a separately fitted rotation for each displayed block. Holding the input tensor fixed separates the local basis change from upstream quantization effects and complements the end-to-end view in Figure 8. Because rotation acts on the full feature dimension before the 512-channel window is selected, a change in local peak height need not imply a change in total activation energy. Together, the paired-local and end-to-end views distinguish direct rotation-induced redistribution from the activation geometry observed along the quantized forward path.

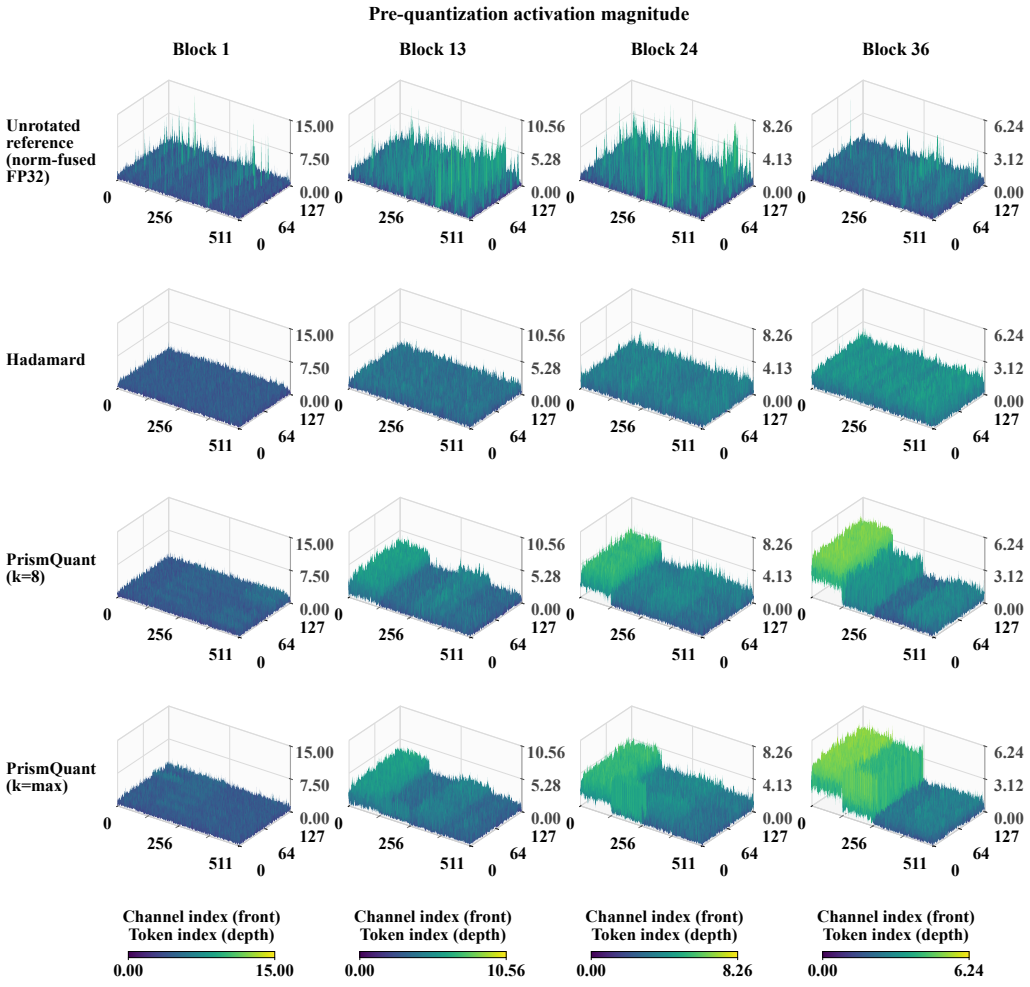

Figure 8: **Local end-to-end activation magnitudes at `q_proj` on Qwen3-8B-Base.** Columns show Blocks 1, 13, 24, and 36; rows show the unrotated reference, Hadamard, PrismQuant ($k = 8$), and PrismQuant ($k = \max$). The window is sample 0, tokens 0–127, and channels 0–511, with $g = 128$. Heights represent absolute input activations before quantization. Linear heights and square-root colors share limits across all four rows within each column; the pale plane marks $z = 0$. The rotated rows include upstream quantization effects.

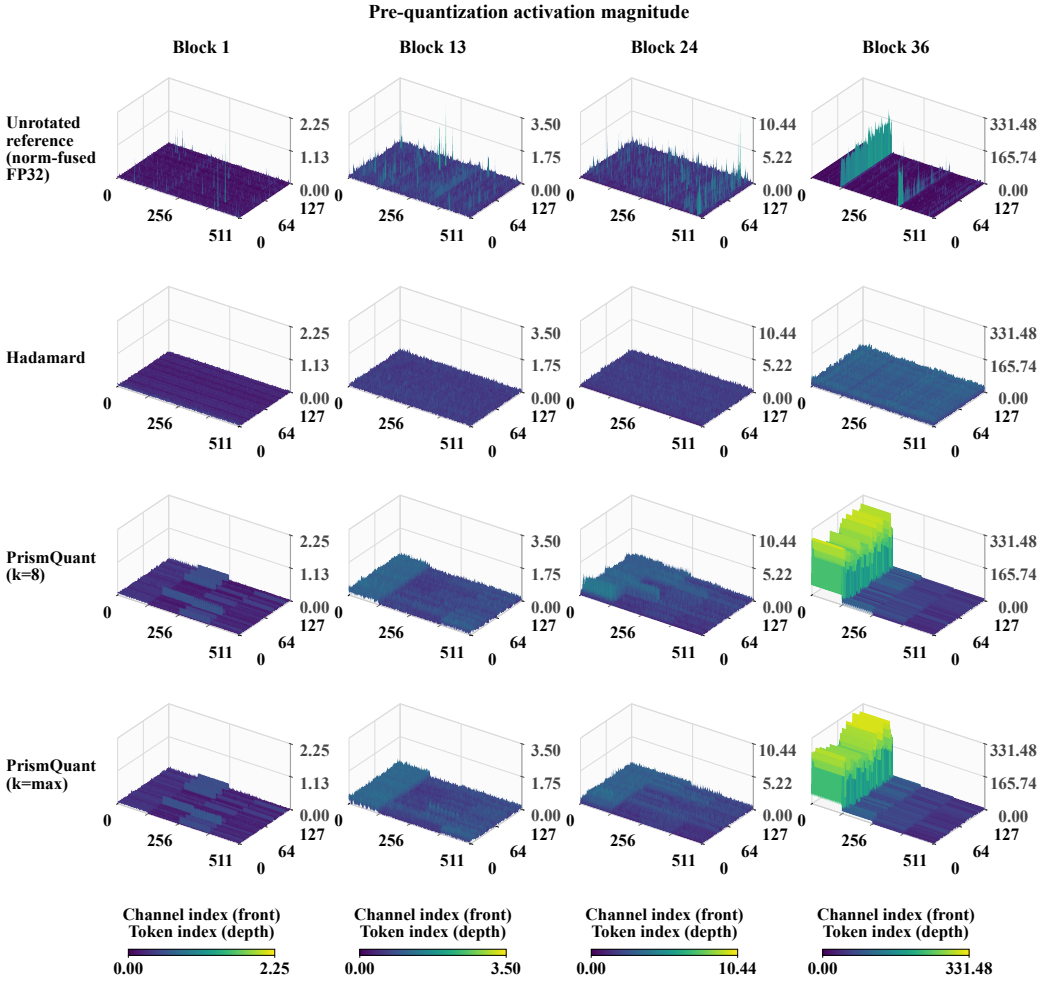

Figure 9: **Local end-to-end activation magnitudes at `down_proj` on Qwen3-8B-Base.** The rotated rows are observed after the online R_4 transform and before activation quantization. Columns show Blocks 1, 13, 24, and 36; rows show the unrotated reference, Hadamard, PrismQuant ($k = 8$), and PrismQuant ($k = \max$). We retain sample 0, tokens 0–127, and channels 0–511, corresponding to four groups with $g = 128$. Heights are absolute activations, not centered residuals. Each column shares linear height limits and square-root color mapping across its four rows, with a pale $z = 0$ plane.

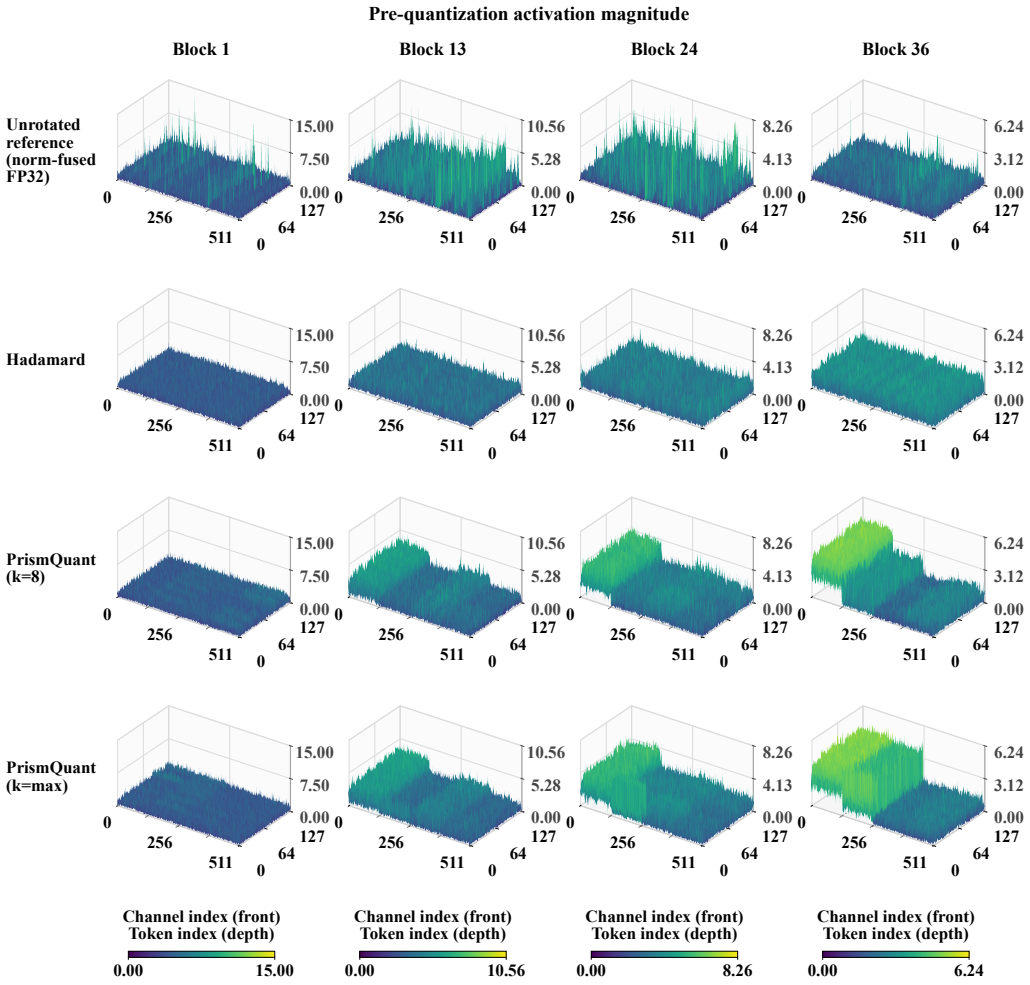

Figure 10: **Paired-local activation magnitudes at `q_proj` on Qwen3-8B-Base.** All methods transform the same captured input tensor within each block, isolating the local effect of rotation. Columns show Blocks 1, 13, 24, and 36; rows show the unrotated reference, Hadamard, PrismQuant ($k = 8$), and PrismQuant ($k = \max$). The displayed window contains sample 0, tokens 0–127, and channels 0–511, with $g = 128$ and no pooling. Heights are linear and colors use square-root mapping, with identical local limits across the four rows of each column; the pale plane marks $z = 0$.