GLF-Q: GLOBAL-LOCAL FEATURE-BASED QUANTIZATION FOR VISION TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Post-training quantization (PTQ) efficiently compresses Vision Transformers (ViTs) without retraining, yet suffers severe accuracy degradation at low bit-widths. Existing optimization-based PTQ methods guide block reconstruction via either soft logits or second-order Hessian proxies. Logit supervision is prone to overfitting on limited calibration data, while Hessian approximations incur structural truncation errors. To address these limitations, we propose **GLF-Q**, a novel PTQ framework guided by Global-Local Feature alignment. GLF-Q propagates quantized block outputs through downstream full-precision layers to align penultimate-layer representations under local output regularization, providing downstream feature supervision without explicitly approximating the Hessian or using a Taylor expansion. Furthermore, offline Hadamard transformations are introduced with zero runtime overhead to disperse activation outliers across channels, effectively contracting dynamic ranges and reducing quantization errors. Meanwhile, optimizing this loss via a Straight-Through Estimator (STE) achieves rapid convergence, bypassing continuous relaxation rounding formulations such as AdaRound. Extensive experiments across representative ViT architectures demonstrate that GLF-Q with standard uniform quantizers substantially outperforms state-of-the-art methods under 3-bit quantization on image classification. In addition, GLF-Q exhibits strong out-of-domain calibration robustness and achieves speedups under 8-bit GPU deployment.

1 INTRODUCTION

Vision Transformers (ViTs) (Dosovitskiy et al., 2020) have achieved remarkable success across diverse visual recognition tasks, such as image classification (Touvron et al., 2021), as well as object detection and instance segmentation (He et al., 2017; Cai & Vasconcelos, 2018; Liu et al., 2021), owing to their powerful capability in modeling long-range dependencies. However, their massive model sizes, intensive self-attention computations, and heavy memory bandwidth demands severely limit practical deployment on resource-constrained edge devices. Among various model compression techniques, network quantization has attracted widespread attention, as it converts floating-point weights and activations into low-bit representations, thereby significantly reducing memory footprints and substantially accelerating hardware inference simultaneously.

Existing quantization approaches can be broadly divided into Quantization-Aware Training (QAT) and Post-Training Quantization (PTQ). Although QAT can effectively recover model accuracy through retraining, it requires full labeled data and expensive retraining, severely restricting its practical deployment. In contrast, PTQ requires only a small amount of unlabeled data for calibration, making it a practical and efficient approach.

Depending on whether optimization is required, existing PTQ methods primarily fall into two categories: calibration-only methods (Yuan et al., 2022; Li et al., 2023; Ding et al., 2022) and optimization-based methods (Nagel et al., 2020; Li et al., 2021; Liu et al., 2023; Wu et al., 2025a;b; Hwang et al., 2026). Calibration-only methods adjust quantization parameters solely during the calibration phase. However, in low-bit settings, quantization errors increase substantially, leading to severe accuracy degradation. To overcome this limitation, optimization-based methods perform block-wise reconstruction to compensate for quantization loss, emerging as the prevailing paradigm for ViT PTQ.

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.

In block-wise reconstruction, the choice of supervision signal directly affects the accuracy of the quantized model. Existing reconstruction approaches can be broadly classified into two categories based on their supervisory signals. The first category comprises logit-guided methods (Liu et al., 2023), which seek global supervision by minimizing the prediction difference on output soft logits at the task head. However, fitting output logits on a small calibration set can lead to overfitting and limit generalization, as shown in Appendix D.

Hessian-guided reconstruction (Li et al., 2021; Wu et al., 2025a;b; Hwang et al., 2026) uses second-order loss approximations to weight reconstruction errors according to their estimated impact on the task loss. Since computing the full Hessian is expensive, these methods employ tractable Hessian approximations, such as diagonal or low-rank representations. The accuracy of the estimated loss change depends on both the Hessian approximation and the second-order Taylor approximation. Under low-bit quantization, larger perturbations may make higher-order effects more relevant and reduce the accuracy of the second-order approximation.

These limitations motivate evaluating feature differences directly through the full-precision downstream network, without explicitly constructing a Hessian approximation or truncating a Taylor expansion. We combine penultimate-layer feature alignment with local reconstruction regularization to improve generalization on limited calibration data.

Motivated by these insights, we propose GLF-Q, an efficient post-training quantization framework for Vision Transformers based on Global-Local Feature alignment. GLF-Q introduces a *Global-Local Feature (GLF) loss*, which combines penultimate feature alignment with local reconstruction regularization. Furthermore, to address the critical challenge of extreme activation outliers and inter-channel scale disparities in ViTs, we systematically introduce *offline Hadamard transformations* (Ashkboos et al., 2024b) into the ViT PTQ pipeline for the first time. This cheap operation evenly distributes energy concentrated in outlier channels across all dimensions, contracting activation dynamic ranges to effectively suppress quantization errors with zero runtime overhead. In addition, regarding the reconstruction optimization mechanism, GLF-Q avoids continuous rounding relaxation methods such as AdaRound (Nagel et al., 2020). Instead, it uses a basic Straight-Through Estimator (STE) (Bengio et al., 2013) to achieve fast and stable convergence, thereby improving reconstruction efficiency.

Our main contributions are summarized as follows:

- We propose the Global-Local Feature (GLF) loss, which combines global feature alignment with local reconstruction regularization to mitigate calibration overfitting and avoid explicit Hessian approximations.
- To mitigate activation outliers, we introduce offline Hadamard transformations to spread large activation values across channels, reducing quantization errors. For reconstruction optimization, we use a simple STE instead of continuous rounding relaxation to achieve fast convergence.
- We demonstrate practical hardware efficiency and strong out-of-domain generalizability. GLF-Q achieves higher accuracy and lower inference latency under practical 8-bit GPU deployment. It also maintains strong accuracy under out-of-domain calibration.

2 Related Work

Model quantization (Gholami et al., 2022) converts floating-point weights and activations into low-bit representations to reduce memory usage and accelerate inference. Existing quantization methods can be classified into Quantization-Aware Training (QAT), which requires retraining, and Post-Training Quantization (PTQ), which does not. Although QAT methods (Esser et al., 2019; Li et al., 2022; Liang et al., 2026) achieve competitive accuracy, their reliance on full labeled data and expensive retraining severely limit practical deployment.

In contrast, PTQ calibrates models using only a small set of unlabeled data without retraining, making it an efficient and popular approach. Depending on whether optimization is involved, PTQ methods broadly fall into calibration-only methods (Yuan et al., 2022; Ding et al., 2022; Li et al., 2023; Wu et al., 2024; Zhong et al., 2024; Moon et al., 2024; Fu et al., 2025; Jiang et al., 2026) and

optimization-based methods (Nagel et al., 2020; Li et al., 2021; Wei et al., 2022; Liu et al., 2023; Zhong et al., 2023; Yang et al., 2024; Ma et al., 2024; Wu et al., 2025b;a; Hwang et al., 2026).

Calibration-only methods determine quantization parameters directly during a forward calibration pass without backpropagation. RepQ-ViT (Li et al., 2023) uses scale reparameterization to convert complex activation quantizers into hardware-friendly forms for efficient inference. AdaLog (Wu et al., 2024) addresses heavy-tailed activations via dynamic log-base tuning paired with a progressive search strategy. UQ-ViT (Jiang et al., 2026) introduces DeMax and NormQuant to accommodate extreme activation distributions with uniform quantizers. Despite these efforts, calibration-only methods suffer substantial accuracy loss in low-bit settings, while those employing specialized non-uniform formats remain difficult to deploy on general hardware.

To recover accuracy, optimization-based methods perform block-wise reconstruction to reduce quantization loss. BRECQ (Li et al., 2021) optimizes weight rounding using second-order error analysis across residual blocks. QDrop (Wei et al., 2022) randomly drops activation quantization during reconstruction to enhance model flatness and noise tolerance. PD-Quant (Liu et al., 2023) minimizes prediction differences on soft classification logits for global guidance. DopQ-ViT (Yang et al., 2024) handles activation outliers during reconstruction via LayerNorm input reparameterization and specialized non-uniform quantizers. APHQ-ViT (Wu et al., 2025b) couples an average perturbation Hessian loss with MLP reconstruction. FIMA-Q (Wu et al., 2025a) constructs a composite Hessian loss by modeling both diagonal and low-rank cross-channel dependencies. LS-ViT (Hwang et al., 2026) formulates Hessian proxy estimation as a least-squares problem over calibration gradients, achieving rapid reconstruction with single-pass backpropagation. Despite these advances, existing reconstruction methods still face limitations. Logit-guided methods can overfit limited calibration data, while Hessian-guided methods rely on approximations that may become less accurate under low-bit quantization.

3 METHOD

3.1 PRELIMINARIES AND MOTIVATION

Logit-Guided Block Reconstruction. In block-wise reconstruction, parameter optimization relies heavily on the supervisory signal. To incorporate global task guidance, logit-guided methods, represented by PD-Quant (Liu et al., 2023), minimize the prediction difference on output soft logits at the final task head. However, fitting output logits on a small calibration set can lead to overfitting and limit generalization.

Hessian-Guided Block Reconstruction. Let θ denote the parameters of the pretrained model and $\Delta\theta$ be the quantization perturbation. The change in task loss can be approximated via a second-order Taylor expansion:

$$\mathbb{E}\left[\mathcal{L}(\theta+\Delta\theta)\right]-\mathbb{E}\left[\mathcal{L}(\theta)\right]\approx\mathbb{E}\left[\Delta\theta^{\top}g^{(\theta)}+\frac{1}{2}\Delta\theta^{\top}H^{(\theta)}\Delta\theta\right], \tag{1}$$

where $g^{(\theta)}=\nabla_{\theta}\mathcal{L}$ is the gradient and $H^{(\theta)}=\nabla^{2}_{\theta}\mathcal{L}$ is the Hessian matrix. Since a well-converged model satisfies $g^{(\theta)}\approx 0$, existing reconstruction-based PTQ methods map parameter perturbations to the module output perturbation Δy via the chain rule:

$$\mathcal{L}_{\mathrm{H}}(\Delta y)=\frac{1}{2}\Delta y^{\top}H^{(y)}\Delta y, \tag{2}$$

where $H^{(y)}=\nabla^{2}_{y}\mathcal{L}$ is the Hessian matrix with respect to the module output y. Compared with vanilla MSE, $\mathcal{L}_{\mathrm{H}}$ weights quantization errors according to directional task sensitivity.

Because computing the dense matrix $H^{(y)}$ is intractable, existing methods commonly adopt diagonal, low-rank, or Fisher matrix approximations (Li et al., 2021; Wu et al., 2025a;b; Hwang et al., 2026). However, the accuracy of these proxies depends on the second-order Taylor approximation and the structural assumptions used to estimate curvature. Under low-bit quantization, larger perturbations may reduce the accuracy of the local quadratic approximation, motivating direct measurement of downstream feature differences.

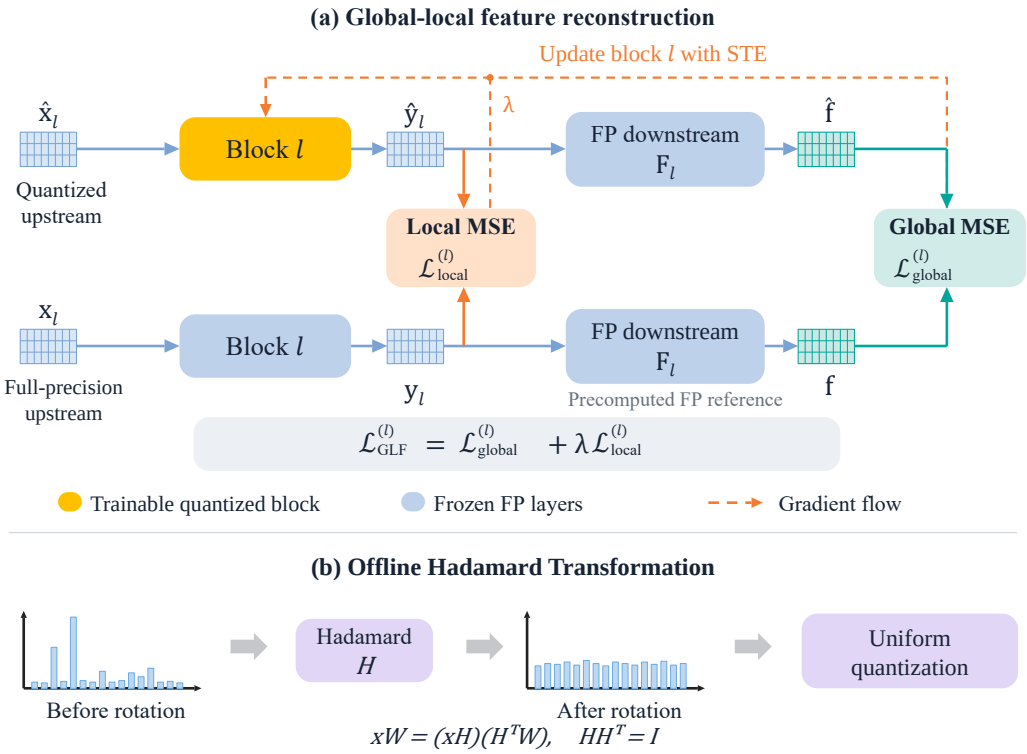

Figure 1: Overview of the proposed GLF-Q framework. (a) Global-local feature reconstruction. For block l, the quantized block output $\hat{y}_l$ is propagated through the frozen downstream full-precision network F_l to match the precomputed full-precision reference feature f, under local reconstruction regularization. Gradients are backpropagated directly to update the quantized block via STE. (b) Offline Hadamard transformation. Orthogonal transformations are fused offline into adjacent linear weights to suppress activation outliers and balance channel distributions with zero runtime overhead.

3.2 Global-Local Feature Loss

To address the two limitations discussed above, we introduce the Global-Local Feature (GLF) loss, illustrated in Figure 1(a). Specifically, the GLF loss directly leverages global penultimate-layer feature representations (Wang et al., 2021) to evaluate downstream quantization loss under local regularization.

When reconstructing block l, preceding quantized blocks are frozen, with their cached activations serving as input $\hat{x}_l$. We cache the global feature $f \in \mathbb{R}^C$ and local block output $y_l \in \mathbb{R}^{N\times C}$ on $\mathcal{D}_{\text{cal}}$ using the full-precision teacher after offline transformations. During reconstruction, the dequantized block output $\hat{y}_l \in \mathbb{R}^{N\times C}$ is dynamically computed from $\hat{x}_l$ and fed into the frozen downstream full-precision suffix F_l:

$$\hat{f} = F_l(\hat{y}_l), \tag{3}$$

where $\hat{f} \in \mathbb{R}^C$ denotes the global penultimate representation immediately preceding the task head. For image classification, f is the final class token representation for ViT and DeiT, and the globally averaged token representation for Swin. The global feature loss is formulated as

$$\mathcal{L}^{(l)}_{\text{global}} = \mathbb{E}_{x\sim\mathcal{D}_{\text{cal}}}\left[\frac{1}{C}\left\|\hat{f} - f\right\|_2^2\right]. \tag{4}$$

This loss measures the feature differences caused by quantization after propagation through the full-precision downstream network. Concurrently, we introduce the local feature loss as a regularizer:

$$\mathcal{L}^{(l)}_{\text{local}} = \mathbb{E}_{x\sim\mathcal{D}_{\text{cal}}}\left[\frac{1}{NC}\left\|\hat{y}_l - y_l\right\|_F^2\right]. \tag{5}$$

The GLF loss for block l is defined as

$$\mathcal{L}_{\mathrm{GLF}}^{(l)} = \mathcal{L}_{\mathrm{global}}^{(l)} + \lambda \mathcal{L}_{\mathrm{local}}^{(l)}, \tag{6}$$

where $\lambda > 0$ is the regularization coefficient controlling the strength of local regularization. In practice, each loss term is normalized by its value on the first reconstruction minibatch, with the normalization factors fixed throughout optimization.

During optimization of block l, the downstream suffix F_l remains frozen, yet gradients backpropagate through F_l via STE to update this block's parameters θ_l and quantization scale s_l. Once reconstructed, block l is frozen in its quantized state, and the pipeline advances sequentially to the next block.

3.3 GLF-Q Framework

As shown in Figure 1 and Algorithm 1 in Appendix C, GLF-Q follows a block-wise quantization pipeline. First, we apply offline Hadamard transformations to mitigate activation outliers with zero runtime overhead. We then perform MLP reconstruction by replacing GELU with strictly non-negative ReLU to expand the effective dynamic range. Finally, we carry out block reconstruction guided by the GLF loss.

Offline Hadamard Transformation. Directly quantizing Vision Transformers remains challenged by channel scale disparities and activation outliers (Darcet et al., 2024; Sun & Wu, 2026), which widen the dynamic range and increase quantization errors. To disperse activation outliers across channels without additional inference overhead, we apply an offline Hadamard transformation (Ashkboos et al., 2024b) prior to reconstruction, as depicted in Figure 1(b). Specifically, let $M = I - \frac{1}{C}\mathbf{1}\mathbf{1}^\top \in \mathbb{R}^{C\times C}$ denote the mean-subtraction projection matrix, where I is the identity matrix and $\mathbf{1} \in \mathbb{R}^C$ is the all-ones vector, such that the centered activation satisfies $x_{\mathrm{c}} = xM$. Following SliceGPT (Ashkboos et al., 2024a), both the mean centering and affine operations of LayerNorm are absorbed into adjacent linear layers:

$$\mathrm{LN}(x)W + b = \mathrm{RMSNorm}(x_{\mathrm{c}})\widetilde{W} + \widetilde{b}, \tag{7}$$

where the preceding layer absorbs mean-centering:

$$\widetilde{W}_{\mathrm{prev}} = W_{\mathrm{prev}}M, \qquad \widetilde{b}_{\mathrm{prev}} = b_{\mathrm{prev}}M, \tag{8}$$

and the subsequent layer absorbs the scaling γ and bias β:

$$\widetilde{W} = \mathrm{Diag}(\gamma)W, \qquad \widetilde{b} = \beta W + b. \tag{9}$$

This transformation converts LayerNorm into standard RMSNorm. We then apply a randomized Hadamard transform R_1 to rotate the residual stream:

$$xW = (xR_1)(R_1^\top W), \tag{10}$$

where $R_1 R_1^\top = I$. Both R_1 and $R_1^\top$ are fused into weights offline before quantization, distributing activation outliers across channels. For multi-head self-attention, head-wise orthogonal rotations are applied to QK and VO projections to eliminate outliers without cross-head interference. Full network equivalence, coordinate alignment, and generalized Kronecker constructions for non-2^k dimensions are detailed in Appendix B.

MLP Reconstruction. For nonlinear activations, the asymmetric negative tail of GELU wastes the dynamic range of low-bit quantizers and causes quantization loss. Following APHQ-ViT (Wu et al., 2025b), before block reconstruction, we conduct an MLP reconstruction step. We replace GELU with ReLU in the MLP module to avoid allocating quantization levels to negative values, and then reconstruct the MLP to reduce the changes caused by activation replacement.

Efficient Block Reconstruction via STE. Unlike conventional reconstruction methods that introduce continuous relaxation variables and require tens of thousands of iterations, such as AdaRound (Nagel et al., 2020), we directly optimize full-precision weights θ_l and quantization scales s_l using STE (Bengio et al., 2013) under the GLF loss in Eq. (6). This direct formulation converges within only 3000 iterations, significantly accelerating reconstruction while achieving superior low-bit accuracy.

Table 1: Top-1 accuracy comparison across representative Vision Transformers on ImageNet. "*" denotes results reimplemented using the FIMA-Q framework, as QDrop and PD-Quant were originally designed for CNNs. "Opt." indicates whether a PTQ method is optimization-based (✓) or calibration-only (×). "Spec." indicates reliance on a specialized non-uniform quantizer (✓) versus a standard uniform quantizer (×). The best results within each bit-width setting are highlighted in **boldface**.

Method	Opt.	Spec.	W/A	ViT-S	ViT-B	DeiT-T	DeiT-S	DeiT-B	Swin-S	Swin-B
Full-Prec	-	-	32/32	81.39	84.54	72.21	79.85	81.80	83.23	85.27
PTQ4ViT (Yuan et al., 2022)	×	✓	3/3	0.10	0.10	3.50	0.10	31.06	28.69	20.13
RepQ-ViT (Li et al., 2023)	×	✓	3/3	0.10	0.10	0.10	0.10	0.10	0.10	0.10
AdaLog (Wu et al., 2024)	×	✓	3/3	13.88	37.91	31.56	24.47	57.47	64.41	69.75
I&S-ViT (Zhong et al., 2023)	✓	✓	3/3	45.16	63.77	41.52	55.78	73.30	74.20	69.30
DopQ-ViT (Yang et al., 2024)	✓	✓	3/3	54.72	65.76	44.71	59.26	74.91	74.77	69.63
QDrop* (Wei et al., 2022)	✓	×	3/3	41.05	74.75	46.88	50.95	72.97	74.67	76.57
PD-Quant* (Liu et al., 2023)	✓	×	3/3	40.52	75.72	53.23	60.72	74.54	74.59	76.71
APHQ-ViT (Wu et al., 2025b)	✓	×	3/3	63.17	76.31	55.42	68.76	76.31	76.10	78.14
FIMA-Q (Wu et al., 2025a)	✓	×	3/3	64.09	77.63	55.55	69.13	76.54	77.26	78.82
LS-ViT (Hwang et al., 2026)	✓	×	3/3	64.10	77.65	55.72	69.41	76.57	77.39	79.40
GLF-Q (Ours)	✓	×	3/3	**67.40**	**78.18**	**56.48**	**71.55**	**77.20**	**78.10**	**80.45**
PTQ4ViT (Yuan et al., 2022)	×	✓	4/4	42.57	30.69	36.96	34.08	64.39	76.09	74.02
APQ-ViT (Ding et al., 2022)	×	✓	4/4	47.95	41.41	47.94	43.55	67.48	77.15	76.48
RepQ-ViT (Li et al., 2023)	×	✓	4/4	65.05	68.48	57.43	69.03	75.61	79.45	78.32
ERQ (Zhong et al., 2024)	×	✓	4/4	68.91	76.63	60.29	72.56	78.23	80.74	82.44
IGQ-ViT (Moon et al., 2024)	×	✓	4/4	73.61	79.32	62.45	74.66	79.23	80.98	83.14
AdaLog (Wu et al., 2024)	×	✓	4/4	72.75	79.68	63.52	72.06	78.03	80.77	82.47
UQ-ViT (Jiang et al., 2026)	×	×	4/4	68.34	72.07	59.71	72.13	76.59	79.93	81.96
I&S-ViT (Zhong et al., 2023)	✓	✓	4/4	74.87	80.07	65.21	75.81	79.97	81.17	82.60
DopQ-ViT (Yang et al., 2024)	✓	✓	4/4	75.69	80.95	65.54	75.84	80.13	81.71	83.34
QDrop* (Wei et al., 2022)	✓	×	4/4	71.84	82.63	65.27	72.64	79.96	81.21	82.99
PD-Quant* (Liu et al., 2023)	✓	×	4/4	72.98	82.72	66.23	74.98	79.90	81.28	82.92
OASQ (Ma et al., 2024)	✓	×	4/4	72.88	76.59	66.31	76.00	78.83	81.02	82.46
APHQ-ViT (Wu et al., 2025b)	✓	×	4/4	76.07	82.41	66.66	76.40	80.21	81.81	83.42
FIMA-Q (Wu et al., 2025a)	✓	×	4/4	76.68	83.04	66.84	76.87	80.33	81.82	83.60
LS-ViT (Hwang et al., 2026)	✓	×	4/4	76.67	83.08	67.05	76.89	80.41	81.82	83.62
GLF-Q (Ours)	✓	×	4/4	**77.82**	**83.22**	**67.37**	**77.34**	**80.53**	**82.01**	**83.80**

4 EXPERIMENTS

We evaluate GLF-Q on image classification, object detection, and instance segmentation. We further conduct ablation studies, assess out-of-domain calibration robustness, and compare inference latency and training time. Additional experiments are provided in the appendix.

4.1 EXPERIMENTAL SETUP

Datasets and Models. For image classification, we evaluate on ImageNet (Deng et al., 2009) across ViT (Dosovitskiy et al., 2020), DeiT (Touvron et al., 2021), and Swin (Liu et al., 2021). For object detection and instance segmentation, we conduct experiments on COCO (Lin et al., 2014) using Mask R-CNN (He et al., 2017) and Cascade Mask R-CNN (Cai & Vasconcelos, 2018) with Swin backbones. All pretrained full-precision Vision Transformers are obtained from the `timm` library[1], while pretrained detection and segmentation models are sourced from MMDetection (Chen et al., 2019).

Implementation Details. We apply standard channel-wise uniform quantization to weights and layer-wise uniform quantization to activations, including Softmax outputs, as detailed in Appendix A. Following reconstruction-based PTQ methods (Li et al., 2021; Wei et al., 2022; Wu et al., 2025b;a), we randomly sample 1024 unlabeled images from ImageNet for image classification and 256 unlabeled images from COCO for object detection and instance segmentation as the calibration sets. We set the batch size, learning rate for activation quantization, learning rate for tuning weight, and reconstruction iterations as 32, 2e-4, 2e-5, and 3000, respectively. The regularization

[1] `https://github.com/huggingface/pytorch-image-models`

Table 2: Object detection and instance segmentation performance on COCO with Mask R-CNN and Cascade Mask R-CNN under 4-bit quantization.

Method	Opt.	Spec.	W/A	Mask R-CNN				Cascade Mask R-CNN			
				Swin-T		Swin-S		Swin-T		Swin-S	
				AP^b	AP^m	AP^b	AP^m	AP^b	AP^m	AP^b	AP^m
Full-Precision	-	-	32/32	46.0	41.6	48.5	43.3	50.4	43.7	51.9	45.0
PTQ4ViT (Yuan et al., 2022)	×	✓	4/4	6.9	7.0	26.7	26.6	14.7	13.5	0.5	0.5
APQ-ViT (Ding et al., 2022)	×	✓	4/4	23.7	22.6	**44.7**	40.1	27.2	24.4	47.7	41.1
RepQ-ViT (Li et al., 2023)	×	✓	4/4	36.1	36.0	44.2	40.2	47.0	41.1	49.3	43.1
ERQ (Zhong et al., 2024)	×	✓	4/4	36.8	36.6	43.4	40.7	47.9	42.1	50.0	43.6
I&S-ViT (Zhong et al., 2023)	✓	✓	4/4	37.5	36.6	43.4	40.3	48.2	42.0	50.3	43.6
DopQ-ViT (Yang et al., 2024)	✓	✓	4/4	37.5	36.5	43.5	40.4	48.2	42.1	50.3	43.7
QDrop* (Wei et al., 2022)	✓	×	4/4	36.2	35.4	41.6	39.2	47.0	41.3	49.0	42.5
APHQ-ViT (Wu et al., 2025b)	✓	×	4/4	38.9	38.1	44.1	41.0	**48.9**	**42.7**	50.3	43.7
FIMA-Q (Wu et al., 2025a)	✓	×	4/4	38.7	37.8	44.2	**41.1**	48.7	42.5	50.4	43.7
GLF-Q (Ours)	✓	×	4/4	**39.6**	**38.2**	44.3	**41.1**	48.8	42.6	**50.6**	**44.0**

coefficient λ in Eq. (6) is set to 2 across all experiments. For detection and segmentation, the global term averages the normalized bbox and mask RoI feature losses. For Swin, where hierarchical patch merging hinders exact RMSNorm conversion, we retain LayerNorm without residual Hadamard, pairing head-wise QK/VO Hadamard rotations with LayerNorm input reparameterization (Yang et al., 2024). More implementation details are provided in Appendix G.

4.2 Experiments on Image Classification

We first evaluate the performance of our method on the classification task on ImageNet in terms of Top-1 accuracy, compared to the state-of-the-art PTQ approaches. We report results across various representative Transformer architectures, including ViT, DeiT, and Swin, under 4-bit and 3-bit settings.

As displayed in Table 1, GLF-Q achieves strong performance across diverse architectures, with particularly significant gains under 3-bit quantization. Concretely, for 4-bit quantization, while some existing methods suffer noticeable accuracy degradation, our method maintains robust performance across all architectures. Under the challenging 3-bit quantization, the performance of competing methods degrades severely, with methods such as PTQ4ViT (Yuan et al., 2022) and RepQ-ViT (Li et al., 2023) suffering severe accuracy degradation. In comparison, our proposed GLF-Q maintains robust accuracy, outperforming the second-best approach by 3.30% on ViT-S.

It is worth noting that many compared approaches such as PTQ4ViT (Yuan et al., 2022), RepQ-ViT (Li et al., 2023), AdaLog (Wu et al., 2024), and DopQ-ViT (Yang et al., 2024) attempt to boost performance by designing specific quantizers, which however are generally difficult to implement on hardware in practice. In contrast, our method relies strictly on a standard uniform quantizer while achieving superior accuracy, facilitating hardware-friendly deployment on general-purpose platforms.

4.3 Experiments on Object Detection and Instance Segmentation

We evaluate 4-bit quantization on COCO (Lin et al., 2014) using Mask R-CNN (He et al., 2017) and Cascade Mask R-CNN (Cai & Vasconcelos, 2018) with Swin backbones. As shown in Table 2, GLF-Q achieves competitive performance across object detection and instance segmentation, obtaining the best results on several model configurations while using standard uniform quantization. These results demonstrate the effectiveness of GLF-Q on dense prediction tasks.

Table 3: Top-1 accuracy comparison (%) with different quantization losses and an additional baseline on ImageNet under 3-bit quantization. †: FIMA-Q + MR + Hadamard with AdaRound, 20k iterations.

Setting	ViT-S	ViT-B	DeiT-T	DeiT-S	DeiT-B	Swin-S	Swin-B
LF	60.48	77.06	53.16	70.16	76.76	76.66	78.46
DPLR-FIM (Wu et al., 2025a)	63.83	77.15	54.94	70.67	76.60	77.30	79.19
Logits + LF (Liu et al., 2023)	65.16	76.89	55.26	70.71	76.87	76.80	78.58
GF	66.09	77.17	54.06	70.41	76.69	77.75	80.30
GLF (Ours)	**67.40**	**78.18**	**56.48**	**71.55**	**77.20**	**78.10**	**80.45**
FIMA-Q + MR + Had.†	66.88	76.13	55.29	69.81	76.24	77.37	79.42

Table 4: Top-1 accuracy comparison (%) with different framework components on ImageNet under 3-bit quantization. "GLF" denotes Global-Local Feature loss, "MR" denotes MLP Reconstruction, and "Hadamard" denotes Offline Hadamard Transformation.

Method	ViT-S	ViT-B	DeiT-T	DeiT-S	DeiT-B	Swin-S	Swin-B
GLF only	60.37	72.50	55.28	68.75	75.79	76.92	79.31
GLF + MR	63.53	73.60	55.24	69.00	75.79	77.96	80.09
GLF + Hadamard	65.04	77.16	55.69	71.30	76.91	77.63	79.81
GLF-Q	**67.40**	**78.18**	**56.48**	**71.55**	**77.20**	**78.10**	**80.45**

4.4 Ablation Studies

On different quantization losses. We compare five reconstruction losses within the same 3-bit framework using STE for 3,000 iterations: local feature loss (LF), DPLR-FIM from FIMA-Q (Wu et al., 2025a), Logits + LF from PD-Quant (Liu et al., 2023), global feature loss (GF), and GLF in Eq. (6). We additionally evaluate FIMA-Q (Wu et al., 2025a) with MLP reconstruction and offline Hadamard transformations using AdaRound for 20,000 iterations, reported separately in Table 3. As shown in Table 3, GLF achieves the highest mean accuracy of 72.77%, exceeding DPLR-FIM, Logits + LF, and GF by 1.38%, 1.30%, and 0.98%, respectively. With MR and Hadamard, FIMA-Q achieves 71.59% mean accuracy using AdaRound for 20k iterations, 1.17% below GLF-Q using STE for 3k iterations.

On main components in GLF-Q. To evaluate the contribution of main components in our framework, we conduct an ablation study on MLP Reconstruction and Offline Hadamard Transformation under 3-bit quantization on ImageNet. As shown in Table 4, the full combination achieves the highest accuracy across all evaluated architectures. Adding MR to GLF improves accuracy on five of the seven architectures, with a slight decrease on DeiT-T and no change on DeiT-B. Adding Hadamard transformations improves accuracy across all seven architectures.

4.5 Experiments on Out-of-Domain Calibration

Table 5 reports ImageNet accuracy after calibration on 1024 unlabeled SUN397 images (Xiao et al., 2010). GLF-Q outperforms FIMA-Q (Wu et al., 2025a) on all three models at both bit-widths, with gains of up to 31.97% on W3A3 DeiT-S. GLF also outperforms DPLR-FIM within our framework at both bit-widths.

4.6 Analysis of Inference Efficiency

Since quantization below 8 bits requires specialized hardware (Li et al., 2021; Zhong et al., 2024), we evaluate real-world deployment under 8-bit quantization. As shown in Table 6, GLF-Q reduces inference latency by 15% on average compared to FIMA-Q (Wu et al., 2025a), while improving accuracy across all evaluated models. For ViT and DeiT models, RMSNorm removes the mean-

Table 5: ImageNet top-1 accuracy (%) with SUN397 calibration. †: DPLR-FIM in our framework.

Method	3-bit			4-bit		
	ViT-S	DeiT-S	Swin-S	ViT-S	DeiT-S	Swin-S
FIMA-Q (Wu et al., 2025a)	41.86	36.03	69.95	73.37	75.89	80.72
DPLR-FIM†	47.80	65.58	70.77	73.88	76.02	80.51
GLF-Q (Ours)	**58.72**	**68.00**	**74.71**	**75.34**	**76.46**	**81.07**

Table 6: Comparison of inference latency and Top-1 accuracy under 8-bit quantization. Latency (ms) is measured on a single NVIDIA H800 GPU with a batch size of 64 using NVIDIA TensorRT (NVIDIA Corporation, 2024), and Top-1 accuracy is evaluated on ImageNet.

Method	Latency			Top-1 Accuracy		
	ViT-S	DeiT-S	Swin-S	ViT-S	DeiT-S	Swin-S
FIMA-Q (Wu et al., 2025a)	2.95	2.95	6.83	79.19	78.72	83.06
GLF-Q (Ours)	**2.50**	**2.49**	**5.88**	**80.35**	**79.07**	**83.08**

Table 7: Comparison of training time (minutes) on ImageNet under 3-bit quantization.

Method	ViT-S	ViT-B	DeiT-T	DeiT-S	DeiT-B	Swin-S	Swin-B
FIMA-Q (Wu et al., 2025a)	48	87	44	49	88	115	135
LS-ViT (Hwang et al., 2026)	40	56	41	40	56	101	104
GLF-Q (Ours)	**21**	**34**	**19**	**20**	**34**	**57**	**63**

centering operation in LayerNorm, while ReLU simplifies activation computation compared with GELU. Swin models retain LayerNorm but also benefit from the simpler ReLU activation. These computational simplifications can contribute to the observed latency reductions. Deployment and timing details are provided in Appendix G.

4.7 Analysis of Training Efficiency

To evaluate practical training efficiency, we compare training time against FIMA-Q (Wu et al., 2025a) and LS-ViT (Hwang et al., 2026) under 3-bit quantization on ImageNet. All training times are measured on a single NVIDIA H800 GPU. For all methods, the reported time includes calibration and block reconstruction. For GLF-Q, it also includes MLP reconstruction. As shown in Table 7, GLF-Q is approximately 2.0 to 2.6 times faster than FIMA-Q and 1.6 to 2.2 times faster than LS-ViT across the seven architectures. GLF-Q converges within 3,000 iterations using simple STE, whereas FIMA-Q and LS-ViT use AdaRound with 20,000 reconstruction iterations.

5 Conclusion

In this paper, we proposed GLF-Q, an efficient post-training quantization framework for Vision Transformers. To address calibration overfitting in logit-guided reconstruction and the approximation limitations of Hessian-based objectives, we introduce the Global-Local Feature loss, which combines penultimate-layer feature alignment through the full-precision downstream network with local reconstruction regularization. Furthermore, offline Hadamard transformations disperse activation outliers to suppress clipping errors with zero runtime overhead, while basic STE optimization achieves rapid convergence. Extensive experiments demonstrate that GLF-Q with standard uniform quantizers substantially outperforms state-of-the-art methods under 3-bit quantization on image classification. GLF-Q also exhibits strong out-of-domain calibration robustness and delivers speedups under 8-bit GPU deployment.

AI USE STATEMENT

In this work, we used generative AI tools for assisting in the writing of proofs, implementing methods, and assisting with translation. We have not used generative AI tools for generating synthetic datasets, developing theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, cleaning and reformatting datasets, supporting qualitative and thematic data analysis, or interpreting results. Additionally, we used generative AI tools to edit the manuscript to improve readability and to assist with literature search. All AI-assisted text, derivations, and code were thoroughly reviewed, tested, and verified for correctness by the authors. The authors take full responsibility for the final content of this work, including all text, code, claims, and artifacts.

ETHICS STATEMENT

This work adheres to the ICLR Code of Ethics. Our research on Vision Transformer quantization promotes green computing by reducing memory and computational overhead. All experiments are conducted entirely on publicly available benchmark datasets, involving no human subjects, sensitive personal data, or IRB concerns. As a general model compression algorithm, it introduces no domain-specific bias, and we advocate for responsible and compliant AI deployment.

REPRODUCIBILITY STATEMENT

To ensure reproducibility, full algorithmic formulations, uniform quantization setups, and offline Hadamard transformations are detailed in Section 3 and Appendix A–B. All hyperparameter settings, calibration sample sizes, and implementation details across different architectures are comprehensively described in Section 4.1. Our code will be made publicly available upon paper acceptance.

REFERENCES

Saleh Ashkboos, Maximilian Croci, Marcelo Gennari do Nascimento, Torsten Hoefler, and James Hensman. Slicegpt: Compress large language models by deleting rows and columns. In *International Conference on Learning Representations*, volume 2024, pp. 11682–11701, 2024a.

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

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

Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 6154–6162. Ieee, 2018.

Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *International Conference on Learning Representations*, 2024.

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. Ieee, 2009.

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.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Steven K Esser, Jeffrey L McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S Modha. Learned step size quantization. *arXiv preprint arXiv:1902.08153*, 2019.

Minghao Fu, Hao Yu, Jie Shao, Junjie Zhou, Ke Zhu, and Jianxin Wu. Quantization without tears. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4462–4472. IEEE, 2025.

Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. A survey of quantization methods for efficient neural network inference. In *Low-power computer vision*, pp. 291–326. Chapman and Hall/CRC, 2022.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

Hyunha Hwang, Xuan Truong Nguyen, and Hyuk-Jae Lee. Ls-vit: Least-squares hessian based block reconstruction for low-bit post-training quantization of vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 33588–33597, 2026.

Tao Jiang, Yucheng Jiang, Xiwen Yao, Gong Cheng, and Junwei Han. Uq-vit: Harmonizing extreme activations with hardware-friendly uniform quantization in vision transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 22354–22362, 2026.

Yanjing Li, Sheng Xu, Baochang Zhang, Xianbin Cao, Peng Gao, and Guodong Guo. Q-vit: Accurate and fully quantized low-bit vision transformer. *Advances in neural information processing systems*, 35:34451–34463, 2022.

Yuhang Li, Ruihao Gong, Xu Tan, Yang Yang, Peng Hu, Qi Zhang, Fengwei Yu, Wei Wang, and Shi Gu. Brecq: Pushing the limit of post-training quantization by block reconstruction. *arXiv preprint arXiv:2102.05426*, 2021.

Zhikai Li, Junrui Xiao, Lianwei Yang, and Qingyi Gu. Repq-vit: Scale reparameterization for post-training quantization of vision transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 17181–17190. IEEE, 2023.

Guang Liang, Xinyao Liu, and Jianxin Wu. Gplq: A general, practical, and lightning qat method for vision transformers. *Advances in Neural Information Processing Systems*, 38:7033–7057, 2026.

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. Springer, 2014.

Jiawei Liu, Lin Niu, Zhihang Yuan, Dawei Yang, Xinggang Wang, and Wenyu Liu. Pd-quant: Post-training quantization based on prediction difference metric. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24427–24437. IEEE, 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 *2021 IEEE/CVF international conference on computer vision (ICCV)*, pp. 9992–10002. Ieee, 2021.

Yuexiao Ma, Huixia Li, Xiawu Zheng, Feng Ling, Xuefeng Xiao, Rui Wang, Shilei Wen, Fei Chao, and Rongrong Ji. Outlier-aware slicing for post-training quantization in vision transformer. In *Forty-first International Conference on Machine Learning*, 2024.

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

Markus Nagel, Rana Ali Amjad, Mart Van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? adaptive rounding for post-training quantization. In *International conference on machine learning*, pp. 7197–7206. PMLR, 2020.

NVIDIA Corporation. *NVIDIA TensorRT*, 2024. `https://developer.nvidia.com/tensorrt`.

Peilin Sun and Jianxin Wu. Nonlinear bipolar compensation: Handling outliers in post-training quantization. *arXiv preprint arXiv:2605.16423*, 2026.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pp. 10347–10357. PMLR, 2021.

Guo-Hua Wang, Yifan Ge, and Jianxin Wu. Distilling knowledge by mimicking features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):8183–8195, 2021.

Xiuying Wei, Ruihao Gong, Yuhang Li, Xianglong Liu, and Fengwei Yu. Qdrop: Randomly dropping quantization for extremely low-bit post-training quantization. *arXiv preprint arXiv:2203.05740*, 2022.

Zhuguanyu Wu, Jiaxin Chen, Hanwen Zhong, Di Huang, and Yunhong Wang. Adalog: Post-training quantization for vision transformers with adaptive logarithm quantizer. In *European Conference on Computer Vision*, pp. 411–427. Springer, 2024.

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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14891–14900. IEEE, 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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9686–9695. IEEE, 2025b.

Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pp. 3485–3492. IEEE, 2010.

Lianwei Yang, Haisong Gong, Haokun Lin, Yichen Wu, Caifeng Shan, Zhenan Sun, and Qingyi Gu. Dopq-vit: towards distribution-friendly and outlier-aware post-training quantization for vision transformers. *arXiv preprint arXiv:2408.03291*, 2024.

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*, pp. 191–207. Springer, 2022.

Yunshan Zhong, Jiawei Hu, Mengzhao Chen, Rongrong Ji, et al. I&s-vit: An inclusive & stable method for pushing the limit of post-training vits quantization. *arXiv preprint arXiv:2311.10126*, 2023.

Yunshan Zhong, Jiawei Hu, You Huang, Yuxin Zhang, and Rongrong Ji. Erq: Error reduction for post-training quantization of vision transformers. In *Forty-first International Conference on Machine Learning*, 2024.

APPENDIX

A UNIFORM QUANTIZER FORMULATION

For efficient hardware deployment, we employ the standard uniform quantizer. Given bit-width b, uniform quantization maps a continuous tensor x to its low-bit integer representation x^{q}:

$$x^{\mathrm{q}} = \mathrm{clip}\left(\left\lfloor \frac{x}{s} \right\rceil + z, 0, 2^b - 1\right), \tag{11}$$

where $\lfloor \cdot \rceil$ denotes the round-to-nearest operator, $\mathrm{clip}(\cdot, 0, 2^b - 1)$ restricts values to $[0, 2^b - 1]$, $s \in \mathbb{R}^+$ is the quantization scale, and $z \in \mathbb{Z}$ is the zero-point offset:

$$s = \frac{\max(x) - \min(x)}{2^b - 1}, \quad z = \mathrm{clip}\left(\left\lfloor -\frac{\min(x)}{s} \right\rceil, 0, 2^b - 1\right). \tag{12}$$

The dequantized tensor $\hat{x}$ approximates the original tensor x via:

$$\hat{x} = s \times (x^{\mathrm{q}} - z). \tag{13}$$

B COMPREHENSIVE DETAILS AND EQUIVALENCE PROOFS FOR OFFLINE HADAMARD TRANSFORMATIONS

This section describes the conversion from LayerNorm to RMSNorm, offline Hadamard transformations and their equivalence, and Hadamard construction for dimensions that are not powers of two. We use the row-vector convention throughout.

B.1 LAYERNORM-TO-RMSNORM CONVERSION

Let C denote the channel dimension. Define the mean-centering matrix and centered activation as:

$$M = I - \frac{1}{C}\mathbf{1}\mathbf{1}^\top \in \mathbb{R}^{C \times C}, \qquad x_{\mathrm{c}} = xM, \tag{14}$$

where $\mathbf{1} \in \mathbb{R}^C$ is the all-ones vector. LayerNorm can be expressed using affine-free RMSNorm as:

$$\mathrm{LN}(x) = \mathrm{RMSNorm}(xM)\,\mathrm{Diag}(\gamma) + \beta, \tag{15}$$

where $\gamma, \beta \in \mathbb{R}^C$ are the affine scaling and bias. RMSNorm retains the original ϵ, while the affine parameters are folded into the subsequent linear layer:

$$\widetilde{W} = \mathrm{Diag}(\gamma)W, \qquad \widetilde{b} = \beta W + b. \tag{16}$$

For a standard pre-LN residual sublayer $x^+ = x + F(\mathrm{LN}(x))$, post-centering yields $x_{\mathrm{c}}^+ = x^+M = (x + F(\mathrm{LN}(x)))M = x_{\mathrm{c}} + F(\mathrm{LN}(x))M$. To maintain $x_{\mathrm{c},l} = x_l M$ across layers, we fold centering into the patch embedding and the output projections of attention and MLP branches:

$$E_{\mathrm{patch}} \leftarrow E_{\mathrm{patch}}M, \quad b_E \leftarrow b_E M,$$
$$W_{\mathrm{out}} \leftarrow W_{\mathrm{out}}M, \quad b_{\mathrm{out}} \leftarrow b_{\mathrm{out}}M.$$

The class token and positional embeddings are also centered as $t \leftarrow tM$ and $p \leftarrow pM$. These updates preserve the features after the final normalization.

B.2 RESIDUAL HADAMARD TRANSFORMATIONS

Following the computational invariance established in SliceGPT (Ashkboos et al., 2024a), we apply a shared rotation to the residual stream. We select an orthogonal transformation matrix $R_1 \in \mathbb{R}^{C \times C}$ satisfying $R_1^\top R_1 = R_1 R_1^\top = I$ and define the rotated residual activation as:

$$x_{\mathrm{c}}^{\mathrm{rot}} = x_{\mathrm{c}} R_1. \tag{17}$$

Affine-free RMSNorm satisfies orthogonal equivariance (Ashkboos et al., 2024b):

$$\mathrm{RMSNorm}(x_{\mathrm{c}}R_1) = \frac{x_{\mathrm{c}}R_1}{\sqrt{\frac{1}{C}\|x_{\mathrm{c}}R_1\|_2^2 + \epsilon}} = \frac{x_{\mathrm{c}}R_1}{\sqrt{\frac{1}{C}\|x_{\mathrm{c}}\|_2^2 + \epsilon}} = \mathrm{RMSNorm}(x_{\mathrm{c}})R_1, \tag{18}$$

Table 8: Parameter transformation rules for the centered and affine-folded Vision Transformer.

Network Position	Parameter Transformation Rule
Input Patch Embedding	$E_{\text{patch}} \to E_{\text{patch}} R_1\,, \quad b_E \to b_E R_1$
Class Token & Positional Embeddings	$t \to tR_1\,, \quad p \to pR_1$
QKV Projection & MLP First Layer	$W_{\text{in}} \to R_1^\top W_{\text{in}}\,, \quad b_{\text{in}}$ (unchanged)
Attention Output Projection & MLP Second Layer	$W_{\text{out}} \to W_{\text{out}} R_1\,, \quad b_{\text{out}} \to b_{\text{out}} R_1$
Final Classification Head	$W_{\text{head}} \to R_1^\top W_{\text{head}}\,, \quad b_{\text{head}}$ (unchanged)

because orthogonal transformations preserve Euclidean norms. Table 8 summarizes the corresponding offline parameter transformations.

Input and output projections absorb $R_1^\top$ and R_1, respectively, so both residual branches share the same rotation. RMSNorm equivariance allows this rotation to propagate through the network, while the classification head absorbs its inverse, preserving the logits.

B.3 Head-wise QK and VO Transformations

Head-wise QK Rotation. Let $Q_h, K_h \in \mathbb{R}^{N\times C_h}$ denote the query and key states of attention head h, where $C_h = C/N_h$. For an orthogonal rotation $R_{2,h} \in \mathbb{R}^{C_h\times C_h}$, we define $Q'_h = Q_h R_{2,h}$ and $K'_h = K_h R_{2,h}$, with the rotation folded into the query and key projections offline. Then

$$Q'_h {K'_h}^\top = (Q_h R_{2,h})(K_h R_{2,h})^\top = Q_h R_{2,h} R_{2,h}^\top K_h^\top = Q_h K_h^\top\,. \tag{19}$$

Because the inner products entering Softmax remain unchanged, the resulting attention probabilities are also unchanged.

Head-wise VO Rotation. Let $P_h \in \mathbb{R}^{N\times N}$ denote the attention probability matrix, $V_h \in \mathbb{R}^{N\times C_h}$ the value states, and $W_{O,h} \in \mathbb{R}^{C_h\times C}$ the row slice of the output projection corresponding to head h. Under the row-vector convention, this head contributes $(P_h V_h)W_{O,h}$ to the attention output. For an orthogonal rotation $R_{3,h} \in \mathbb{R}^{C_h\times C_h}$, we define $V'_h = V_h R_{3,h}$ and fold the inverse rotation into the output projection as $W'_{O,h} = R_{3,h}^\top W_{O,h}$. Then

$$(P_h V'_h)W'_{O,h} = (P_h V_h R_{3,h})(R_{3,h}^\top W_{O,h}) = P_h V_h W_{O,h}\,. \tag{20}$$

Summing these contributions over all heads and retaining the output projection bias preserves the multi-head self-attention output.

B.4 Randomized Hadamard Construction

Following QuaRot (Ashkboos et al., 2024b), we construct normalized randomized Hadamard matrices for the dimensions used in our models. For $d = 2^k$, we start with $H_1 = [1]$ and recursively construct

$$H_{2n} = \begin{bmatrix} H_n & H_n \\ H_n & -H_n \end{bmatrix}$$

until the dimension reaches d. We then define the normalized randomized Hadamard matrix as

$$R_d = \frac{DH_d}{\sqrt{d}}\,, \tag{21}$$

where D is a diagonal matrix with independent entries uniformly sampled from $\{-1, +1\}$.

For $d = 12 \cdot 2^k$, we combine a fixed Hadamard matrix H_{12} with H_{2^k} using the Kronecker product:

$$R_d = \frac{D\left(H_{12}^\top \otimes H_{2^k}\right)}{\sqrt{d}}\,. \tag{22}$$

This construction supports hidden dimensions such as 192, 384, and 768 without padding or truncation. Since $H_m^\top H_m = mI_m$ and $D^\top D = I_d$, both constructions satisfy $R_d^\top R_d = I_d$.

C Overall Pipeline of GLF-Q

The overall pipeline of GLF-Q is summarized in Algorithm 1.

Algorithm 1 Overall Pipeline of GLF-Q

Require: Pretrained model $\mathcal{M}$; unlabeled calibration data $\mathcal{D}_{\text{cal}}$; bit-widths (b_w, b_a); reconstruction iterations $T = 3000$; regularization coefficient $\lambda = 2$.

Ensure: Quantized model $\widehat{\mathcal{M}}$.

```
Initialize teacher M_fp and student M_s from M.
Apply identical offline transformations to both models following Section 3.3 and the architecture-specific settings in Section 4.1.
Freeze M_fp; replace student MLP GELU with ReLU and reconstruct against M_fp.
Insert and calibrate student uniform quantizers; apply the Swin reparameterization when applicable.
for each reconstruction block l in forward order do
    Cache student inputs x̂_l from the quantized prefix and teacher targets (y_l, f) on D_cal.
    Let F_l be the frozen teacher suffix; optimize only the current student block.
    for t = 1, ..., T do
        Sample matching batches of (x̂_l, y_l, f).
        Compute the current quantized block output ŷ_l from x̂_l and obtain f̂ = F_l(ŷ_l).
        Evaluate L_global^(l) and L_local^(l) on the minibatch using Eqs. (4) and (5).
        if t = 1 then
            Cache the detached loss values, each clamped below at 10^-12, as fixed normalization factors.
        end if
        Normalize both loss terms by their cached factors and combine them using Eq. (6).
        Backpropagate through F_l and the quantizers using STE; update θ_l and s_l with Adam.
    end for
end for
```

D Effect of Loss Functions on Generalization

To evaluate generalization under limited calibration data, Table 9 compares the top-1 accuracy of logit and feature supervision on the calibration set and the ImageNet validation set. Logits + LF combines logit supervision with local feature regularization from PD-Quant (Liu et al., 2023), while GLF denotes the proposed Global-Local Feature loss. Both variants follow the 3-bit setting in Table 3 and differ only in the loss function.

Table 9: Top-1 accuracy (%) on the calibration set and ImageNet validation set under 3-bit quantization.

	Logits + LF		GLF	
Model	Calibration	Validation	Calibration	Validation
ViT-S	86.91	65.16	87.70	67.40
ViT-B	91.11	76.89	91.11	78.18
DeiT-T	77.05	55.26	76.66	56.48
DeiT-S	90.04	70.71	89.65	71.55
DeiT-B	96.00	76.87	96.00	77.20
Swin-S	87.40	76.80	87.70	78.10
Swin-B	92.38	78.58	92.48	80.45
Mean	88.70	71.47	88.76	72.77

With nearly identical mean calibration accuracy (88.76% for GLF and 88.70% for Logits + LF), GLF yields higher validation accuracy across all seven architectures. On average, GLF improves

validation accuracy by 1.30% and narrows the performance gap between the calibration and validation sets from 17.23% to 15.99%. These results suggest better generalization with GLF under limited calibration data.

E Stability across Random Seeds

Table 10 reports multi-seed ImageNet top-1 accuracy for GLF-Q across seven architectures under 3-bit and 4-bit quantization.

Table 10: Multi-seed top-1 accuracy (%) of GLF-Q on ImageNet under 3-bit and 4-bit quantization, reported as mean ± standard deviation over three random seeds.

W/A	**ViT-S**	**ViT-B**	**DeiT-T**	**DeiT-S**	**DeiT-B**	**Swin-S**	**Swin-B**
3/3	67.34 ± 0.12	78.21 ± 0.04	56.17 ± 0.27	71.67 ± 0.10	77.22 ± 0.10	78.07 ± 0.06	80.44 ± 0.03
4/4	77.69 ± 0.15	83.12 ± 0.10	67.22 ± 0.16	77.40 ± 0.06	80.51 ± 0.04	82.05 ± 0.08	83.81 ± 0.01

F Additional Ablation Studies

Sensitivity to the regularization coefficient. We evaluate GLF-Q with $\lambda \in \{1, 2, 4\}$ under 3-bit quantization on ImageNet. As shown in Table 11, mean top-1 accuracy across seven architectures ranges from 72.77% to 72.81%. These results suggest limited sensitivity to λ within the evaluated range.

Table 11: Top-1 accuracy (%) of GLF-Q with different regularization coefficients under 3-bit quantization on ImageNet.

λ	ViT-S	ViT-B	DeiT-T	DeiT-S	DeiT-B	Swin-S	Swin-B	Mean
1	67.53	78.29	56.31	71.47	77.33	78.22	80.49	72.81
2	67.40	78.18	56.48	71.55	77.20	78.10	80.45	72.77
4	67.48	78.18	56.61	71.59	77.22	78.14	80.35	72.80

G Implementation Details

G.1 MLP Reconstruction

MLP reconstruction is performed block-wise in floating-point precision before quantization to compensate for the error introduced by replacing GELU with ReLU. The teacher retains GELU, while the student uses ReLU. Both use identical coordinate transformations. Each student MLP is optimized using cached teacher inputs x and outputs y with the objective

$$\mathcal{L}_{\mathrm{MR}} = \frac{1}{NC} \left\| f_{\mathrm{ReLU}}(x) - y \right\|_F^2 + \frac{2}{NC} \left\| f_{\mathrm{ReLU,clip}}(x) - y \right\|_F^2 , \tag{23}$$

where N and C denote the number of output tokens and channels, respectively. The function $f_{\mathrm{ReLU,clip}}$ applies upper clipping to the student's FC2 input, with a fixed threshold for each block set to the 99th percentile of positive activations at the corresponding teacher FC2 input over the calibration data.

Each MLP is optimized for 20,000 iterations using Adam without weight decay. The learning rate follows an independent cosine schedule for each block, decaying from 4×10^{-5} to zero. Only the weights and biases of the current student MLP's FC1 and FC2 layers are updated. All other parameters remain frozen.

G.2 Object Detection and Segmentation

To extend GLF-Q to object detection and instance segmentation, we use intermediate bbox and mask RoI features before the prediction layers as global supervision targets. The bbox and mask branches use up to 1000 and 128 student RPN proposals per image, respectively, preserving their original order. Teacher and student paths use identical RoI coordinates for each loss. The proposals and teacher reference features are cached once and reused throughout reconstruction.

Backbone reconstruction combines local, bbox RoI, and mask RoI feature MSE losses with weights of 2, 0.5, and 0.5, respectively. Each loss term is normalized by its value at the first reconstruction iteration, with the normalization factor held fixed for that block. FPN reconstruction uses only the normalized local MSE with a weight of 2, while the RPN and RoI heads undergo calibration-only quantization.

G.3 TensorRT Deployment and Latency Measurement

To support TensorRT's standard INT8 path, asymmetric quantization representations that are not directly compatible are exported in floating-point form and requantized using a common scheme. We use symmetric absmax quantization per output channel for weights and per tensor for activations.

All latencies are measured on the same NVIDIA H800 80GB GPU using TensorRT 10.9, with an input resolution of 224×224 and a batch size of 64. After 50 warm-up runs, we time 200 inference runs using CUDA events and report the mean batch latency. Timing covers only GPU engine execution, excluding data loading, preprocessing, and data transfers. Engines are built with INT8 enabled and FP32/TF32 fallback allowed.