MIRRORQUANT: LEARNABLE HOUSEHOLDER ROTATIONS FOR LOW-BIT LLM QUANTIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Orthogonal rotations have become an effective technique for low-bit LLM quantization by expressing hidden representations in more quantization-friendly bases. However, existing approaches predominantly parameterize learned rotations as dense matrices, incurring quadratic learned state and costly optimization. Their deployment efficiency largely depends on foldability: at non-foldable interfaces, dense rotations must remain on the inference path, with both quadratic storage and computation of the rotation dimension. To overcome these limitations, we propose MirrorQuant, a unified framework for efficient and learnable rotations across Transformer interfaces. MirrorQuant parameterizes learned rotations using short chains of Householder reflections, a construction that preserves exact orthogonality, reduces the learned state to linear in the rotation dimension for fixed chain depth, and enables learnable log-linear online execution without materializing dense rotation matrices. We further introduce projection-guided learning, which complements representation reconstruction by penalizing quantization distortion after the immediate downstream linear projection. Experiments across 23 matched settings spanning model families, scales, and low-bit configurations demonstrate accuracy and efficiency gains. MirrorQuant improves 0-shot$_9$ accuracy over DartQuant in every setting, with gains up to 1.81%, while its online rotation reduces prefill latency by 29.6% relative to a dense implementation. Source code can be found in the supplementary materials.

1 INTRODUCTION

Large language models (LLMs) have demonstrated remarkable capabilities across a wide range of language and reasoning tasks (Touvron et al., 2023; Team, 2024; Yang et al., 2024), yet their substantial storage and computational demands remain major obstacles to efficient deployment. Quantization maps weights and activations to low-bit representations, reducing model storage and memory traffic while enabling efficient low-precision computation (Dettmers et al., 2022; Xiao et al., 2023; Lin et al., 2025). It has therefore become one of the most practical techniques for compressing and deploying LLMs. However, aggressively quantizing both weights and activations often degrades performance, making it difficult to balance deployment efficiency and model quality.

Among quantization paradigms, post-training quantization (PTQ) provides a practical way to adapt pretrained models to low precision using only a small calibration set, without full-model retraining (Frantar et al., 2022; Xiao et al., 2023; Shao et al., 2024). However, low-bit weight–activation PTQ remains challenging. Highly non-uniform activation distributions and salient outliers can substantially enlarge the quantization range, leaving most values coarsely represented at limited precision (Dettmers et al., 2022; Xiao et al., 2023). To mitigate this issue, recent methods introduce function-preserving transformations that reshape intermediate representations into forms that are more amenable to quantization. Orthogonal rotations are a representative choice: they redistribute activation magnitudes across channels while preserving the full-precision function through corresponding transformations of the weights. At foldable interfaces, these rotations can be absorbed into adjacent weights and therefore incur no additional inference-time overhead.

Early rotation-based PTQ methods for quantizing large language models primarily used fixed fast transforms. QuIP and QuIP# (Chee et al., 2023; Tseng et al., 2024) use randomized incoherence processing for weight quantization, while QuaRot (Ashkboos et al., 2024) applies randomized

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.

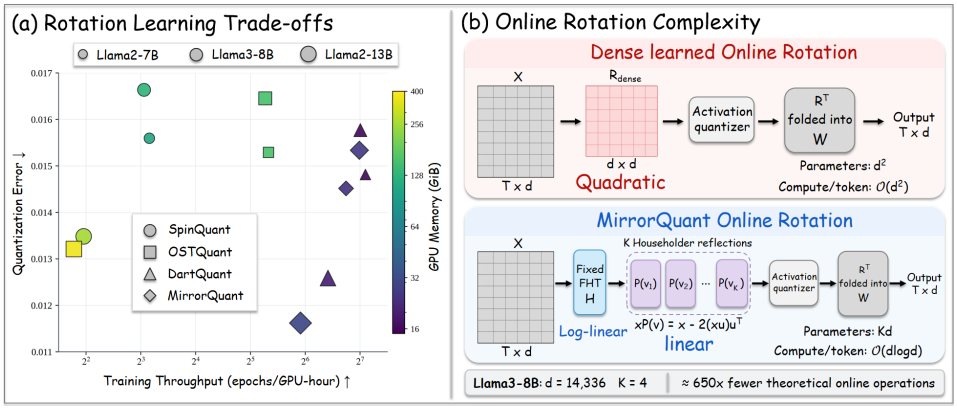


Figure 1: **Efficient rotation learning and online execution.** (a) Trade-offs in quantization error, training throughput, and peak GPU memory under W4A4KV4 across three model scales. (b) Compared with an explicit dense online realization, the fixed Hadamard transform with K Householder reflections requires about $650\times$ fewer theoretical rotation operations on Llama-3-8B ($K = 4$).

Hadamard rotations across residual, attention, MLP, and KV-cache interfaces. Later work advanced both rotation structures and optimization. DuQuant (Lin et al., 2024) combines blockwise rotations with permutations, ParoQuant (Liang et al., 2025) learns lightweight Givens rotations with channel-wise scaling, and DFRot (Xiang & Zhang, 2024) refines rotations using an activation-weighted objective. RUQuant (Liu et al., 2026) and HARP (Zagitov et al., 2026) construct learnable rotations from compact orthogonal factors. Among dense learned-rotation methods, SpinQuant (Liu et al., 2025) performs end-to-end quantization-aware optimization, OSTQuant (Hu et al., 2025) jointly optimizes rotations and scaling, and DartQuant (Shao et al., 2025) improves calibration efficiency through distribution-driven local learning. Within this line, dense rotations are learned mainly at foldable interfaces, whereas online interfaces retain fixed fast transforms.

This division reflects a central efficiency trade-off in model-specific rotation learning. A learned rotation R can adapt the representation basis to the model and quantizer, thereby mitigating the effect of activation outliers more effectively than a fixed transform. Existing methods, however, typically obtain this adaptability by parameterizing R as a dense matrix. During calibration, dense R requires quadratic learned state and costly full-matrix optimization, leading to the trade-off among quantization error, training throughput, and peak GPU memory shown in Figure 1(a). At deployment, whether the learned transform incurs additional inference cost depends on foldability. A foldable R can be absorbed into adjacent weights, whereas at an online interface, its activation-side transform must remain on the inference path. As shown in Figure 1(b), deploying a dense learned R online incurs quadratic state and computation, making model-specific online rotations impractical. Beyond efficiency, existing objectives either optimize aggregate model behavior or match local representations and distributions, without explicitly distinguishing quantization errors by how they are transformed by the subsequent linear projection. Errors of similar magnitude in representation space may therefore affect downstream computation very differently.

Motivated by these limitations, we propose MirrorQuant, an efficient framework that combines a compact Householder parameterization with projection-guided learning. Instead of optimizing a dense matrix, MirrorQuant represents each learnable rotation as a fixed orthogonal base followed by a short chain of K Householder reflections, with only the K Householder vectors optimized during calibration. Because the base and every reflection are orthogonal, their product remains exactly orthogonal. This construction reduces the number of learnable parameters from $\mathcal{O}(d^2)$ to $\mathcal{O}(Kd)$, which scales linearly with the rotation dimension d for fixed K. Within the standard four-rotation placement, the same formulation accommodates both foldable and online rotations. At foldable interfaces, the compact parameterization reduces calibration cost relative to dense optimization, and the learned rotation is subsequently absorbed into adjacent weights. At an online interface, combining a fast Hadamard base with the learned Householder chain yields a model-specific rotation with $\mathcal{O}(d \log d)$ online computation for fixed K, instead of the $\mathcal{O}(d^2)$ cost of a dense learned rotation. Because this compact family has limited adaptation capacity, its Householder budget should prioritize errors that matter most to downstream computation instead of treating directions equally. Accordingly, MirrorQuant combines full-space representation reconstruction with interface-specific projection guidance: the former preserves the overall representation, while the latter prioritizes quantization errors according to their effects after the subsequent linear projection.

Extensive experiments across model families, parameter scales, and low-bit weight–activation–KV-cache configurations demonstrate the effectiveness of MirrorQuant. Across all 23 matched settings, including three Llama-3-70B settings using a fixed DartQuant-derived R_1 base, MirrorQuant improves 0-shot$_9$ accuracy over DartQuant in every comparison, with gains up to 1.81%. On Llama-3-8B, the learned online rotation improves 0-shot$_9$ accuracy from 62.34 to 62.60 over the fixed FHT while adding only 4.3% prefill and 1.0% decode latency; relative to the dense online realization, it reduces these latencies by 29.6% and 5.9%, respectively.

2 PRELIMINARIES

In this section, we introduce the notation and background needed to describe our method. Let $X \in \mathbb{R}^{T\times d}$ denote an activation matrix whose rows are token representations, and let $W \in \mathbb{R}^{d\times m}$ denote the linear map that consumes X.

2.1 POST-TRAINING QUANTIZATION (PTQ)

Let $f(\cdot;\Theta)$ be a pretrained full-precision network. Post-training quantization (PTQ) determines low-bit quantization parameters from a small calibration set while keeping the pretrained weights fixed. For symmetric uniform quantization, each quantization group g uses a scale $s_g > 0$:

$$\mathcal{Q}_{s_g}(x) = s_g \,\mathrm{clip}(\mathrm{round}(x/s_g), q_{\min}, q_{\max}), \tag{1}$$

where $q_{\min}$ and $q_{\max}$ are determined by the bit-width. We write $\mathcal{Q}$ when the quantization grid is fixed or its specific parameters are irrelevant. A central challenge in joint weight–activation quantization is that activation outliers can enlarge the scale required by a group, thereby increasing the rounding error for its non-outlier coordinates.

2.2 ORTHOGONAL ROTATIONS FOR QUANTIZATION

For an orthogonal matrix $R \in \mathbb{R}^{d\times d}$ satisfying $R^\top R = I$, a linear projection is unchanged in full precision:

$$XW = (XR)(R^\top W). \tag{2}$$

Thus, R can redistribute the coordinates of X before quantization, while $R^\top$ can be absorbed into the following linear weight whenever the network structure permits. This property allows rotation-based PTQ methods to reduce the impact of activation outliers on quantization without changing the original full-precision function.

Quantization, however, breaks this exact equivalence. After rotating and quantizing X, we map the quantized representation back to the original basis, $\widehat{X}_R = \mathcal{Q}(XR)R^\top$, and define the resulting representation error as $E_R(X) = \widehat{X}_R - X$. For a following linear projection W, the induced output error is

$$\widehat{X}_R W - XW = E_R(X)W. \tag{3}$$

These two errors capture different aspects of quantization. The reconstruction error, $\|E_R(X)\|_F^2$, treats all representation directions equally, whereas the error after the following projection, $\|E_R(X)W\|_F^2$, weights different error directions according to $WW^\top$:

$$\|E_R(X)\|_F^2 = \mathrm{tr}\big(E_R(X)^\top E_R(X)\big), \tag{4}$$

$$\|E_R(X)W\|_F^2 = \mathrm{tr}\big(E_R(X)^\top E_R(X)WW^\top\big). \tag{5}$$

The two objectives are therefore equivalent for every error matrix only in a special case.

Proposition 2.1 (Reconstruction-projection equivalence)**.** *For a fixed linear operator W, there exists $c \geq 0$ such that*

$$\|EW\|_F^2 = c\|E\|_F^2 \quad \textit{for every } E \in \mathbb{R}^{T\times d} \quad \Longleftrightarrow \quad WW^\top = cI_d. \tag{6}$$

Hence, minimizing reconstruction error alone does not generally minimize the error seen by the following projection. This distinction motivates our learning objective: we preserve the complete intermediate representation while also accounting for the error transmitted to the next linear projection when such a projection is explicitly available. The proof is provided in Appendix C.2.

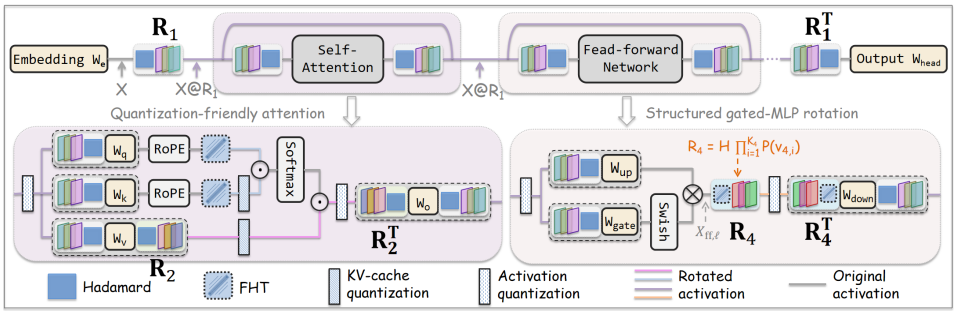


Figure 2: **Deployment of MirrorQuant rotations across a Transformer block.** R_1 defines the residual-stream basis, $R_{2,\ell}$ rotates the attention context immediately before the output projection, and R_3 applies a fixed normalized Hadamard transform to the query/key paths. These transforms are either folded into adjacent weights or fixed. At the gated-MLP interface, $R_{4,\ell} = H \prod_{i=1}^{K_4} P(\mathbf{v}_{4,\ell,i})$ remains on the activation path, while $R_{4,\ell}^\top$ is folded into the effective down-projection weight.

For later use, we define the normalized reconstruction and projection errors as

$$\mathcal{L}_{\text{ref}}(X;R) = \frac{\|E_R(X)\|_F^2}{\|X\|_F^2 + \epsilon}, \quad \mathcal{L}_{\text{proj}}(X;R,W) = \frac{\|E_R(X)W\|_F^2}{\|XW\|_F^2 + \epsilon}, \tag{7}$$

where ϵ=1e-6. The first measures how much the quantized rotation changes the representation itself, while the second measures how much of that error is passed through the following linear projection.

2.3 Rotation Placement and Deployment Constraint

We adopt the four rotation sites of prior learned-rotation PTQ methods (Liu et al., 2025; Shao et al., 2025). MirrorQuant retains these sites while redesigning learned-rotation parameterization and optimization. As shown in Fig. 2, R_1 changes the residual-stream basis, R_2 rotates the attention context before the output projection, R_3 applies a fixed normalized Hadamard transform to queries and keys, and R_4 rotates gated-MLP activations between the elementwise product and the down projection.

The key deployment distinction is that R_1 and R_2 are foldable, whereas the activation-side R_4 is not. After learning, R_1 and R_2 can be absorbed into adjacent linear weights, requiring no separate online rotation. Because R_4 follows the gated MLP's elementwise product, its forward transform cannot be absorbed into the preceding gate and up projections. Its transpose $R_4^\top$ can be folded into W_{down}, but R_4 itself must still be applied during inference. Thus, a dense learned R_4 would require $\mathcal{O}(d_f^2)$ additional rotation state and online computation, where d_f is the gated-MLP intermediate dimension. This motivates a learnable R_4 parameterization that remains efficient online.

3 Method

Building on the deployment asymmetry described above, MirrorQuant combines a compact Householder parameterization with projection-guided progressive learning. Householder chains enable efficient rotation learning and online execution without dense learned matrices, while projection guidance prioritizes errors according to their distortion after the immediate downstream linear projection. We first learn and freeze the foldable R_1 and R_2, then learn the online R_4 from gated-MLP states regenerated under the resulting upstream activation-quantization configuration.

3.1 Efficient Learnable Rotations via Householder Chains

We first address the parameterization challenge caused by this deployment asymmetry. As established above, R_1 and R_2 can be folded into adjacent linear weights after calibration, while R_3 remains fixed. In contrast, R_4 follows the element-wise product in the gated MLP and cannot be fully folded, leaving an activation-side rotation at inference. A dense R_4 therefore incurs quadratic storage and computation, directly offsetting the efficiency gains of low-bit quantization.

Single Householder Reflection. To avoid this quadratic cost, we start from a fixed orthogonal base and augment it with a short learnable Householder chain. We begin with a single Householder reflection. For a nonzero vector $v \in \mathbb{R}^d$, let

$$u = \frac{v}{\|v\|_2}, \qquad P(v) = I - 2uu^\top. \tag{8}$$

Since $P(v)^\top P(v) = I$, the reflection is exactly orthogonal. Although $P(v)$ is a dense $d \times d$ matrix if explicitly formed, neither the matrix nor its product with an activation needs to be materialized. For a row vector $x \in \mathbb{R}^{1\times d}$,

$$xP(v) = x - 2x(uu^\top) = x - 2(xu)u^\top. \tag{9}$$

The last equality follows from *associativity of matrix multiplication*. **Consequently, the reflection requires only $\mathcal{O}(d)$ computation.** It requires storing only one length-d Householder vector, reducing the learned state from quadratic for a dense rotation to linear.

A single Householder reflection is determined by one direction and represents a reflection across a single hyperplane. Although it provides exact orthogonality with linear storage and computation, it represents only a restricted subset of orthogonal transformations. This motivates composing multiple reflections to increase the capacity of the learned rotation while retaining the same efficient structure. Each additional reflection introduces only one length-d vector and $\mathcal{O}(d)$ computation.

From a Single Reflection to a Householder Chain. We define a depth-K Householder chain as $R_{\text{hh}} = \prod_{i=1}^{K} P(v_i)$. The product is never explicitly materialized. Instead, the reflections are applied sequentially:

$$x_0 = x, \qquad x_i = x_{i-1} - 2(x_{i-1}u_i)u_i^\top, \quad i = 1, \dots, K. \tag{10}$$

This preserves the same computational advantage for the full chain: storing K Householder vectors requires $\mathcal{O}(Kd)$ learned state, and applying the chain requires $\mathcal{O}(Kd)$ computation. Since every factor is orthogonal, their product is also exactly orthogonal.

We construct the rotation as the composition of a fixed orthogonal base B and the Householder chain:

$$R = BR_{\text{hh}} = B\prod_{i=1}^{K} P(v_i). \tag{11}$$

For fixed B and depth K, the corresponding compact rotation family is

$$\mathcal{H}_K(B) = \left\{ B\prod_{i=1}^{K} P(v_i) \;:\; v_i \in \mathbb{R}^d \setminus \{0\} \right\}. \tag{12}$$

Here, the fixed base provides the initial orthogonal transformation, while the Householder chain introduces a model-specific learnable correction without requiring a dense learned matrix. For the online MLP rotation, we use a Hadamard transform as the base,

$$R_4 = H\prod_{i=1}^{K_4} P(v_{4,i}). \tag{13}$$

The Hadamard transform costs $\mathcal{O}(d_f \log d_f)$, while the learned reflections cost $\mathcal{O}(K_4 d_f)$. Since K_4 is a fixed small depth independent of d_f, the total online complexity satisfies

$$\mathcal{O}(d_f \log d_f + K_4 d_f) = \mathcal{O}(d_f \log d_f). \tag{14}$$

The resulting rotation therefore requires $\mathcal{O}(K_4 d_f)$ learned state and $\mathcal{O}(d_f \log d_f)$ online computation, compared with the $\mathcal{O}(d_f^2)$ storage and computation required by a dense R_4.

The same parameterization also benefits the foldable R_1 and R_2. Although these rotations introduce no additional matrix multiplication after deployment, learning them as dense matrices still requires $\mathcal{O}(d^2)$ calibration parameters together with their gradients and optimizer states. We therefore use the same Householder parameterization for R_1 and R_2, reducing their learnable state to $\mathcal{O}(Kd)$ during calibration while preserving exact orthogonality. R_3 remains fixed. The number of reflections K directly controls the additional learned state and, for R_4, the online computation.

Finally, we initialize the Householder chain close to the identity so that the full rotation starts near its fixed base B. Simply shrinking v does not achieve this because $P(\alpha v) = P(v)$ for any nonzero scalar α. We therefore initialize consecutive reflections as nearly canceling pairs. For unit vectors u_j and z_j satisfying $z_j \perp u_j$, the two normals are initialized as u_j and $(u_j + \delta z_j)/\|u_j + \delta z_j\|_2$, with $\delta = 10^{-3}$. Identical normals cancel exactly, since $P(u_j)P(u_j) = I$, while the small perturbation keeps the chain close to, but not exactly at, the identity.

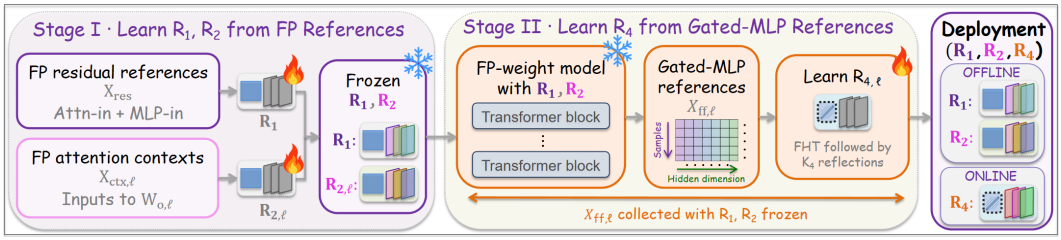


Figure 3: **Progressive learning of MirrorQuant rotations.** Stage I learns the shared R_1 from full-precision residual-stream references and each layer-wise $R_{2,\ell}$ from attention contexts. After fixing R_1 and R_2, we enable the corresponding upstream activation quantizers and regenerate the gated-MLP states for Stage II. At deployment, R_1 and R_2 are folded offline, while the Hadamard transform and K_4 Householder reflections remain online.

3.2 Projection-Guided Learning

The compact Householder parameterization reduces learning and deployment costs; with a fixed reflection budget, its learnable capacity should be directed toward errors that matter most downstream. Representation reconstruction is a natural objective, yet it treats all error directions equally. As shown by Proposition 2.1, reconstruction and projection errors are generally inequivalent, so representation errors of similar magnitude can have substantially different effects after the following linear projection. We therefore combine full-space representation reconstruction with interface-specific projection guidance whenever an explicit linear consumer is available.

We optimize the rotations in two stages, as illustrated in Fig. 3. Stage I learns the foldable R_1 and R_2 from full-precision references. After fixing these rotations, we enable the upstream activation quantizers and regenerate the gated-MLP states used to learn R_4 in Stage II. This aligns the R_4 calibration inputs with the upstream rotation and activation-quantization configuration.

Stage I: Learning the Foldable Rotations. Although R_1 and $R_{2,\ell}$ act at different interfaces, Stage I learns both with a unified reconstruction-and-projection objective. We represent each cached sample as (X, g, W), where $g \in \{0, 1\}$ controls projection guidance. When $g = 1$, W denotes the corresponding linear consumer; when $g = 0$, the projection term is disabled. Given a reference set $\mathcal{D}$, a fixed base B, a reflection depth K, and a loss weight λ, we solve

$$R^\star = \arg\min_{R \in \mathcal{H}_K(B)} \mathbb{E}_{(X,g,W)\sim\mathcal{D}} \left[\mathcal{L}_{\text{ref}}(X; R) + \lambda g\, \mathcal{L}_{\text{proj}}(X; R, W)\right]. \tag{15}$$

For R_1, the reference set $\mathcal{D}_1$ collects full-precision residual-stream inputs to self-attention and gated-MLP sublayers across layers. Every sample contributes to $\mathcal{L}_{\text{ref}}$; a self-attention sample at layer ℓ additionally uses $g = 1$ and $W = W_{v,\ell}$, whereas a gated-MLP sample uses $g = 0$.

For each $R_{2,\ell}$, the reference set $\mathcal{D}_{2,\ell}$ contains full-precision attention contexts immediately before $W_{o,\ell}$. Every sample uses $g = 1$ and $W = W_{o,\ell}$, guiding $R_{2,\ell}$ by the error delivered to the output projection. Each rotation is optimized independently by updating only its Householder vectors; model weights, fixed bases, cached references, and quantizer parameters remain unchanged. For grouped-query attention, we use the head-balanced objective described in Appendix B.1.

Stage II: Learning the Online MLP Rotation. With R_1 and R_2 fixed, we enable activation quantization at the corresponding upstream interfaces and rerun the calibration data to obtain

$$X_{\text{ff},\ell} = \text{SiLU}(X_\ell W_{\text{gate},\ell}) \odot (X_\ell W_{\text{up},\ell}), \tag{16}$$

where X_ℓ is the gated-MLP input produced by the fixed Stage-I rotation-and-quantization configuration. We collect these states in $\mathcal{D}_{\text{ff},\ell}$ and learn each $R_{4,\ell}$ independently.

At this interface, the down projection is the immediate linear consumer and provides a local measure of how representation errors enter the block output. After folding the Stage-I residual rotation, let $\widetilde{W}_{\text{down},\ell} = W_{\text{down},\ell} R_1$ denote the full-precision effective down-projection weight, with $\widetilde{W}_{\text{down},\ell} = U_\ell \Sigma_\ell V_\ell^\top$. The resulting projection error can be expressed as

$$\left\| E_R(X_{\text{ff},\ell}) \widetilde{W}_{\text{down},\ell} \right\|_F^2 = \sum_i \sigma_{\ell,i}^2 \left\| E_R(X_{\text{ff},\ell}) \mathbf{u}_{\ell,i} \right\|_2^2, \tag{17}$$

where $\mathbf{u}_{\ell,i}$ and $\sigma_{\ell,i}$ are the corresponding left singular vector and singular value. Thus, the down projection assigns greater importance to errors along directions with larger singular values.

To reduce the cost of evaluating the full projection metric, we retain the leading r singular components and define $G_\ell = U_{\ell,1:r}\Sigma_{\ell,1:r}$. This factor is used only during calibration and does not enter the deployed model. Because G_ℓ covers only a subspace of the gated-MLP representation, we complement the projected error with full-space reconstruction and optimize

$$R^\star_{4,\ell} = \arg\min_{R\in\mathcal{H}_{K_4}(H)} \mathbb{E}_{X\sim\mathcal{D}_{\mathrm{ff},\ell}} \left[\|E_R(X)G_\ell\|_F^2 + \lambda_{\mathrm{act}} \|E_R(X)\|_F^2 \right]. \tag{18}$$

The first term prioritizes errors along the leading projection-sensitive directions, while the second constrains the complete representation. Appendix C.2 analyzes the term omitted by the fixed-rank approximation, and Appendix B.2 provides the optimization settings.

Deployment. At deployment, the learned R_1 and R_2 are folded into adjacent weights, while R_3 remains a fixed normalized Hadamard transform on the query–key path. At the gated-MLP interface, the final down-projection weight is $\widehat{W}_{\mathrm{down},\ell} = R_{4,\ell}^\top \widetilde{W}_{\mathrm{down},\ell}$, while the activation-side $R_{4,\ell}$ is executed as an FHT followed by K_4 Householder reflections. Thus, among the learned rotations, only R_4 remains online, and MirrorQuant avoids evaluating any dense learned rotation during inference.

4 Experiments

Models and tasks. We evaluate Llama-2-7B/13B/70B, Llama-3-8B/70B, and Qwen2.5-1.5B/3B/7B/14B, covering two model families and scales from 1.5B to 70B parameters. We report perplexity (PPL) on WikiText-2, Penn Treebank (PTB), and C4, together with zero-shot accuracy on WinoGrande, SocialIQA, PIQA, OpenBookQA, LAMBADA, HellaSwag, ARC-Easy, ARC-Challenge, and MMLU. The arithmetic mean across these nine tasks is denoted 0-shot$_9$. Evaluation details and dataset references are provided in Appendix B.3.

Baselines and implementation. We compare MirrorQuant with conventional PTQ baselines, including RTN, SmoothQuant, and GPTQ, as well as rotation-based methods, including QuaRot, SpinQuant, OSTQuant, and DartQuant. DartQuant serves as the primary matched baseline because MirrorQuant retains its rotation placement and quantization protocol. We evaluate three low-bit configurations: W4A4KV4, W4A4KV16, and W4A8KV16, where W, A, and KV denote the weight, activation, and KV-cache bit widths. For controlled comparison, all data-driven methods use the same 128 WikiText-2 sequences of length 2048, with quantizers, clipping rules, group sizes, tokenizers, and evaluation code matched across methods. Implementation details appear in Appendix B.2.

4.1 Quantization Accuracy

Model-scale evaluation. Table 1 evaluates five models from Llama-2 and Llama-3, ranging from 7B to 70B parameters, under three low-bit configurations. We use a Hadamard base for R_1 by default, while Llama-3-70B uses the fixed DartQuant-derived base analyzed in Appendix D.3. With these base choices, MirrorQuant achieves the highest 0-shot$_9$ accuracy in 14 of 15 settings and outperforms DartQuant in every matched comparison, with gains of 0.61 points on average and up to 1.23 points. It also reduces mean PPL by 0.29 on average, supporting the effectiveness of MirrorQuant across model scales and quantization settings.

Task-level evaluation. Table 5 expands the largest-scale evaluation to all nine zero-shot tasks on Llama-3-70B under the three low-bit settings. Conditional on the fixed DartQuant-derived R_1 base, MirrorQuant achieves the highest average accuracy in every setting, outperforming DartQuant by 0.86, 1.23, and 0.22 points under W4A4KV4, W4A4KV16, and W4A8KV16, respectively. It also ranks first on five, six, and four individual tasks. These results show that, conditional on the fixed base, the aggregate improvements persist across quantization settings and are supported by gains on multiple downstream benchmarks rather than a single task.

Model-family evaluation. Table 2 evaluates four Qwen2.5 models ranging from 1.5B to 14B parameters under W4A4KV16 and W4A8KV16. MirrorQuant outperforms DartQuant in all eight matched settings, with a maximum 0-shot$_9$ accuracy gain of 1.81% on Qwen2.5-3B under W4A4KV16. The improvements remain consistent across every evaluated Qwen2.5 size and both quantization settings. Together with the preceding Llama results, this confirms that the gains extend across model families, parameter scales, and low-bit configurations.

Table 1: **Main results on Llama-2 and Llama-3.** PPL is the arithmetic mean over WikiText-2, PTB, and C4; 0-shot$_9$ is the mean accuracy (%) over nine tasks. Best and second-best quantized results in each matched setting are shown in bold and underlined, respectively.

Bits (W-A-KV)	Method	Llama-2-7B		Llama-2-13B		Llama-2-70B		Llama-3-8B		Llama-3-70B	
		PPL ↓	0-shot$_9$ ↑	PPL ↓	0-shot$_9$ ↑	PPL ↓	0-shot$_9$ ↑	PPL ↓	0-shot$_9$ ↑	PPL ↓	0-shot$_9$ ↑
16-16-16	Full precision	16.88	61.16	20.85	64.28	11.09	69.53	8.92	66.04	6.19	72.70
4-4-4	RTN	3070.58	29.47	1989.04	28.96	2073.27	29.25	397.61	30.80	405.67	30.66
	SmoothQuant	9978.11	29.39	1957.88	28.62	1823.00	29.64	745.14	29.87	398.43	30.14
	GPTQ	11615.20	29.95	2151.95	29.47	1925.89	28.90	740.99	29.74	382.05	30.27
	QuaRot	30.06	56.31	25.77	60.01	11.81	67.18	12.91	56.35	12.55	58.94
	SpinQuant	23.60	56.50	24.05	61.02	11.94	67.44	11.64	60.09	10.18	64.76
	OSTQuant	20.44	57.25	23.56	61.91	11.67	68.11	11.04	61.06	8.09	67.84
	DartQuant	19.82	58.04	23.51	61.82	11.59	68.33	10.88	61.60	8.37	68.35
	MirrorQuant	**19.43**	**58.58**	**22.81**	**62.79**	**11.54**	**68.71**	**10.79**	**62.60**	**8.04**	**69.21**
4-4-16	RTN	2735.48	29.63	2140.10	29.58	2041.80	29.06	344.30	30.13	388.92	30.23
	SmoothQuant	11447.59	29.83	2422.87	29.30	1616.87	29.36	604.37	29.50	420.92	30.91
	GPTQ	11302.87	29.80	2732.73	29.54	1779.99	29.00	625.94	29.69	388.89	30.61
	QuaRot	20.76	56.99	24.82	61.12	11.68	67.25	12.11	58.20	12.09	59.78
	SpinQuant	21.29	54.67	24.21	60.12	11.55	67.71	11.47	60.83	9.63	66.06
	OSTQuant	19.86	57.97	23.34	61.97	11.98	68.29	10.72	62.37	7.98	68.03
	DartQuant	19.16	58.16	23.35	62.01	11.55	68.18	10.68	61.90	8.25	68.92
	MirrorQuant	**18.81**	**58.92**	**22.57**	**62.86**	**11.49**	**68.70**	**10.57**	**62.78**	**7.94**	**70.15**
4-8-16	RTN	20.40	58.43	24.77	62.00	11.65	68.13	10.96	62.65	7.53	69.27
	SmoothQuant	3177.09	59.00	23.97	62.53	11.83	67.69	11.79	61.65	7.33	70.40
	GPTQ	3946.93	58.82	23.15	62.50	11.90	67.41	11.47	62.35	7.31	70.50
	QuaRot	17.97	59.72	21.87	63.10	11.20	69.20	9.59	64.79	6.92	70.85
	SpinQuant	17.39	**60.63**	21.80	63.77	11.30	69.39	9.51	64.89	6.69	71.76
	OSTQuant	17.85	60.38	21.92	63.67	11.29	69.49	9.52	64.66	6.68	71.78
	DartQuant	17.84	59.85	21.85	63.68	11.19	69.54	9.50	65.12	6.67	71.99
	MirrorQuant	**17.29**	60.58	**21.24**	**63.79**	**11.14**	**69.57**	**9.49**	**65.22**	**6.64**	**72.21**

4.2 Training and Online Efficiency

Tables 3 and 4 evaluate rotation-learning and online efficiency, respectively. Table 3 reports optimization throughput and peak GPU memory on Llama-2-13B and Llama-3-8B. Using the shared calibration set, we report completed epochs per GPU-hour to characterize per-epoch optimization efficiency under each method's prescribed schedule. Table 4 reports quantization quality and inference cost at different Householder depths, alongside fixed-FHT and dense online references. The checkpoint, quantized operators, inputs, backend, and other rotations are fixed across all online measurements. Prefill and decode latency are reported in milliseconds and milliseconds per token, respectively, and memory in GiB. Complete results and benchmark protocols are provided in Appendix B.5 and Appendix B.4.

Compared with SpinQuant and OSTQuant, MirrorQuant achieves more than $3.2\times$ the per-epoch training throughput and reduces peak GPU memory by at least 78.9%, while jointly learning R_1, R_2, and the online R_4. For online deployment, $K_4 = 4$ provides the best overall quantization quality and is used by default. It adds only 4.3% prefill and 1.0% decode latency over the fixed FHT, while reducing them by 29.6% and 5.9%, respectively, relative to the dense online reference. Its peak memory remains within 0.005 GiB of the fixed FHT and is 0.378 GiB below the dense reference.

Table 2: **Cross-family evaluation on Qwen2.5 base models.** We report perplexity (PPL; lower is better) and nine-task zero-shot accuracy (0-shot$_9$; higher is better) at four model scales. Best and second-best quantized results are shown in bold and underlined, respectively.

Bits (W-A-KV)	Method	Qwen2.5-1.5B		Qwen2.5-3B		Qwen2.5-7B		Qwen2.5-14B	
		PPL ↓	0-shot$_9$ ↑	PPL ↓	0-shot$_9$ ↑	PPL ↓	0-shot$_9$ ↑	PPL ↓	0-shot$_9$ ↑
16-16-16	Full precision	14.09	59.55	12.18	62.67	10.51	67.26	8.84	69.97
4-4-16	QuaRot	17.48	51.87	14.45	56.86	12.20	62.12	10.12	67.10
	DartQuant	17.32	52.23	14.53	56.58	12.20	62.48	9.90	67.32
	MirrorQuant	**16.69**	**53.75**	**14.10**	**58.39**	**11.85**	**63.50**	**9.83**	**68.25**
4-8-16	QuaRot	14.82	57.88	12.63	62.13	**10.75**	66.27	9.10	69.42
	DartQuant	14.79	58.10	12.63	61.97	10.76	66.53	**9.09**	68.90
	MirrorQuant	**14.75**	**58.21**	**12.60**	**62.14**	**10.75**	**66.67**	9.10	**69.94**

Table 3: **Rotation-Learning Efficiency.** Training throughput and peak memory.

Metric	Method	Llama-2-13B	Llama-3-8B
Throughput ↑	SpinQuant	3.88	8.33
	OSTQuant	3.43	38.62
	DartQuant	**85.31**	**128.57**
	MirrorQuant	60.23	126.91
Memory ↓	SpinQuant	256.56	126.68
	OSTQuant	374.32	153.97
	DartQuant	**20.37**	**16.39**
	MirrorQuant	30.64	26.68

Table 4: **Online-Rotation Efficiency.** Accuracy and inference cost on Llama-3-8B.

Online R_4	K_4	Quantization Quality		Inference Cost ↓		
		PPL ↓	0-shot$_9$ ↑	Prefill	Decode	Memory
Dense rotation	–	–	–	835.23	192.60	18.311
Fixed FHT	0	10.83	62.34	**563.64**	**179.45**	**17.928**
MirrorQuant	2	10.81	62.41	575.79	180.38	17.930
	4	**10.79**	**62.60**	587.85	181.23	17.933
	8	**10.79**	62.53	616.03	182.46	17.939

Table 5: **Task-level zero-shot accuracy (%) on Llama-3-70B.** Best and second-best quantized results in each setting are shown in bold and underlined, respectively.

Bits (W-A-KV)	Method	WG	SIQA	PIQA	OBQA	LAMB.	HS	ARC-E	ARC-C	MMLU	Avg.
16-16-16	Full precision	80.82	50.61	84.44	48.40	79.31	84.99	85.98	64.25	75.47	72.70
4-4-4	RTN	51.78	33.37	54.73	28.80	3.16	29.23	28.83	22.70	23.32	30.66
	SmoothQuant	50.51	34.85	51.14	27.80	3.07	29.04	29.00	22.35	23.52	30.14
	GPTQ	49.41	34.24	51.63	25.40	3.63	29.68	29.92	24.66	23.89	30.27
	QuaRot	66.85	43.35	77.04	41.00	66.50	70.47	72.01	46.76	46.45	58.94
	SpinQuant	73.80	43.81	79.38	43.40	72.56	81.20	76.81	54.10	57.81	64.76
	OSTQuant	73.48	45.55	81.99	45.40	75.78	**83.06**	80.05	56.31	**68.97**	67.84
	DartQuant	**77.35**	47.75	81.94	46.20	**76.31**	82.22	80.35	55.80	67.23	68.35
	MirrorQuant	76.95	**48.11**	**82.70**	**46.80**	75.96	82.44	**83.25**	**59.64**	67.02	**69.21**
4-4-16	RTN	49.88	33.06	53.37	26.60	3.43	29.26	29.84	23.04	23.63	30.23
	SmoothQuant	52.57	33.78	52.18	27.00	4.31	29.16	32.07	23.46	23.65	30.91
	GPTQ	51.54	33.98	52.83	27.00	3.96	29.43	29.63	23.55	23.54	30.61
	QuaRot	68.51	45.04	77.75	41.00	67.49	71.79	72.53	46.93	46.97	59.78
	SpinQuant	73.56	44.52	79.54	46.00	74.13	81.00	79.42	56.23	60.13	66.06
	OSTQuant	73.64	46.32	81.56	45.40	75.61	**83.19**	80.13	57.17	**69.29**	68.03
	DartQuant	75.77	47.75	**82.97**	45.80	76.50	82.41	81.44	58.45	69.19	68.92
	MirrorQuant	**77.82**	**48.11**	82.21	**48.60**	**77.43**	82.92	**85.02**	**60.84**	68.38	**70.15**
4-8-16	RTN	76.09	47.49	81.94	46.60	79.00	81.13	81.44	58.45	71.31	69.27
	SmoothQuant	78.45	49.08	82.10	46.80	**79.25**	82.64	83.16	60.67	71.46	70.40
	GPTQ	78.69	49.18	82.92	46.80	78.83	82.36	82.91	61.52	71.31	70.50
	QuaRot	79.32	48.52	83.62	47.60	78.46	84.11	83.16	60.07	72.82	70.85
	SpinQuant	80.27	49.90	84.22	**49.20**	78.58	84.42	83.54	61.77	73.97	71.76
	OSTQuant	79.51	50.16	84.31	48.76	78.63	84.48	83.97	62.20	**74.04**	71.78
	DartQuant	79.64	50.41	**84.39**	**49.20**	78.69	**84.55**	84.43	62.63	73.96	71.99
	MirrorQuant	**80.51**	**50.77**	84.33	48.40	78.91	84.38	**85.23**	**63.74**	73.66	**72.21**

4.3 Ablation Study

Table 7 evaluates the cumulative contributions of R_1, R_2, and R_4 on Llama-3-8B under W4A4KV4, with all other settings fixed. Relative to DartQuant, learning the foldable R_1 improves 0-shot$_9$ accuracy by 0.32 points, and adding the attention-side R_2 raises the gain to 0.74 points while further reducing PPL. The complete MirrorQuant with learned online R_4 reaches a 1.00-point accuracy gain and reduces PPL from 10.88 to 10.79. Table 9 further examines how R_4 is learned: projection guidance improves accuracy over reconstruction alone, cache regeneration improves both metrics over a stale cache, and the truncated metric obtains similar results to the full projection. Together, these results demonstrate the complementary effects of the proposed rotations and support the evaluated learning choices for R_4. Task-level results and sensitivity analyses are provided in Appendix D.

5 Conclusion

In this paper, we presented MirrorQuant, an efficient framework for learning foldable and online rotations in low-bit LLM quantization. By composing fixed orthogonal bases with short Householder chains, MirrorQuant reduces the learnable state of dense rotations from quadratic to linear for fixed depth while preserving exact orthogonality. Projection-guided objectives penalize distortion after immediate linear projections, while progressive calibration learns the online rotation from representations consistent with the fixed upstream configuration. Foldable rotations are absorbed into weights, and the online rotation combines an FHT with learned reflections for log-linear complexity at fixed depth. Across all 23 matched settings under the stated base choices, MirrorQuant outperforms DartQuant in 0-shot$_9$ accuracy by up to 1.81%. On Llama-3-8B, the online rotation improves quantization quality over the fixed FHT with only 4.3% prefill and 1.0% decode latency overhead, while reducing these latencies by 29.6% and 5.9% relative to a dense realization.

Reproducibility Statement

Section 3 provides the complete formulation of MirrorQuant, and Section 4 describes the evaluated models, datasets, baselines, quantization settings, and metrics. Appendix B documents reference-cache construction, optimization and initialization settings, evaluation procedures, and hardware measurement protocols. Theoretical analyses are provided in Appendix C, with additional ablations and detailed results reported in Appendices D and E.

Ethics Statement

This work studies efficient post-training quantization of publicly available pretrained language models and does not involve human-subject studies, collection of personal data, or deployment in sensitive application domains. Our experiments use publicly available models, datasets, and evaluation benchmarks under standard research protocols. We do not identify additional ethical concerns specific to the proposed methodology.

AI Use Statement

Generative AI tools were used to assist with language editing and manuscript consistency checks, including improvements to grammar, clarity, and presentation. The authors reviewed and revised all AI-assisted text and take full responsibility for the methods, experiments, results, and conclusions presented in this work.

References

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. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: reasoning about physical commonsense in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pp. 7432–7439, 2020.

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

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the AI2 reasoning challenge. *CoRR*, abs/1803.05457, 2018.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. Llm.int8(): 8-bit matrix multiplication for transformers at scale. *CoRR*, abs/2208.07339, 2022.

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

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. GPTQ: accurate post-training quantization for generative pre-trained transformers. *CoRR*, abs/2210.17323, 2022.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

Xing Hu, Yuan Cheng, Dawei Yang, Zhixuan Chen, Zukang Xu, Jiangyong Yu, Chen Xu, Zhihang Yuan, Zhe Jiang, and Sifan Zhou. Ostquant: Refining large language model quantization with orthogonal and scaling transformations for better distribution fitting. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025.

Yesheng Liang, Haisheng Chen, Song Han, and Zhijian Liu. Paroquant: Pairwise rotation quantization for efficient reasoning LLM inference. *CoRR*, abs/2511.10645, 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. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Xingyu Dang, and Song Han. AWQ: activation-aware weight quantization for LLM compression and acceleration. *CoRR*, abs/2306.00978, 2023.

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. In *Proceedings of Machine Learning and Systems (MLSys)*, 2025.

Han Liu, Haotian Gao, Changya Li, Feng Zhang, Xiaotong Zhang, Wei Wang, and Hong Yu. Ruquant: Towards refining uniform quantization for large language models. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 843–854, 2026.

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 *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025.

Yuexiao Ma, Huixia Li, Xiawu Zheng, Feng Ling, Xuefeng Xiao, Rui Wang, Shilei Wen, Fei Chao, and Rongrong Ji. Affinequant: Affine transformation quantization for large language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024.

Mitchell P. Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. The penn treebank: Annotating predicate argument structure. In *Human Language Technology: Proceedings of a Workshop (HLT)*, 1994.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? A new dataset for open book question answering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2381–2391, 2018.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2016.

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. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pp. 8732–8740, 2020.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social iqa: Commonsense reasoning about social interactions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4462–4472, 2019.

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

Yuantian Shao, Yuanteng Chen, Peisong Wang, Jianlin Yu, Jing Lin, Yiwu Yao, Zhihui Wei, and Jian Cheng. Dartquant: Efficient rotational distribution calibration for LLM quantization. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Yuxuan Sun, Ruikang Liu, Haoli Bai, Han Bao, Kang Zhao, Yuening Li, Jiaxin Hu, Xianzhi Yu, Lu Hou, Chun Yuan, Xin Jiang, Wulong Liu, and Jun Yao. Flatquant: Flatness matters for LLM quantization. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 267, 2025.

Llama Team. The llama 3 herd of models. *CoRR*, abs/2407.21783, 2024.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288, 2023.

Albert Tseng, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa. Quip#: Even better LLM quantization with hadamard incoherence and lattice codebooks. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 235, pp. 48630–48656, 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 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1648–1665, 2023.

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

Guangxuan Xiao, Ji Lin, Mickaël Seznec, Hao Wu, Julien Demouth, and Song Han. Smoothquant: Accurate and efficient post-training quantization for large language models. In *Proceedings of International Conference on Machine Learning (ICML)*, volume 202, pp. 38087–38099, 2023.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *CoRR*, abs/2412.15115, 2024.

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. *CoRR*, abs/2304.01089, 2023.

Artur Zagitov, Gleb Molodtsov, and Aleksandr Beznosikov. HARP: hadamard-preconditioned adaptive rotation processor for extreme LLM quantization. *CoRR*, abs/2605.29843, 2026.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 4791–4800, 2019.

APPENDIX CONTENTS

Appendix A. Related Work
Appendix B. Implementation and Measurement Details
Appendix B.1 Calibration-Cache Construction
Appendix B.2 Optimization and Initialization
Appendix B.3 Evaluation Protocol
Appendix B.4 Online R_4 Benchmark Protocol
Appendix B.5 Complete Rotation-Learning Efficiency Results
Appendix C. Proofs and Computational Complexity
Appendix C.1 Paired-Reflection Initialization and Geometry
Appendix C.2 Projection-Metric Analysis
Appendix C.3 Parameter and Arithmetic Costs
Appendix D. Additional Experimental Analyses
Appendix D.1 Component Ablation and Task-Level Results
Appendix D.2 R_2 Reflection-Depth Ablation
Appendix D.3 R_1 Base Sensitivity and Progressive Construction
Appendix E. Detailed Llama Task- and Corpus-Level Results

A RELATED WORK

Post-training quantization and representation transformation. Post-training quantization (PTQ) compresses pretrained LLMs without full-model retraining. For weight-only quantization, GPTQ (Frantar et al., 2022) uses approximate second-order reconstruction to compensate for rounding error, AWQ (Lin et al., 2023) rescales activation-salient channels to protect important weights, and SpQR (Dettmers et al., 2024) retains a sparse set of sensitive weights in higher precision. Joint weight–activation quantization is further complicated by activation outliers that dominate the low-bit dynamic range. LLM.int8() (Dettmers et al., 2022) isolates such dimensions through mixed precision; SmoothQuant (Xiao et al., 2023) applies an equivalent scaling transformation; Outlier Suppression+ (Wei et al., 2023) combines equivalent shifting and scaling; and RPTQ (Yuan et al., 2023) reduces inter-channel range variation through reordering and grouping. OmniQuant (Shao et al., 2024) jointly learns equivalent transformations and weight-clipping parameters, while AffineQuant (Ma et al., 2024) optimizes equivalent affine transformations.

Orthogonal rotations and structured parameterizations. Orthogonal rotations preserve vector geometry, and function-preserving rotations can often be fused into adjacent linear weights. QuIP (Chee et al., 2023) and QuIP# (Tseng et al., 2024) use randomized incoherence processing for weight quantization, while QuaRot (Ashkboos et al., 2024) applies randomized Hadamard rotations across residual, attention, MLP, and KV-cache paths. SpinQuant (Liu et al., 2025) learns dense orthogonal rotations, OSTQuant (Hu et al., 2025) jointly optimizes orthogonal and scaling transformations, and DartQuant (Shao et al., 2025) improves rotation-learning efficiency through distribution-aware local calibration. Other structured transformations include the blockwise rotations and permutations of DuQuant (Lin et al., 2024), the Kronecker-structured affine maps of FlatQuant (Sun et al., 2025), and the Givens rotations with channelwise scaling of ParoQuant (Liang et al., 2025). DFRot (Xiang & Zhang, 2024) refines rotations with a weighted objective for massive activations, RUQuant (Liu et al., 2026) combines Householder reflections and Givens rotations to refine uniform activation quantization, and HARP (Zagitov et al., 2026) uses Hadamard-preconditioned block-orthogonal stages. In contrast, MirrorQuant applies short Householder chains to R_1, R_2, and R_4, combines representation reconstruction with interface-specific projection guidance, and executes the learned online R_4 in $\mathcal{O}(d_f \log d_f)$ time for fixed K_4.

B IMPLEMENTATION AND MEASUREMENT DETAILS

B.1 CALIBRATION-CACHE CONSTRUCTION

The reference caches for R_1 and R_2 are collected once from the original full-precision model. The R_1 cache contains residual-stream states entering the self-attention and gated-MLP sublayers. All entries contribute to representation reconstruction, while self-attention entries additionally measure the error propagated through the corresponding value projection $W_{v,\ell}$.

For each layer ℓ, the R_2 cache contains full-precision attention contexts immediately before the output projection $W_{o,\ell}$. These references measure both representation reconstruction and the resulting error after $W_{o,\ell}$. Let h_q be the number of query heads, d_h the dimension of each head, and $g(j)$ the KV group associated with query head j. For a fixed layer, the full rotation is assembled from headwise blocks as

$$R_{2,\ell} = \mathrm{blkdiag}_{j=1}^{h_q} R_{2,g(j)}, \qquad R_{2,g} = B_{2,g} \prod_{i=1}^{K_2} P(v_{2,g,i}) \in \mathbb{R}^{d_h \times d_h}. \tag{19}$$

Here $B_{2,g}$ is the fixed orthogonal base for group g. All query heads in the same KV group share the learned block $R_{2,g}$. Thus $K_2 = 4$ denotes four reflections per KV group; the layerwise $R_{2,\ell}$ repeats these blocks across the concatenated attention width $h_q d_h$. Equation (15) gives the single-rotation template: for $R_{2,\ell}$, its constraint $R \in \mathcal{H}_K(B)$ applies separately to each learned block $R_{2,g} \in \mathcal{H}_{K_2}(B_{2,g})$, not to the assembled $R_{2,\ell}$. The reconstruction and $W_{o,\ell}$ projection losses are evaluated on the assembled block-diagonal rotation and the concatenated attention context.

Let $\mathcal{G}_g$ be the set of query heads sharing KV group g, with $r = |\mathcal{G}_g|$, and let $X_{n,j} \in \mathbb{R}^{1\times d_h}$ be the attention context of token n at query head j. With $\mathcal{Q}_A$ denoting the activation quantizer, define

$$e_{n,j} = \frac{\|\mathcal{Q}_A(X_{n,j}R_{2,g(j)}) - X_{n,j}R_{2,g(j)}\|_2^2}{\|X_{n,j}\|_2^2 + \epsilon}, \quad \bar{e}_{n,g} = \frac{1}{r}\sum_{j\in\mathcal{G}_g} e_{n,j}. \tag{20}$$

Because $R_{2,g(j)}$ is orthogonal, $e_{n,j}$ equals the normalized quantization error after mapping the activation back with $R_{2,g(j)}^\top$. For N cached tokens and G KV groups, the head-balancing regularizer is

$$\mathcal{L}_{\mathrm{bal}} = \frac{1}{NG}\sum_{n=1}^{N}\sum_{g=1}^{G} \frac{\frac{1}{r}\sum_{j\in\mathcal{G}_g}(e_{n,j} - \bar{e}_{n,g})^2}{\bar{e}_{n,g}^2 + \epsilon}. \tag{21}$$

This term penalizes relative variation in quantization error among query heads sharing a KV group, so a large error in one head cannot be hidden by the group average. Writing $X_n = [X_{n,1}, \ldots, X_{n,h_q}] \in \mathbb{R}^{1\times h_q d_h}$ for the concatenated attention context, the complete R_2 learning objective is

$$\mathcal{L}_{2,\ell}(R_{2,\ell}) = \mathbb{E}_{X\sim\mathcal{D}_{2,\ell}}\left[\mathcal{L}_{\mathrm{ref}}(X; R_{2,\ell}) + \lambda_2\mathcal{L}_{\mathrm{proj}}(X; R_{2,\ell}, W_{o,\ell})\right] + \lambda_{\mathrm{bal}}\mathcal{L}_{\mathrm{bal}}. \tag{22}$$

We set $\lambda_2 = 0.01$ for every evaluated model. For multi-head attention, $\lambda_{\mathrm{bal}} = 0$; for grouped-query attention, $\lambda_{\mathrm{bal}} = 0.03$. Both objectives include the full $W_{o,\ell}$ projection term, which is evaluated after concatenating the quantization errors of all query heads. The head-balancing regularizer is an additional GQA term, not a replacement for projection guidance.

After Stage I, R_1 and R_2 are inserted and frozen, and activation quantization is enabled at the corresponding upstream interfaces. We rerun the calibration data and collect the pre-down-projection gated-MLP states $X_{\mathrm{ff},\ell}$ used to learn R_4. Model weights remain in full precision during this cache-construction pass. Consequently, the R_4 cache reflects the fixed upstream rotations and activation-quantization configuration, while excluding effects of upstream weight quantization at deployment.

B.2 Optimization and Initialization

Comparison protocol. We compare MirrorQuant with round-to-nearest (RTN), SmoothQuant (Xiao et al., 2023), GPTQ (Frantar et al., 2022), QuaRot (Ashkboos et al., 2024), SpinQuant (Liu et al., 2025), OSTQuant (Hu et al., 2025), and DartQuant (Shao et al., 2025). DartQuant serves as the primary matched baseline because MirrorQuant retains its rotation placement and quantization protocol while replacing dense rotation adaptation with the proposed Householder parameterization and projection-guided learning. Quantizers, clipping rules, group sizes, tokenizers, and evaluation code are matched wherever applicable. FlatQuant (Sun et al., 2025) is discussed separately because it jointly changes the affine transformation and fused-kernel stack, whereas our tables provide a controlled comparison of rotation learning under a shared quantization pipeline rather than a system-level comparison across heterogeneous implementations.

Optimization setup. Unless otherwise specified, rotation learning uses 128 WikiText-2 samples with seed 0. Learning R_1 and R_2 uses sequences of length 2048, while the R_4 calibration cache

retains 128 tokens per sample following the same token-selection rule as DartQuant. We optimize only the Householder parameters for 20 epochs using Adam with $(\beta_1, \beta_2) = (0.9, 0.999)$ and learning rates of 0.01 for R_1 and R_2 and 0.003 for R_4. Pretrained weights, selected orthogonal bases, and quantizer settings remain fixed, and rounding follows the straight-through estimator used by the matched DartQuant implementation. For R_1, `global_loss_weight` and `pref_aq_lambda` are set to 1, while the scale, zero-point, range, and rounding regularization weights are zero. The projection weight `v_lambda` is 0.001 for Llama-2-7B, 0.01 for Llama-2-13B/70B, 0.10 for Llama-3-8B/70B and Qwen2.5-7B, and 0.05 for Qwen2.5-1.5B/3B/14B. For R_2, Llama-2 models use `aq`, whereas Llama-3 and Qwen2.5 use `gqa_aq`; both include the W_o projection term in Eq. (22) with $\lambda_2 = 0.01$. The `gqa_aq` objective additionally uses `gqa_balance_lambda`$= 0.03$ and `tail_lambda`$= 0$. We set `tail_tau` to 0.95 for Llama-3-8B/70B and Qwen2.5-7B and to 0.90 for Qwen2.5-1.5B/3B/14B; these values have no effect when `tail_lambda`$= 0$. For R_4, λ_{act} in Eq. (18) is set to 0.05 for Llama-2 and 0.10 for Llama-3 and Qwen2.5.

During rotation learning, activations use asymmetric 4-bit quantization with a 0.9 clipping ratio. The R_1 and R_4 activation groups span the full feature dimension, whereas R_2 uses groups of 128. At evaluation, weights use symmetric 4-bit quantization with group size 128 and activations use the same asymmetric setting at their evaluated bit width. The 4-bit key/value cache uses asymmetric quantization with group size 128, while the KV16 settings retain the cache at 16-bit precision. The truncated R_4 metric uses the leading $r = 256$ singular components of the full-precision effective weight $\widetilde{W}_{\text{down},\ell} = W_{\text{down},\ell} R_1$. This rank is fixed across layers and throughout R_4 learning.

We use an even Householder depth and initialize each consecutive pair with normals $\mathbf{u}$ and $(\mathbf{u} + \delta\mathbf{z})/\|\mathbf{u} + \delta\mathbf{z}\|_2$, where $\mathbf{z} \perp \mathbf{u}$ and $\delta = 10^{-3}$. The resulting transform is exactly orthogonal and begins near, rather than exactly at, its selected base. Unless varied in an ablation, every evaluated model uses $K_1 = K_2 = K_4 = 4$ and deploys (R_1, R_2, R_4). Qwen2.5-14B uses the same rotation depth, initialization, and learning-rate schedule as Qwen2.5-7B. The default base is Hadamard. For Llama-3-70B, R_1 uses the orthogonal R_1 from the matched DartQuant configuration as a fixed base; the Householder parameter and optimization costs reported for MirrorQuant are conditional on this fixed base and do not include its construction. This model-specific exception is analyzed in Appendix D.3. R_4 uses the fixed normalized Hadamard/FHT base in all evaluated configurations.

B.3 Evaluation Protocol

We evaluate the publicly released Llama-2 (Touvron et al., 2023), Llama-3 (Team, 2024), and Qwen2.5 (Yang et al., 2024) checkpoints. Each full-precision model and all of its quantized variants use the same tokenizer, preprocessing, sample order, context handling, and evaluation code.

Perplexity. We report perplexity on WikiText-2 (Merity et al., 2017), Penn Treebank (PTB) (Marcus et al., 1994), and C4 (Raffel et al., 2020). The main tables report the arithmetic mean across the three corpora. For the evaluated Llama models, Appendix E provides the individual corpus values.

Zero-shot accuracy. We evaluate WinoGrande (Sakaguchi et al., 2020), SocialIQA (Sap et al., 2019), PIQA (Bisk et al., 2020), OpenBookQA (Mihaylov et al., 2018), LAMBADA (Paperno et al., 2016), HellaSwag (Zellers et al., 2019), ARC-Easy and ARC-Challenge (Clark et al., 2018), and MMLU (Hendrycks et al., 2021) without demonstrations. The arithmetic mean across these nine tasks is denoted 0-shot$_9$. All methods use identical task configurations and scoring procedures. Appendix E reports the task-level results for the evaluated Llama models.

B.4 Online R_4 Benchmark Protocol

Table 4 measures Llama-3-8B under W4A4KV4 on one NVIDIA RTX A6000 using CUDA 11.8, PyTorch 2.6.0+cu118, Transformers 4.44.2, and eager Hugging Face execution. The checkpoint, quantized linear kernels, input, and decoding backend are fixed across all measurements. The fixed FHT is the primary runtime baseline for measuring the overhead of the learned reflections. The dense row is a structural cost reference that applies a dense $d_f \times d_f$ online transform with the same activation shape and numerical precision; it is not a separately optimized quantization configuration, so no quantization-quality result is assigned to it. Prefill uses batch size 1 and 2048 input tokens with KV-cache construction enabled. Decoding performs 128 cached one-token forward passes with greedy token selection and reports their mean latency per token. Timings use synchronized CUDA events. We discard three warm-up runs and report the median of ten measured runs.

Table 6: **Complete comparison of rotation-learning throughput, peak GPU memory, and normalized value-projection output error.** Best and second-best results for each model and metric are shown in bold and underlined, respectively.

Cost	Method	Llama-2-7B	Llama-2-13B	Llama-3-8B
Throughput ↑ (epochs/GPU-hour)	SpinQuant	8.90	3.88	8.33
	OSTQuant	40.24	3.43	38.62
	DartQuant	**137.14**	**85.31**	**128.57**
	MirrorQuant	<u>107.25</u>	<u>60.23</u>	<u>126.91</u>
Peak memory ↓ (GiB)	SpinQuant	106.08	256.56	126.68
	OSTQuant	157.12	374.32	153.97
	DartQuant	**16.37**	**20.37**	**16.39**
	MirrorQuant	<u>24.58</u>	<u>30.64</u>	<u>26.68</u>
Output error ↓	SpinQuant	0.01560	0.01348	0.01664
	OSTQuant	0.01529	0.01321	0.01645
	DartQuant	<u>0.01482</u>	<u>0.01259</u>	<u>0.01577</u>
	MirrorQuant	**0.01452**	**0.01162**	**0.01534**

Measurements begin after model construction and deployment preparation, and exclude model loading, rotation construction, quantizer fitting, weight packing, and graph compilation. CUDA peak-memory statistics are reset before each run. Peak memory is the maximum allocated device memory observed during prefill or decoding, followed by the median across measured runs; it excludes host memory, reserved CUDA memory, and allocations made before the timed benchmark.

B.5 Complete Rotation-Learning Efficiency Results

Table 6 extends the rotation-learning comparison in Table 3 to all three models and reports normalized output error. Throughput denotes completed optimization epochs per GPU-hour under each method's prescribed schedule. Because the methods optimize different rotation sets, this metric characterizes per-epoch optimization efficiency rather than identical end-to-end workloads. Memory denotes peak GPU allocation during rotation learning. Output error averages the normalized value-projection error $\mathcal{L}_{\mathrm{proj}}(X; R, W_v)$ over the shared residual-stream calibration references and associated value projections. Detailed optimization schedules are provided in Appendix B.2.

Although DartQuant attains higher throughput and lower peak memory under its shorter two-rotation schedule, MirrorQuant additionally learns the online R_4 and achieves 60.23–126.91 epochs per GPU-hour while keeping peak memory below 31 GiB. Compared with SpinQuant and OSTQuant, it achieves at least $2.66\times$ the per-epoch throughput and reduces peak GPU memory by at least 76.8% across the three models. It also achieves the lowest normalized output error on all three models.

C Proofs and Computational Complexity

C.1 Paired-Reflection Initialization and Geometry

This section characterizes the paired-reflection initialization used in Section 3. It explains why consecutive nearly identical Householder reflections initialize the learned correction close to the identity while preserving exact orthogonality.

Paired near-base initialization. For unit $\mathbf{u}_j$, choose $\mathbf{z}_j \perp \mathbf{u}_j$ and define $\mathbf{v}_j(\epsilon) = (\mathbf{u}_j + \epsilon\mathbf{z}_j)/\|\mathbf{u}_j + \epsilon\mathbf{z}_j\|_2$ and $R(\epsilon) = B\prod_{j=1}^{M} P(\mathbf{u}_j)P(\mathbf{v}_j(\epsilon))$. Let $Q_j(\epsilon) = P(\mathbf{u}_j)P(\mathbf{v}_j(\epsilon))$. Every $Q_j(\epsilon)$ and B are orthogonal, which immediately gives $R(\epsilon)^\top R(\epsilon) = I$.

Let θ_j be the angle between $\mathbf{u}_j$ and $\mathbf{v}_j(\epsilon)$. Orthogonality of $\mathbf{u}_j$ and $\mathbf{z}_j$ gives

$$\sin\theta_j = \frac{|\epsilon|\|\mathbf{z}_j\|_2}{\sqrt{1+\epsilon^2\|\mathbf{z}_j\|_2^2}}. \tag{23}$$

The product of two reflections is a rotation by $2\theta_j$ on $\mathrm{span}\{\mathbf{u}_j, \mathbf{z}_j\}$ and the identity on its orthogonal complement. Therefore,

$$\|Q_j(\epsilon) - I\|_2 = 2\sin\theta_j = 2\frac{|\epsilon|\|\mathbf{z}_j\|_2}{\sqrt{1+\epsilon^2\|\mathbf{z}_j\|_2^2}}. \tag{24}$$

Telescoping the product $\prod_j Q_j(\epsilon)$ and using the unit spectral norm of every orthogonal prefix yields

$$\|R(\epsilon) - B\|_2 \le \sum_{j=1}^{M} \|Q_j(\epsilon) - I\|_2. \tag{25}$$

Substituting the single-pair expression above and using $\sqrt{1+a^2} \ge 1$ gives

$$\|R(\epsilon) - B\|_2 \le 2|\epsilon| \sum_{j=1}^{M} \|\mathbf{z}_j\|_2. \tag{26}$$

It remains to compute the first-order direction. Differentiating the normalized perturbation at zero gives $\mathbf{v}_j'(0) = \mathbf{z}_j$ and hence

$$\frac{\mathrm{d}}{\mathrm{d}\epsilon} P(\mathbf{v}_j(\epsilon))\Big|_{\epsilon=0} = -2(\mathbf{z}_j \mathbf{u}_j^\top + \mathbf{u}_j \mathbf{z}_j^\top). \tag{27}$$

Using $P(\mathbf{u}_j)\mathbf{u}_j = -\mathbf{u}_j$ and $P(\mathbf{u}_j)\mathbf{z}_j = \mathbf{z}_j$, we obtain $Q_j'(0) = 2(\mathbf{u}_j \mathbf{z}_j^\top - \mathbf{z}_j \mathbf{u}_j^\top)$. Since $Q_j(0) = I$ for every j, the product rule gives

$$R'(0) = 2B \sum_{j=1}^{M} (\mathbf{u}_j \mathbf{z}_j^\top - \mathbf{z}_j \mathbf{u}_j^\top). \tag{28}$$

Finite-angle interpretation. For unit normals $\mathbf{u}$ and $\mathbf{v}$, the product $P(\mathbf{u})P(\mathbf{v})$ fixes their common orthogonal complement and acts on their span as a rotation through twice the angle between the normal lines (up to orientation). Consequently, every planar rotation can be represented by a reflection pair, while identical normals yield the identity. This is the finite-angle counterpart of the local statement above.

C.2 Projection-Metric Analysis

Proof of Proposition 2.1. Let $A = WW^\top$. If $A = cI_d$, then for every E,

$$\|EW\|_F^2 = \operatorname{tr}(E^\top E A) = c \operatorname{tr}(E^\top E) = c\|E\|_F^2. \tag{29}$$

Conversely, suppose Eq. (6) holds for every E. For any $\mathbf{z} \in \mathbb{R}^d$, choose an E whose first row is $\mathbf{z}^\top$ and whose remaining rows are zero. Then $\mathbf{z}^\top A \mathbf{z} = c\mathbf{z}^\top \mathbf{z}$ for every $\mathbf{z}$, which implies $A = cI_d$. If A is not a scaled identity, it has two unit eigenvectors $\mathbf{u}_i$ and $\mathbf{u}_j$ with distinct eigenvalues λ_i and λ_j. Constructing E_a and E_b from these vectors gives $\|E_a\|_F = \|E_b\|_F = 1$ but $\|E_a W\|_F^2 = \lambda_i \neq \lambda_j = \|E_b W\|_F^2$, proving the claim.

Truncated down-projection metric. Let $X_{\mathrm{ff},\ell}$ denote a cached gated-MLP activation generated under the fixed Stage-I rotation-and-activation-quantization configuration. After applying $R_{4,\ell}$, quantizing the rotated activation, and mapping it back to the original basis, its reconstruction is

$$\widehat{X}_{\mathrm{ff},\ell} = \mathcal{Q}\left(X_{\mathrm{ff},\ell} R_{4,\ell}\right) R_{4,\ell}^\top. \tag{30}$$

Define the corresponding activation-space error as

$$E_\ell = \widehat{X}_{\mathrm{ff},\ell} - X_{\mathrm{ff},\ell}. \tag{31}$$

Following Stage I, let

$$\widetilde{W}_{\mathrm{down},\ell} = W_{\mathrm{down},\ell} R_1 \tag{32}$$

denote the full-precision effective down-projection weight. At deployment, $R_{4,\ell}^\top$ is further folded into this weight, giving $\widehat{W}_{\mathrm{down},\ell} = R_{4,\ell}^\top \widetilde{W}_{\mathrm{down},\ell}$. The resulting error at the down-projection output is therefore

$$E_\ell \widetilde{W}_{\mathrm{down},\ell}, \tag{33}$$

with squared magnitude $\|E_\ell \widetilde{W}_{\mathrm{down},\ell}\|_F^2$.

Let $\widetilde{W}_{\mathrm{down},\ell} \in \mathbb{R}^{d_f \times d}$ and $q = \min(d_f, d)$. Its thin singular value decomposition is

$$\widetilde{W}_{\mathrm{down},\ell} = U_\ell \Sigma_\ell V_\ell^\top, \tag{34}$$

where the singular values satisfy $\sigma_{\ell,1} \geq \sigma_{\ell,2} \geq \cdots$. Since $q > 256$ for all evaluated layers, we fix the truncated rank to $r = 256$. We retain the corresponding leading singular components and define

$$G_\ell = U_{\ell,1:r} \Sigma_{\ell,1:r}. \tag{35}$$

By the unitary invariance of the Frobenius norm,

$$\begin{aligned} \left\| E_\ell \widetilde{W}_{\mathrm{down},\ell} \right\|_F^2 &= \| E_\ell U_\ell \Sigma_\ell \|_F^2 \\ &= \sum_{i=1}^{q} \sigma_{\ell,i}^2 \, \| E_\ell \mathbf{u}_{\ell,i} \|_2^2 , \end{aligned} \tag{36}$$

where $\mathbf{u}_{\ell,i}$ is the i-th left singular vector of $\widetilde{W}_{\mathrm{down},\ell}$. The truncated metric retains the leading terms:

$$\| E_\ell G_\ell \|_F^2 = \sum_{i=1}^{r} \sigma_{\ell,i}^2 \, \| E_\ell \mathbf{u}_{\ell,i} \|_2^2 . \tag{37}$$

Subtracting Eq. (37) from Eq. (36) gives

$$\begin{aligned} 0 &\leq \left\| E_\ell \widetilde{W}_{\mathrm{down},\ell} \right\|_F^2 - \| E_\ell G_\ell \|_F^2 \\ &= \sum_{i=r+1}^{q} \sigma_{\ell,i}^2 \, \| E_\ell \mathbf{u}_{\ell,i} \|_2^2 \\ &\leq \sigma_{\ell,r+1}^2 \sum_{i=r+1}^{q} \| E_\ell \mathbf{u}_{\ell,i} \|_2^2 \\ &\leq \sigma_{\ell,r+1}^2 \, \| E_\ell \|_F^2 . \end{aligned} \tag{38}$$

Thus, $\| E_\ell G_\ell \|_F^2$ is exactly the contribution of the retained leading singular components to the squared effective down-projection error. Equation (38) bounds the omitted contribution using the first discarded singular value and the total activation-space error.

At the fixed rank $r = 256$, the retained metric is optimal in the following worst-case spectral-norm sense. Let

$$A_\ell = \widetilde{W}_{\mathrm{down},\ell} \widetilde{W}_{\mathrm{down},\ell}^\top, \qquad A_{\ell,r} = G_\ell G_\ell^\top = U_{\ell,1:r} \Sigma_{\ell,1:r}^2 U_{\ell,1:r}^\top.$$

The eigenvalues of A_ℓ are $\sigma_{\ell,i}^2$. Therefore, the Eckart–Young–Mirsky theorem gives

$$A_{\ell,r} \in \arg \min_{M:\, \mathrm{rank}(M) \leq r} \| A_\ell - M \|_2 , \qquad \| A_\ell - A_{\ell,r} \|_2 = \sigma_{\ell,r+1}^2. \tag{39}$$

Because $A_{\ell,r}$ is positive semidefinite and has rank at most r, it is feasible for the factorized class $M = GG^\top$. Moreover, for any symmetric matrix D,

$$\sup_{E \neq 0} \frac{\left| \mathrm{tr}(E^\top E D) \right|}{\| E \|_F^2} = \| D \|_2, \tag{40}$$

where equality is attained by placing an eigenvector corresponding to an eigenvalue of largest magnitude of D in one row of E. Consequently, for any factor G of rank at most r,

$$\sup_{E \neq 0} \frac{\left| \| E \widetilde{W}_{\mathrm{down},\ell} \|_F^2 - \| EG \|_F^2 \right|}{\| E \|_F^2} = \left\| A_\ell - GG^\top \right\|_2 . \tag{41}$$

Equation (39) therefore shows that G_ℓ is an optimal rank-r factor for minimizing the worst-case discrepancy between the exact and truncated projection metrics.

The R_4 objective combines the retained low-rank metric with an activation-space term:

$$\mathcal{L}_{4,\ell} = \| E_\ell G_\ell \|_F^2 + \lambda_{\mathrm{act}} \, \| E_\ell \|_F^2 , \qquad \lambda_{\mathrm{act}} > 0. \tag{42}$$

The first term weights errors along the leading left-singular directions of $\widetilde{W}_{\mathrm{down},\ell}$ by their singular values, while the second measures activation error in all directions, including those omitted by the truncated metric. This matches the R_4 objective in Eq. (18).

C.3 Parameter and Arithmetic Costs

For a token matrix $X \in \mathbb{R}^{n\times d}$, one Householder factor is applied as

$$XP(\mathbf{v}) = X - 2(X\mathbf{u})\mathbf{u}^\top. \tag{43}$$

Computing $X\mathbf{u}$ and the subsequent rank-one update requires approximately $2nd$ multiply-adds. A depth-K chain therefore stores Kd scalar entries and requires approximately

$$2nKd \tag{44}$$

multiply-adds, without materializing a dense $d \times d$ matrix. In comparison, applying a dense learned rotation requires $\mathcal{O}(nd^2)$ arithmetic and $\mathcal{O}(d^2)$ stored entries.

The learned R_1 and R_2 rotations are absorbed into adjacent linear weights and therefore introduce no additional online transformation. The activation-side R_4 remains online and is implemented as

$$R_4 = H \prod_{i=1}^{K_4} P(\mathbf{v}_{4,i}), \tag{45}$$

where H is the fixed normalized Hadamard operator used by the matched implementation, satisfying $H^\top H = I$, and is applied by its FHT routine without materializing a dense matrix. For n tokens and gated-MLP width d_f, the FHT requires $\mathcal{O}(nd_f \log d_f)$ arithmetic, while the Householder chain requires approximately $2nK_4d_f$ multiply-adds. Under the fixed-depth setting used throughout this work, the total asymptotic complexity is therefore

$$C_{R_4}(n) := \mathcal{O}(nd_f \log d_f). \tag{46}$$

Relative to the fixed-Hadamard baseline, the learned component adds approximately

$$2nK_4d_f \tag{47}$$

multiply-adds and stores K_4d_f scalar entries. In contrast, a dense learned online rotation requires $\mathcal{O}(nd_f^2)$ arithmetic and $\mathcal{O}(d_f^2)$ storage.

Under the operation-count convention used in Figure 1, we count the FHT as $nd_f \log_2 d_f$ operations and the Householder chain as $2nK_4d_f$ operations. The resulting dense-to-structured ratio is

$$\frac{nd_f^2}{nd_f \log_2 d_f + 2nK_4d_f} = \frac{d_f}{\log_2 d_f + 2K_4}. \tag{48}$$

For Llama-3-8B, $d_f = 14{,}336$ and $K_4 = 4$, yielding a ratio of approximately 657, which we report as $650\times$ in Figure 1.

These expressions characterize arithmetic and storage. Table 4 reports measured latency and peak memory, which also reflect kernel fusion, memory traffic, tensor shape, and hardware utilization.

D Additional Experimental Analyses

This appendix expands the component studies in Section 4.3. It reports task-level component results, controlled ablations of the R_4 objective and calibration procedure, the effect of the R_2 reflection depth, and the R_1 base comparison for Llama-3-70B. Appendix E provides detailed task- and corpus-level results for the evaluated Llama models.

D.1 Component Ablation and Task-Level Results

Table 7 reports the aggregate component ablation referenced in Section 4.3. The rows follow a cumulative construction from the matched DartQuant baseline: a checkmark indicates that the corresponding DartQuant rotation is replaced by the proposed learned rotation, while an unmarked site retains the DartQuant configuration rather than removing the rotation.

Table 8 expands this ablation across its nine zero-shot tasks and reports average PPL over WikiText-2, PTB, and C4. MirrorQuant-R_1 replaces only DartQuant's residual rotation, MirrorQuant-$R_{1,2}$ replaces both foldable rotations, and the complete MirrorQuant additionally enables the learned online MLP rotation R_4. WG, SIQA, PIQA, OBQA, LAMB., HS, and ARC-E/C abbreviate WinoGrande, SocialIQA, PhysicalIQA, OpenBookQA, LAMBADA, HellaSwag, and ARC-Easy/Challenge, respectively. The complete configuration uses $K_1 = K_2 = K_4 = 4$.

Table 7: **Component ablation on Llama-3-8B under W4A4KV4.** PPL is averaged over WikiText-2, PTB, and C4. Checkmarks denote proposed rotations introduced cumulatively; unmarked sites retain the corresponding DartQuant configuration. Green subscripts report improvements over DartQuant: ↓ for PPL and ↑ for 0-shot$_9$.

Configuration	MirrorQuant component			PPL ↓	0-shot$_9$ ↑
	Residual R_1	Attention R_2	Online MLP R_4		
DartQuant	–	–	–	10.88	61.60
MirrorQuant-R_1	✓	–	–	$10.87_{\downarrow 0.01}$	$61.92_{\uparrow 0.32}$
MirrorQuant-$R_{1,2}$	✓	✓	–	$\underline{10.83}_{\downarrow 0.05}$	$\underline{62.34}_{\uparrow 0.74}$
MirrorQuant	✓	✓	✓	$\mathbf{10.79}_{\downarrow 0.09}$	$\mathbf{62.60}_{\uparrow 1.00}$

Table 8: **Task-level component analysis on Llama-3-8B under W4A4KV4, corresponding to Table 7.** PPL is averaged over WikiText-2, PTB, and C4. Best and second-best results are bolded and underlined, respectively.

Configuration	WG	SIQA	PIQA	OBQA	LAMB.	HS	ARC-E	ARC-C	MMLU	PPL	Avg.
DartQuant	69.53	**45.50**	76.82	<u>43.80</u>	71.22	75.23	71.42	46.08	**54.76**	10.88	61.60
MirrorQuant-R_1	69.69	44.88	<u>78.35</u>	41.80	**71.92**	**75.88**	73.65	48.04	53.05	10.87	61.92
MirrorQuant-$R_{1,2}$	<u>70.24</u>	<u>45.45</u>	**78.40**	**44.40**	<u>71.86</u>	75.51	<u>75.84</u>	<u>48.21</u>	51.15	<u>10.83</u>	<u>62.34</u>
MirrorQuant	**71.43**	45.39	77.09	43.40	71.24	<u>75.52</u>	**76.47**	**49.32**	<u>53.56</u>	**10.79**	**62.60**

Table 9: **Ablation of the R_4 objective, calibration cache, and projection metric on Llama-3-8B under W4A4KV4.** PPL is reported on WikiText-2, PTB, and C4, with the mean taken over the three corpora. Zero-shot accuracy is averaged over the nine downstream tasks. Row D is the default MirrorQuant configuration. All results use seed 0.

ID	Objective	R_4 cache	Projection metric	WikiText-2	PTB	C4	Mean PPL ↓	0-shot$_9$ ↑
A	Reconstruction	Regenerated	–	7.40	12.96	12.01	10.79	62.22
B	Reconstruction + projection	Stale	Truncated G_ℓ	7.44	13.13	12.06	10.88	62.43
C	Reconstruction + projection	Regenerated	Full $\widetilde{W}_{\mathrm{down},\ell}$	7.39	12.96	12.02	10.79	62.65
D	Reconstruction + projection	Regenerated	Truncated G_ℓ	7.40	12.98	11.99	10.79	62.60

ID	WG	SIQA	PIQA	OBQA	LAMB.	HS	ARC-E	ARC-C	MMLU	Avg. ↑
A	69.80	46.37	77.53	43.00	71.55	75.53	73.78	48.21	54.21	62.22
B	70.26	45.19	77.48	42.00	71.26	74.89	77.15	49.23	54.45	62.43
C	69.61	44.42	78.73	44.40	71.78	75.54	75.08	50.26	54.03	62.65
D	71.43	45.39	77.09	43.40	71.24	75.52	76.47	49.32	53.56	62.60

R_4 objective and calibration ablation. Table 9 isolates three design choices in learning the online R_4: the projection-guided objective, regeneration of the calibration cache after fixing the upstream rotations, and the truncated projection metric. All configurations use the same Llama-3-8B checkpoint, W4A4KV4 quantization setting, calibration data, and optimization schedule.

Relative to reconstruction alone, the complete configuration increases 0-shot$_9$ accuracy from 62.22 to 62.60 while retaining the same mean PPL of 10.79. Using a stale cache increases mean PPL to 10.88 and reduces accuracy to 62.43, supporting regeneration of the calibration representations after the upstream configuration is fixed. Finally, the full and truncated projection metrics obtain $10.79/62.65$ and $10.79/62.60$ in mean PPL/accuracy, respectively. Thus, in this controlled seed-0 experiment, the truncated metric preserves mean PPL to the reported precision with only a 0.05-point difference in average accuracy relative to the full projection.

D.2 R_2 Reflection-Depth Ablation

We vary the even Householder depth $K_2 \in \{2, 4, 6, 8\}$ for R_2 on Llama-3-8B under W4A4KV4 using the MirrorQuant-$R_{1,2}$ configuration, keeping R_4 as the fixed FHT and all other rotation settings unchanged. Consecutive reflection pairs use the same near-base initialization rule and seed, so all configurations start near the same selected base. Table 10 shows that the nine-task average increases from 62.14 at $K_2 = 2$ to its highest value of 62.34 at $K_2 = 4$. Increasing the depth to 6 or 8 provides no further gain in this setting, supporting $K_2 = 4$ as the default budget.

Table 10: **Effect of Householder depth K_2 on Llama-3-8B under W4A4KV4.** Consecutive reflection pairs use the same near-base initialization rule and seed, with all other rotation settings fixed. Best and second-best results are bolded and underlined, respectively; ties share formatting.

K_2	WG	SIQA	PIQA	OBQA	LAMB.	HS	ARC-E	ARC-C	MMLU	Avg.
2	**71.43**	**46.62**	77.04	42.60	70.64	**75.65**	73.74	47.70	<u>53.88</u>	<u>62.14</u>
4	<u>70.24</u>	45.45	**78.40**	**44.40**	**71.86**	75.51	<u>75.84</u>	<u>48.21</u>	51.15	**62.34**
6	69.46	45.14	<u>77.58</u>	43.20	<u>71.53</u>	<u>75.61</u>	**75.97**	46.25	53.62	62.04
8	69.78	<u>45.96</u>	76.22	<u>43.80</u>	70.99	75.16	72.77	**48.63**	**54.32**	61.96

D.3 R_1 Base Sensitivity and Progressive Construction

For Llama-3-70B, we pre-specify the R_1 from the matched DartQuant configuration as the fixed base independently of downstream task results. This largest-scale protocol is designed to evaluate whether the proposed Householder corrections provide additional gains over a strong existing rotation without relearning its base. Table 11 subsequently provides a post-hoc sensitivity analysis of this protocol choice; it is not used to select the reported configuration. All rows share the same optimization samples, quantizer, and evaluation pipeline. Accordingly, the comparison measures the learned Householder correction conditional on the fixed base and does not attribute the base-construction cost to MirrorQuant.

Table 11: **Controlled analysis of the R_1 base and the progressive construction of MirrorQuant on Llama-3-70B under W4A4KV4.** Panel (a) reports a post-hoc sensitivity comparison between two fixed bases using the complete method. Panel (b) fixes the DartQuant base and progressively introduces the proposed rotation-learning stages. All values are nine-task zero-shot accuracies (%).

R_1 base	Rotation configuration	0-shot$_9$ ↑
(a) Post-hoc sensitivity to the R_1 base		
Hadamard	MirrorQuant	67.59
DartQuant	MirrorQuant	**69.21**
(b) Progressive construction with the DartQuant base		
DartQuant	DartQuant baseline	68.35
DartQuant	Fixed DartQuant R_1 + projection-guided R_2	68.75
DartQuant	Learned R_1 and R_2	<u>68.95</u>
DartQuant	**MirrorQuant with learned R_4**	**69.21**

As a post-hoc sensitivity check, Panel (a) shows that the complete method obtains 67.59 with the Hadamard base and 69.21 with the pre-specified DartQuant-derived base under W4A4KV4; this comparison is not used for model selection. With the latter base fixed, Panel (b) isolates the subsequent construction: learning the projection-guided R_2, jointly learning R_1 and R_2, and adding the learned online R_4 progressively raise 0-shot$_9$ from the DartQuant baseline of 68.35 to 68.75, 68.95, and 69.21, respectively. These results characterize the conditional benefit of the proposed corrections and should not be interpreted as including the cost of constructing the DartQuant-derived base.

E Detailed Llama Task- and Corpus-Level Results

Table 1 reports aggregate perplexity and zero-shot accuracy for the evaluated Llama models, while Table 5 provides the task-level Llama-3-70B results in the main text. This appendix reports task- and corpus-level breakdowns for the remaining Llama models together with corpus-level results for Llama-3-70B. All entries follow the common protocol in Appendix B.3.

Table 12: Complete zero-shot results on Llama-2-7B. All values are accuracies (%); higher is better. Best and second-best quantized results in each setting are shown in bold and underlined, respectively.

Bits (W-A-KV)	Method	WG	SIQA	PIQA	OBQA	LAMB.	HS	ARC-E	ARC-C	MMLU	Avg.
16-16-16	Full precision	69.06	46.16	79.05	44.20	73.90	76.02	74.54	46.33	41.21	61.16
4-4-4	RTN	48.54	34.54	51.63	24.20	0.80	27.95	28.96	24.66	23.96	29.47
	SmoothQuant	46.17	34.60	50.71	25.20	1.53	28.17	29.63	24.83	23.71	29.39
	GPTQ	50.67	33.67	53.59	24.60	1.53	28.26	29.00	25.00	23.24	29.95
	QuaRot	64.96	44.68	75.90	39.00	67.81	71.67	69.70	40.78	32.29	56.31
	SpinQuant	63.85	42.43	77.09	39.20	68.48	71.93	70.08	41.98	33.49	56.50
	OSTQuant	66.22	42.68	76.88	40.20	69.36	72.76	**72.56**	40.96	33.62	57.25
	DartQuant	65.43	**45.70**	76.88	41.40	70.79	**73.54**	70.96	43.34	34.35	58.04
	MirrorQuant	**67.17**	44.27	**77.69**	**42.40**	**71.61**	73.48	70.66	**43.69**	**36.21**	**58.58**
4-4-16	RTN	48.93	33.88	54.30	23.60	1.14	28.11	28.03	24.66	24.04	29.63
	SmoothQuant	49.17	34.75	54.52	23.80	2.02	28.43	29.34	23.04	23.36	29.83
	GPTQ	49.96	34.34	53.05	24.20	1.84	28.06	29.04	24.57	23.12	29.80
	QuaRot	63.77	44.11	77.20	40.00	69.78	72.44	69.87	42.49	33.25	56.99
	SpinQuant	63.61	41.15	73.78	36.40	67.32	71.24	66.12	40.10	32.32	54.67
	OSTQuant	65.43	44.32	77.09	40.20	71.36	**73.65**	**73.99**	41.81	33.89	57.97
	DartQuant	**67.40**	**45.14**	**77.69**	40.20	71.80	73.54	70.92	42.75	34.01	58.16
	MirrorQuant	67.09	45.09	77.53	**43.40**	**72.19**	73.62	71.21	**44.20**	**35.92**	**58.92**
4-8-16	RTN	65.98	43.76	77.75	43.60	71.57	73.59	69.40	**45.73**	34.49	58.43
	SmoothQuant	67.88	43.76	77.91	43.00	71.28	74.31	71.04	44.45	37.38	59.00
	GPTQ	66.30	43.65	77.75	41.40	71.28	74.65	71.59	45.39	37.34	58.82
	QuaRot	68.67	45.65	78.13	42.40	73.51	74.67	72.81	43.60	38.06	59.72
	SpinQuant	68.75	45.24	78.62	44.20	73.67	**75.06**	73.91	45.14	**41.04**	**60.63**
	OSTQuant	68.82	45.09	78.56	43.40	74.09	74.95	**76.09**	43.60	38.86	60.38
	DartQuant	**69.30**	45.70	78.51	42.00	73.70	74.94	73.86	43.86	36.75	59.85
	MirrorQuant	68.43	**46.11**	**78.84**	**44.60**	**74.15**	74.98	73.86	44.88	39.40	60.58

Table 13: Complete perplexity results on Llama-2-7B. Mean is the arithmetic mean over WikiText-2, PTB, and C4; lower is better. Best and second-best quantized results in each setting are shown in bold and underlined, respectively.

Bits (W-A-KV)	Method	WikiText-2	PTB	C4	Mean
16-16-16	Full precision	5.47	37.91	7.26	16.88
4-4-4	RTN	509.11	8121.84	580.80	3070.58
	SmoothQuant	696.09	28583.66	654.57	9978.11
	GPTQ	695.32	33506.11	644.18	11615.20
	QuaRot	6.33	75.18	8.68	30.06
	SpinQuant	6.45	55.49	8.86	23.60
	OSTQuant	5.99	47.16	8.18	20.44
	DartQuant	**5.98**	45.35	**8.13**	19.82
	MirrorQuant	5.99	**44.16**	**8.13**	**19.43**
4-4-16	RTN	433.17	7256.03	517.23	2735.48
	SmoothQuant	591.90	33112.47	638.39	11447.59
	GPTQ	554.49	32761.27	592.85	11302.87
	QuaRot	6.14	47.75	8.40	20.76
	SpinQuant	6.45	48.61	8.82	21.29
	OSTQuant	5.95	45.51	8.11	19.86
	DartQuant	5.94	43.51	**8.04**	19.16
	MirrorQuant	**5.93**	**42.47**	**8.04**	**18.81**
4-8-16	RTN	6.45	46.17	8.58	20.40
	SmoothQuant	7.68	9515.18	8.41	3177.09
	GPTQ	8.75	11823.55	8.48	3946.93
	QuaRot	**5.60**	40.76	7.56	17.97
	SpinQuant	**5.60**	39.07	**7.51**	17.39
	OSTQuant	5.61	40.41	7.54	17.85
	DartQuant	5.61	40.38	7.54	17.84
	MirrorQuant	5.61	**38.73**	7.54	**17.29**

Table 14: Complete zero-shot results on Llama-2-13B. All values are accuracies (%); higher is better. Best and second-best quantized results in each setting are shown in bold and underlined, respectively.

Bits (W-A-KV)	Method	WG	SIQA	PIQA	OBQA	LAMB.	HS	ARC-E	ARC-C	MMLU	Avg.
16-16-16	Full precision	72.14	47.39	80.52	45.20	76.73	79.38	77.44	49.15	50.53	64.28
4-4-4	RTN	48.38	34.24	51.03	24.40	0.10	27.03	25.84	25.68	23.95	28.96
	SmoothQuant	47.28	34.80	48.53	23.20	0.02	27.21	26.43	27.22	22.93	28.62
	GPTQ	52.41	34.34	50.33	24.80	0.14	27.54	27.57	24.83	23.30	29.47
	QuaRot	67.48	44.17	77.86	43.80	71.16	75.20	72.77	44.54	43.09	60.01
	SpinQuant	69.14	44.32	**78.94**	42.60	73.63	75.70	74.28	46.08	44.45	61.02
	OSTQuant	69.22	44.78	78.45	**44.60**	**75.45**	76.80	76.27	45.90	45.76	61.91
	DartQuant	70.88	45.34	78.78	43.60	74.66	76.81	74.54	46.08	45.73	61.82
	MirrorQuant	**70.96**	**46.52**	78.89	**44.60**	75.18	**76.95**	**76.85**	**48.21**	**46.99**	**62.79**
4-4-16	RTN	50.43	35.16	50.54	26.80	0.14	27.15	26.94	25.60	23.49	29.58
	SmoothQuant	49.33	34.90	48.86	25.60	0.17	27.26	26.22	27.90	23.43	29.30
	GPTQ	49.72	34.80	49.24	27.20	0.08	27.49	27.65	25.68	24.04	29.54
	QuaRot	69.22	44.17	78.94	43.20	73.06	76.39	73.15	46.50	45.48	61.12
	SpinQuant	69.77	43.60	77.42	42.00	72.31	75.05	71.68	44.71	44.54	60.12
	OSTQuant	68.43	45.55	79.16	44.40	**76.29**	76.70	76.33	45.82	45.04	61.97
	DartQuant	69.69	46.01	**79.49**	43.40	74.58	77.21	74.71	47.95	45.06	62.01
	MirrorQuant	**69.85**	**46.06**	78.89	**45.20**	75.33	**77.45**	**76.81**	**48.38**	**47.74**	**62.86**
4-8-16	RTN	70.64	44.42	79.33	45.00	73.70	76.85	73.27	47.95	46.85	62.00
	SmoothQuant	71.59	45.75	79.11	45.20	76.25	76.39	73.23	47.35	47.89	62.53
	GPTQ	71.51	45.39	78.78	45.00	75.80	76.78	73.57	48.29	47.36	62.50
	QuaRot	71.59	46.52	**79.98**	44.00	76.32	78.51	75.55	46.76	48.68	63.10
	SpinQuant	**72.06**	46.21	79.82	**45.80**	76.52	78.67	76.64	48.46	49.72	63.77
	OSTQuant	71.51	46.01	79.92	44.20	**77.61**	78.23	**77.37**	48.38	**49.83**	63.67
	DartQuant	71.90	46.88	79.76	45.00	76.25	78.30	76.98	48.63	49.39	63.68
	MirrorQuant	71.51	**46.98**	79.60	45.20	76.48	**78.89**	77.23	**48.98**	49.20	**63.79**

Table 15: Complete perplexity results on Llama-2-13B. Mean is the arithmetic mean over WikiText-2, PTB, and C4; lower is better. Best and second-best quantized results in each setting are shown in bold and underlined, respectively.

Bits (W-A-KV)	Method	WikiText-2	PTB	C4	Mean
16-16-16	Full precision	4.88	50.94	6.73	20.85
4-4-4	RTN	1137.94	3737.47	1091.70	1989.04
	SmoothQuant	1088.82	3352.04	1432.78	1957.88
	GPTQ	1292.76	3367.65	1795.44	2151.95
	QuaRot	5.61	63.85	7.85	25.77
	SpinQuant	5.53	58.87	7.74	24.05
	OSTQuant	5.28	58.02	7.37	23.56
	DartQuant	5.29	57.89	7.36	23.51
	MirrorQuant	**5.25**	**55.85**	**7.32**	**22.81**
4-4-16	RTN	1311.05	3605.48	1503.78	2140.10
	SmoothQuant	1584.82	3563.99	2119.79	2422.87
	GPTQ	1942.09	3678.60	2577.51	2732.73
	QuaRot	5.43	61.45	7.58	24.82
	SpinQuant	5.57	59.20	7.87	24.21
	OSTQuant	5.24	57.43	7.35	23.34
	DartQuant	5.25	57.50	7.31	23.35
	MirrorQuant	**5.21**	**55.23**	**7.27**	**22.57**
4-8-16	RTN	5.45	61.37	7.49	24.77
	SmoothQuant	5.23	59.35	7.33	23.97
	GPTQ	5.24	56.84	7.36	23.15
	QuaRot	5.01	53.66	6.94	21.87
	SpinQuant	**4.99**	53.48	6.92	21.80
	OSTQuant	**4.99**	53.85	6.93	21.92
	DartQuant	**4.99**	53.62	6.93	21.85
	MirrorQuant	**4.99**	**51.84**	**6.90**	**21.24**

Table 16: Complete zero-shot results on Llama-2-70B. All values are accuracies (%); higher is better. Best and second-best quantized results in each setting are shown in bold and underlined, respectively.

Bits (W-A-KV)	Method	WG	SIQA	PIQA	OBQA	LAMB.	HS	ARC-E	ARC-C	MMLU	Avg.
16-16-16	Full precision	77.98	49.13	82.75	48.80	79.58	83.80	81.02	57.51	65.20	69.53
4-4-4	RTN	50.83	33.73	49.67	25.00	0.88	25.96	24.79	28.24	24.12	29.25
	SmoothQuant	51.30	34.14	50.00	27.20	0.29	26.14	26.64	26.96	24.11	29.64
	GPTQ	50.28	33.42	50.92	23.00	0.23	26.00	25.88	25.94	24.41	28.90
	QuaRot	75.61	47.65	80.96	47.40	78.85	81.63	77.95	54.27	60.30	67.18
	SpinQuant	75.53	47.03	81.28	47.80	79.14	81.78	78.79	53.92	61.68	67.44
	OSTQuant	76.58	**48.21**	82.32	46.60	79.58	81.70	80.51	56.23	61.28	68.11
	DartQuant	**77.43**	48.06	82.64	46.60	79.33	82.23	79.84	56.48	**62.33**	68.33
	MirrorQuant	75.93	47.95	**82.97**	**48.80**	**80.09**	**82.51**	**80.56**	**57.34**	62.26	**68.71**
4-4-16	RTN	48.46	33.52	51.03	24.40	0.12	26.32	25.67	28.24	23.76	29.06
	SmoothQuant	51.22	34.19	50.76	24.20	0.27	26.55	26.68	26.62	23.76	29.36
	GPTQ	49.09	33.78	49.84	25.20	0.27	25.73	26.09	26.79	24.25	29.00
	QuaRot	75.14	47.34	81.72	47.40	79.20	81.92	78.66	53.07	60.79	67.25
	SpinQuant	**76.72**	47.08	82.05	47.40	79.27	81.88	78.83	54.61	61.59	67.71
	OSTQuant	76.24	47.95	82.32	47.20	79.20	82.52	80.09	56.31	62.78	68.29
	DartQuant	75.85	48.00	82.26	46.40	79.33	82.63	80.18	56.48	62.53	68.18
	MirrorQuant	75.61	**48.36**	**82.70**	**48.00**	**79.82**	**82.89**	**81.10**	**56.91**	**62.95**	**68.70**
4-8-16	RTN	77.03	46.62	80.96	48.00	78.69	81.53	81.27	58.28	60.75	68.13
	SmoothQuant	77.51	46.98	81.34	47.00	78.94	80.07	80.05	56.66	60.66	67.69
	GPTQ	76.87	46.32	81.01	46.60	79.51	80.20	79.67	56.14	60.39	67.41
	QuaRot	77.51	48.46	82.32	49.20	79.90	83.44	80.98	56.57	64.43	69.20
	SpinQuant	77.58	48.77	83.08	48.60	80.05	**83.52**	80.85	57.59	**64.46**	69.39
	OSTQuant	77.66	48.72	83.14	49.00	80.01	83.30	81.12	58.03	64.41	69.49
	DartQuant	**77.82**	48.62	**83.35**	**49.40**	**80.19**	83.16	80.81	58.11	64.39	69.54
	MirrorQuant	77.74	**48.98**	82.81	48.80	80.05	83.38	**81.44**	**58.62**	64.30	**69.57**

Table 17: Complete perplexity results on Llama-2-70B. Mean is the arithmetic mean over WikiText-2, PTB, and C4; lower is better. Best and second-best quantized results in each setting are shown in bold and underlined, respectively.

Bits (W-A-KV)	Method	WikiText-2	PTB	C4	Mean
16-16-16	Full precision	3.32	24.25	5.71	11.09
4-4-4	RTN	2189.73	2339.50	1690.58	2073.27
	SmoothQuant	1950.74	1930.27	1588.00	1823.00
	GPTQ	2128.66	1890.06	1758.95	1925.89
	QuaRot	3.90	25.33	6.21	11.81
	SpinQuant	3.74	25.99	6.08	11.94
	OSTQuant	3.67	25.30	6.03	11.67
	DartQuant	3.67	25.10	6.01	11.59
	MirrorQuant	**3.65**	**24.96**	**6.00**	**11.54**
4-4-16	RTN	2311.23	2093.37	1720.80	2041.80
	SmoothQuant	1527.98	1892.86	1429.77	1616.87
	GPTQ	1915.45	1895.39	1529.13	1779.99
	QuaRot	3.85	25.03	6.17	11.68
	SpinQuant	3.71	24.90	6.04	11.55
	OSTQuant	3.65	26.28	6.01	11.98
	DartQuant	3.64	25.03	5.99	11.55
	MirrorQuant	**3.63**	**24.86**	**5.98**	**11.49**
4-8-16	RTN	3.82	25.04	6.09	11.65
	SmoothQuant	3.77	25.61	6.12	11.83
	GPTQ	3.76	25.82	6.11	11.90
	QuaRot	3.42	24.38	5.80	11.20
	SpinQuant	**3.40**	24.71	**5.79**	11.30
	OSTQuant	**3.40**	24.68	**5.79**	11.29
	DartQuant	3.41	24.37	**5.79**	11.19
	MirrorQuant	3.41	**24.23**	**5.79**	**11.14**

Table 18: Complete zero-shot results on Llama-3-8B. All values are accuracies (%); higher is better. Best and second-best quantized results in each setting are shown in bold and underlined, respectively.

Bits (W-A-KV)	Method	WG	SIQA	PIQA	OBQA	LAMB.	HS	ARC-E	ARC-C	MMLU	Avg.
16-16-16	Full precision	73.32	47.19	80.96	45.00	75.63	79.14	77.82	53.24	62.06	66.04
4-4-4	RTN	53.12	32.96	53.75	27.20	2.99	28.99	29.76	24.66	23.79	30.80
	SmoothQuant	50.99	32.80	51.52	26.80	2.02	28.74	28.49	23.63	23.82	29.87
	GPTQ	50.36	34.29	53.32	25.40	2.23	28.39	27.95	22.27	23.37	29.74
	QuaRot	64.01	43.24	74.70	38.20	63.94	70.39	66.84	41.13	44.70	56.35
	SpinQuant	67.64	44.68	76.06	41.60	68.25	74.38	71.80	45.48	50.88	60.09
	OSTQuant	67.48	44.37	76.44	42.14	69.90	75.17	74.46	46.76	52.78	61.06
	DartQuant	69.53	**45.50**	76.82	**43.80**	71.22	75.23	71.42	46.08	**54.76**	61.60
	MirrorQuant	**71.43**	45.39	**77.09**	43.40	**71.24**	**75.52**	**76.47**	**49.32**	53.56	**62.60**
4-4-16	RTN	49.41	33.62	52.23	27.20	3.43	29.24	28.49	24.15	23.44	30.13
	SmoothQuant	48.78	33.78	51.96	24.40	2.33	28.27	28.70	23.89	23.42	29.50
	GPTQ	50.51	33.37	51.41	25.40	2.66	28.52	27.86	24.57	22.95	29.69
	QuaRot	66.46	43.76	76.06	38.60	66.27	72.27	70.24	41.89	48.21	58.20
	SpinQuant	68.90	44.83	77.15	**43.80**	68.95	74.50	72.39	46.50	50.43	60.83
	OSTQuant	69.06	45.34	77.53	42.80	**72.54**	75.64	75.19	**48.49**	54.78	62.37
	DartQuant	70.72	45.60	**77.64**	41.60	71.57	75.95	71.68	46.33	56.00	61.90
	MirrorQuant	**70.80**	**45.91**	77.48	42.80	71.98	**76.26**	**75.84**	47.87	**56.09**	**62.78**
4-8-16	RTN	70.88	45.91	79.00	44.20	68.15	77.35	74.71	49.32	54.35	62.65
	SmoothQuant	73.48	45.24	74.92	43.80	71.69	76.99	69.07	44.03	55.60	61.65
	GPTQ	72.93	44.83	75.14	43.80	72.91	76.77	72.43	45.99	56.36	62.35
	QuaRot	73.64	46.26	80.14	**44.80**	74.23	77.67	75.80	50.68	59.86	64.79
	SpinQuant	**74.19**	46.37	79.60	43.40	74.93	77.99	76.05	51.45	60.04	64.89
	OSTQuant	72.93	46.37	79.87	43.60	74.19	75.67	**79.17**	51.45	58.66	64.66
	DartQuant	73.95	46.06	**80.36**	44.60	75.32	77.93	75.59	51.37	**60.94**	65.12
	MirrorQuant	73.32	**46.42**	80.03	44.40	**75.59**	**78.27**	77.31	**51.54**	60.06	**65.22**

Table 19: Complete perplexity results on Llama-3-8B. Mean is the arithmetic mean over WikiText-2, PTB, and C4; lower is better. Best and second-best quantized results in each setting are shown in bold and underlined, respectively.

Bits (W-A-KV)	Method	WikiText-2	PTB	C4	Mean
16-16-16	Full precision	6.14	11.18	9.45	8.92
4-4-4	RTN	307.07	340.00	545.76	397.61
	SmoothQuant	432.06	489.43	1313.93	745.14
	GPTQ	432.83	502.92	1287.22	740.99
	QuaRot	8.91	15.20	14.61	12.91
	SpinQuant	8.01	13.78	13.12	11.64
	OSTQuant	7.54	13.34	12.24	11.04
	DartQuant	7.46	13.08	12.10	10.88
	MirrorQuant	**7.40**	**12.98**	**11.99**	**10.79**
4-4-16	RTN	250.85	295.17	486.89	344.30
	SmoothQuant	331.27	439.00	1042.84	604.37
	GPTQ	314.80	478.90	1084.13	625.94
	QuaRot	8.39	14.22	13.72	12.11
	SpinQuant	7.94	13.52	12.94	11.47
	OSTQuant	7.31	12.96	11.90	10.72
	DartQuant	7.32	12.84	11.87	10.68
	MirrorQuant	**7.26**	**12.68**	**11.77**	**10.57**
4-8-16	RTN	7.60	13.16	12.11	10.96
	SmoothQuant	8.30	14.48	12.60	11.79
	GPTQ	8.04	13.92	12.45	11.47
	QuaRot	6.54	11.75	10.49	9.59
	SpinQuant	**6.49**	11.75	**10.29**	9.51
	OSTQuant	6.50	11.78	**10.29**	9.52
	DartQuant	6.50	11.72	**10.29**	9.50
	MirrorQuant	6.50	**11.66**	10.31	**9.49**

Table 20: Complete perplexity results on Llama-3-70B. Mean is the arithmetic mean over WikiText-2, PTB, and C4; lower is better. Best and second-best quantized results in each setting are shown in bold and underlined, respectively.

Bits (W-A-KV)	Method	WikiText-2	PTB	C4	Mean
16-16-16	Full precision	2.86	8.53	7.17	6.19
4-4-4	RTN	397.15	398.61	421.26	405.67
	SmoothQuant	358.71	396.99	439.58	398.43
	GPTQ	378.32	380.60	387.23	382.05
	QuaRot	7.68	15.40	14.58	12.55
	SpinQuant	6.12	12.18	12.24	10.18
	OSTQuant	4.91	**9.83**	9.54	8.09
	DartQuant	5.03	10.43	9.66	8.37
	MirrorQuant	**4.83**	9.89	**9.41**	**8.04**
4-4-16	RTN	382.37	370.90	413.49	388.92
	SmoothQuant	401.13	498.16	363.48	420.92
	GPTQ	340.20	462.90	363.57	388.89
	QuaRot	7.41	14.80	14.06	12.09
	SpinQuant	5.97	11.65	11.26	9.63
	OSTQuant	4.82	**9.76**	9.36	7.98
	DartQuant	4.95	10.30	9.49	8.25
	MirrorQuant	**4.76**	9.78	**9.27**	**7.94**
4-8-16	RTN	4.46	9.54	8.60	7.53
	SmoothQuant	4.17	9.36	8.45	7.33
	GPTQ	4.16	9.33	8.44	7.31
	QuaRot	3.81	9.02	7.94	6.92
	SpinQuant	3.50	8.86	7.71	6.69
	OSTQuant	3.49	8.83	7.72	6.68
	DartQuant	**3.48**	8.83	**7.70**	6.67
	MirrorQuant	3.51	**8.68**	7.73	**6.64**