REVQ: ROTATION-ENHANCED PRODUCT VECTOR QUANTIZATION WITH LOW-RANK RESIDUAL COMPENSATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Joint low-bit quantization of weight and activation could substantially reduce the storage and computational costs of Transformer-based architectures and convolutional neural networks (CNNs). However, existing methods cannot simultaneously minimize the layer-wise local quantization errors caused by joint misalignment between weight and activation distributions and address accumulation of errors across layers. In this paper, we formulate the cross-layer and local-layer errors for joint weight and activation quantization, and consequently, propose a rotation-based two-stage framework named REVQ for low-bit quantization to simultaneously address the errors. Specifically, we develop learnable groupwise rotation transformations parametrized by the Lie algebra of the special orthogonal group SO(g) to learn the coordinate system of parameter (weight and activation) space for achieving product vector quantization and reducing local-layer quantization error. Moreover, we design cross-layer low-rank residual compensation to bridge rotation-based quantization in adjacent linear layers with learned low-rank compensation via Haar wavelets for minimizing cross-layer quantization error. Extensive experiments demonstrate that REVQ achieves state-of-the-art performance in various tasks of image classification on ImageNet and CIFAR-100, object detection and instance segmentation on COCO, and language modeling on Wikipedia using a wide range of Transformer-based architectures and CNNs.

1 INTRODUCTION

Transformer-based architectures and convolutional neural networks (CNNs) are prevailing in visual recognition and language modeling, but usually suffer from substantial storage, memory, and inference costs for deployment. Network quantization reduces these costs without altering network architectures by representing model weights and activations with lower-precision numerical formats. In practice, quantized models are obtained through either post-training quantization (PTQ) that quantizes a pretrained model using calibration data without training or quantization-aware training (QAT) that simulates quantization during optimization. This paper focuses on the challenging problem of low-bit quantization (equal to or lower than 4 bits) for both weights and activations. For example, vision Transformers (ViTs) often exhibit heavy-tailed, layer-dependent activations, while language models like BERT contain persistent channel outliers and rare large activations (Liu et al., 2023b; Sun et al., 2024). These distributions are difficult to fit with the axis-aligned grids of low-bit scalar quantizers, and cause unacceptable quantization error and performance loss.

Although QAT adapts model parameters to quantized computation, scalar QAT without coordinate transformations still uses an axis-aligned quantization grid. Existing methods alleviate distribution mismatch with fixed or learned rotations and other equivalent transformations (Ashkboos et al., 2024; Liu et al., 2024b; Sun et al., 2025). However, these transformations have been studied primarily for weight-only or activation-only quantization, and the interaction between weight and activation quantization has been neglected. In Figure 1, compared with weight-only or activation-only quantization, we find heavier activation tails than weight tails and larger output errors in ViT-S and ViT-B under joint W4A4 quantization. Since quantization error propagates through the network blocks, blockwise reconstruction with clean full-precision inputs cannot explicitly correct the cross-layer error caused by perturbation in inputs. There is still a lack of a joint weight and activation quantiza-

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.

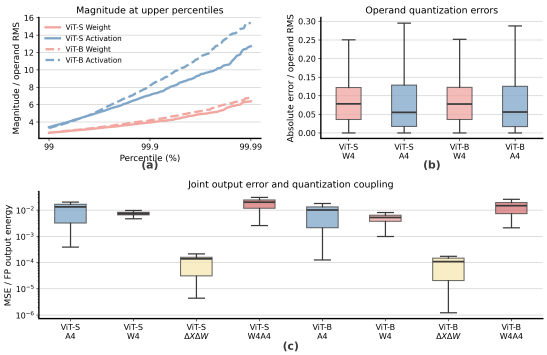


Figure 1: Diagnostics for ViT-S and ViT-B under W4A4 quantization: RMS-normalized upper-tail magnitudes, local operand quantization errors, and output errors, including the coupling term $\Delta X \Delta W$.

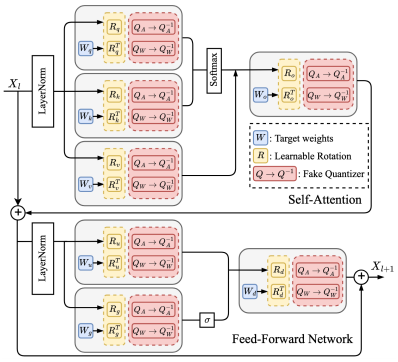


Figure 2: REVQ Transformer block. Learnable rotations are inserted before the target linear projections, followed by fake quantizers.

tion paradigm that can simultaneously adapt to weight and activation distributions and compensate for the cross-layer quantization error propagated through the network.

To address the problem, we propose REVQ, a two-stage framework that combines learnable groupwise rotations with cross-layer low-rank residual compensation to achieve high-performance joint low-bit quantization of weights and activations. Specifically, we first formulate the cross-layer and local-layer errors arising from joint weight and activation quantization. Subsequently, we develop learnable groupwise rotation transformations parameterized by the Lie algebra of the the special orthogonal group $\mathrm{SO}(g)$ for product vector quantization to minimize the local-layer error, and design cross-layer low-rank residual compensation to compensate for the cross-layer error propagated through the network caused by activation quantization.

Our contributions are summarized as follows.

- We propose REVQ, a rotation-based two-stage framework for joint low-bit quantization of weights and activations. REVQ simultaneously addresses the local-layer error with learnable groupwise rotations in $\mathrm{SO}(g)$ and minimizes cross-layer error with cross-layer low-rank residual compensation.
- We develop learnable groupwise rotation transformations parametrized by the Lie algebra of $\mathrm{SO}(g)$ to learn the coordinate system of parameter (weight and activation) space for reducing local-layer quantization error.
- We design cross-layer low-rank residual compensation to bridge rotation-based quantization in adjacent linear layers with learned low-rank compensation via Haar wavelets for minimizing cross-layer quantization error.

We evaluate REVQ on various vision and language tasks, including image classification on ImageNet and CIFAR-100, object detection and instance segmentation on COCO, and language modeling on Wikipedia, using a wide range of backbones such as Transformer-based architectures (*i.e.*, vision Transformers, Swin Transformers, and BERT) and CNNs like ResNet18, ResNet50, MobileNet-V2, and EfficientNet-V2. Extensive experimental results demonstrate that REVQ is comparable or superior to recent state-of-the-art methods and even full-precision baselines. Remarkably, on CIFAR-100, REVQ raises W4A4 performance above the pretrained FP baseline across all evaluated ViT and Swin models by jointly training rotations and weights as shown in Table 2.

2 Related Work

2.1 Low-bit Quantization for Transformers

Low-bit quantization is primarily achieved for Transformers via post-training quantization (PTQ) or quantization-aware training (QAT) (Jacob et al., 2018). QAT methods for vision Transformers

focus on quantization distortion and optimization instability. Q-ViT combines information rectification with distribution-guided distillation (Li et al., 2022). OFQ addresses weight oscillation during quantization training (Liu et al., 2023a), whereas Quantization Variation uses multi-crop distillation to supervise the quantized student model (Huang et al., 2024). For language models, LLM-QAT studies data-free QAT (Liu et al., 2024a), while QuEST explores 1-bit weight and activation quantization (Panferov et al., 2025). Many of these methods use teacher-based distillation to improve distribution alignment or training stability. Our method instead uses only full-precision outputs on clean full-precision inputs as the reconstruction target, and jointly trains groupwise rotations and weights so that the learned coordinate system of the parameter space adapts to the quantization grid.

2.2 Product Codebooks and Quantization Transformations

Vector quantization represents vectors with a finite codebook, while product quantization decomposes the representation into subspaces with separate codebooks. Optimized product-quantization methods adapt this decomposition to the data (Ge et al., 2013; Kalantidis & Avrithis, 2014). QuIP# combines incoherence processing with lattice codebooks for weight quantization (Tseng et al., 2024), whereas FSQ uses scalar levels to form an implicit Cartesian-product codebook (Mentzer et al., 2024). In this paper, we focus on groupwise quantization, which defines scales and zero points for scalar rounding rather than explicit vector-codeword search.

Equivalent transformations have also been used to improve the compatibility of weights and activations with low-bit quantizers. SmoothQuant redistributes quantization difficulty through scaling (Xiao et al., 2023). QuaRot (Ashkboos et al., 2024) and SpinQuant (Liu et al., 2024b) use randomized and learned rotations, respectively. DuQuant combines rotations with channel permutations (Lin et al., 2024). DFRot refines rotational transformations to suppress long-tailed activations (Xiang & Zhang, 2025), whereas DartQuant performs rotational distribution calibration (Shao et al., 2026). OSTQuant combines orthogonal and scaling transformations (Hu et al., 2025), while FlatQuant learns transformations that flatten weight and activation distributions (Sun et al., 2025). ParoQuant uses pairwise rotations (Liang et al., 2026); and ButterflyQuant employs learnable orthogonal butterfly transforms (Xu et al., 2025). However, most of the above methods focus on quantization for large language models (LLMs), whereas our method is applicable to both Transformer-based architectures and CNNs. Moreover, unlike the rotation matrices used in these methods, REVQ parameterizes group-wise rotations through the Lie algebra SO and jointly optimizes rotations and weights during blockwise QAT.

2.3 Low-rank Residual Correction

LoRA introduces low-rank factors for parameter-efficient adaptation (Hu et al., 2022). LoftQ (Li et al., 2024) jointly searches for a quantized backbone and a low-rank initialization, while LQ-LoRA (Guo et al., 2024) decomposes a pretrained weight matrix into quantized and trainable low-rank components. These methods primarily target parameter-efficient downstream adaptation rather than cross-layer quantization error. Other methods use low-rank factors to reconstruct quantization error directly. LQER approximates weight quantization error with truncated SVD, and its activation-aware variant, L^2QER, incorporates activation statistics before factorization (Zhang et al., 2024). QERA minimizes expected layer-output error for fixed quantized weights and derives exact and diagonal solutions based on second moments of the input (Zhang et al., 2025). These approaches target activation-dependent output error at an individual layer.

REVQ instead uses cross-layer low-rank residual compensation to initialize corrections at adjacent linear-layer interfaces after rotation training. With rotations and projection weights frozen, it refines the low-rank factors using cross-layer supervision. The corrections are applied after weight quantization in rotated coordinates. For residual compensation, We compare the SVD initialization with the importance-aware Haar wavelet initialization.

3 Proposed Method

In this section, we present REVQ, a two-stage framework for Transformer quantization. We first introduce the problem formulation in Section 3.1. Subsequently, we elaborate on shared rotation design in Section 3.2, residual compensation in Section 3.3, and two-stage optimization in Section 3.4.

Table 1: Instantiated layers and selected layers for residual compensation.

Architectures	Linear Layers	Residual Compensation
ViT	Fused QKV projection, Attention output projection, and MLP projections	Fused QKV projection, first MLP projection
Swin / BERT	Query, Key, Value, Attention output, and FFN/MLP projections	Query, first FFN/MLP projection
CNN	Convolutional and linear layers, excluding the classification head	All instantiated convolutional and non-classification linear layers

3.1 Problem Formulation and Motivation

We focus on quantizing the weights and activations in linear layers for feature transformation in Transformer-based architectures and convolutional neural networks (CNNs), as summarized in Table 1. Consider an arbitrary linear layer that produces a d_{out}-channel output activation $Y \in \mathbb{R}^{n \times d_{\text{out}}}$ from a d_{in}-channel input activation $X \in \mathbb{R}^{n \times d_{\text{in}}}$ with weight matrix $W \in \mathbb{R}^{d_{\text{in}} \times d_{\text{out}}}$, where n is the number of activation vectors to be processed, *e.g.*, the number of tokens for Transformer-based models.

Problem Formulation. In QAT, the weight W is quantized to $\widehat{W}$ in b_w bits with a signed symmetric quantizer $Q_w(\cdot)$, and the quantized output activation $\widehat{Y} = \widehat{X}\widehat{W}$ is obtained from the quantized input activation $\widehat{X}$ in b_a bits with a dynamic asymmetric quantizer $Q_a(\cdot)$.

$$Q_w(x) = s_w(x)\ \text{clip}\left(\left\lfloor \frac{x}{s_w(x)} \right\rceil, -(2^{b_w-1}-1), 2^{b_w-1}-1\right), \tag{1}$$

$$Q_a(x) = s_a(x)\left[\text{clip}\left(\left\lfloor \frac{x}{s_a(x)} + z_a(x) \right\rceil, 0, 2^{b_a}-1\right) - z_a(x)\right], \tag{2}$$

where s_w and s_a are the scales varying with the weight and activation.

The orthogonal rotation transformation R is employed to admit an identity transform $XW = (XR)(R^\top W)$ on the full-precision linear mapping function. This allows the quantization coordinates not to alter the full-precision linear mapping. The quantized output activation is

$$\widehat{Y} = Q_a(\widehat{X}R)Q_w(R^\top W). \tag{3}$$

Let $E^X = X - \widehat{X}$ denote the error between the full precision activation X and its quantized version $\widehat{X}$ propagated from the previous layer. We obtain the quantization errors for the rotated weight $R^\top W$ and activation $\widehat{X}R$.

$$E^A = Q_a(\widehat{X}R) - \widehat{X}R, \qquad E^W = Q_w(R^\top W) - R^\top W. \tag{4}$$

From (3) and (4), the quantization error for Y can be approximated by omitting the high-order products $E^X R E^W$ and $E^A E^W$ associated with E^X, E^A, and E^W.

$$\widehat{Y} - Y \approx \underbrace{-E^X W}_{\text{cross-layer}} + \underbrace{E^A R^\top W + XRE^W}_{\text{local-layer}}. \tag{5}$$

Here, the local-layer term depends on E^A and E^W associated with the rotation R, and the cross-layer term is determined by E^X propagated from previous layer.

Motivation. According to (5), the local-layer errors depend on the misalignment between the parameter distribution and the quantization grid, while the cross-layer term reflects the propagated quantization error on input activations that cannot be explicitly recovered by local-layer reconstruction. This motivates the main contributions of rotation-enhanced product vector quantization and cross-layer low-rank residual compensation in this paper.

First, we introduce in Section 3.2 the learnable orthogonal rotation based on Lie algebra to learn rotations for adjusting the coordinate system of the parameter space and reducing the local-layer error. We partition the input dimension into G groups of size g such that $d_{\text{in}} = G \times g$. For a weight

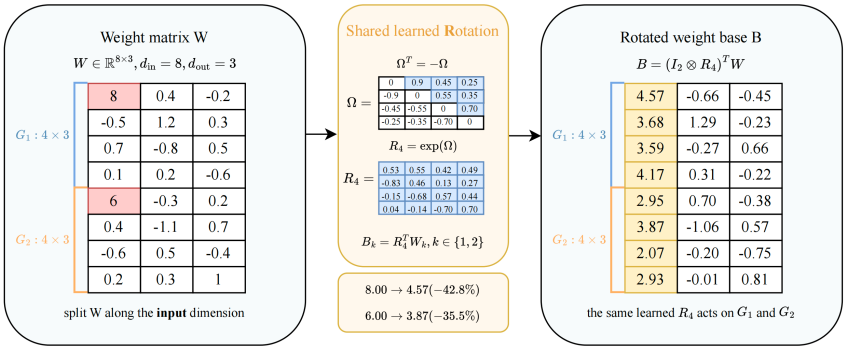


Figure 3: The impact of groupwise orthogonal rotation.

group u and activation group v, let $v_- = \min(\min_i(v_i), 0)$ and $v_+ = \max(\max_i(v_i), 0)$. The quantization scale $s_w(u)$ for the weight and quantization scale $s_a(v)$ and zero point $z_a(v)$ for the activation are determined by

$$s_w(u) = \frac{\max(\|u\|_\infty, \epsilon)}{2^{b_w-1} - 1}, s_a(v) = \max\left(\frac{v_+ - v_-}{2^{b_a} - 1}, \epsilon\right), z_a(v) = \text{clip}\left(-\left\lfloor \frac{v_-}{s_a(v)} \right\rceil, 0, 2^{b_a} - 1\right). \quad (6)$$

Second, we further analyze the cross-layer error and achieve cross-layer residual compensation for the rotation-enhanced quantization via learning in the low-rank or Haar wavelet space in Section 3.3.

3.2 Rotation-Enhanced Product Vector Quantization

Different from existing rotation-based methods, we develop learnable groupwise rotation transformations via the Lie algebra of the special orthogonal group (Hall, 2015; Absil et al., 2008) to enhance product vector quantization by simultaneously adapting the orientation of the scale-product codebook to weight and activation and preserving its structure.

$$\widehat{Y} = Q_a\left(\widehat{X} R(\boldsymbol{\theta})\right) Q_w\left(R(\boldsymbol{\theta})^\top W\right), \quad (7)$$

where $R(\boldsymbol{\theta})$ is parameterized by the $g \times g$ matrix of unconstrained learnable parameters $\boldsymbol{\theta}$ to learn admissible rotations $R_g(\boldsymbol{\theta}) \in \text{SO}(g)$ through a skew-symmetric generator $A(\boldsymbol{\theta})$ in Lie algebra $\mathfrak{so}(g)$.

$$R(\boldsymbol{\theta}) = I_G \otimes R_g(\boldsymbol{\theta}), \quad \text{with } R_g(\boldsymbol{\theta}) = \exp(A(\boldsymbol{\theta})). \quad (8)$$

Here, $\otimes$ is the Kronecker product to expand R_g into the $d_{\text{in}} \times d_{\text{in}}$ full matrix. The Lie algebra $\mathfrak{so}(g)$ is the tangent space of the Lie group $\text{SO}(g)$ at the identity, and the matrix exponential maps elements of $\mathfrak{so}(g)$ to admissible rotations in $\text{SO}(g)$. In (8), $A(\boldsymbol{\theta})$ is formulated by combining the Frobenius-orthonormal basis $\{\Omega_{ij}\}_{1\le i<j\le g}$ of $\mathfrak{so}(g)$ with $\boldsymbol{\theta}$.

$$A(\boldsymbol{\theta}) = \sum_{1\le i<j\le g} \theta_{ij}\Omega_{ij} \in \mathfrak{so}(g). \quad (9)$$

Here, θ_{ij} is the (i, j)-th element of $\boldsymbol{\theta}$. $\Omega_{ij} = \frac{1}{\sqrt{2}}(e_i e_j^\top - e_j e_i^\top)$ is constructed from the standard basis vectors $\{e_i\}_{1\le i\le g}$ of $\mathbb{R}^g$ and satisfies the skew-symmetric property of $A(\boldsymbol{\theta})^\top = -A(\boldsymbol{\theta})$. Therefore, orthogonality is enforced for the proposed learnable rotations $R_g(\boldsymbol{\theta})$ and $R(\boldsymbol{\theta})$ without introducing an additional penalty.

$$R_g^\top R_g = \exp(A^\top)\exp(A) = \exp(-A)\exp(A) = I_g, \qquad R^\top R = I_{d_{\text{in}}}. \quad (10)$$

Note that $R_g(\boldsymbol{\theta})$ uses $g(g-1)/2$ rotation coordinates per projection. Figure 3 illustrates an example to show that rotation exerts a smoothing effect by altering the coordinate systems of the parameter space. Refer to Appendix A.1 for more details on invariance analysis. This includes the proof that a learned rotation can reduce coordinate peaks in the calibration distribution, thereby tightening the rounding-error bound while preserving the vector norm.

3.3 CROSS-LAYER LOW-RANK RESIDUAL COMPENSATION

To address cross-layer errors, we further introduce residual compensation between selected linear layers. For ViT models, we apply compensation to the weights of two types of linear layers, *i.e.*, i) the first linear layer of the MLP after the attention module within each Transformer block, and ii) the attention input projection layer of the next block. For CNN models, we add compensation to every designated quantization target, including convolutional and fully connected layers. Details are provided in Table 1.

Specifically, in (3), we introduce the learnable residual compensation ΔB into the rotated and quantized version of weight.

$$\widehat{Y}' = Q_a(\widehat{X}R)[Q_w(R^\top W) + \Delta B], \tag{11}$$

ΔB is learned over the calibration batches indexed by m. It is initialized using the cross-layer term formulated in (5) and then obtained by solving a weighted rank-r regression problem.

$$\min_{\Delta B} \left[\sum_m \left\| Q_a(\widehat{X}_{j,m} R_{j,m}) \Delta B - E^X W \right\|_F^2 + \rho \|\Delta B\|_F^2 \right], \text{ s.t. } \operatorname{rank}(\Delta B) \le r. \tag{12}$$

To enforce the low-rank constraint on ΔB, we design two strategies for residual initialization.
First, we consider using SVD (Golub & Reinsch, 1970):

$$\Delta B_{\mathrm{SVD}} = UV, \qquad U \in \mathbb{R}^{d_{\mathrm{in}} \times r}, \quad V \in \mathbb{R}^{r \times d_{\mathrm{out}}}. \tag{13}$$

Second, we compute the Hessian of the activations within each group to sort the channels in descending order and employ a block-diagonal orthogonal Haar wavelet transform H to initialize rank-r factors in two bands.

$$\Delta B_{\mathrm{Haar}} = \Pi^\top H^\top \left(P_L U_L V_L + P_H U_H V_H \right). \tag{14}$$

Here, Π is the within-group permutation, P_L, P_H embed the two frequency bands (Mallat, 1989), and U_L, V_L, U_H, V_H denote the low-rank residual factors obtained from the low- and high-frequency bands, respectively. Note that r denotes the rank assigned to each frequency band for Haar-based initialization. Refer to Appendix A.2 for more details on the derivation of the residual initialization and the trainable compensation parameters, the whitened SVD solution, and the sequential Haar "low-to-high" initialization.

3.4 TWO-STAGE OPTIMIZATION

Finally, we present the two-stage optimization for QAT with each stage traversing the network in a block-wise manner and updating only the active block. After the first-stage rotation training, we restore the checkpoint and freeze the rotations and weights. We then initialize the residual factors and refine them through residual compensation. Appendix A.4 provides the complete algorithm.

Stage 1: Learning rotation and weight. Each layer learns its own groupwise rotation initialized to the identity by optimizing the loss function $\mathcal{L}_{\mathrm{rot}}$.

$$\mathcal{L}_{\mathrm{rot}} = \mathcal{L}_{\mathrm{task}} + \lambda_{\mathrm{out}} \mathcal{L}_{\mathrm{out}} + \lambda_{\mathrm{cos}} \mathcal{L}_{\mathrm{cos}}. \tag{15}$$

Here, $\mathcal{L}_{\mathrm{task}}$ denotes the cross-entropy loss for image classification in vision models and the masked language modeling (MLM) loss for BERT. $\mathcal{L}_{\mathrm{out}}$ is the local-layer output reconstruction loss and encourages the quantized output $\widehat{Y}$ to match the unquantized output $Y_{\mathrm{loc}} = \widehat{X}W$, and $\mathcal{L}_{\mathrm{cos}} = 1 - \mathrm{mean}_i[\widehat{\mathbf{y}}_i^\top \mathbf{y}_i / (\|\widehat{\mathbf{y}}_i\|_2 \|\mathbf{y}_i\|_2)]$ measures the cosine similarity between the the i-th rows $\mathbf{y}_i$ and $\widehat{\mathbf{y}}_i$ of the corresponding output matrices in the feature space.

Stage 2: Residual compensation. In this stage, we add low-rank residual compensation to the quantized weights. We optimize $\mathcal{L}_{\mathrm{compen}} = \mathcal{L}_{\mathrm{task}} + \lambda_{\mathrm{out}} \mathcal{L}_{\mathrm{out}} + \lambda_{\mathrm{cos}} \mathcal{L}_{\mathrm{cos}} + \lambda_{\mathrm{clean}} \mathcal{L}_{\mathrm{clean}}$. Here $\mathcal{L}_{\mathrm{clean}}$ represents the reconstruction error between the compensated quantized output $\widehat{Y}'$ and the full-precision reference target $Y = XW$, as formulated in (12).

4 EXPERIMENTS

We evaluate REVQ on vision and masked language modeling (MLM) tasks. We report the main results for rotation-enhanced quantization and residual compensation, followed by ablations on rotation mechanisms, the selection of residual compensation target interfaces, and channel ordering for Haar compensation. All experiments are conducted on a single NVIDIA RTX 3090 GPU.

Table 2: Main classification results on ImageNet-1K and CIFAR-100. The *Trainable* column specifies whether rotations and weights are optimized.

Dataset	Bit-width W–A	Trainable		ViT-S		ViT-B		ViT-L		Swin-T		Swin-S		Swin-B	
		Weight	Rotation	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
ImageNet-1K	FP baseline	✗	✗	81.40	96.19	85.20	97.55	85.90	97.80	81.18	95.52	83.21	96.25	85.16	97.50
	W4A8	✗	✗	79.96	95.59	84.49	**97.31**	85.6	97.74	80.61	95.2	82.67	96.02	84.75	97.38
	W4A8	✗	✓	80.17	95.58	**84.56**	97.28	85.72	97.76	**80.76**	95.27	82.76	96.1	84.78	97.42
	W4A8	✓	✓	**80.45**	**95.73**	84.42	97.25	**85.83**	**97.83**	80.7	**95.28**	**82.78**	**96.13**	**84.85**	**97.42**
	W4A4	✗	✗	79.23	95.20	84.23	**97.26**	85.43	97.70	80.38	95.14	82.58	95.89	84.55	97.31
	W4A4	✗	✓	79.57	95.25	84.27	97.17	85.53	97.73	80.56	**95.19**	82.57	95.99	84.60	97.35
	W4A4	✓	✓	**80.00**	**95.55**	**84.27**	97.16	**85.71**	**97.75**	**80.63**	95.12	**82.61**	**96.03**	**84.63**	**97.35**
	W3A3	✗	✗	54.33	78.63	79.30	94.91	83.21	96.81	74.38	91.81	79.29	94.42	80.98	95.86
	W3A3	✗	✓	62.31	85.35	79.61	95.01	83.66	97.04	75.62	92.48	79.60	94.71	81.65	96.20
	W3A3	✓	✓	**69.97**	**90.04**	**80.58**	**95.44**	**84.22**	**97.25**	**76.90**	**93.29**	**80.24**	**94.93**	**82.26**	**96.41**
CIFAR-100	FP baseline	✗	✗	86.87	98.20	91.55	99.17	94.16	99.28	87.12	98.19	88.21	98.74	92.78	99.33
	W4A4	✗	✗	85.35	97.69	90.80	98.99	93.87	99.20	86.38	98.03	87.32	98.67	92.12	99.2
	W4A4	✗	✓	85.78	97.87	90.83	98.99	93.89	99.20	86.68	98.01	87.6	98.55	92.28	99.28
	W4A4	✓	✓	**87.37**	**98.14**	**92.24**	**99.23**	**94.20**	**99.25**	**87.89**	**98.38**	**88.52**	**98.79**	**92.87**	**99.29**

4.1 Experimental Settings

Datasets and Models. For image classification, we evaluate ViT-S/B/L (Steiner et al., 2021; Dosovitskiy et al., 2021) and Swin-T/S/B (Liu et al., 2021) on ImageNet-1K (Deng et al., 2009) and CIFAR-100 (Krizhevsky, 2009). We additionally evaluate ResNet-18/50 (Wightman et al., 2021; He et al., 2015), EfficientNet-V2 (Tan & Le, 2021) and MobileNet-V2 (Sandler et al., 2018) on CIFAR-100. We report Top-1 and Top-5 accuracy for all classification results. For masked language modeling, we evaluate BERT-Large (Devlin et al., 2018) on English Wikipedia (Wikimedia Foundation, 2026), reporting masked-token perplexity and token accuracy. We provide detailed dataset splits and masking protocols in Appendix A.3. For downstream vision tasks, we evaluate Swin-T/S (Liu et al., 2021) with Mask R-CNN on COCO2017 (Lin et al., 2014). Training follows the detector configuration and preprocessing pipeline specified in the MMDetection (Chen et al., 2019) configuration. Evaluation uses the official COCO bounding-box and instance-segmentation metrics, including Box AP, Mask AP, Box AP^{50}, and Mask AP^{50}.

Quantization Settings. We primarily conduct joint 4-bit weight and 4-bit activation quantization (W4A4) experiments. On ImageNet, we additionally evaluate ViT models under the W4A8 and W3A3 settings. Unless otherwise specified, all experiments use a group size of $g = 32$. We adopt the groupwise signed symmetric uniform weight quantizer and dynamic asymmetric uniform activation quantizer defined in Section 3.1.

Implementation Details. We use AdamW with linear warm-up and cosine decay for both stages, and we set model-specific learning rates for the target weights to account for differences in optimization dynamics across architectures. During residual compensation, the rotations and projection weights are frozen, while the low-rank residual factors are optimized with a learning rate of 5×10^{-5}. The calibration samples for the residual compensation stage are aligned with the corresponding first-stage training data. More training details including dataset-specific processing and learning rates are provided in Appendix A.3.

4.2 Main Results

As listed in Table 1, we quantize the main feature transformation layers of each architecture. For ViT, we quantize the fused QKV projection, the attention output projection, and the two MLP projections. For Swin and BERT, we quantize the Query, Key, Value, attention output, and FFN/MLP projections. For CNNs, we quantize convolutional and linear layers, excluding the final classification head. CNN results are reported in Table 7 in Appendix A.5. We additionally provide visualizations of COCO downstream tasks and BERT-Large masked-language-modeling results in Appendix A.9.

Image Classification. Table 2 reports Top-1 and Top-5 accuracy on ImageNet-1K and CIFAR-100. We evaluate W4A8, W4A4 and W3A3 quantization on ImageNet and W4A4 quantization on

Table 3: Comparison Top-1 on ImageNet between REVQ and representative quantization methods.

Method	W–A	ViT-S	ViT-B	Swin-T	Swin-S	Swin-B
Full-Precision	W32A32	81.40	85.20	81.18	83.21	85.16
PTQ4ViT (Yuan et al., 2022)	W4A4	42.57	30.69	–	76.09	74.02
APQ-ViT (Ding et al., 2022)	W4A4	47.95	41.41	–	77.15	76.48
RepQ-ViT (Li et al., 2023)	W4A4	65.05	68.48	73.00	71.90	69.00
GPLQ (Liang et al., 2025)	W4A4	–	–	79.50	82.00	84.00
Ours (Rotation-QAT)	W4A4	**80.00**	**84.27**	**80.63**	**82.61**	**84.63**
RepQ-ViT (Li et al., 2023)	W3A3	–	–	52.70	65.90	65.8
GPLQ (Liang et al., 2025)	W3A3	–	–	74.2	78.3	80.3
Ours (Rotation-QAT)	W3A3	**69.97**	**80.58**	**76.90**	**80.24**	**82.26**

Table 4: Comparison of residual compensation methods on ImageNet-1K and CIFAR-100.

Dataset	Method	ViT-S		ViT-B		ViT-L		Swin-T		Swin-S		Swin-B	
		Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
ImageNet-1K	W4A8 (Rotation-QAT)	80.45	95.73	84.42	97.25	85.83	97.83	80.70	**95.28**	82.78	**96.13**	**84.85**	97.42
	+ SVD (r=1)	80.45	**95.74**	84.40	97.25	85.86	97.83	80.73	95.21	82.79	96.08	84.76	97.41
	+ Haar (r=1)	**80.51**	95.72	**84.42**	**97.25**	**85.89**	**97.84**	**80.79**	95.21	**82.81**	96.08	84.76	**97.42**
	W4A4 (Rotation-QAT)	80.00	95.55	84.27	97.17	**85.71**	97.75	80.63	95.12	**82.61**	**96.03**	84.63	97.35
	+ SVD (r=1)	80.05	95.48	84.34	**97.21**	85.65	97.76	80.64	95.14	82.52	96.03	84.64	97.35
	+ Haar (r=1)	**80.11**	**95.60**	**84.35**	97.16	85.67	**97.78**	**80.69**	**95.15**	82.53	96.02	**84.66**	**97.36**
CIFAR-100	W4A4 (Rotation-QAT)	87.37	98.14	92.24	99.23	94.20	99.25	87.89	98.38	88.52	98.79	**92.87**	99.29
	+ SVD (r=1)	87.91	98.22	**92.34**	99.21	94.25	99.35	88.01	98.40	88.49	98.86	92.78	**99.30**
	+ Haar (r=1)	**87.92**	**98.26**	92.27	**99.37**	**94.26**	**99.37**	**88.13**	**98.42**	**88.60**	**98.89**	92.83	99.25

CIFAR-100. Under W4A4 quantization, joint optimization improves Top-1 over direct quantization by up to 0.77 percentage points (pp) on ImageNet and 2.02 pp on CIFAR. The improvement becomes more pronounced at W3A3. Table 3 compares our method with representative ViT quantization methods on ImageNet. The "Rotation-QAT" means jointly training rotations and weights which is the first stage in out method. Under W4A4, we verify that Rotation-QAT achieves the highest Top-1 across all five evaluated architectures. Under W3A3, our method also outperforms the compared methods on all Swin models.

BERT-Large. Table 8 shows that Rotation-QAT substantially narrows the gap to the full-precision baseline. Rank-1 residual compensation further reduces perplexity and improves token accuracy with Haar initialization.

COCO Detection and Segmentation. Detailed COCO2017 detection and segmentation results are reported in Table 9. Compared with representative ViT quantization methods, our rotation-based W4A4 approach achieves the smallest performance degradation relative to the full-precision models.

4.3 Cross-Layer Residual Compensation

For a controlled comparison, we apply residual compensation to the W4A4 Rotation-QAT checkpoint of each model, evaluating rank-1 SVD Initialization and rank-1 importance-aware Haar initialization. As shown in Table 4, residual compensation improves the CIFAR-100 Top-1 accuracy on four of the six architectures, with Haar achieving the best results on ViT-S/L and Swin-T/S. On ImageNet, the gains are modest and architecture-dependent. Haar method improves ViT-S Top-1 under W4A4 by up to 0.11 pp but slightly degrades performance on ViT-L and Swin-S. These results indicate that rotation-based QAT leaves limited space for subsequent residual error correction.

4.4 Ablation Studies

We separate the contributions of learnable rotations and residual compensation through three ablation studies. Detailed metric definitions and analysis are provided in Appendix A.8.

Rotation Mechanism. Under the W4A4 setting, we examine three QAT strategies: training only the projection weights (Weight-Only), training only the rotation parameters (Rotation-Only), and

jointly optimizing both (Rotation-QAT). As shown in Figure 4 and Tables 10 and 11, Rotation-Only can improve performance for most evaluated models, while Rotation-QAT generally achieves the best performance, demonstrating the benefit of jointly adapting the projection weights and rotations.

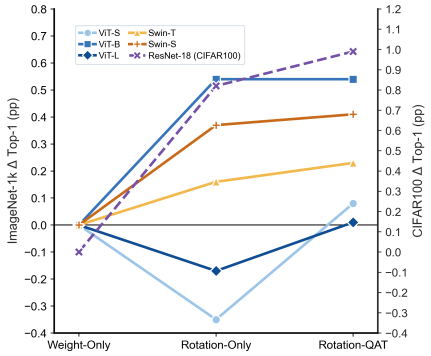


Figure 4: The impact of Rotation and QAT mechanism.

Selection of Target Interfaces. Following the selection criteria described in Table 1 and Section 3.3, we compare three target-layer selection strategies for rank-1 Haar compensation on the Swin-T: (i) select all valid linear layers in the Swin models specified in Table 1; (ii) select the ten valid linear layers with the the highest cross-layer error scores; and (iii) select the ten deepest valid linear layers in the model. We compute and aggregate the cross-layer and local-layer errors before and after compensation across the calibration batches and samples, and compare the resulting error changes. Negative values indicate error reduction. Table 5 shows that selective compensation is more effective than activating all valid linear layers. With other budgets held equal, Deep-10 shows better performance than Top-10, highlighting the importance of addressing architecture-dependent quantization errors in deeper layers. All three strategies reduce the cross-layer error while increasing the local-layer error, suggesting that the residual factors primarily correct errors accumulated from preceding layers.

Effect of Channel Ordering For Haar. We evaluate the effect of channel ordering in Haar compensation on ViT-S. We compare no channel ordering (None) with within-group ordering based on the Hessian of activations (Importance). Table 6 shows that, given the same budget, importance-based ordering can improve accuracy and increase the proportion of target interfaces that benefit from compensation. Moreover, the aggregate deviation from the full precision reference outputs changes from a slight increase to a slight decrease, indicating more consistent residual correction.

Table 5: The impact of the interface selection policies in Haar transformation.

Policy	Num of # Interfaces	Cross-layer ΔErr (%)	Local-layer ΔErr (%)	Top-1	Top-5
All-20	20	−0.70	+12.49	80.38	95.07
Top-10	10	−1.89	+9.94	80.48	95.12
Deep-10	10	−2.06	+6.08	**80.63**	95.12

Table 6: The impact of the Importance channel ordering in Haar transformation.

Ordering	Top-1	Top-5	Cross-layer ΔErr (%)	Improved (%)
None	87.84	98.20	+0.00392	73.9
Importance	**87.92**	**98.26**	−0.00053	**87.0**
Difference	+0.08	+0.06	−0.00445	+13.1

5 Conclusion

In this paper, we propose REVQ, a two-stage framework for joint weight and activation quantization that combines groupwise rotation learning with selective low-rank residual compensation. Learnable rotation provides more consistent improvements across different models and tasks, whereas the gain from residual compensation depends on the selection of target layers and the remaining capacity for error correction after rotation training. Given that our current method is a two-stage optimization, our future work will investigate the joint optimization of rotation and compensation. Moreover, because the two residual initialization methods differ in their learnable parameter structures, we plan to study additional equal-budget residual designs and further investigate the selection of target layers for compensation.

AI USE STATEMENT

In our work, AI tools were used for two main purposes. (i) Polishing and checking the manuscript's grammar and language, including correcting grammatical errors, refining expression, and improving clarity. (ii) Assisting with code writing for the research. They also helped with peripheral tasks such as writing visualization scripts. All AI-assisted content and code were reviewed and verified.

REFERENCES

P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2008.

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 *Advances in Neural Information Processing Systems*, volume 37, pp. 100213–100240, 2024.

Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, KC Jimmy, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. Mmdetection: Open mmlab detection toolbox and benchmark. *ArXiv*, abs/1906.07155, 2019. URL https://api.semanticscholar.org/CorpusID:189927886.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, 2018. URL http://arxiv.org/abs/1810.04805.

Yifu Ding, Haotong Qin, Qinghua Yan, Zhenhua Chai, Junjie Liu, Xiaolin Wei, and Xianglong Liu. Towards accurate post-training quantization for vision transformer. In *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 5380–5388, 2022. URL https://doi.org/10.1145/3503161.3547826.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.

Tiezheng Ge, Kaiming He, Qifa Ke, and Jian Sun. Optimized product quantization for approximate nearest neighbor search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2946–2953, June 2013.

Gene H. Golub and Christian Reinsch. Singular value decomposition and least squares solutions. *Numerische Mathematik*, 14(5):403–420, 1970.

Han Guo, Philip Greengard, Eric Xing, and Yoon Kim. LQ-loRA: Low-rank plus quantized matrix decomposition for efficient language model finetuning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=xw29VvOMmU.

Brian C. Hall. *Lie Groups, Lie Algebras, and Representations: An Elementary Introduction*. Springer, 2 edition, 2015. doi: 10.1007/978-3-319-13467-3.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Xing Hu, Yuan Cheng, Dawei Yang, Zhixuan Chen, Zukang Xu, JiangyongYu, XUCHEN, Zhihang Yuan, Zhe jiang, and Sifan Zhou. OSTQuant: Refining large language model quantization with orthogonal and scaling transformations for better distribution fitting. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=rAcgDBdKnP.

Xijie Huang, Zhiqiang Shen, Pingcheng Dong, and Kwang-Ting Cheng. Quantization variation: A new perspective on training transformers with low-bit precision. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=MHfoA0Qf6g.

Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference . In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2704–2713, 2018. URL https://doi.ieeecomputersociety.org/10.1109/CVPR.2018.00286.

Yannis Kalantidis and Yannis Avrithis. Locally optimized product quantization for approximate nearest neighbor search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2321–2328, June 2014.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Yanjing Li, Sheng Xu, Baochang Zhang, Xianbin Cao, Peng Gao, and Guodong Guo. Q-vit: Accurate and fully quantized low-bit vision transformer. In *Advances in Neural Information Processing Systems*, volume 35, pp. 34451–34463, 2022.

Yixiao Li, Yifan Yu, Chen Liang, Nikos Karampatziakis, Pengcheng He, Weizhu Chen, and Tuo Zhao. Loftq: LoRA-fine-tuning-aware quantization for large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=LzPWWPAdY4.

Zhikai Li, Junrui Xiao, Lianwei Yang, and Qingyi Gu. Repq-vit: Scale reparameterization for post-training quantization of vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17227–17236, 2023.

Guang Liang, Xinyao Liu, and Jianxin Wu. GPLQ: A general, practical, and lightning QAT method for vision transformers. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=58Vr1KOWG9.

Yesheng Liang, Haisheng Chen, Zihan Zhang, Song Han, and Zhijian Liu. ParoQuant: Pairwise Rotation Quantization for Efficient Reasoning LLM Inference. In *International Conference on Learning Representations (ICLR)*, 2026.

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 *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=mp8u2Pcmqz.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755, 2014.

Shih-Yang Liu, Zechun Liu, and Kwang-Ting Cheng. Oscillation-free quantization for low-bit vision transformers. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 21813–21824, 2023a. URL https://proceedings.mlr.press/v202/liu23w.html.

Yijiang Liu, Huanrui Yang, Zhen Dong, Kurt Keutzer, Li Du, and Shanghang Zhang. Noisyquant: Noisy bias-enhanced post-training activation quantization for vision transformers. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20321–20330, 2023b. URL https://ieeexplore.ieee.org/document/10203639/.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, and Vikas Chandra. Llm-qat: Data-free quantization aware training for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 467–484, 2024a. URL https://aclanthology.org/2024.findings-acl.26.

Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. Spinquant: Llm quantization with learned rotations. *ArXiv*, abs/2405.16406, 2024b. URL https://api.semanticscholar.org/CorpusID:270062819.

Stephane G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1989.

Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite scalar quantization: VQ-VAE made simple. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=8ishA3LxN8.

Andrei Panferov, Jiale Chen, Soroush Tabesh, Mahdi Nikdan, and Dan Alistarh. QuEST: Stable training of LLMs with 1-bit weights and activations. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=I0Ux2nAN6u.

Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, 2018. URL https://api.semanticscholar.org/CorpusID:4555207.

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

Andreas Steiner, Alexander Kolesnikov, , Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? data, augmentation, and regularization in vision transformers. *arXiv preprint arXiv:2106.10270*, 2021.

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

Yuxuan Sun, Ruikang Liu, Haoli Bai, Han Bao, Kang Zhao, Yuening Li, JiaxinHu, Xianzhi Yu, Lu Hou, Chun Yuan, Xin Jiang, Wulong Liu, and Jun Yao. Flatquant: Flatness matters for LLM quantization, 2025. URL https://openreview.net/forum?id=pxGucWt9vM.

Mingxing Tan and Quoc Le. Efficientnetv2: Smaller models and faster training. In *International conference on machine learning*, pp. 10096–10106, 2021.

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 41st International Conference on Machine Learning*, 2024.

Ross Wightman, Hugo Touvron, and Hervé Jégou. Resnet strikes back: An improved training procedure in timm. *arXiv e-prints*, pp. arXiv:2110.00476, 2021.

Wikimedia Foundation. Wikimedia downloads, 2026. URL https://dumps.wikimedia.org. Accessed: 2026-09-23.

Jingyang Xiang and Sai Qian Zhang. DFRot: Achieving outlier-free and massive activation-free for rotated LLMs with refined rotation. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=WzGypILLDb.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. Smoothquant: accurate and efficient post-training quantization for large language models. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

Bingxin Xu, Zhen Dong, Oussama Elachqar, and Yuzhang Shang. Butterflyquant: Ultra-low-bit llm quantization through learnable orthogonal butterfly transforms. *ArXiv*, abs/2509.09679, 2025. URL https://api.semanticscholar.org/CorpusID:281252400.

Zhihang Yuan, Chenhao Xue, Yiqi Chen, Qiang Wu, and Guangyu Sun. Ptq4vit: Post-training quantization for vision transformers with twin uniform quantization. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XII*, pp. 191–207, 2022. URL https://doi.org/10.1007/978-3-031-19775-8_12.

Cheng Zhang, Jianyi Cheng, George A. Constantinides, and Yiren Zhao. Lqer: low-rank quantization error reconstruction for llms. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24, 2024.

Cheng Zhang, Jeffrey T. H. Wong, Can Xiao, George Anthony Constantinides, and Yiren Zhao. QERA: an analytical framework for quantization error reconstruction. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=LB5cKhgOTu.

A Appendix

A.1 Rotation Invariance and Quantization-Error Analysis

Mathematical Analysis of Rotation Invariance The rotation proposed in Section 3.2 can preserve the floating-point mapping, and the orthogonality gives

$$\|XR(\boldsymbol{\theta})\|_F = \|X\|_F, \qquad \|R(\boldsymbol{\theta})^\top W\|_F = \|W\|_F. \tag{16}$$

Thus, rotation preserves the operand norms while changing their coordinatewise distributions.

For the signed symmetric weight quantizer defined in (6), consider a weight group u. Under nearest rounding, the coordinatewise quantization error is at most $s_w(u)/2$, and hence

$$\|Q_w(u) - u\|_2 \le \frac{\sqrt{g}}{2} s_w(u) = \frac{\sqrt{g}\max(\|u\|_\infty, \epsilon)}{2(2^{b_w-1}-1)} \tag{17}$$

where g is the group size. For an orthogonal group rotation R_g, the vector norm is unchanged and

$$\|uR_g\|_\infty \ge \frac{\|uR_g\|_2}{\sqrt{g}} = \frac{\|u\|_2}{\sqrt{g}}. \tag{18}$$

Consequently, a learned rotation can reduce coordinate peaks for the calibration distribution and thereby tighten the rounding-error bound while preserving the vector norm.

For asymmetric activation quantization, the relevant range includes zero, with $v_- = \min(\min_i v_i, 0)$ and $v_+ = \max(\max_i v_i, 0)$. The quantization step satisfies $(2^{b_a}-1)s_a(v) \ge v_+ - v_-$. When the values lie within the quantization range, nearest rounding contributes at most $\sqrt{g}s_a(v)/2$ in Euclidean error. Including zero in the range avoids unnecessary saturation for one-sided activation groups.

Quantization-Error Analysis. Let $E^X = X - \widehat{X}$ denote the error between full precision activation X and its quantized version $\widehat{X}$ propagated from the previous layer.

The complete expansion of the quantization error between quantized output $\widehat{Y}$ and full precision output Y is:

$$\widehat{Y} - Y = -E^X W + E^A R^\top W + XRE^W - E^X RE^W + E^A E^W. \tag{19}$$

The first term is the cross-layer error. The second and third terms are the local-layer activation- and weight-quantization error, respectively. Dropping the two product terms gives the first-order decomposition in (5). This approximation does not require the quantizers to be smooth.

Because R is orthogonal, the omitted terms satisfy

$$\left\|-E^X RE^W + E^A E^W\right\|_F \le \left(\|E^X\|_F + \|E^A\|_F\right)\|E^W\|_2, \tag{20}$$

A.2 Derivation of Cross-Layer Residual Compensation

This section derives the calibration statistics and initialization procedure in Section 3.3. To make the derivation easier to follow, we define some intermediate variables in this section:

$$\widehat{Z} = Q_a(\widehat{X}R), \qquad Y_{\text{loc}} = \widehat{X}W, \qquad T = E^X W. \tag{21}$$

Here W is the unquantized weight. For calibration batch m, define

$$\nu_m = \max(\text{mean}[(Y_{\text{loc},m})^2], \epsilon), \qquad \widetilde{Z}_m = \widehat{Z}_m/\sqrt{\nu_m}, \qquad \widetilde{T}_m = T_m/\sqrt{\nu_m}. \tag{22}$$

Furthermore, define accumulated calibration statistics

$$K = \sum_m \widetilde{Z}_m^\top \widetilde{Z}_m, \qquad C = \sum_m \widetilde{Z}_m^\top \widetilde{T}_m. \tag{23}$$

Thus, the regularized rank-constrained initialization objective in (12) can be further transformed as

$$\mathcal{J}_\rho(\Delta B) = \sum_m \|\widetilde{Z}_m \Delta B - \widetilde{T}_m\|_F^2 + \rho\|\Delta B\|_F^2 = \text{tr}(\Delta B^\top K_\rho \Delta B) - 2\,\text{tr}(\Delta B^\top C) + \text{Const}, \tag{24}$$

where $K_\rho = K + \rho I$ and $\rho > 0$. Completing the square gives

$$\mathcal{J}_\rho(\Delta B) = \|K_\rho^{1/2}\Delta B - K_\rho^{-1/2}C\|_F^2 + \text{Const.} \tag{25}$$

Because multiplication by the invertible matrix $K_\rho^{1/2}$ preserves rank, the rank-r initialization is

$$\Delta B^{(0)} = K_\rho^{-1/2}\left[K_\rho^{-1/2}C\right]_r, \tag{26}$$

where $[\cdot]_r$ denotes the rank-r truncated SVD approximation. The resulting matrix $\Delta B^{(0)}$ is used as the initialization of the SVD residual ΔB_{SVD}. Its two factors are initialized by splitting the retained singular values between U and V so that $UV = \Delta B^{(0)}$.

For the Haar Initialization, channels are sorted within each group by the Hessian of activations. To be mentioned, because K_{ii} records the accumulated normalized second moment of channel i, the sorting is performed by decreasing K_{ii}. Let Π be the within-group permutation and H be the block-diagonal orthogonal Haar wavelet transform, which maps each adjacent pair (a, b) to $(a + b)/\sqrt{2}$ and $(a - b)/\sqrt{2}$. Define

$$\mathcal{U} = H\Pi, \qquad K^{\mathcal{U}} = \mathcal{U}K\mathcal{U}^\top, \qquad C^{\mathcal{U}} = \mathcal{U}C. \tag{27}$$

Let $P_L, P_H \in \mathbb{R}^{d_{\text{in}}\times(d_{\text{in}}/2)}$ select the low- and high-frequency bands, respectively. We write the residual in the same coordinates as the Haar form in (14):

$$\Delta B = \mathcal{U}^\top\left(P_L\Delta B_L + P_H\Delta B_H\right), \tag{28}$$

with a separate rank-r constraint on each band. For $a, b \in \{L, H\}$, define

$$K_{ab}^{\mathcal{U}} = P_a^\top K^{\mathcal{U}} P_b, \qquad C_a^{\mathcal{U}} = P_a^\top C^{\mathcal{U}}, \qquad K_{a,\rho} = K_{aa}^{\mathcal{U}} + \rho I.$$

Starting with the high-frequency band set to zero, fit the low-frequency band using

$$\Delta B_L^{(0)} = K_{L,\rho}^{-1/2}\left[K_{L,\rho}^{-1/2}C_L^{\mathcal{U}}\right]_r. \tag{29}$$

Then update the high-frequency cross statistic and solve its conditional problem:

$$\widetilde{C}_H^{\mathcal{U}} = C_H^{\mathcal{U}} - K_{HL}^{\mathcal{U}}\Delta B_L^{(0)}, \qquad \Delta B_H^{(0)} = K_{H,\rho}^{-1/2}\left[K_{H,\rho}^{-1/2}\widetilde{C}_H^{\mathcal{U}}\right]_r. \tag{30}$$

This sequential procedure defines an initialization of the two Haar bands.

A.3 Training and Calibration Details

For masked-language-modeling, we use BERT-Large (Devlin et al., 2018) and the English Wikipedia corpus provided by the Hugging Face dataset (Wikimedia Foundation, 2026). We apply a deterministic document-level split to the downloaded snapshot, where 99% of the documents are used for training and the remaining 1% for validation. Training examples use dynamic masked-language-modeling with a masking probability of 0.15. Validation masks are generated deterministically from the example index and a fixed seed.

For ImageNet experiments, we use batch size 32, 500 training steps per block, and 50 warm-up steps. The learning rates are 1×10^{-5} for weights and 1×10^{-4} for rotation parameters. For CIFAR-100 experiments, we use 300 training steps per block and 25 warm-up steps, with a weight learning rate 3×10^{-6}. For BERT-Large, we use 50 training steps per block and 5 warm-up steps, with a weight learning rate 1×10^{-6}.

ImageNet uses its standard training and validation splits. For CIFAR-100, since we uniformly use ImageNet-pretrained models, we first adapt these models for 10 epochs by replacing the classification head. The learning rates for the backbone parameters and the head parameters are 1×10^{-5} and 5×10^{-4}, respectively. For Wikipedia, the deterministic 99%/1% document split described above is reused by every quantization variant.

For residual compensation, calibration examples are drawn only from the training split of the corresponding rotation stage experiment. We use 64 calibration batches and at most 16,384 activation rows per target layer on ImageNet, and 32 batches with at most 8,192 rows per target layer on CIFAR-100 and BERT-Large. An activation row denotes one flattened token or patch position used to accumulate regression statistics in residual compensation. Rotations and weights are frozen, and residual factors are optimized with AdamW with a learning rate of 5×10^{-5}.

A.4 ALGORITHM: TWO-STAGE OPTIMIZATION OF REVQ

Algorithm 1 Two-stage optimization of REVQ

Input: FP model M_0, QAT data $\mathcal{D}_{\text{qat}}$, calibration data $\mathcal{D}_{\text{cal}}$, quantization target blocks $\mathcal{B}$, group size g, selected target layers $\mathcal{S}$, budgets $T_{\text{rot}}, T_{\text{res}}$
Output: Rotation checkpoint M_{rot}, selected residual factors

Stage 1: Rotation-enhanced product-VQ QAT
Partition each input dimension into G groups of size g and initialize each group rotation to the identity.
for each target block $\ell \in \mathcal{B}$ **do**
  for $t = 1, \ldots, T_{\text{rot}}$ **do**
    Construct a separate $R_p = I_{G_p} \otimes R_{g,p}$ for each projection p in block ℓ using (8).
    Run the full precision reference pass and the current quantized network, then compute $\mathcal{L}_{\text{rot}}$ for the active block.
    Update active-block rotations and, for Rotation-QAT, designated projection weights.
  end for
end for
Save M_{rot} and freeze its rotations and projection weights.
Stage 2: Residual compensation
for each target block $\ell \in \mathcal{B}$ in network order **do**
  Collect paired full precision reference and current inputs, then initialize residuals at selected target layers in block ℓ following the selection criteria described in Section 3.3.
  for $t = 1, \ldots, T_{\text{res}}$ **do**
    Run quantized and full precision passes, adding residuals after quantized weight.
    Update active-block residual factors using optimization function provided in Section 3.4.
  end for
end for
return M_{rot} and residual factors in selected target linear layers.

A.5 Experiments on CNN Models

Table 7 reports CIFAR-100 classification results under different quantization and training configurations, including rank-1 SVD and Haar compensation for all four CNN models. For each CNN, we replace all convolutional and linear layers except the final classification head. Standard convolutions use the rotational quantization module, whereas depthwise and other grouped convolutions use a fallback quantizer without rotational transformation. Thus, the stem, backbone blocks, projection layers, and intermediate linear layers are quantized, while the final "fc" or "classifier" layer remains in full precision.

Across the four architectures evaluated under W4A4, training the rotations alone consistently achieves higher Top-1 accuracy than training the weights alone. On ResNet-18 and ResNet-50, both compensation variants further reduce the Top-1 accuracy gap relative to the full-precision models, with comparable performance between rank-1 SVD and Haar compensation.

Table 7: Main classification results on CNN models on CIFAR-100.

Bit-width W–A	Trainable		ResNet18		ResNet50		EfficientNet-V2		MobileNet-V2	
	Weight	Rotation	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
FP baseline	✗	✗	78.59	94.42	83.90	96.52	88.91	97.72	63.24	87.25
W16A4	✗	✗	75.74	92.96	77.51	93.72	86.86	97.27	57.64	84.10
W4A16	✗	✗	76.36	92.96	78.32	94.03	82.78	95.61	29.44	55.84
W4A4	✗	✗	72.06	90.62	73.64	91.51	78.84	94.21	25.89	51.63
W4A4	✓	✗	77.04	93.96	79.55	94.59	87.77	97.40	54.63	81.67
W4A4	✗	✓	77.86	**94.44**	81.50	95.45	88.23	97.48	54.84	81.91
W4A4	✓	✓	78.03	94.21	81.30	95.67	88.49	97.72	**56.71**	**82.97**
+ SVD ($r = 1$)	–	–	78.13	94.27	82.69	96.20	88.58	97.90	54.00	80.92
+ Haar ($r = 1$)	–	–	**78.24**	94.40	**82.71**	**96.22**	**88.62**	**97.96**	53.84	80.78

A.6 Experiments on BERT-Large

Table 8 provides the detailed results for BERT-Large on Wikipedia.

Table 8: BERT-Large masked-language-modeling results on Wikipedia.

W–A	Method	PPL ↓	ΔPPL	Token Acc. ↑	ΔAcc
W32A32	FP baseline	9.43	–	61.50	–
W4A4	Weight-Only	9.75	−0.32	60.69	−0.81
W4A4	Rotation-Only	11.65	−2.22	58.50	−3.01
W4A4	+ SVD (r=1)	11.01	−1.58	59.07	−2.43
W4A4	+ Haar (r=1)	10.99	−1.56	59.14	−2.36
W4A4	Ours (Rotation-QAT)	9.72	−0.29	60.72	−0.78
W4A4	+ SVD (r=1)	9.46	−0.03	60.99	−0.51
W4A4	+ Haar (r=1)	9.46	−0.03	61.02	−0.48

A.7 Downstream Detection and Segmentation

Table 9 provides the detailed COCO2017 results for Swin-T and Swin-S under the 3× schedule, including box and mask AP as well as AP at an IoU threshold of 0.5. We compare both rotation-based variants with full-precision references and existing W4A4 quantization methods. Both variants achieve higher box and mask AP than the listed quantization baselines on both backbones.

A.8 Ablation Analysis

Rotation Mechanism. Figure 4 compares three matched W4A4 settings, Tables 10 and 11 show more ablation results. "Weight-Only" optimizes the projection weights and keeps rotation frozen. "Rotation-Only" optimizes the groupwise rotations while keeping the weights fixed, whereas

Table 9: Comparison of object detection and instance segmentation results on COCO2017.

Model	W–A	Method	Box AP ↑	Mask AP ↑	Box AP-50 ↑	Mask AP-50 ↑
Swin-T (3×)	W32A32	Full-Precision (Liu et al., 2021)	0.460	0.416	0.682	0.653
	W4A4	PTQ4ViT (Yuan et al., 2022)	0.069	0.070	–	–
	W4A4	APQ-ViT (Ding et al., 2022)	0.237	0.226	–	–
	W4A4	RepQ-ViT (Li et al., 2023)	0.361	0.360	–	–
	W4A4	GPLQ (Liang et al., 2025)	0.401	0.389	–	–
	W4A4	Ours (Rotation-Only)	**0.455**	**0.413**	**0.677**	**0.646**
	W4A4	Ours (Rotation-QAT)	0.454	0.410	0.673	0.643
Swin-S (3×)	W32A32	Full-Precision (Liu et al., 2021)	0.482	0.432	0.699	0.671
	W4A4	PTQ4ViT (Yuan et al., 2022)	0.267	0.266	–	–
	W4A4	APQ-ViT (Ding et al., 2022)	0.447	0.401	–	–
	W4A4	RepQ-ViT (Li et al., 2023)	0.426	0.400	–	–
	W4A4	GPLQ (Liang et al., 2025)	0.434	0.413	–	–
	W4A4	Ours (Rotation-Only)	0.479	0.429	0.694	0.667
	W4A4	Ours (Rotation-QAT)	**0.480**	**0.430**	**0.695**	**0.668**

Table 10: Ablation on rotation mechanism for W4A4 quantization on ImageNet.

Dataset	W–A	Trainable		ViT-S		ViT-B		ViT-L		Swin-T		Swin-S	
		Weight	Rotation	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
ImageNet-1K	W4A4	✓	✗	79.92	**95.56**	83.73	96.89	85.70	**97.78**	80.40	95.11	82.20	95.73
	W4A4	✗	✓	79.57	95.25	84.27	**97.17**	85.53	97.73	80.56	**95.19**	82.57	95.99
	W4A4	✓	✓	**80.00**	95.55	**84.27**	97.16	**85.71**	97.75	**80.63**	95.12	**82.61**	**96.03**

"Rotation-QAT" optimizes both. Rotation-Only improves Swin-T and Swin-S but lowers ViT-S and ViT-L. Joint training removes these negative cases, retains the Swin gains, and increases the CIFAR-100 gain from approximately $+0.82$ to $+1.00$ points.

Selection of Target Interfaces. Table 5 evaluates where to attach a rank-1 Haar residual in ImageNet Swin-T. "All-20" activates all 20 valid interfaces in Swin-T. "Top-10" selects 10 interfaces by measuring the cross-layer error on the valid interfaces. "Deep-10" activates the 10 deepest interfaces in the model. Three policies use the same checkpoint, residual rank, optimizer, and update budget. The calibration data are split into disjoint fitting and evaluation subsets.

The two ΔErr columns measure gains on cross-layer error and local-layer error, respectively, which defined in (5). For policy p, let $\mathcal{S}_p$ denote its target compensation layer set. At layer j, let $E_{j,\text{before}}$ and $E_{j,\text{after}}$ be the squared errors accumulated over the calibration samples before and after compensation. We report

$$G_p^w = \frac{\sum_{j\in\mathcal{S}_p}(E_{j,\text{before}} - E_{j,\text{after}})}{\sum_{j\in\mathcal{S}_p} E_{j,\text{before}}} \times 100\%. \tag{31}$$

Cross-layer gain uses the full precision reference output, whereas local-layer gain uses the local reconstruction target. This aggregation weights each target layer's relative reduction by its pre-compensation error energy.

Each policy is evaluated over its own active layer set. The reported gains therefore compare relative reconstruction improvements on the selected layers, rather than total error reductions over a common set of interfaces.

All-20 reduces the aggregate deviation from full-precision outputs by 0.70%, while Top-10 and Deep-10 reach 1.89% and 2.06%, respectively. Since Top-10 and Deep-10 use the same number of residuals, the comparison shows that the propagated-error score does not outperform the simple depth rule in this experiment. Moreover, All three policies increase local-layer error, indicating that the residual corrects error accumulated from preceding layers. The final Top-1 and Top-5 differences are small and are auxiliary evidence rather than the basis for selecting interfaces.

Residual Initialization and Effect of Channel Ordering For Haar. A rank-r SVD residual on a $d_{\text{in}} \times d_{\text{out}}$ matrix uses $r(d_{\text{in}} + d_{\text{out}})$ trainable parameters. The two-band Haar residual uses approximately $r(d_{\text{in}} + 2d_{\text{out}})$ parameters when both bands have rank r. For a square layer of width

Table 11: Ablation on rotation mechanism for W4A4 quantization on CIFAR.

Dataset	W–A	Trainable		ViT-B		Swin-T		ResNet-18		ResNet-50		EffNet-V2		MobileNet-V2	
		Weight	Rotation	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
CIFAR-100	W4A4	✓	✗	92.00	99.17	87.66	**98.39**	77.04	93.96	79.55	94.59	87.77	97.40	54.63	81.67
	W4A4	✗	✓	90.83	98.99	86.68	98.01	77.86	**94.44**	**81.50**	95.45	88.23	97.48	54.84	81.91
	W4A4	✓	✓	**92.24**	**99.23**	**87.89**	98.38	**78.03**	94.21	81.30	**95.67**	**88.49**	**97.72**	**56.71**	**82.97**

d and rank 1, this is about $2d$ parameters for SVD and $3d$ for Haar, so equal-rank SVD and Haar are not equal-budget comparisons.

Table 6 isolates the channel order before the Haar transform on CIFAR-100 ViT-S. "None" does not use any ordering; "Importance" sorts channels by activation second moment before applying the same transform and residual fitting procedure. Both rows use the same Rotation-QAT checkpoint, group size, rank, calibration data, optimizer, and evaluation set. Shadow-output error measures the aggregate reduction in deviation from the full-precision interface outputs, as defined above. Improved is the percentage of evaluated interfaces for which this deviation decreases after compensation.

Importance ordering increases Top-1 from 87.84% to 87.92% and Top-5 from 98.20% to 98.26%. It changes the aggregate Shadow-output error from a small increase ($+0.00392\%$) to a small reduction (-0.00053%), while the fraction of improved interfaces rises from 73.9% to 87.0%. The ordering therefore makes Haar correction more consistently useful across interfaces in this setting. It does not establish a general performance ordering between Haar and SVD, because their parameter budgets differ.

A.9 Qualitative Detection and Masked-Language-Modeling Examples

COCO Detection Examples. Figure 5 presents qualitative COCO examples in a fixed top-to-bottom order. Each row compares the ground-truth annotations with direct W4A4 and Rotation-QAT W4A4 predictions. The displayed counts use score ≥ 0.3 and IoU ≥ 0.5 for visualization.

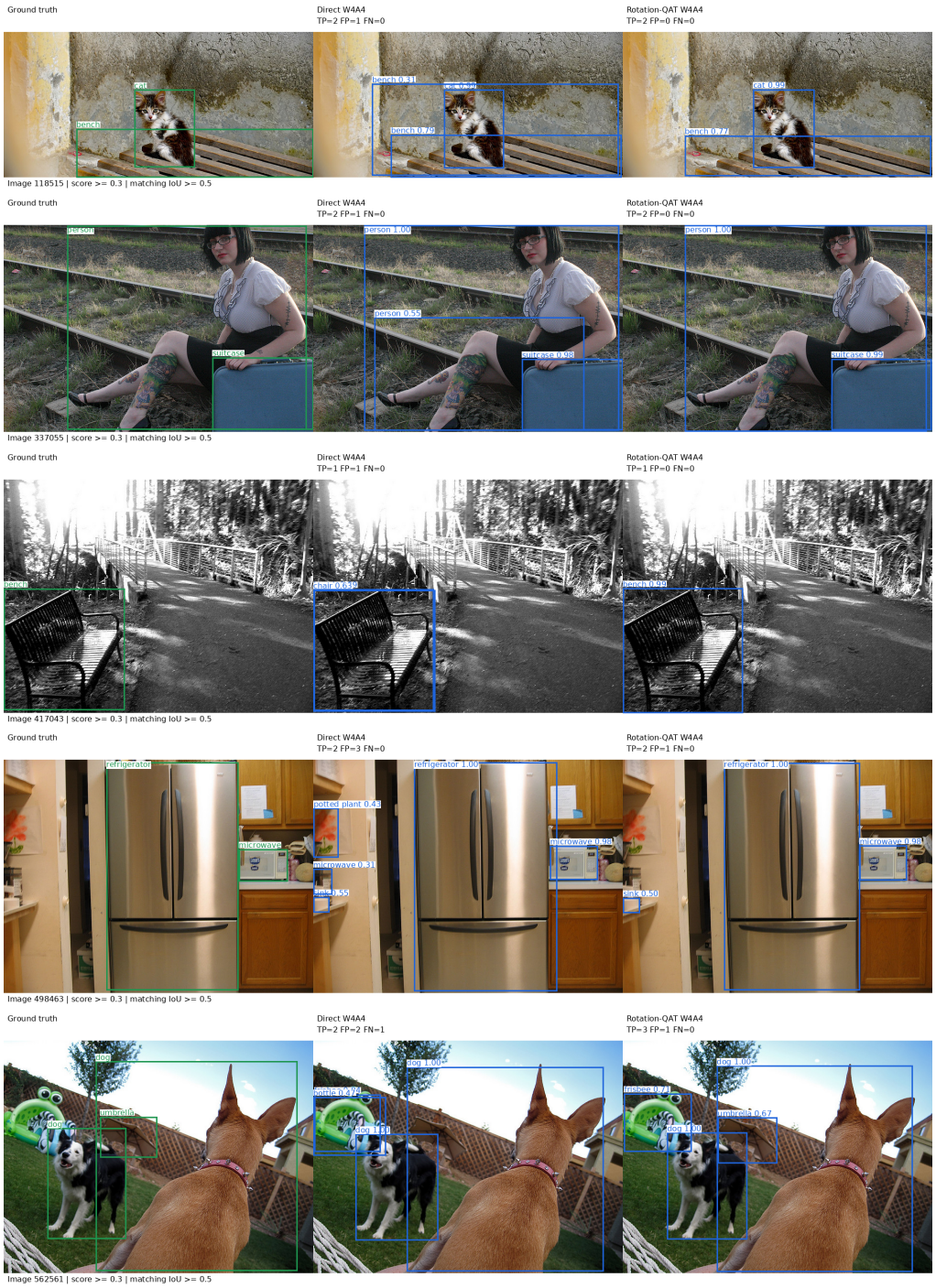


Figure 5: Qualitative COCO2017 detection examples. Each strip shows the ground truth, direct W4A4 predictions, and Rotation-QAT W4A4 predictions from left to right.

BERT-Large Masked-Language-Modeling Examples. The following text boxes transcribe representative Wikipedia MLM cases from the qualitative output. "Original" is the ground-truth token, while "Direct" and "Rotation-QAT" are the predictions under the two W4A4 configurations. Cases are grouped by whether Rotation-QAT improves, regresses, or leaves the prediction unchanged.

Improved cases.

Case 418. colin greenwood [MASK] his lyrics as "a running commentary on what is happening in the world..."
Original: described Direct: describes Rotation-QAT: described

Case 256. singer michael stipe, who gave him advice about how to [MASK] with fame.
Original: deal Direct: cope Rotation-QAT: deal

Case 106. the chief areas of his scientific interests are: history of dogmas, dogmatic [MASK], theology of god, philosophy of st.
Original: theology Direct: ##s Rotation-QAT: theology

Case 282. yorke and jonny greenwood contributed to the 2004 band aid 20 single "do they know it is christmas?", [MASK] by godrich.
Original: produced Direct: written Rotation-QAT: produced

Case 384. with the smile, yorke has [MASK] a fender mustang bass with a fingerstyle technique.
Original: used Direct: played Rotation-QAT: used

Case 204. yorke said he [MASK] he would become a rock star after seeing the queen guitarist brian may on television for the first time at the age of eight.
Original: knew Direct: thought Rotation-QAT: knew

Case 65. in 2013, william began to shift his career [MASK] to mainland china.
Original: focus Direct: back Rotation-QAT: focus

Case 296. pitchfork likened it to the eraser and wrote that yorke's vocals "[MASK] so perfectly that it feels like this is his band".
Original: work Direct: blend Rotation-QAT: work

Regressed and unchanged cases.

Case 198 (regressed). according to yorke, [MASK] last surgery was "botched", giving him a drooping eyelid.
Original: the Direct: the Rotation-QAT: his

Case 325 (regressed). the soundtrack was composed of field recordings [MASK] in the english countryside and played on speakers at different heights with different frequency ranges.
Original: made Direct: made Rotation-QAT: taken

Case 0 (unchanged). [MASK] socialist party (bolivia) (1970s)
Original: workers Direct: workers Rotation-QAT: workers