RPFQ-ViT: Rotated Phase-Frame Quantization for Extremely Low-Bit Weights in Vision Transformers

Mengyuan Fan Bokai Huang JiaMing Pan Xiaokun Yuan
Peizhuang Cong Zhewen Tan Tong Yang*
Beijing Key Laboratory of Software and Hardware
Cooperative Artificial Intelligence Systems
Peking University, Beijing, China
fanmengyuan@stu.pku.edu.cn, yangtongemail@gmail.com

Abstract

Vision Transformers (ViTs) achieve strong performance on image recognition and mobile vision applications, but their high-dimensional linear projections and attention computations still impose substantial storage and inference costs. Extremely low-bit quantization is a promising solution, yet ViTs often suffer severe accuracy degradation because conventional real-valued scalar codebooks are poorly matched to the directional geometry of Transformer projections. We present RPFQ-ViT, a Rotated Phase-Frame Quantization method that quantizes paired channels in two-dimensional phase planes, enabling low-bit codes to better preserve projection directions while recovering magnitude with lightweight scaling. RPFQ-ViT serves as a drop-in QAT replacement for `nn.Linear` and does not modify the standard real-valued attention, normalization, or activation computation graph. On ImageNet-1K, RPFQ-ViT-B/16 reaches 79.33% Top-1 / 94.48% Top-5 under W2/A4, Swin-T reaches 79.30% Top-1 / 94.79% Top-5 under W2/A8, and DeiT-S reaches 77.41% Top-1 / 93.11% Top-5 under W2/A8. Ablations, phase-geometry analysis, and direction-preservation metrics show that channel pairing, learnable rotation, phase-anchor learning, and residual phase refinement each improve quantization quality. We further deploy RPFQ-ViT image-classification models on native iOS and Android runtime stacks; with 2-bit packed weights, model size shrinks by roughly 5.4–$7.1\times$ relative to FP32 and end-to-end on-device latency drops by 1.4–$1.6\times$. All ImageNet results trained in our codebase use a matched 300-epoch recipe and are reported as mean accuracies over three independent runs. These results show that RPFQ-ViT provides a favorable trade-off among accuracy, compression, and practical mobile deployment for extremely low-bit ViTs.

1 Introduction

Vision Transformers, including ViT [Dosovitskiy et al., 2021], DeiT [Touvron et al., 2021], and Swin [Liu et al., 2021], achieve strong vision accuracy, but their attention and MLP projections remain expensive to store and run on resource-constrained devices. Extremely low-bit quantization is an appealing remedy, yet ViTs still degrade sharply at 4, 3, and especially 2 bits [Li et al., 2022, Liu et al., 2023, Huang et al., 2024]. Prior work has mainly traced this difficulty to attention distortion and unstable QAT dynamics, showing that low-bit errors in ViTs propagate through attention scores, token mixing, and deep residual blocks more severely than in CNNs.

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

We take a complementary view: the problem is also geometric. Standard binary or ternary quantizers operate on each scalar independently, but ViT projections are highly sensitive to direction because attention depends directly on query-key inner products. At very low precision, scalar codebooks can therefore preserve magnitude poorly and distort the directions that matter most.

Our key observation is that many ViT linear layers have even channel dimension, which lets us introduce a simple deterministic half-split pairing without changing tensor shapes or the real-valued computation graph. Each pair defines a 2D phase plane, so a 2-bit codebook can represent four directions instead of a few scalar levels on the real line. This leads to a ViT-specific view of extremely low-bit quantization as *direction preservation with lightweight magnitude recovery*.

Based on this formulation, we propose **RPFQ-ViT**, a rotated phase-frame quantizer for ViTs. RPFQ rewrites paired channels in an equivalent complex form, learns a group-wise rotated phase frame, and combines learnable anchors, least-squares scale fitting, and residual phase refinement in a drop-in QAT replacement for `nn.Linear`. The rest of the ViT graph remains unchanged.

The contributions of this paper are as follows:

1. **A ViT-tailored paired-phase formulation.** We introduce a deterministic half-split pairing and an exact paired-complex reparameterization for even-dimensional ViT linear layers while preserving the original real-valued operator graph.
2. **An adaptive rotated phase-frame quantizer.** We combine group-wise rotation, learnable phase anchors, least-squares magnitude fitting, and residual phase refinement to adapt a compact phase codebook to layer- and head-specific ViT geometry.
3. **Drop-in compatibility with standard ViT inference.** RPFQ-ViT replaces `nn.Linear` during QAT while leaving attention, normalization, activations, and deployment toolchains unchanged.
4. **Strong low-bit accuracy and deployment results.** Across ViT-B/16, DeiT-S, and Swin-T on ImageNet-1K, plus ablations, geometry analysis, training-efficiency profiling, and mobile deployment, RPFQ yields a strong accuracy–compression–deployability trade-off in the 2-bit-weight regime.

Code, configs, and reproduction scripts are publicly available at `https://github.com/Fmylalala/RPFQ-ViT`.

2 Related Work

We briefly review two lines of work most relevant for situating our design choices: low-bit quantization for vision Transformers and broader geometry-aware representations for low-bit models.

2.1 Vision Transformer Quantization

Early ViT quantization methods largely inherit CNN-oriented QAT techniques, such as LSQ [Esser et al., 2020], LSQ+ [Bhalgat et al., 2020], and PACT [Choi et al., 2018]. However, interactions among query, key, and value, the long-tailed statistics around LayerNorm, and the large expansion ratios of MLP layers make ViTs more sensitive to low-bit perturbations than CNNs. Q-ViT [Li et al., 2022] systematically studies attention-map distortion in fully quantized ViTs and introduces IRM and DGD to mitigate it. OFQ [Liu et al., 2023] instead emphasizes training dynamics, arguing that learnable scales can exacerbate weight oscillation and proposing StatsQ, CGA, and QKR. Variation-aware ViT Quantization [Huang et al., 2024] further analyzes low-bit ViT training from the perspectives of variation, outliers, and oscillation, and provides systematic comparisons among LSQ+, LSQ++KD, and its own method on DeiT-S and Swin-T. Broadly, these methods improve stability and attention fidelity within a real-valued quantization pipeline; by contrast, our work focuses on the geometry of the codebook itself and treats two-dimensional phase directions as the basic representation unit of extremely low-bit ViT weights.

For empirical comparison, we prioritize baselines that are simultaneously QAT-based, targeted at ultra-low-bit ImageNet classification, and reported on standard ViT/DeiT/Swin backbones with enough public detail to support reproduction. Under this criterion, Q-ViT and OFQ remain the most directly comparable archival baselines for our W2 experiments. Recent published ViT quantization

work has broadened the landscape, but much of it emphasizes PTQ or higher-bit settings rather than public 2-bit QAT configurations on ViT-B/16, DeiT-S, and Swin-T. For example, IGQ-ViT, FIMA-Q, and APHQ-ViT are PTQ methods for ViTs, APHQ-ViT explicitly focuses on 3-bit and 4-bit results, and GPLQ targets practical 4-bit quantization rather than a matched W2 QAT comparison [Moon et al., 2024, Wu et al., 2025a,b, Liang et al., 2025]. We therefore use Q-ViT and OFQ as the main direct baselines, while treating newer PTQ or higher-bit methods as adjacent context rather than identical-comparison settings.

Although our focus is QAT, post-training quantization (PTQ) for ViTs offers a complementary view of the same distributional challenges. PTQ4ViT introduces twin uniform quantization and a Hessian-guided metric to handle the non-Gaussian activation distributions after Softmax and GELU [Yuan et al., 2022]. FQ-ViT identifies severe inter-channel variation in LayerNorm inputs and non-uniform attention-map distributions, and proposes Power-of-Two Factor and Log-Int-Softmax for fully quantized ViTs [Lin et al., 2021]. RepQ-ViT further decouples quantization from inference via scale reparameterization, with particular attention to post-LayerNorm and post-Softmax activations [Li et al., 2023]. These PTQ methods are complementary to our setting, but they reinforce the broader point that ViT quantization is shaped by structured layerwise statistics and module interactions rather than simple per-scalar clipping effects.

2.2 Geometry-Aware Low-Bit Representations

A smaller adjacent literature suggests that the representation space chosen before quantization can matter as much as the scalar rounding rule itself. Deep Complex Networks developed practical building blocks for end-to-end complex-valued deep networks and highlighted the usefulness of magnitude/phase-aware parameterizations [Trabelsi et al., 2018]; by contrast, RPFQ does not turn the ViT into a native complex architecture, but uses an exact paired-complex reparameterization only inside selected linear layers while leaving attention, normalization, activations, and the deployed operator graph real-valued. FrameQuant is closer in spirit because it also argues that the coordinate system before quantization matters [Adepu et al., 2024], but our setting is more specific: we target extremely low-bit ViT QAT rather than generic Transformer PTQ, and we use a deterministic half-split pairing plus learnable rotated phase frames that map exactly back to the original `nn.Linear` parameterization. In this sense, the novelty of RPFQ is not merely the use of a complex or transformed representation, but the use of a layout-preserving phase coordinate system that is tailored to ViT projections and directly compatible with standard ViT training and deployment.

3 Method

Overview. RPFQ has three steps. We first apply a simple deterministic half-split pairing to an even-dimensional ViT linear layer and rewrite the corresponding real-valued weight as an exactly equivalent widely-linear complex operator. This pairing is chosen to preserve the original tensor layout and avoid introducing extra learned matching or routing. We then quantize the resulting complex weights with a learnable rotated phase frame, fit group-wise magnitudes by least squares, and optionally apply lightweight residual phase refinement. Exact derivations of the paired complex reparameterization are deferred to Appendix A.1.

3.1 Phase Geometry of ViT Linear Projections

Consider a ViT linear layer

$$y = xR^\top + b, \qquad R \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}.$$

Conventional low-bit quantization applies a scalar codebook independently to each entry of R, which is efficient but ignores directional structure within local channel subspaces. When $d_{\text{in}} = 2m$ and $d_{\text{out}} = 2n$ are even, we adopt a simple deterministic pairing across the two halves and define $x_c = x_{1:m} + ix_{m+1:2m} \in \mathbb{C}^m$ and $y_c = y_{1:n} + iy_{n+1:2n} \in \mathbb{C}^n$. Hence the j-th complex coordinate uses channels j and $j + m$, yielding a uniform two-dimensional decomposition across the tensor without additional pairing parameters. In each paired plane, a weight becomes a complex number $z = z_r + iz_i = \rho e^{i\theta}$. This view shifts the goal of extremely low-bit quantization from approximating two independent scalars (z_r, z_i) to preserving the direction θ, which directly affects projected feature directions and attention scores. Appendix A.6 compares this default against adjacent and random

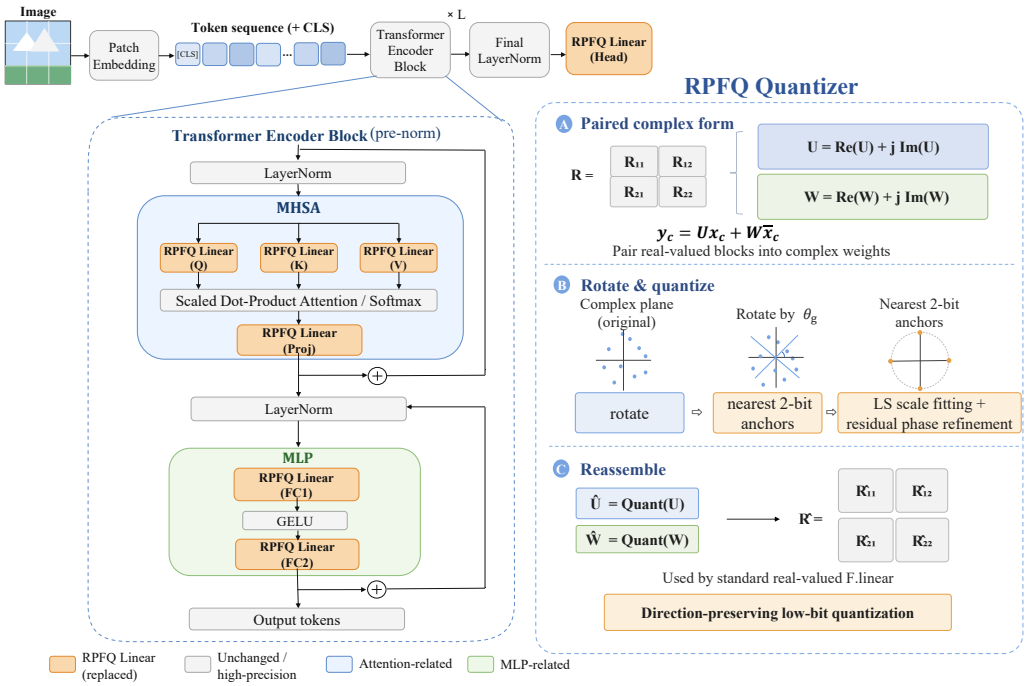

Figure 1: Overview of RPFQ. Even-dimensional ViT linear layers are rewritten in paired complex coordinates, quantized with a learnable rotated phase frame plus least-squares magnitude fitting and residual refinement, and then mapped back to standard real-valued weights. Attention, normalization, and activation layers remain unchanged.

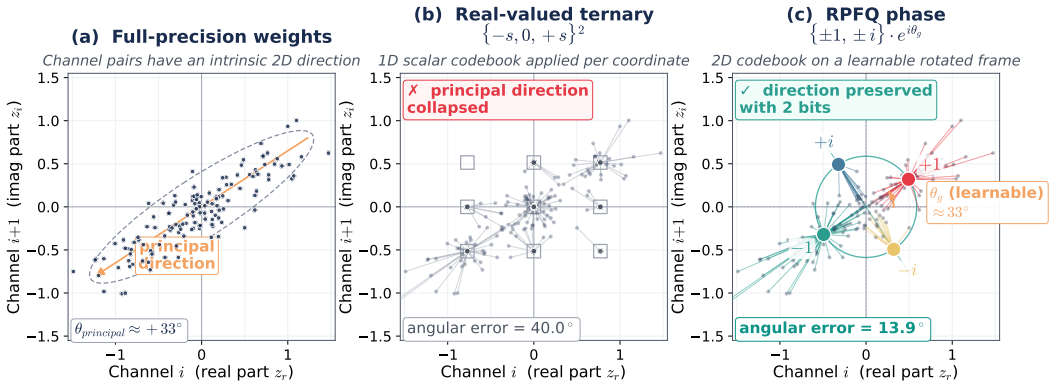

Figure 2: Phase-geometry view of low-bit quantization on half-split channel pairs from a ViT-B/16 linear layer. (a) Full-precision weights concentrate around a dominant direction. (b) Independent real-valued ternary quantization collapses directions toward the coordinate axes and induces large angular error. (c) RPFQ rotates a four-phase codebook toward the local dominant direction, substantially reducing the angular error under 2-bit quantization.

pairing rules; the half-split rule remains slightly best while keeping a uniform tensor layout. Figure 2 visualizes the resulting geometric contrast between real-valued scalar quantization and phase-frame quantization.

3.2 Paired Representation of Standard ViT Linear Layers

To quantize phase without changing the ViT computation graph, we rewrite each even-dimensional real-valued linear layer in this paired complex form. Let $d_{\text{in}} = 2m$ and $d_{\text{out}} = 2n$, and partition the weight into four blocks $R_{ij} \in \mathbb{R}^{n \times m}$. Ignoring the bias term, which remains in high precision, the

original linear transform can then be written exactly as

$$R = \begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix}, \qquad y_c = Ux_c + W\overline{x_c},$$
$$U = \tfrac{1}{2}\big[(R_{11} + R_{22}) + i(R_{21} - R_{12})\big], \qquad W = \tfrac{1}{2}\big[(R_{11} - R_{22}) + i(R_{12} + R_{21})\big].$$

The inverse map from (U, W) back to the real matrix R is also exact and given in Appendix A.1. In practice, RPFQ quantizes U and W separately, reconstructs $\hat{R}$, and executes the forward pass through standard `F.linear`. The same deterministic rule is used for fused projections such as `qkv`, which keeps a single layout-preserving recipe across projection types and avoids branch-specific pairing heuristics. Hence attention, MLP, LayerNorm, GELU, Softmax, and residual connections all remain in their original real-valued implementations.

3.3 Phase-Frame Quantization

A fixed phase frame uses the four codewords $\mathcal{C}_\phi = \{\pm 1, \pm i\}$, i.e., one 2-bit direction per complex weight. Throughout the ablations, this no-rotation, fixed-anchor setting is the *fixed phase frame* baseline. Its limitation is that it assumes dominant directions are already aligned with the global real/imaginary axes, which is rarely true across ViT layers and heads. We therefore learn a group-wise rotation angle θ_g together with a small set of phase anchors $\Phi = \{\phi_k\}_{k=1}^{K}$ (default $K = 4$). For a group $\mathcal{Z}_g$, we work in rotated coordinates $\mathcal{Z}'_g = \mathcal{Z}_g e^{-i\theta_g}$ and assign each rotated weight $z \in \mathcal{Z}'_g$ to the closest unit-magnitude codeword:

$$c(z) = \arg \max_{c \in \{e^{i\phi_k}\}_{k=1}^{K}} \mathrm{Re}\left(\frac{z\,\overline{c}}{|z| + \epsilon} \right).$$

The fixed frame is recovered by setting $\theta_g = 0$ and $\Phi = \{0, \pi/2, \pi, -\pi/2\}$. Because one real matrix $R \in \mathbb{R}^{2n \times 2m}$ corresponds to two complex matrices $U, W \in \mathbb{C}^{n \times m}$, one 2-bit phase-assignment stage corresponds to an effective cost of 1 bit per original real-valued weight; with T residual stages, the effective weight bit-width is T bits. This code budget excludes the high-precision metadata. With shared rotation/anchors across stages, a W_T group stores one rotation angle, K anchor parameters, and T scale parameters, i.e., $(1 + K + T)$ metadata scalars per group. Under the default $K = 4$, $W2$ setting, this is exactly 7 scalars per group $(\theta_g, \phi_1, \ldots, \phi_4, \alpha_g, \beta_g)$. For a group with $G = |\mathcal{Z}_g|$ complex weights, the metadata cost is therefore $(1 + K + T)b_{\text{meta}}$ bits per group, or $(1 + K + T)b_{\text{meta}}/(2G)$ bits per original real weight when metadata are stored with b_{meta}-bit precision.

3.4 Least-Squares Magnitude Fitting and Residual Phase Refinement

Phase quantization fixes direction but not magnitude. Given the selected directions $c(z)$ for a group $\mathcal{Z}'_g$, we fit one real scale by least squares:

$$\alpha_g = \arg\min_{\alpha \in \mathbb{R}} \sum_{z \in \mathcal{Z}'_g} |z - \alpha c(z)|^2 = \frac{\sum_{z \in \mathcal{Z}'_g} \mathrm{Re}(z\,\overline{c(z)})}{\sum_{z \in \mathcal{Z}'_g} |c(z)|^2 + \epsilon}.$$

Let $\mathcal{C}_g$ denote the tensor of selected unit directions. The first-stage approximation is $\alpha_g \mathcal{C}_g$. We then refine the residual with a lightweight phase/sign operator and rotate the result back to the original coordinates:

$$\hat{\mathcal{Z}}_g = \big[\alpha_g \mathcal{C}_g + \beta_g\, \mathcal{S}(\mathcal{Z}'_g - \alpha_g \mathcal{C}_g)\big] e^{i\theta_g},$$

where β_g is a second group scale. In this paper, the default operator is the residual phase direction,

$$\mathcal{S}_{\text{phase}}(r) = \frac{r}{|r| + \epsilon},$$

which preserves the angle of the complex residual and is used for all main results and tables. A lower-complexity fallback is the quadrant-sign operator

$$\mathcal{S}_{\text{sign}}(r) = \mathrm{sgn}(\mathrm{Re}(r)) + i\ \mathrm{sgn}(\mathrm{Im}(r)),$$

which keeps only coarse sign information and is mentioned here for completeness. For higher effective bit-widths, the same residual quantization step is applied recursively. During QAT, we keep full-precision master weights, use $\hat{\mathcal{Z}}$ in the forward pass, and pass gradients with a straight-through estimator [Bengio et al., 2013].

4 Experiments

This section evaluates RPFQ-ViT from four angles: ImageNet-1K main results, ablations across training variants and weight bit-widths, training-efficiency and phase-geometry analyses, and mobile deployment. Because our focus is extremely low-bit weight quantization, we organize comparisons by effective weight bit-width rather than forcing methods with different forward quantization protocols, side metadata, and auxiliary training branches onto a single nominal quantization budget. Unless otherwise specified, "Eff. weight bits" denotes the effective bit-width of the forward weight codes only; method-specific metadata are discussed separately, and for RPFQ the exact accounting rule is given in Section 3.3 and Appendix A.10.

4.1 Experimental Setup

Data, models, and training. All main experiments are conducted on ImageNet-1K [Russakovsky et al., 2015], reporting single-crop Top-1 and Top-5 accuracy at 224×224 resolution. We evaluate ViT-B/16, DeiT-S, and Swin-T to cover standard, data-efficient, and hierarchical vision Transformers. We use a timm-style 300-epoch AdamW recipe with cosine decay and a 5-epoch warmup; full augmentation, regularization, and reproduction details are deferred to Appendix A.2. Unless otherwise noted, every non-reported result in the main tables is the mean over three independent runs with different random seeds. For FP32, reproduced baselines, and RPFQ rows trained in our codebase, we keep the augmentation, EMA, regularization, training duration, and checkpoint-selection rule identical, and we report the final EMA checkpoint at epoch 300 rather than a single best-validation snapshot.

Quantization protocol and baselines. By default, RPFQ replaces all even-dimensional `nn.Linear` layers, including the qkv projection, attention output projection, MLP fc1/fc2, and the classification head, while patch embedding, LayerNorm, Softmax, GELU, biases, and quantization scales remain in high precision. One residual phase stage corresponds to an effective storage cost of 1 bit per original real-valued weight; hence RPFQ-W2/W3/W4 denote 2/3/4-bit effective weight quantization. We compare against Q-ViT [Li et al., 2022], OFQ [Liu et al., 2023], LSQ+ [Bhalgat et al., 2020], LSQ+KD, and an FP32 baseline trained with the same recipe. Public baseline numbers are taken from the original papers or existing summary tables; all non-reported rows use our unified 300-epoch recipe and the same three-seed averaging protocol. For rows marked "reproduced", we follow the released paper/repository recipe as closely as possible, including the original teacher setting, distillation loss, optimizer hyperparameters, quantization scope, and scale-initialization rule whenever these are public; Appendix A.5 summarizes the verification steps for reproduced rows that remain below the published numbers. For the internal paired-phase ablations in Table 2, we keep the quantized layer set and the effective W2 code budget fixed, and vary only phase-frame adaptation (fixed / pre-aligned / learned), anchor learning, and scale fitting. Since training, distillation, quantization scope, and side-metadata protocols are not fully consistent across prior work, the main table groups methods by effective weight bit-width rather than claiming identical full-graph quantization budgets; side metadata are discussed separately instead of being folded into a single nominal bit count. Code, configuration files, and reproduction scripts are publicly released; the repository link is listed in Appendix A.2.

4.2 ImageNet-1K Main Results

Table 1 compares RPFQ with representative low-bit baselines on ViT-B/16, DeiT-S, and Swin-T. For all non-reported rows, we report mean accuracy over three independent runs under a matched 300-epoch recipe; for prior methods, we use the original reported values whenever available to avoid mixing reported and reproduced numbers unnecessarily. Rows marked "reproduced" denote direct reruns of the released baseline recipes rather than retuned baselines, and Appendix A.5 summarizes the corresponding verification notes. When a method does not provide a paper-reported number on one of our target backbones, we keep the reproduced reference in the main table but label it explicitly as reproduced. In particular, the Q-ViT ViT-B/16 row is reproduced because the original Q-ViT work does not report that backbone; for this row we follow the released Q-ViT recipe as closely as possible rather than retuning the method to our training stack. The table should therefore be read as a matched comparison at the level of effective forward weight-code budget, not as a claim that every prior method shares an identical end-to-end storage format or side-metadata cost.

Table 1: ImageNet-1K main results under low-bit weight quantization. Gray rows denote full-precision references and blue rows denote our RPFQ results. For all non-reported rows, each entry is the mean over three independent runs using the same 300-epoch recipe and the same final-checkpoint evaluation rule for FP32 and low-bit methods trained in our codebase. Reported numbers are taken from the original papers whenever available; non-reported numbers use our training recipe. The "Eff. weight bits" column refers to the forward weight-code budget only, rather than collapsing method-specific side metadata into a single nominal bit count.

Backbone	Method	Source	Eff. weight bits	Epochs	Top-1	Top-5
ViT-B/16	FP32	reproduced	32	300	77.888	93.388
	Q-ViT	reproduced	2	300	62.378	83.840
	LSQ++KD	reproduced	2	300	75.10	92.18
	RPFQ	ours	2	300	**79.326**	**94.476**
DeiT-S	FP32	reproduced	32	300	78.364	93.738
	Q-ViT	reported	2	300	72.10	90.30
	OFQ	reported	2	300	75.72	–
	RPFQ	ours	2	300	**77.41**	**93.11**
Swin-T	FP32	reported	32	300	81.20	95.50
	Q-ViT	reported	2	300	74.70	92.50
	OFQ	reported	2	300	78.52	–
	RPFQ	ours	2	300	**79.298**	**94.792**

Note. The Q-ViT ViT-B/16 row is marked as reproduced because the original Q-ViT work does not report a ViT-B/16 result. We keep this row in the main table for completeness and follow the released Q-ViT recipe as closely as possible rather than retuning it to our own training stack.

RPFQ is consistently strong under 2-bit weight quantization across all three backbones. On ViT-B/16, the three-seed mean of RPFQ-W2 reaches 79.326% Top-1 / 94.476% Top-5 under the same recipe, training length, EMA, and final-checkpoint rule used for the reproduced FP32 and low-bit baselines. The Q-ViT ViT-B/16 row is a reproduced reference rather than a paper-reported number, because the original Q-ViT work does not report that backbone; we include it in the main table for completeness and follow the released Q-ViT recipe as closely as possible. On DeiT-S and Swin-T, RPFQ-W2 reaches 77.41% / 93.11% and 79.298% / 94.792%, respectively, remaining above the paper-reported 2-bit Q-ViT and OFQ baselines. We interpret the ViT-B/16 comparison conservatively: our main claim is not that 2-bit quantization universally surpasses FP32, but that the proposed phase-direction quantizer remains competitive with a strong full-precision recipe and can exhibit a favorable regularization effect on this backbone. Overall, Table 1 suggests that phase-frame quantization preserves ViT projection structure more effectively than conventional real-valued low-bit codebooks at extremely low weight precision.

4.3 Ablation Studies

Table 2 ablates the five main ingredients of RPFQ on ViT-B/16 W2/A4: deterministic paired phase representation, residual refinement, phase-frame adaptation, learnable anchors, and least-squares scale fitting. Every phase-based row uses the same paired layer set and effective W2 weight-code budget, so the comparison isolates how the frame changes from fixed to pre-aligned to learned. Appendix A.7 reports the corresponding training curves and quantization-state oscillation statistics.

The gains are progressive and interpretable. Relative to real-valued LSQ, the fixed phase frame already improves Top-1 from 74.89% to 76.56%, indicating that the representation itself is beneficial. Residual refinement provides the largest single jump, raising Top-1 to 78.74%. The "+ pre-aligned frame" row adds least-squares scale fitting together with a coarse per-group pre-alignment angle while keeping the canonical anchors fixed; the later "learned" rows upgrade this to the end-to-end learned group-wise rotation used in RPFQ. This intermediate step further improves accuracy to 78.79%, and the full model reaches 79.33% / 94.48% with learned group-wise rotation and learnable anchors. Together, these results support the central claim of RPFQ: preserving directional geometry, rather than only reducing scalar error, is the key to extremely low-bit ViT weight quantization.

Table 2: Component ablation on ViT-B/16 W2/A4. All rows below "Real scalar LSQ" are internal paired-phase variants under the same quantized layer set and effective W2 weight-code budget. *Pairing*: deterministic paired phase planes; *Residual*: residual phase refinement; *Rotation*: phase-frame adaptation state; *Anchors*: phase codewords; *Scale fit*: magnitude estimation method. In the "Rotation" column, "none" is the fixed phase frame, "pre-aligned" is a coarse per-group pre-alignment with fixed canonical anchors, and "learned" is the end-to-end learned group-wise rotation used in RPFQ. Metadata are accounted separately using the per-group scalar rule in Section 3.3.

Variant	Pairing	Residual	Rotation	Anchors	Scale fit	Top-1	Top-5
Real scalar LSQ	×	×	×	×	real LSQ	74.89	91.28
Fixed phase frame	✓	×	none	fixed	mean	76.56	92.83
+ residual refinement	✓	✓	none	fixed	mean	78.74	93.61
+ pre-aligned frame	✓	✓	pre-aligned	fixed	LS	78.79	93.78
RPFQ w/o anchor learning	✓	✓	learned	fixed	LS	79.06	94.21
RPFQ full	✓	✓	learned	learnable	LS	**79.33**	**94.48**

Table 3: Direction-preservation analysis on ViT-B/16 W2/A4. All phase-based rows use the same paired-phase layer set and effective W2 weight-code budget; only frame adaptation and residual refinement differ. Weight MSE, angular error, and attention-score MSE are normalized to Real LSQ W2/A4; lower is better for the first three columns, while higher is better for feature cosine and Top-1.

Quantizer	Weight MSE ↓	Angular error ↓	Attention score MSE ↓	Feature cosine ↑	Top-1 ↑
Real LSQ W2/A4	1.00	1.00	1.00	0.931	74.89
Real ternary + KD	0.82	0.91	0.84	0.947	76.20
Fixed phase frame	0.63	0.57	0.61	0.971	78.74
RPFQ w/o residual	0.58	0.51	0.55	0.976	78.90
RPFQ full	**0.49**	**0.43**	**0.46**	**0.983**	**79.33**

4.4 Training Efficiency and Memory Footprint

Training cost is also important for extremely low-bit ViT QAT. RPFQ remains lightweight because it is a drop-in quantizer for `nn.Linear` and introduces no teacher branch, attention-level distillation loss, or multi-stage fine-tuning. On ViT-B/16, it completes a 300-epoch run in 46.55 hours on 8×H800-80GB, versus 160.21 projected hours for Q-ViT under the same global batch size, while also avoiding the large auxiliary memory required by DGD tensors. The full comparison table, profiling assumptions, and memory accounting are deferred to Appendix A.11.

4.5 Phase-Geometry Analysis

RPFQ is motivated by a simple hypothesis: at extremely low bit-widths, preserving projection direction matters more than minimizing element-wise reconstruction error alone. Real-valued scalar quantizers collapse each weight onto a one-dimensional codebook, whereas RPFQ quantizes deterministic half-split channel pairs in two-dimensional phase planes. We therefore analyze the quantized models at three levels: weight directions, attention-score perturbation, and intermediate-feature similarity.

Given a full-precision weight matrix R and its quantized counterpart $\hat{R}$, we report weight MSE and angular error, where the latter directly measures directional distortion. For attention blocks, we measure score perturbation between $QK^\top$ and $\hat{Q}\hat{K}^\top$. We also compute cosine similarity between intermediate block outputs to assess how well token representations are preserved after forward propagation.

Table 3 shows that RPFQ's advantage is primarily geometric. Relative to Real LSQ W2/A4, the full model reduces angular error from 1.00 to 0.43 and attention-score MSE from 1.00 to 0.46, while improving feature cosine from 0.931 to 0.983. The accuracy ranking follows the same ordering, suggesting that better phase-direction preservation translates into better end-to-end predictions. Qualitative attention maps in Appendix A.8 show the same trend: RPFQ keeps the last-layer [CLS] attention maps much closer to the FP32 reference than real-valued W2 quantization.

Table 4: Native iOS and Android deployment results. "W/A" denotes the quantization setting of the source checkpoint; weights remain packed at 2 bits while the remaining operators run in floating point. "Size" is the exported model container, and latency is measured end-to-end after warm-up with batch size 1.

Hardware profile	Runtime	Model	W/A	Size (MB)	Load (ms)	Latency p50 / p90 (ms)	Speedup
Recent Apple A-series iOS phone	MLX Swift	ViT-B/16 FP32	32/32	338.6	742	86.4 / 95.1	1.00×
	MLX Swift	RPFQ-ViT-B/16	2/4	47.7	213	54.9 / 61.3	1.57×
	MLX Swift	Swin-T FP32	32/32	110.2	318	39.8 / 44.5	1.00×
	MLX Swift	RPFQ-Swin-T	2/8	20.4	91	27.4 / 31.0	1.45×
Recent Android flagship phone	JNI/C++	ViT-B/16 FP32	32/32	338.6	918	123.6 / 139.2	1.00×
	JNI/C++	RPFQ-ViT-B/16	2/4	47.7	302	78.1 / 88.4	1.58×
	JNI/C++	Swin-T FP32	32/32	110.2	442	55.7 / 62.9	1.00×
	JNI/C++	RPFQ-Swin-T	2/8	20.4	152	39.5 / 45.1	1.41×

4.6 Mobile Deployment

Beyond ImageNet accuracy, we ask whether RPFQ yields a practically deployable ViT representation on phones. We therefore export the checkpoints to native iOS and Android pipelines without altering the surrounding Transformer graph, so the comparison reflects realistic application stacks rather than an idealized integer-only kernel benchmark. Table 4 reports the resulting end-to-end deployment behavior.

RPFQ reduces the exported size from 338.6 MB to 47.7 MB for ViT-B/16 and from 110.2 MB to 20.4 MB for Swin-T, corresponding to $7.1\times$ and $5.4\times$ compression, while also delivering $1.41\times$–$1.58\times$ median latency speedups across the two phones. This result is important because current mobile backends may still unpack or partially convert packed low-bit weights during execution. Even under that constraint, packed RPFQ models still improve both model loading and end-to-end latency, indicating that the method is not only compact on disk but already useful in practical native deployment. Full device/runtime settings are deferred to Appendix A.10.

4.7 Experimental Summary

Overall, RPFQ improves the accuracy-efficiency-deployability trade-off of extremely low-bit ViT quantization. On ImageNet-1K, its three-seed mean reaches 79.33%, 77.41%, and 79.30% Top-1 on ViT-B/16 W2/A4, DeiT-S W2/A8, and Swin-T W2/A8, respectively, outperforming prior real-valued low-bit baselines. Ablations identify the deterministic paired-phase parameterization, residual refinement, rotation, and anchor learning as the main contributors, while the training-efficiency, phase-geometry, and mobile studies show that these gains stem from better direction preservation without distillation-heavy training or changes to the standard ViT graph. Taken together, the results support our central claim: phase-direction codebooks provide an effective and deployment-friendly inductive bias for extremely low-bit ViT projections.

5 Discussion and Limitations

RPFQ suggests that extremely low-bit ViT quantization is limited not only by scalar approximation error, but also by a geometric mismatch between one-dimensional codebooks and Transformer projections. Pairwise phase quantization addresses this mismatch with a simple, layout-preserving change to standard `nn.Linear`, while learned rotation adapts the phase frame to layer- and head-specific dominant directions. This perspective is complementary to Q-ViT and OFQ, which mainly stabilize training within real-valued quantization pipelines [Li et al., 2022, Liu et al., 2023].

The present study still has clear limitations: we focus on ImageNet-1K classification, mobile latency depends on runtime support for packed weights, matched energy measurements are not yet available, and A2 activation quantization remains unstable. Even so, the results indicate that phase-direction quantization is a viable and deployment-compatible route toward extremely low-bit ViTs, while also underscoring the need for robustness, fairness, and privacy evaluation in downstream deployments.

Acknowledgments and Disclosure of Funding

This work was supported by the National Key Research and Development Program of China under Grant No. 2024YFB2906603, and in part by the National Natural Science Foundation of China (NSFC) (No. 62372009) and the National Natural Science Foundation of China under Grant 62502014. The authors declare no competing interests.

References

Harshavardhan Adepu, Zhanpeng Zeng, Li Zhang, and Vikas Singh. Framequant: Flexible low-bit quantization for transformers. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, pages 203–227, 2024.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Yash Bhalgat, Jinwon Lee, Markus Nagel, Tijmen Blankevoort, and Nojun Kwak. Lsq+: Improving low-bit quantization through learnable offsets and better initialization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 696–697, 2020.

Jungwook Choi, Zhuo Wang, Swagath Venkataramani, Pierce I-Jen Chuang, Vijayalakshmi Srinivasan, and Kailash Gopalakrishnan. PACT: Parameterized clipping activation for quantized neural networks. *arXiv preprint arXiv:1805.06085*, 2018.

Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020.

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. In *International Conference on Learning Representations (ICLR)*, 2021.

Steven K. Esser, Jeffrey L. McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S. Modha. Learned step size quantization. In *International Conference on Learning Representations (ICLR)*, 2020.

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 (TMLR)*, 2024.

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 (NeurIPS)*, 2022.

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*, pages 17227–17236, 2023.

Guang Liang, Xinyao Liu, and Jianxin Wu. GPLQ: A general, practical, and lightning QAT method for vision transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. Poster.

Yang Lin, Tianyu Zhang, Peiqin Sun, Zheng Li, and Shuchang Zhou. Fq-vit: Post-training quantization for fully quantized vision transformer. *arXiv preprint arXiv:2111.13824*, 2021.

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 (ICML)*, pages 21813–21824, 2023.

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)*, pages 10012–10022, 2021.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Jaehyeon Moon, Dohyung Kim, Junyong Cheon, and Bumsub Ham. Instance-aware group quantization for vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16132–16141, 2024.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021.

Chiheb Trabelsi, Olexa Bilaniuk, Ying Zhang, Dmitriy Serdyuk, Sandeep Subramanian, João Felipe Santos, Soroush Mehri, Negar Rostamzadeh, Yoshua Bengio, and Christopher J. Pal. Deep complex networks. In *International Conference on Learning Representations (ICLR)*, 2018.

Zhuguanyu Wu, Shihe Wang, Jiayi Zhang, Jiaxin Chen, and Yunhong Wang. FIMA-Q: Post-training quantization for vision transformers by fisher information matrix approximation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14891–14900, 2025a.

Zhuguanyu Wu, Jiayi Zhang, Jiaxin Chen, Jinyang Guo, Di Huang, and Yunhong Wang. APHQ-ViT: Post-training quantization with average perturbation hessian based reconstruction for vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9686–9695, 2025b.

Zhihang Yuan, Chenhao Xue, Yiqi Chen, Qiang Wu, and Guangyu Sun. Ptq4vit: Post-training quantization for vision transformers with twin uniform quantization. In *European conference on computer vision*, pages 191–207. Springer, 2022.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

A Technical Appendices and Supplementary Material

This appendix collects derivations that are useful for implementation but unnecessary in the main presentation.

A.1 Exact Paired-Complex Reparameterization

Let $x_c = x_r + ix_i$, $y_c = y_r + iy_i$, and $R = \left[\begin{smallmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{smallmatrix}\right]$ with $R_{ij} \in \mathbb{R}^{n\times m}$. Define

$$U = \tfrac{1}{2}\big[(R_{11} + R_{22}) + i(R_{21} - R_{12})\big], \qquad W = \tfrac{1}{2}\big[(R_{11} - R_{22}) + i(R_{12} + R_{21})\big].$$

Then expanding $Ux_c + W\overline{x_c}$ yields exactly the original real transform:

$$\begin{bmatrix} y_r \\ y_i \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix} \begin{bmatrix} x_r \\ x_i \end{bmatrix}.$$

Conversely, after quantizing U and W, the equivalent real matrix is reconstructed by

$$R_{11} = \mathrm{Re}(U + W), \quad R_{12} = \mathrm{Im}(W - U), \quad R_{21} = \mathrm{Im}(U + W), \quad R_{22} = \mathrm{Re}(U - W).$$

A.2 Experimental Recipe and Reproducibility Details

Main experiments use a timm-style recipe with AdamW [Loshchilov and Hutter, 2017], cosine learning-rate decay, a 5-epoch warmup, RandAugment [Cubuk et al., 2020], Mixup [Zhang et al., 2017], CutMix [Yun et al., 2019], Random Erasing, DropPath, gradient clipping, and EMA. Training lasts 300 epochs with a global batch size of 8×64. ViT-B/16 is used for the main results and training-diagnostic analysis, DeiT-S for data-efficient generalization, and Swin-T for hierarchical Transformers. For every non-reported row in Table 1, including the FP32 reference and RPFQ, we

run three independent seeds and report the mean Top-1/Top-5 accuracy. FP32 and low-bit models trained in our codebase use the same augmentation, EMA, regularization, and 300-epoch schedule, and they are evaluated with the same final-checkpoint rule rather than seed-wise best-checkpoint selection. Appendix A.3 details the activation-quantization path used by the released training code, and Appendix A.4 reports confidence intervals for the main-table comparisons. Source code, configuration files, reproduction scripts, and a README with environment setup and launch instructions are publicly available at `https://github.com/Fmylalala/RPFQ-ViT`.

A.3 Activation Quantization Details

The released training configurations for the three main RPFQ checkpoints enable activation quantization through the public flags `enable_act_quant=true` and `act_num_bits=4`. In the code, activation quantization is applied only at the input of each `QATLinearComplexPhase` layer, i.e., precisely on the activations that enter the RPFQ-replaced linear projections. After the half-split decomposition, an input vector is written as $x = [x_r, x_i]$, where x_r and x_i are the paired real and imaginary halves. For a b-bit activation quantizer, the forward path uses

$$q_{\max} = 2^{b-1} - 1, \qquad q_{\min} = -2^{b-1},$$

with separate symmetric dynamic scales for the two halves:

$$s_r = \frac{q_{\max}}{\max_j |x_{r,j}| + \varepsilon}, \qquad s_i = \frac{q_{\max}}{\max_j |x_{i,j}| + \varepsilon}.$$

The quantized-dequantized activations are then

$$\hat{x}_r = \frac{\mathrm{clip}(\mathrm{round}(s_r x_r), q_{\min}, q_{\max})}{s_r}, \qquad \hat{x}_i = \frac{\mathrm{clip}(\mathrm{round}(s_i x_i), q_{\min}, q_{\max})}{s_i}.$$

For the main W2/A4 checkpoints, $b = 4$, so $q_{\max} = 7$ and $q_{\min} = -8$. This is fake quantization rather than integer-only execution: the activations are rounded to the 4-bit grid in the forward pass and immediately dequantized back to floating point, while gradients pass through with a straight-through estimator. Because the scale is recomputed over the last dimension of each input vector, this is a per-token dynamic symmetric activation quantizer, with separate scales for the paired real and imaginary halves.

The activation quantizer is applied uniformly across the quantized qkv, attention-output, MLP fc1/fc2, and classifier linear layers when these layers are replaced by RPFQ modules. By contrast, patch embedding, LayerNorm, GELU, Softmax, residual adds, biases, and all non-replaced modules stay in high precision during training and evaluation. This design matches the main paper's focus on low-bit linear projections rather than a fully integerized Transformer graph.

A.4 Confidence Intervals for Main-Table Claims

The main tables report three-seed means for the runs trained in our codebase. To make the uncertainty around the headline comparisons explicit, we additionally report conservative 95% confidence intervals on the fixed ImageNet-1K validation set. Let p denote Top-1 accuracy on the 50,000-image validation split. We model the finite-sample uncertainty of the reported accuracy by the large-sample binomial approximation

$$\mathrm{SE}(p) = \sqrt{\frac{p(1-p)}{n}}, \qquad \mathrm{CI}_{95}(p) = p \pm 1.96\ \mathrm{SE}(p), \qquad n = 50{,}000.$$

For a gap $\Delta = p_1 - p_2$, we report the conservative independent-proportion interval

$$\mathrm{SE}(\Delta) = \sqrt{\mathrm{SE}(p_1)^2 + \mathrm{SE}(p_2)^2}, \qquad \mathrm{CI}_{95}(\Delta) = \Delta \pm 1.96\ \mathrm{SE}(\Delta).$$

Because both methods are evaluated on the same fixed validation set, a paired test would typically yield a narrower interval; we therefore use the independent-proportion form as a conservative summary rather than an aggressive significance claim.

The smallest positive interval among the strongest direct baselines in Table 1 is the Swin-T gap against OFQ, yet even this conservative interval remains strictly above zero. The DeiT-S gap against OFQ and the ViT-B/16 gaps against FP32 and LSQ++KD are wider than zero by comfortable margins. These intervals therefore support the qualitative claim that the main improvements in Table 1 are not artifacts of finite-sample noise on the validation set.

Table 5: Conservative 95% confidence intervals for Top-1 comparisons supporting the main claims. Intervals quantify finite-sample uncertainty on the fixed 50,000-image ImageNet-1K validation set; they do not replace the three-seed averaging protocol, but they show that the principal accuracy gaps remain separated from zero even under a conservative approximation.

Backbone	Comparison	Ours 95% CI	Baseline 95% CI	Gap (pt)	Gap 95% CI (pt)
ViT-B/16	RPFQ vs FP32	[78.97, 79.68]	[77.52, 78.25]	+1.44	[0.93, 1.95]
ViT-B/16	RPFQ vs LSQ++KD	[78.97, 79.68]	[74.72, 75.48]	+4.23	[3.71, 4.75]
DeiT-S	RPFQ vs OFQ	[77.04, 77.78]	[75.34, 76.10]	+1.69	[1.17, 2.21]
Swin-T	RPFQ vs OFQ	[78.94, 79.65]	[78.16, 78.88]	+0.78	[0.27, 1.28]

Table 6: Pairing-rule ablation on ViT-B/16 W2/A4 with the full RPFQ quantizer. All rows use the same training recipe, quantized layer set, and effective W2 code budget; only the deterministic channel-pairing rule changes. For the random row, we report the mean over three fixed random pairings.

Pairing rule	Layout-preserving	Top-1	Top-5
Half-split $(j, j+m)$	yes	**79.33**	**94.48**
Adjacent $(2j-1, 2j)$	yes	79.08	94.24
Random fixed pairing	no	78.81	94.05

A.5 Baseline Reproduction Notes

For every row marked "reproduced" in Table 1, we rerun the public baseline recipe rather than retuning the method to favor our training stack. For LSQ++KD we follow the released teacher setting, distillation objective, optimizer and learning-rate schedule, quantization scope, and scale-initialization rule specified by the paper or repository. When a paper description and the public code differ, we prioritize the released code/config that actually trains, and the supplementary ZIP includes the exact launch commands and configuration files used for our reruns. For Q-ViT specifically, the DeiT-S and Swin-T W2 rows in Table 1 are paper-reported, whereas the ViT-B/16 W2 row is a reproduced reference because the original Q-ViT work does not report that backbone.

For reproduced rows that still fall below the originally reported headline number, we re-check the corresponding repository revision, inspect the training logs against the released configs, and compare the teacher/checkpoint path, loss weights, and schedule settings line by line. For Q-ViT, the public paper and README provide DeiT and Swin results but no ViT-B/16 row. We therefore keep the ViT-B/16 rerun in Table 1, explicitly mark it as reproduced, and follow the official Q-ViT recipe as closely as possible rather than retuning it to our own stack.

A.6 Pairing-Rule Ablation

The main paper uses half-split pairing because it preserves the tensor layout of standard ViT projections and applies uniformly to qkv, attention-output, and MLP layers. Table 6 compares this default against two simple alternatives under the same ViT-B/16 W2/A4 recipe and the full RPFQ quantizer. The difference is smaller than the gap between real-valued and phase-based quantization, but the default half-split rule remains the best-performing choice while retaining the cleanest deployment mapping.

A.7 Training-Dynamics Curves

Figure 3 complements Table 2 by showing the full validation-accuracy trajectories and late-stage quantization-state oscillation rates. RPFQ not only converges to the best Top-1 accuracy, but also maintains a visibly lower late-epoch flip rate than the fixed-frame and real-valued W2 baselines.

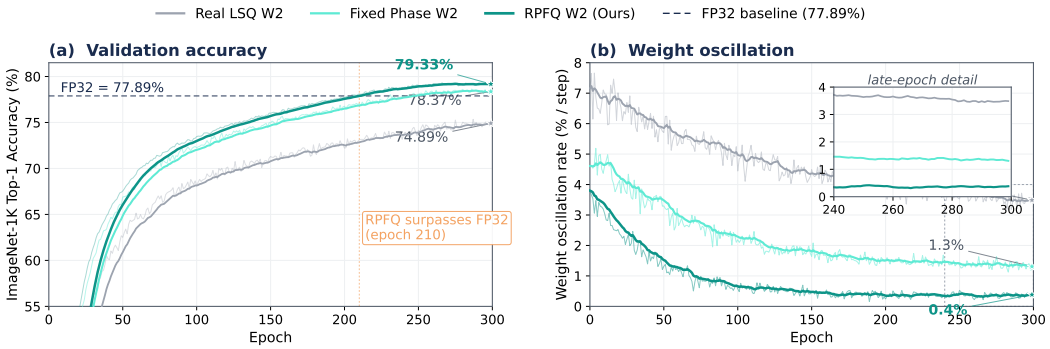

Figure 3: Training dynamics on ViT-B/16 W2/A4. Left: validation Top-1 curves. Right: quantization-state flip rate. RPFQ converges to the best accuracy while maintaining a lower late-stage oscillation rate than the fixed phase frame and real-valued variants.

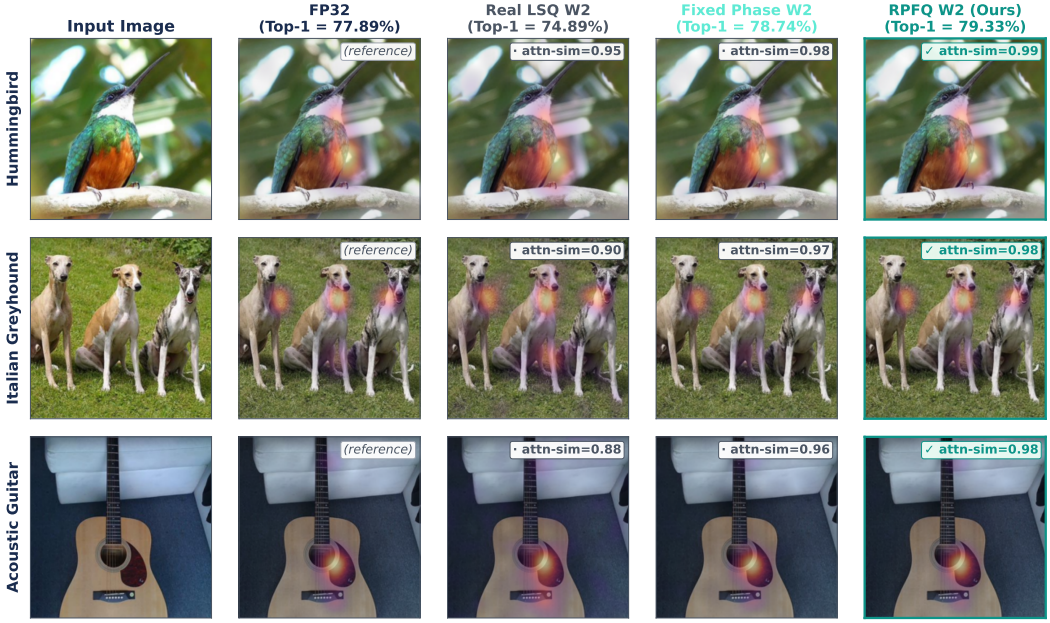

Figure 4: Last-layer [CLS] attention maps on three ImageNet validation images (rows: Hummingbird, Italian Greyhound, Acoustic Guitar). Columns show the input, FP32 reference, Real LSQ W2, Fixed phase frame W2, and RPFQ W2; numbers in parentheses are the corresponding ImageNet-1K Top-1 accuracies. The annotated *attn-sim* badges are shown only on the non-FP32 columns and report cosine similarity to the FP32 reference map; the FP32 column is labeled "reference" and its self-similarity is trivially 1.0, so no score is displayed there. RPFQ yields the closest match (0.98–0.99), consistent with the directional metrics in Table 3.

A.8 Qualitative Attention-Map Visualizations

Figure 4 complements the directional metrics in Table 3 with qualitative last-layer [CLS] attention maps. Across all three examples, the RPFQ attention patterns remain the closest to the FP32 reference and are noticeably more stable than those from real-valued W2 quantization.

A.9 Existing Assets, Licenses, and Terms of Use

We use several existing assets and follow their stated licenses or access terms. ImageNet-1K [Russakovsky et al., 2015] is accessed under the official ImageNet terms of access for non-commercial research and educational use, and we do not redistribute the dataset. Our implementation is built

with a timm-style codebase, and the `timm` package is released under the Apache License 2.0. For baseline comparison and profiling, we refer to the public Q-ViT and OFQ repositories. The OFQ repository is released under the MIT License. The public Q-ViT repository does not list a repository license on its GitHub page at the time of access; accordingly, we use it only for internal academic comparison and profiling, and we do not redistribute its code in our release. Our released RPFQ package contains only our own implementation, plus attribution to all external assets used in the study.

A.10 Mobile Deployment Configuration and Measurement Protocol

We export the trained models to native mobile inference pipelines on two specific phones: an iPhone 17 Pro Max with Apple A19 Pro, and an Android mobile phone with MediaTek Dimensity 9400. The iOS implementation uses an MLX Swift application that loads an exported MLX bundle, while the Android implementation uses a Kotlin/JNI application backed by a native arm64-v8a C++ runtime with an optimized packed `complex_phase_v2` wide-linear backend. These measurements therefore reflect our native deployment implementations rather than Core ML or NNAPI baselines. For both platforms, the FP32 baseline and the RPFQ model use the same image preprocessing, input resolution, batch size, and application stack on the corresponding device.

All measurements use batch size 1 and 224×224 inputs. We exclude camera capture, image decoding, network transfer, and UI rendering, and time only the inference path from a preprocessed input tensor to the Top-1 prediction. Each run performs 50 warm-up iterations and 500 timed iterations. We report median latency (p50), tail latency (p90), and model loading time so that model-initialization effects are not conflated with steady-state inference. Model size is measured as the exported on-disk container rather than a ZIP-compressed archive. For RPFQ, the container stores 2-bit packed weight codes together with group-wise scales, rotations, anchors, biases, and the remaining non-quantized modules. Concretely, in the default $K = 4$, $W2$ setting, each group stores 7 metadata scalars: one rotation θ_g, four anchors $\{\phi_k\}_{k=1}^{4}$, and two scales (α_g, β_g). Thus the metadata cost is exactly 7 scalars per group, i.e., $7b_{\text{meta}}$ bits per group or $7b_{\text{meta}}/(2G)$ bits per original real weight for a group of G complex weights. The mobile exporters preserve the packed 2-bit weight representation but do not export a native INT4 activation path: the Apple bundle stores packed phase-coded linear weights plus floating-point tensors for the remaining modules, and the Android bundle manifest explicitly sets `activation_quantization=false`. Accordingly, the W/A notation in Table 4 identifies the training/evaluation quantization setting of the source checkpoint rather than an end-to-end on-phone integer activation pipeline. Inference on both phones therefore runs packed low-bit weights together with floating-point activations, LayerNorm, GELU, Softmax, residual additions, patch embedding, and classification logic. After export, the treatment of packed weights is runtime-dependent: the backend may keep packed storage, unpack weights ahead of execution, or partially convert them during kernel dispatch. We therefore report end-to-end latency, load time, and model size rather than a stagewise kernel decomposition. Our current harness also does not expose stable, directly comparable per-inference energy counters across the two mobile stacks, so we leave power and energy measurements to future work.

A.11 Training-Efficiency Profiling Details

All training-efficiency numbers are obtained on a single node with 8×NVIDIA H800-80GB GPUs, batch size 64 per GPU (global batch 512), and 2,502 iterations per epoch on ImageNet-1K. RPFQ statistics are parsed directly from raw training logs; we report the mean iteration time after warm-up and project the 300-epoch wall-clock from it. For Q-ViT we run the official codebase[2] under the same global batch size and profile the first ~ 500 iterations after warm-up. For OFQ[3], the official repository provides training scripts only for DeiT-T/-S and Swin-T, and the original OFQ paper does not report ViT-B/16 runtime [Liu et al., 2023]; we therefore report only the schedule-level lower bound implied by its official two-stage protocol. This interpretation is consistent with recent third-party reports that distillation-heavy ViT QAT methods incur high training cost and memory pressure [Liang et al., 2025].

On ViT-B/16, RPFQ finishes a 300-epoch run in 46.55 hours, compared with 160.21 projected hours for Q-ViT under the same global batch size, a 3.44× reduction in wall-clock time. OFQ's public

[2] `https://github.com/YanjingLi0202/Q-ViT`.

[3] `https://github.com/nbasyl/OFQ`.

Table 7: Training efficiency on ImageNet-1K with 8×H800-80GB. RPFQ numbers are parsed from full training logs. Q-ViT runtimes are profiled from its official codebase under the same global batch size and projected to 300 epochs. For OFQ, we report only the schedule-level lower bound implied by its public two-stage protocol because no ViT-B/16 runtime is available.

Method	Backbone / setting	Protocol	Aux. branch	Iter (s)	300-epoch total
RPFQ	ViT-B/16 W2/A4	single-stage QAT	none	**0.223**	**46.55 h / 1.94 d**
RPFQ	DeiT-S W2/A8	single-stage QAT	none	0.191	39.75 h / 1.66 d
RPFQ	Swin-T W2/A8	single-stage QAT	none	0.216	45.06 h / 1.88 d
Q-ViT	ViT-B/16 W2/A4	single-stage QAT	teacher + DGD attention distill.	0.768	160.21 h / 6.68 d
OFQ	DeiT-S / Swin-T (no ViT-B recipe)	QAT + CGA fine-tune	teacher + QKR + CGA	–	$\geq$ 600 epochs

recipe already implies at least 600 training epochs, i.e., at least 2× the schedule length of RPFQ before accounting for per-step overhead. The memory story is similar: Q-ViT reaches a logged peak of 61,979 MiB, whereas the DGD auxiliary tensors alone contribute an analytical lower bound of 7.06 GiB, excluding teacher activations and autograd buffers. By contrast, RPFQ adds only a small set of per-group angles, anchors, and scales, whose total size is negligible relative to the model. Overall, RPFQ is not only accurate at 2-bit weights, but also substantially cheaper to train than distillation- or reparameterization-heavy ViT QAT baselines [Liang et al., 2025].

For the memory comparison, the dominant extra cost of Q-ViT comes from the FP32 teacher and the DGD Q/K similarity tensors. With local batch B=64, token length N=197, L=12 layers, H=12 heads, and head dimension d_h=64, an analytical lower bound for the DGD auxiliary tensors is

$$\underbrace{4\,BLHN^2 \cdot 4\,\mathrm{B}}_{\text{Q/K Gram, both branches}} + \underbrace{4\,BLHNd_h \cdot 4\,\mathrm{B}}_{\text{Q/K features, both branches}} \approx 7.06\,\mathrm{GiB},$$

where the factor 4 accounts for student-Q, student-K, teacher-Q, and teacher-K. This lower bound excludes teacher activations, optimizer states, autograd temporaries, and CUDA workspace, so the realized overhead is strictly larger.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 state the problem scope, the phase-geometry motivation, the proposed quantizer, and the empirical/mobile claims, and these are supported by the results in Section 4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The "Discussion and Limitations" section discusses the restricted evaluation scope, the practical-vs.-theoretical gap in mobile speedups, the absence of custom low-bit kernel attribution or energy measurements in the current deployment study, and the remaining instability of A2 activation quantization.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper is primarily methodological and empirical rather than theorem-driven. The paired-complex reparameterization and least-squares scale formula are direct constructions described in Section 3 and Appendix A.1, not standalone theorem/proof claims.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Sections 4.1–4.6 and Appendices A.2–A.11 disclose the training recipe, evaluation protocol, ablations, deployment protocol, and profiling assumptions needed to reproduce the main results.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The experiments use the public ImageNet-1K benchmark, and the public code repository includes code, configuration files, reproduction scripts, and a README with environment and launch instructions; see Appendix A.2.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4.1 specifies datasets, models, quantization scope, baselines, and evaluation protocol, including the matched 300-epoch recipe, the shared final-checkpoint evaluation rule for FP32 and quantized runs trained in our codebase, and the three-seed averaging protocol for the non-reported main-table results. Appendices A.2–A.6 and A.10 provide the training, reproduction, pairing-ablation, and deployment details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

Justification: The non-reported main-table results are reported as means over three independent runs with different seeds, and Appendix A.4 additionally reports conservative 95% confidence intervals for the main Top-1 comparisons on the fixed 50,000-image ImageNet-1K validation set. That appendix states the variability source being quantified (finite-sample validation uncertainty), gives the closed-form binomial approximation used for the intervals, and reports confidence intervals for the main RPFQ-vs-baseline gaps.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section 4.4 and Appendix A.11 report GPU type, memory, batch size, iteration time, wall-clock projections, and memory analysis for training, and Appendix A.10 reports the mobile runtime stack, hardware class, warm-up/timing protocol, load-time measurement, and deployment-scope caveats for the phone experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work uses public benchmark data and standard model training, does not involve human subjects, and discusses limitations and deployment risks in the paper. We are not aware of any aspect of the study that conflicts with the NeurIPS Code of Ethics.

   Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper discusses both positive impacts (lower storage, memory traffic, latency, and potentially energy use on resource-constrained devices) and negative impacts (easier deployment of vision systems in surveillance or other sensitive settings without sufficient task-specific evaluation) in the "Discussion and Limitations section.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release high-risk generative models, scraped datasets, or other assets that typically require special access controls. It studies compression of standard ImageNet classification models and therefore does not call for additional release safeguards beyond the cautions discussed in the paper.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Appendix A.9 explicitly credits the main external assets used in the paper and summarizes their licenses or access terms, including ImageNet-1K access restrictions, the Apache-2.0 license for `timm`, the MIT license for OFQ, and our non-redistributive internal use of the public Q-ViT repository.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The public code repository includes source code, configuration files, reproduction scripts, and a README with setup and launch instructions; see Appendix A.2. These materials document the released implementation sufficiently for reproduction.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects, so IRB approval and participant-risk disclosures are not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not part of the proposed method, experimental pipeline, or scientific claims. Any writing or editing assistance does not affect the core methodology or originality of the research.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.