FINE-TUNING LEAVES CLUES: CALIBRATION-FREE QUANTIZATION VIA TASK-AWARE BASIS

Anonymous authors
Paper under double-blind review

ABSTRACT

Fine-tuning foundation models has become the standard way to create domain-specific expert models. While calibration-free quantization offers an efficient deployment solution without requiring downstream data, existing methods often suffer significant performance drop at low bit-widths. These methods typically neglect the task-specific information encoded in the task vector, which is defined as the parameter difference between the fine-tuned and pre-trained weights. In this work, we propose **T**ask-aware bas**I**s to **G**uid**E** calibration-f**R**ee **Q**uantization (TIGER-Q). TIGER-Q extracts a task-aware basis from the task vector to serve as a data-free proxy for the Hessian matrix. This basis establishes a task-sensitive subspace that guides the quantization process to minimize error along dimensions critical to the target task. To efficiently solve this objective, we introduce a decoupled continuous-discrete refinement pipeline. It computes closed-form analytical solutions for continuous quantization scales, while employing a local search for discrete weights, coupled with a row-wise acceptance mechanism to maintain monotonic error reduction. Extensive experiments show that TIGER-Q achieves superior performance over state-of-the-art calibration-free methods on both language and vision models.

1 INTRODUCTION

Driven by the emergence of large foundation models (Liu et al., 2019; Radford et al., 2021; Touvron et al., 2023; Kirillov et al., 2023; Devlin et al., 2019), adapting pre-trained models through downstream fine-tuning has become the dominant paradigm (Luo et al., 2024; 2025; Raffel et al., 2020; Dosovitskiy et al., 2020), rather than designing and training new architectures from scratch. This pre-training and fine-tuning workflow produces expert models highly specialized in tasks such as mathematical reasoning (Luo et al., 2025; Azerbayev et al., 2023; Lewkowycz et al., 2022), code generation (Luo et al., 2024; Wang et al., 2021), or instruction following (Ouyang et al., 2022). As these fine-tuned expert models are deployed at scale, their massive memory footprint limit practical application, making post-training quantization (PTQ) a crucial technique for efficient deployment.

Calibration-free quantization (Guo et al., 2022; Badri & Shaji, 2023; Müller et al., 2025) offers a highly efficient PTQ solution by eliminating the need for input calibration data. This makes it practical for deploying expert models, and its value becomes particularly crucial when their fine-tuning datasets are proprietary or inaccessible. However, existing calibration-free methods suffer from a fundamental limitation of failing to incorporate any task-specific information. By relying solely on the statistical properties of the model weights, such as mean or variance, they neglect the most valuable aspect of a fine-tuned model, specifically its task-specific adaptation. Without this crucial task-aware guidance, these methods cannot mitigate the adverse impact of quantization error on downstream performance. This specific adaptation is represented by the parameter difference between a fine-tuned model and its pre-trained initialization, which is commonly defined as the *task vector* in the context of model editing (Ilharco et al., 2023). The task vector precisely encodes the relatively subtle, yet vital, parameter displacements required to acquire specialized behavior. While standard calibration-free methods might achieve a small overall weight distortion, they often disproportionately degrade the critical directions embedded in the task vector. Without task-aware guidance, low-bit quantization can degrade a fine-tuned model's target-task performance, particularly below 4 bits.

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

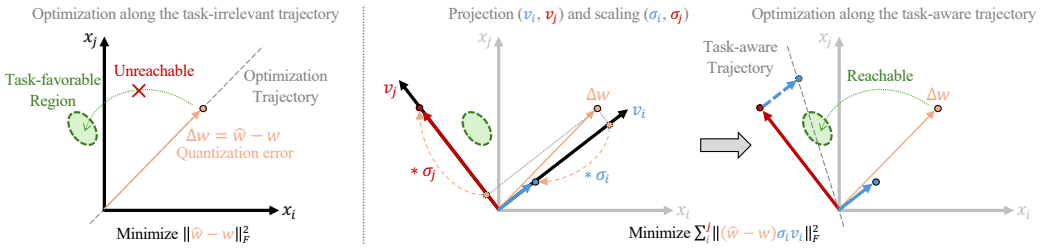


Figure 1: Illustration of the task-aware basis. **Left**: In the original coordinate system spanned by x_i and x_j, the orange vector denotes the quantization error $\Delta w = \widehat{w} - w$, and naive weight reconstruction fails to direct it toward the task-favorable region. **Middle**: The error is projected onto the task-aware basis vectors v_i and v_j, and weighted by their corresponding singular values σ_i and σ_j. **Right**: The resulting task-aware objective guides the optimization trajectory toward the task-favorable region. Dashed curves denote the corresponding optimization trajectories.

In this work, we propose **T**ask-aware bas**I**s to **G**uid**E** calibration-f**R**ee **Q**uantization (TIGER-Q), a novel method designed specifically for fine-tuned expert models. To achieve this, TIGER-Q constructs a task-aware basis from the task vector and uses it as a practical data-free proxy for the input covariance in the quantization objective. As illustrated in Fig. 1, the resulting objective penalizes quantization error along directions identified by the task vector. To efficiently solve this task-aware objective, TIGER-Q introduces a decoupled continuous-discrete refinement pipeline. First, it performs Task-Aware Scale Refinement to analytically compute closed-form optimal updates for the continuous quantization scales. Subsequently, it executes Task-Aware Integer Weight Refinement, systematically adjusting the discrete integer weights through a task-aware objective-driven local search with strict row-wise acceptance. TIGER-Q successfully transforms calibration-free quantization from purely statistical weight reconstruction into task-aware error guidance, effectively preserving the expert model's specialized capabilities without requiring any calibration data.

We evaluate TIGER-Q on various expert models fine-tuned from powerful foundation models including language and vision models under low-bit weight quantization. Extensive experiments validate the superiority of TIGER-Q, which significantly outperforms state-of-the-art (SOTA) calibration-free PTQ baselines. For example, TIGER-Q yields an over 11.62% average performance gain relative to the most competitive calibration-free baseline on 3-bit RoBERTa-Base (Liu et al., 2019) on the GLUE benchmark (Wang et al., 2019).

Our main contributions are summarized as follows:

1) We propose TIGER-Q, a novel calibration-free quantization framework that leverages task vectors to construct the task-aware basis for minimizing task-aware error, thereby protecting the specialized capabilities of fine-tuned expert models during low-bit quantization without any calibration data.

2) We design a decoupled continuous-discrete refinement pipeline to optimize the quantization objective, comprising Task-Aware Scale Refinement for analytical updating quantization scales and Task-Aware Integer Weight Refinement for optimizing integer weights via an objective-driven local search.

3) Extensive evaluations across diverse language and vision architectures demonstrate that TIGER-Q significantly outperforms state-of-the-art calibration-free baselines, especially at low bit-widths.

2 Related Work

Quantization. Model quantization is a fundamental technique for reducing the memory footprint and accelerating the inference of large neural networks. Post-Training Quantization (PTQ) has become the dominant paradigm. Standard PTQ methods typically freeze the pre-trained weights and calibrate the quantization parameters using a small calibration dataset. Advanced PTQ techniques (Liu et al., 2025; Xiao et al., 2023; Lin et al., 2024; Frantar et al., 2022; Shao et al., 2024; Ding et al., 2025), including GPTQ (Frantar et al., 2022) and AWQ (Lin et al., 2024), have demonstrated remarkable success in compressing Large Language Models (LLMs) and Vision Transform-

ers (ViTs) by analyzing the activation patterns or Hessian matrices derived from the calibration data. However, the efficacy of these methods relies on the availability of calibration data. When expert models are fine-tuned on proprietary, private, or confidential datasets, obtaining such representative data would be restricted, limiting the applicability of calibration-based PTQ methods.

Calibration-free quantization. Calibration-free quantization emerges as a method that compresses models without requiring any calibration data, relying solely on their inherent weight parameters. The most straightforward method is Round-to-Nearest (RTN), which directly rounds weights using min-max bounds. Existing calibration-free methods tackle quantization error through distinct mathematical strategies: HQQ (Badri & Shaji, 2023) formulates half-quadratic optimizations, SINQ (Müller et al., 2025) applies Sinkhorn-style normalization, and Ashkboos et al. (2024) employ orthogonal Hadamard rotations (Chee et al., 2023; Tseng et al., 2024) to uniformly spread outliers. Despite their efficiency, these methods are fundamentally task-agnostic. These methods treat the network as generic numerical tensors and optimize based on task-agnostic mathematical heuristics like weight statistics. Without task-specific guidance, these methods can degrade target-task performance under aggressive settings such as 3-bit quantization.

Foundation models and task vectors. It is common to develop domain-specific AI by adapting open-source pre-trained models (Liu et al., 2019; Devlin et al., 2019; Touvron et al., 2023) through downstream fine-tuning. Recent studies in model editing and task arithmetic (Ilharco et al., 2023; Lu et al., 2024; Yu et al., 2024; Cheng et al., 2025; Tang et al., 2024; Yang et al., 2024) have revealed that the parameter difference between a fine-tuned model and its pre-trained initialization, defined as the *task vector*, captures the crucial semantics learned during the adaptation phase. While task vectors are widely used for model merging via task arithmetic, their potential as a supervisory signal for compression remains largely unexplored. TIGER-Q bridges this gap. Inspired by Cheng et al. (2025), who reveals that task vectors inherently represent an approximate linear subspace for task inputs, we utilize these vectors to construct task-aware subspaces instead of using them for traditional merging. We use the subspace to explicitly guide the quantization process. It prevents quantization errors from corrupting the task-specific information, maintaining task performance. We note that some delta-compression researches (Kim et al., 2025; Ping et al., 2024) could easily be confused with the scope of our research. Thus, we distinguish TIGER-Q from these delta-compression methods, which quantize task vectors to reduce storage for model merging. TIGER-Q instead quantizes model weights and uses the task vector as a data-free signal to guide weight quantization.

3 METHOD

3.1 PRELIMINARIES

Uniform quantization. Uniform quantization is widely adopted for network compression due to its hardware efficiency. For a full-precision weight $w \in \mathbb{R}^{M\times N}$, the b-bit uniform quantization maps continuous values to integer weight w^q and reconstructs the de-quantized weight $\widehat{w}$ via:

$$w^q = \text{clip}\left(\lfloor\frac{w}{s}\rceil + z, 0, 2^b - 1\right), \quad \widehat{w} = s \cdot (w^q - z) \approx w, \tag{1}$$

where $\lfloor\cdot\rceil$ denotes the round-to-nearest operator and $\text{clip}(x, l, u) = \min(\max(x, l), u)$. The quantization scale s and zero-point z are determined by the weight bounds w_{ub} and w_{lb}:

$$s = \frac{w_{ub} - w_{lb}}{2^b - 1}, \quad z = \lfloor -\frac{w_{lb}}{s}\rceil. \tag{2}$$

Our work builds upon the group-wise quantization format, where s and z are shared within localized parameter groups of size g, such that $s, z \in \mathbb{R}^{M\times G}$ with $G = N/g$.

Task vectors. In the current paradigm of foundation models, downstream models are typically adapted from a pre-trained backbone. For a pre-trained weight $w_0 \in \mathbb{R}^{M\times N}$ and its fine-tuned counterpart $w \in \mathbb{R}^{M\times N}$, the *task vector* is defined as their parameter difference:

$$\tau := w - w_0. \tag{3}$$

While τ captures the precise parametric displacements crucial for task specialization, existing calibration-free methods focus solely on reconstructing w and neglect this task-specific signal. In our work, we utilize τ as an intrinsic, data-free prior to guide the quantization optimization.

3.2 Guiding Quantization Optimization via Task-Aware Basis

To mitigate the capability degradation on target tasks, the quantization process should be optimized with respect to the downstream task loss $\mathcal{L}(\cdot)$. By applying a second-order Taylor expansion around the fine-tuned full-precision weights w, and assuming the model has converged to a local optimum ($\nabla\mathcal{L}(w) \approx 0$), the expected performance drop is dominated by the Hessian matrix $H_w = \nabla^2\mathcal{L}(w)$:

$$\mathbb{E}[\mathcal{L}(\widehat{w}) - \mathcal{L}(w)] \approx \mathbb{E}[\frac{1}{2}\mathrm{vec}(\widehat{w} - w)^\top H_w \mathrm{vec}(\widehat{w} - w)] \tag{4}$$

$$\approx \frac{1}{2}\mathrm{vec}(\widehat{w} - w)^\top (\mathbb{E}[xx^\top] \otimes I_{M\times M})\mathrm{vec}(\widehat{w} - w), \tag{5}$$

where $\mathrm{vec}(\cdot)$ flattens a matrix, $\otimes$ is the Kronecker product, and $x \in \mathbb{R}^{N\times 1}$ is the layer input. Following standard PTQ practices (Nagel et al., 2020; Li et al., 2021; Frantar et al., 2022; Frantar & Alistarh, 2022), the Hessian is approximated as $\mathbb{E}[xx^\top] \otimes I_{M\times M}$.

Since calibration-free quantization strictly prohibits access to downstream data, the exact computation of the input covariance $\mathbb{E}[xx^\top]$ is impossible. To build a practical data-free proxy, we relate $\mathbb{E}[xx^\top]$ to the observable task vector τ. During downstream fine-tuning with the plain gradient descent (*e.g.*, SGD), τ accumulates gradient updates:

$$\tau = -\sum_{u=1}^{T} \gamma_u \, \delta_u x_u^\top, \tag{6}$$

where u denotes the optimization step, γ_u represents the learning rate, and $\delta_u \in \mathbb{R}^{M\times 1}$ is the output gradient. The Gram matrix of τ can be expanded and seperated into the self- and cross-step terms:

$$\tau^\top\tau = \sum_u \gamma_u^2 \|\delta_u\|_2^2 x_u x_u^\top + \sum_{u\neq v} \gamma_u\gamma_v x_u(\delta_u^\top\delta_v)x_v^\top = \sum_u \gamma_u^2\|\delta_u\|_2^2 x_u x_u^\top + R_{\mathrm{cross}}. \tag{7}$$

Taking the expectation over sampled training examples gives

$$\mathbb{E}[\tau^\top\tau] \approx \sum_u \gamma_u^2 \mathbb{E}[\|\delta_u\|_2^2]\mathbb{E}[x_u x_u^\top] + \mathbb{E}[R_{\mathrm{cross}}]$$

$$\approx \lambda\,\mathbb{E}[xx^\top] + \mathbb{E}[R_{\mathrm{cross}}] \approx \lambda\,\mathbb{E}[xx^\top], \tag{8}$$

where $\lambda = \sum_{u=1}^{T}\gamma_u^2\mathbb{E}[\|\delta_u\|_2^2]$. The first approximation assumes weak dependence between the squared gradient norm and the input outer product, while the second assumes limited drift in input statistics across optimization steps. For the third approximation, under *i.i.d.* sampling and approximately zero-mean stochastic updates, $\mathbb{E}[R_{\mathrm{cross}}]$ is expected to be small (Xu et al., 2026). The observed $\tau^\top\tau$ is a single realization of this distribution, motivating $\tau^\top\tau \approx \lambda\mathbb{E}[xx^\top]$ in practice. Details of the theoretical derivations and conditions for the three approximations are provided in Appendix A. Under the approximations in Eq. 8, we use $\tau^\top\tau$ as a practical data-free proxy for the input covariance in the quantization objective in Eq. 5:

$$\arg\min_{s,w^q} \mathbb{E}[\mathcal{L}(\widehat{w}) - \mathcal{L}(w)] \approx \arg\min_{s,w^q} \frac{1}{2\lambda}\mathrm{Tr}\big((\widehat{w} - w)\tau^\top\tau(\widehat{w} - w)^\top\big), \tag{9}$$

where $\mathrm{Tr}(\cdot)$ sums the diagonal elements of the matrix. Although $\tau^\top\tau$ does not exactly recover $\mathbb{E}[xx^\top]$ under these approximations in Eq. 8, strict equality is not required. As discussed in Table. 6, capturing the task-relevant structure and directions of $\mathbb{E}[xx^\top]$ is sufficient for $\tau^\top\tau$ to serve as a practical data-free proxy. By performing an SVD for τ, *i.e.*, $\tau = U_r\Sigma_r V_r^\top$, we can further simplify Eq. 9 into the task-aware objective:

$$\arg\min_{s,w^q} \|(\widehat{w} - w)V_r\Sigma_r\|_F^2 = \arg\min_{s,w^q} \sum_{i=1}^{r} \sigma_i^2 \, \|(\widehat{w} - w)v_i\|_F^2, \tag{10}$$

where σ_i denotes the i-th singular value in Σ_r, and $v_i \in \mathbb{R}^{N\times 1}$ denotes the i-th column of V_r. Eq. 10 formally defines the *task-aware basis*: $\sigma_i v_i \in \mathbb{R}^{N\times 1}$. The orthogonal vectors v_i span a task-sensitive subspace, weighted by their importance σ_i. Intuitively, the optimization objective aims to minimize the projection of the quantization error onto this task subspace, since errors within this space tend to degrade task performance. Therefore, our optimization objective is formulated as:

$$\arg\min_{s,w^q} \quad \sum_{i=1}^{r} \underbrace{\sigma_i^2 \, \|(\widehat{w} - w)v_i\|_F^2}_{\text{task-aware error term}} + \underbrace{\alpha\|\widehat{w} - w\|_F^2}_{\text{reconstruction error term}} + \underbrace{\beta\|s - s^{init}\|_F^2}_{\text{stabilization term}}, \tag{11}$$

where $\alpha, \beta > 0$ are regularization hyperparameters, and s^{init} denotes the initial quantization scale.

3.3 TASK-AWARE QUANTIZATION REFINEMENT

We design task-aware optimization by updating the quantization scales s and the integer weights w^q. Due to the distinct continuous and discrete properties of these parameters, we propose a decoupled approach consisting of Task-Aware Scale Refinement and Task-Aware Integer Weight Refinement.

Task-Aware Scale Refinement. We first freeze the integer weights w^q to analytically solve for the optimal continuous quantization scales s. For row m, the reconstructed weight is $\widehat{w}_m = s_m \widetilde{w}_m = [s_{m,1}\widetilde{w}_{m,1}, \ldots, s_{m,G}\widetilde{w}_{m,G}]^\top$. Here, $s_m \in \mathbb{R}^{G\times 1}$ contains the group-wise scales, and $\widetilde{w}_{m,t} = w^q_{m,t} - z_{m,t}$ ($\widetilde{w}_{m,t} \in \mathbb{R}^{g\times 1}$) denotes the shifted integer weights for the t-th group. The row-wise optimization reduces to a strictly convex quadratic program:

$$\min_{s_m \in \mathbb{R}^G} \quad \sum_{i=1}^{r} \sigma_i^2 \langle s_m \widetilde{w}_m - w_m, v_i\rangle^2 + \alpha \|s_m \widetilde{w}_m - w_m\|_2^2 + \beta \|s_m - s_m^{init}\|_2^2, \tag{12}$$

where $\langle \cdot, \cdot \rangle$ denotes the standard inner product. Setting the derivative of Eq. 12 with respect to s_m to zero yields a linear system $H_m s_m^\star = r_m$, explicitly formulated as:

$$\begin{aligned} H_m &= \alpha D_m + \sum_{i=1}^{r} \sigma_i^2 b_{m,i} b_{m,i}^\top + \beta I_{G\times G}, \\ r_m &= \alpha [\langle \widetilde{w}_{m,1}, w_{m,1}\rangle, \ldots, \langle \widetilde{w}_{m,G}, w_{m,G}\rangle]^\top + \sum_{i=1}^{r} \sigma_i^2 \langle w_m, v_i\rangle b_{m,i} + \beta s_m^{init}, \end{aligned} \tag{13}$$

where $D_m = \mathrm{diag}(\|\widetilde{w}_{m,1}\|_2^2, \ldots, \|\widetilde{w}_{m,G}\|_2^2)$, and $b_{m,i} \in \mathbb{R}^{G\times 1}$ denotes the projection vector whose t-th element is $\langle \widetilde{w}_{m,t}, v_{i,t}\rangle$. The stabilization term $\beta I_{G\times G}$ ensures that H_m is strictly symmetric positive definite. Therefore, the optimal s_m can be solved efficiently via Cholesky decomposition.

Task-Aware Integer Weight Refinement. Building upon the global optimization objective established in Eq. 11, we perform a task-aware integer weights refinement for w^q. To decouple the complex combinatorial optimization, we fix the quantization scales s and zero-points z, and refine the integer weights w^q within a local neighborhood. We restrict the local search to a maximum step size of one integer level, *i.e.*, $\Delta w^q_{m,n} \in \{+1, -1\}$. A unit change in the integer weight induces a perturbation in the reconstructed weight by exactly its corresponding scale, $\Delta \widehat{w}_{m,n} = \Delta w^q_{m,n} \cdot s_{m,t}$, where t denotes the group index corresponding to column n. Since only a single element is updated, the exact change in the objective reduces to a one-dimensional quadratic function:

$$\Delta \mathcal{J}_{m,n}(\Delta w^q_{m,n}) = g_{m,n} \Delta \widehat{w}_{m,n} + d_n (\Delta \widehat{w}_{m,n})^2, \tag{14}$$

where $g_{m,n}$ represents the first-order gradient of $\mathcal{J}$ with respect to $\widehat{w}_{m,n}$, and $d_n = \alpha + \sum_{i=1}^{r} \sigma_i^2 v_{i,n}^2$ represents the corresponding diagonal curvature along the n-th column. We evaluate both ± 1 moves for every parameter, discarding any updates that exceed the valid quantization range $[0, 2^b - 1]$. The optimal local direction $\Delta w^{q\star}_{m,n}$ is chosen to maximize the objective gain: $\mathcal{R}_{m,n} = -\Delta \mathcal{J}_{m,n}(\Delta w^{q\star}_{m,n})$. Elements with negative gains across both directions are assigned a gain of zero.

To maintain optimization stability and avoid an expensive combinatorial search, we aggregate these local evaluations into a sparse update proposal Δw^q. Within each quantization group of size g, we greedily select at most one element that yields the highest positive gain and explicitly set all other proposed updates in that group to zero. Although each proposed element guarantees a local objective decrease independently, applying multiple group-level proposals within the same row simultaneously may increase the overall loss. This occurs because the task-aware penalty $\sum_{i=1}^{r} \sigma_i^2 \|(\widehat{w} - w) v_i\|_F^2$ tightly couples all columns within a row, rendering the simple aggregation of locally greedy group-wise updates potentially suboptimal. To strictly guarantee monotonic convergence, we perform a row-wise acceptance test. For each row m, we calculate the exact objective change $\Delta \mathcal{J}_m$ when applying the combined row proposal $\Delta \widehat{w}_m = [s_{m,1}\Delta w^q_{m,1}, \ldots, s_{m,G}\Delta w^q_{m,G}]^\top$:

$$\Delta \mathcal{J}_m = \alpha \left(2\langle \widehat{w}_m - w_m, \Delta \widehat{w}_m\rangle + \|\Delta \widehat{w}_m\|_2^2\right) + \sum_{i=1}^{r} \sigma_i^2 \left(2\langle \widehat{w}_m - w_m, v_i\rangle \langle \Delta \widehat{w}_m, v_i\rangle + \langle \Delta \widehat{w}_m, v_i\rangle^2\right). \tag{15}$$

A row proposal is accepted and applied only if $\Delta \mathcal{J}_m < 0$. If it is rejected ($\Delta \mathcal{J}_m \geq 0$), we trigger a fallback strategy. It selects only the single group-level update yielding the maximum gain across the entire row. Because this fallback updates only one single element in the row, it mathematically

Table 1: Comparison results of WizardMath (7B and 13B) and WizardCoder (7B and 13B) on the mathematical reasoning and code-generation benchmarks under 4-bit and 3-bit quantization. †: the calibration-based method. The best results of calibration-free methods are highlighted in **bold**.

		WizardMath				**WizardCoder**				
		GSM8K		MATH		HumanEval		MBPP		
	Method	7B	13B	7B	13B	7B	13B	7B	13B	Average
FP	Original	55.34	63.53	10.54	14.14	55.49	62.80	52.00	55.60	46.18
3-Bit	AWQ†	39.88	54.13	6.19	10.12	45.73	57.93	43.60	52.20	38.72
	RTN	32.52	49.13	4.36	8.80	35.37	50.61	37.00	48.40	33.27
	HQQ	22.59	45.87	2.92	5.88	0.00	54.88	0.00	**52.60**	23.09
	Hadamard+RTN	34.19	51.40	4.38	8.74	42.07	51.82	41.80	50.40	35.60
	SINQ	36.39	51.18	5.54	8.50	39.63	57.32	39.20	51.80	36.20
	TIGER-Q (Ours)	**38.59**	**54.73**	**6.70**	**9.48**	**42.68**	**59.75**	**41.80**	52.20	**38.24**
4-Bit	AWQ†	50.94	61.79	9.58	13.74	54.27	64.63	51.40	54.40	45.09
	RTN	48.82	61.49	9.14	13.12	53.05	62.80	49.20	55.20	44.10
	HQQ	49.28	60.58	8.90	13.32	50.61	**64.63**	50.60	54.40	44.04
	Hadamard+RTN	49.89	60.58	8.96	13.38	**53.66**	61.59	51.00	55.40	44.31
	SINQ	49.81	61.18	9.20	13.16	51.83	60.98	51.20	55.20	44.07
	TIGER-Q (Ours)	**50.34**	**61.79**	**9.40**	**13.44**	52.44	61.59	**52.00**	**55.80**	**44.60**

guarantees a decrease in the task-aware objective. Finally, the accepted updates are applied as $w^q \leftarrow w^q + \Delta w^q$, followed by a clipping operation to ensure all codes remain within the valid integer bounds, *i.e.*, $0 \leq w^q \leq 2^b - 1$. By leveraging exact objective changes and row-wise acceptance testing, it drives the quantized weights toward configurations that preserve the crucial task-specific capabilities encoded in the task-aware basis $\sigma_i \cdot v_i$. The processing flow is shown in Appendix D.

After the updating of the discrete codes, we further perform Eq. 13 to re-optimize the quantization scales s, so that the refined w^q is paired with its best task-aware quantization scales.

4 Experiments and Results

4.1 Experimental Setup

Datasets and models. We conduct extensive experiments by performing quantization on diverse models across multiple datasets. For vision tasks, we adopt the evaluation protocol from Ilharco et al. (2023) and quantize ViT-B/32 and ViT-L/14, both derived from CLIP (Radford et al., 2021), across eight standard image classification datasets. For generative language tasks, we select the WizardMath-V1.0 (7B and 13B) and WizardCoder-V1.0 (7B and 13B) families (Luo et al., 2024; 2025). Following a prior protocol (Yu et al., 2024), we evaluate the WizardMath on the GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021) benchmarks for mathematical reasoning capabilities, and evaluate the WizardCoder on the HumanEval (Chen et al., 2021) and MBPP (Austin et al., 2021) benchmarks for code generation capabilities. For discriminative language tasks, we follow Lu et al. (2024) and Tang et al. (2024) to evaluate the encoder model RoBERTa-Base (Liu et al., 2019) and the encoder-decoder model Flan-T5-Base (Raffel et al., 2020) on the GLUE benchmark (Wang et al., 2019) with 8 downstream tasks. All the models to be quantized are task-specific experts, having been individually fine-tuned from their pre-trained versions on their respective datasets, where WizardMath and WizardCoder are fine-tuned from the Llama2 and CodeLlama families (Touvron et al., 2023; Rozière et al., 2023), respectively. All scores are multiplied by 100 for better readability, with higher values indicating better performance across all metrics. More details about metrics are provided in Appendix C.

Comparison methods. We primarily compare our method with calibration-free baselines, including: Round-to-Nearest (RTN), HQQ (Badri & Shaji, 2023), Hadamard (Ashkboos et al., 2024)+RTN, and SINQ (Müller et al., 2025). We additionally include comparisons with AWQ (Lin et al., 2024), a widely adopted advanced calibration-based method, which serve as a strong performance reference to evaluate the relative effectiveness of our calibration-free method. Full Precision (FP) performance is provided as the theoretical upper bound.

Table 2: Comparison results of Flan-T5 and RoBERTa models on eight datasets from GLUE benchmark under 2-bit and 3-bit quantization, respectively. The best results are highlighted in **bold**.

	Method	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST2	STSB	Average
					Flan-T5					
FP	Original	74.98	83.41	87.50	91.49	85.37	85.92	93.58	88.70	86.37
2-Bit	RTN	48.89	2.32	54.16	59.01	60.96	52.34	84.40	32.95	49.38
	HQQ	65.77	52.63	76.47	83.74	78.72	66.06	74.77	40.64	67.35
	Hadamard+RTN	67.78	5.45	74.50	65.31	69.57	70.75	70.52	52.86	59.59
	SINQ	62.22	34.60	61.27	71.00	69.01	66.42	85.78	66.34	64.58
	TIGER-Q (Ours)	**70.75**	**72.94**	**83.33**	**86.71**	**80.86**	**74.36**	**87.50**	**81.67**	**79.76**
					RoBERTa					
FP	Original	55.11	87.40	87.98	92.42	89.46	65.34	94.72	87.70	82.52
3-Bit	RTN	0.00	45.63	69.68	53.65	**69.91**	56.31	59.97	26.00	47.64
	HQQ	0.00	38.26	72.72	49.46	45.52	52.71	70.64	33.88	45.39
	Hadamard+RTN	0.66	39.40	74.52	67.61	56.90	54.87	82.33	49.19	53.18
	SINQ	20.18	71.29	51.31	72.08	56.81	49.81	92.66	53.89	58.50
	TIGER-Q (Ours)	**26.37**	**74.76**	**79.81**	**79.95**	69.63	**59.56**	**92.89**	**78.00**	**70.12**

Implementation details. We apply group-wise weight quantization to the linear layers of the fine-tuned models, employing group sizes of 256 for ViT, 128 for WizardMath, WizardCoder, and RoBERTa, and 64 for Flan-T5. To evaluate the robustness of our task-aware guidance, we include 4-bit, 3-bit, and 2-bit weight quantization settings. For our TIGER-Q, the task vectors are directly extracted by computing the parameter difference between the downstream fine-tuned weights and the original pre-trained backbone weights. We apply SINQ (Müller et al., 2025) for an initial quantization parameters w^q and s then extract the task-aware basis to guide the refinement. We set the regularization hyperparameters α and β to $\frac{1}{MN}$ and 0.02 as default, respectively.

4.2 Comparisons on language tasks

We report the results of generative language tasks in Table 1. At 3 bits, TIGER-Q achieves an average of 38.24% across the eight reported scores, 2.04 points above SINQ (36.20%), the strongest calibration-free baseline, and close to calibration-based AWQ (38.72%). At 4 bits, TIGER-Q achieves an average of 44.60%, 0.29 points above the strongest calibration-free baseline (44.31%). It substantially outperforms calibration-free baselines and approaches the calibration-based AWQ.

We report the results of discriminative language tasks in Table 2. When quantizing fine-tuned models to extreme low-bit formats (*e.g.*, 2-bit for Flan-T5 and 3-bit for RoBERTa), existing calibration-free methods lead to notable performance drops. TIGER-Q gracefully mitigates this damage. For example, it reaches an average score of 79.76 on the 2-bit Flan-T5, outperforming the strongest HQQ (67.35) by 12.41 points. On 3-bit RoBERTa, it reaches 70.12, 11.62 points above the SOTA SINQ (58.50). These results confirm that our TIGER-Q effectively preserve task performance under low-bit quantization. More results of Flan-T5 at 3 bits and RoBERTa at 4 bits quantization are provided in Appendix B.4.

4.3 Comparisons on vision tasks

We present the image classification results for ViT-B/32 and ViT-L/14 in Table 3. The results highlight the advantage of our TIGER-Q under the 3-bit setting. For example, on the ViT-B/32, severe quantization errors cause calibration-free baselines like HQQ and SINQ to suffer catastrophic collapse on sensitive datasets like Cars (dropping to 29.87% and 23.69% respectively). In contrast, by leveraging the task-aware basis to guide optimization toward performance-preserving directions, TIGER-Q mitigates this degradation. Ultimately, it achieves a leading average accuracy of 79.61%, significantly outperforming SINQ by 5.61 points and proving its capability in preserving task performance. More results of ViT-L/14 at 4-bit quantization are provided in Appendix B.4.

Table 3: Comparison results of ViT-B/32 and ViT-L/14 across 8-task vision benchmarks under 4-bit and 3-bit quantization. FP represents the full-precision results. † denotes the calibration-based method. The best results of calibration-free methods are highlighted in **bold**.

	Method	SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD	Average
					ViT-B/32					
FP	Original	75.29	77.68	96.11	99.89	97.46	98.73	99.69	79.41	90.53
3-Bit	AWQ†	58.50	45.13	90.05	99.22	96.62	97.77	99.56	62.98	81.23
	RTN	34.05	17.37	80.89	97.22	94.51	94.65	99.52	53.83	71.51
	HQQ	41.58	29.87	83.41	97.67	94.84	94.87	99.33	52.87	74.31
	Hadamard+RTN	35.69	36.00	86.92	98.04	**96.32**	96.97	99.47	56.65	75.76
	SINQ	41.32	23.69	83.70	97.63	94.94	95.35	99.52	55.85	74.00
	TIGER-Q (Ours)	**54.51**	**40.74**	**88.35**	**98.11**	95.49	**97.11**	**99.53**	**63.03**	**79.61**
4-Bit	AWQ†	73.21	72.53	95.59	99.89	97.36	98.52	99.68	76.70	89.19
	RTN	71.03	69.28	94.73	99.81	97.18	98.56	99.69	75.05	88.17
	HQQ	72.25	71.31	94.75	99.85	97.13	98.56	99.65	76.01	88.69
	Hadamard+RTN	72.37	71.37	95.16	99.85	**97.24**	98.57	99.66	**77.77**	89.00
	SINQ	71.58	71.00	94.63	99.78	97.10	98.33	99.69	75.90	88.50
	TIGER-Q (Ours)	**72.53**	**72.34**	**94.94**	**99.85**	97.17	**98.57**	**99.70**	77.07	**89.02**
					ViT-L/14					
FP	Original	82.32	92.39	97.37	99.89	98.11	99.24	99.69	84.15	94.15
3-Bit	AWQ†	77.88	86.83	96.30	99.74	97.93	98.83	99.60	81.06	92.27
	RTN	73.29	80.16	94.16	99.63	97.68	98.14	99.65	74.68	89.67
	HQQ	75.23	82.56	95.56	99.59	97.52	98.33	99.56	**78.94**	90.91
	Hadamard+RTN	75.60	83.71	**95.62**	**99.81**	97.69	98.27	99.67	77.02	90.92
	SINQ	67.70	68.91	91.48	98.85	96.77	96.18	99.35	70.48	86.21
	TIGER-Q (Ours)	**75.72**	**84.54**	94.75	99.44	**97.75**	**98.33**	**99.68**	77.29	**90.94**

Table 4: Ablation on each component under the 3-bit quantization setting."○" represents setting all σ_i (Eq. 11) to 1.

task-aware basis	scale refinement	integer refinement	ViT-B/32	ViT-L/14
			74.00	86.21
✔	✔		77.45	86.70
✔		✔	76.85	90.41
✔	✔	✔	**79.61**	**90.94**
○	✔	✔	75.22	86.18
	✔	✔	73.90	86.03

Table 5: Analysis of applying TIGER-Q into 3-bit PTQ baselines.

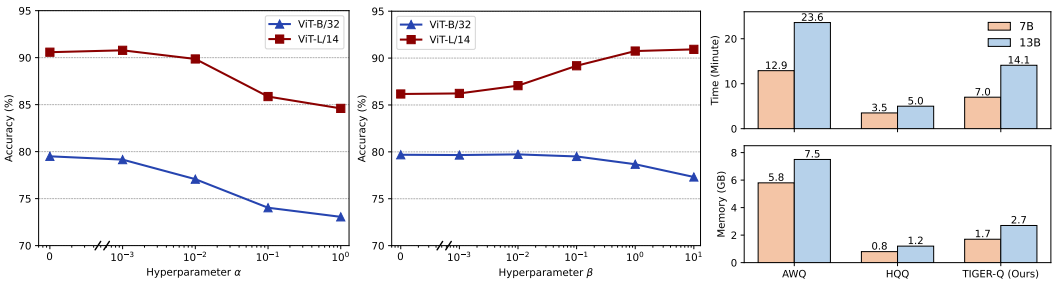
Method	ViT-B/32	ViT-L/14
RTN	$71.51_{(+7.21)}$	$89.67_{(+1.16)}$
HQQ	$74.31_{(+5.45)}$	$90.91_{(+0.95)}$
Hadamard	$75.76_{(+4.84)}$	$90.92_{(+1.49)}$
SINQ	$74.00_{(+5.61)}$	$86.21_{(+4.73)}$

5 Ablation and Analysis

Effectiveness of components. To evaluate the each component of our TIGER-Q, we report the average accuracy results of ViT-B/32 and ViT-L/14 under the 3-bit setting. As shown in Table 4, the baseline averages on ViT-B/32 and ViT-L/14 are 74.00% and 86.21%, respectively. With the task-aware basis (*i.e.*, task-aware error term in Eq. 11), scale refinement alone raises them to 77.45% and 86.70%, while integer-weight refinement alone raises them to 76.85% and 90.41%. Combining both refinements yields 79.61% and 90.94%. Removing the task-aware basis reduces the ViT-B/32 and ViT-L/14 averages from 79.61%/90.94% to 73.90%/86.03%. These severe degradations highlight the critical role of the task-aware basis in preserving target task performance, validating its effectiveness. Furthermore, as shown in the second-to-last row in Table 4, setting all singular-value weights to one degrades the performance to 75.22%/86.18%. It supports that the singular values σ_i play an important role, as they capture the relative importance of different task directions.

Regularization hyperparameters α and β. As observed in Fig. 2a, setting α too high would reduce performance. A larger α makes the objective in Eq. 11 focus mostly on the reconstruction error term, reducing the impact of the task-aware term, which further verify that the task-aware basis is crucial for target task performance. We show the analysis of the hyperparameter β in Fig. 2b. By default, we set it to 0.02 in our experiments and set it to 1 for ViT-L/14 to achieve better results.

Extensibility to baselines. We further assess the plug-and-play capability of TIGER-Q by applying it to several calibration-free baselines. Given that the dequantized weights are formulated as in Eq. 1, we can directly try optimizing them using TIGER-Q. Specifically, we utilize these base-



(a) α for the reconstruction error term (b) β for the stabilization term (c) Time (top) and Memory (bottom)

Figure 2: (a-b) Ablation on regularization hyperparameters α and β for ViT models under the 3-bit setting. (c) Time (minute) and memory (GB) overhead of the quantization process on WizardMath.

Table 6: Analysis of different Hessian proxies on ViT-B/32 under 3-bit quantization. $E[xx^\top]$ is a data-based proxy. CosSim: the mean cosine similarity between each proxy and the true input covariance $E[xx^\top]$. Overlap: dominant eigenspace overlap between each proxy and the $E[xx^\top]$ using the top 25% eigenvectors. R: a randomly initialized matrix. I: the identity matrix.

Proxy	CosSim	Overlap	SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD	Average
$\mathbb{E}[xx^\top]$	1.00	1.00	59.69	46.39	90.13	97.96	96.46	97.51	99.55	67.07	81.84
$R^\top R$	0.09	0.25	43.34	26.68	78.24	95.85	93.70	93.23	99.30	52.34	72.84
I	0.12	0.24	46.21	30.62	82.02	97.00	94.48	95.51	99.40	56.54	75.22
$\tau^\top\tau$ (Ours)	**0.34**	**0.50**	**54.51**	**40.74**	**88.35**	**98.11**	**95.49**	**97.11**	**99.53**	**63.03**	**79.61**

lines to establish the initial quantized integer weights w^q and quantization scales s and then use our task-aware refinement. Table 5 presents the average accuracy on the vision benchmarks, where the base scores represent the original baseline performance and the bracketed values represent the improvements brought by TIGER-Q. TIGER-Q improves the reported average score for both ViT models with each of the four tested calibration-free initializations. The empirical results suggest that TIGER-Q holds potential to act as a post-processing plugin to further improve calibration-free baselines. The full results of Table 5 are provided in Appendix B.1.

Effectiveness of the task-aware proxy. We further evaluate whether $\tau^\top\tau$ can effectively capture the task-relevant direction of the true input covariance $E[xx^\top]$. As shown in Table 6, $\tau^\top\tau$ reaches 0.34 cosine similarity and 0.50 dominant eigenspace overlap with the measured $E[xx^\top]$, higher than 0.12 and 0.24 for the identity proxy and 0.09 and 0.25 for the random proxy, respectively. This stronger direction alignment also translates into better average performance (79.61%), which is higher than the identity proxy (75.22%) and the random proxy (72.84%) and is close to the true $E[xx^\top]$ (81.84%). These results support that $\tau^\top\tau$ captures more of the task-relevant direction of the $E[xx^\top]$ than the two information-free proxies and provides an effective data-free proxy for guiding task-aware quantization in this test. Details of the two metrics are provided in the Appendix B.3.

Processing overhead. We evaluate the time and GPU memory of the quantization process on WizardMath in Fig. 2c. On WizardMath 7B/13B, TIGER-Q takes 7.0/14.1 minutes and uses 1.7/2.7 GB of GPU memory, compared with 3.5/5.0 minutes and 0.8/1.2 GB for the calibration-free HQQ and 12.9/23.6 minutes and 5.8/7.5 GB for the widely adopted calibration-based AWQ. Overall, the overhead of TIGER-Q falls in a practical range.

6 Conclusion

We propose TIGER-Q, a novel calibration-free quantization method that preserves the specialized capabilities of fine-tuned expert models. By extracting a task-aware basis from the task vector, TIGER-Q establishes a data-free proxy for the Hessian matrix to guide the quantization process. We solve this optimization objective through a decoupled pipeline, combining analytical scale refinement with an objective-driven local search for integer weights. Extensive experiments validate that TIGER-Q achieves superior performance over SOTA calibration-free baselines on both vision and language models at low bit-widths.

AI USE STATEMENT

We used Kimi K3 to improve the readability, check for typographical errors, and inspect code for potential bugs. We did not use generative AI to generate citations. The authors reviewed all AI-assisted suggestions and take responsibility for the final content of this work.

ETHICS STATEMENT

We confirm that this work complies with the ICLR Code of Ethics and does not involve human subjects, sensitive data, or any practices that raise ethical concerns.

REPRODUCIBILITY STATEMENT

We describe the components of our method in Sec. 3, the experimental settings in Sec. 4, and theoretical derivations and conditions in Appendix A. We will release the training and testing code upon publication to ensure full reproducibility.

REFERENCES

Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L. Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. Quarot: Outlier-free 4-bit inference in rotated llms. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen McAleer, Albert Q Jiang, Jia Deng, Stella Biderman, and Sean Welleck. Llemma: An open language model for mathematics. *arXiv preprint arXiv:2310.10631*, 2023.

Hicham Badri and Appu Shaji. Half-quadratic quantization of large machine learning models, November 2023.

Jerry Chee, Yaohui Cai, Volodymyr Kuleshov, and Christopher De Sa. Quip: 2-bit quantization of large language models with guarantees. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Runxi Cheng, Feng Xiong, Yongxian Wei, Wanyun Zhu, and Chun Yuan. Whoever started the interference should end it: Guiding data-free model merging via task vectors. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, Proceedings of Machine Learning Research. PMLR / OpenReview.net, 2025.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168, 2021.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

Xin Ding, Xiaoyu Liu, Zhijun Tu, Yun Zhang, Wei Li, Jie Hu, Hanting Chen, Yehui Tang, Zhiwei Xiong, Baoqun Yin, and Yunhe Wang. CBQ: cross-block quantization for large language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

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.

Elias Frantar and Dan Alistarh. Optimal brain compression: A framework for accurate post-training quantization and pruning. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

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

Cong Guo, Yuxian Qiu, Jingwen Leng, Xiaotian Gao, Chen Zhang, Yunxin Liu, Fan Yang, Yuhao Zhu, and Minyi Guo. Squant: On-the-fly data-free quantization via diagonal hessian approximation. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021.

Gabriel Ilharco, Marco Túlio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

Youngeun Kim, Seunghwan Lee, Aecheon Jung, Bogon Ryu, and Sungeun Hong. Task vector quantization for memory-efficient model merging. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20105–20115. IEEE, 2025.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloé Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross B. Girshick. Segment anything. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pp. 3992–4003. IEEE, 2023.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857, 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.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. AWQ: activation-aware weight quantization for on-device LLM compression and acceleration. In *Proceedings of the Seventh Annual Conference on Machine Learning and Systems, MLSys 2024, Santa Clara, CA, USA, May 13-16, 2024*. mlsys.org, 2024.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.

Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. Spinquant: LLM quantization with learned rotations. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

Zhenyi Lu, Chenghao Fan, Wei Wei, Xiaoye Qu, Dangyang Chen, and Yu Cheng. Twin-merging: Dynamic integration of modular expertise in model merging. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jian-Guang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, Yansong Tang, and Dongmei Zhang. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

Ziyang Luo, Can Xu, Pu Zhao, Qingfeng Sun, Xiubo Geng, Wenxiang Hu, Chongyang Tao, Jing Ma, Qingwei Lin, and Daxin Jiang. Wizardcoder: Empowering code large language models with evol-instruct. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

Lorenz K. Müller, Philippe Bich, Jiawei Zhuang, Ahmet Çelik, Luca Benfenati, and Lukas Cavigelli. SINQ: sinkhorn-normalized quantization for calibration-free low-precision LLM weights. *CoRR*, abs/2509.22944, 2025.

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.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744, 2022.

Bowen Ping, Shuo Wang, Hanqing Wang, Xu Han, Yuzhuang Xu, Yukun Yan, Yun Chen, Baobao Chang, Zhiyuan Liu, and Maosong Sun. Delta-come: Training-free delta-compression with mixed-precision for large language models. *Advances in Neural Information Processing Systems*, 37:31056–31077, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, Proceedings of Machine Learning Research, pp. 8748–8763. PMLR, 2021.

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

Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton-Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, and Gabriel Synnaeve. Code llama: Open foundation models for code. *CoRR*, abs/2308.12950, 2023.

Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Peng Gao, Yu Qiao, and Ping Luo. Omniquant: Omnidirectionally calibrated quantization for large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

Anke Tang, Li Shen, Yong Luo, Han Hu, Bo Du, and Dacheng Tao. Fusionbench: A comprehensive benchmark of deep model fusion. *CoRR*, abs/2406.03280, 2024.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971, 2023.

Albert Tseng, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa. Quip#: Even better LLM quantization with hadamard incoherence and lattice codebooks. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, pp. 48630–48656. PMLR / OpenReview.net, 2024.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Yue Wang, Weishi Wang, Shafiq Joty, and Steven CH Hoi. Codet5: Identifier-aware unified pre-trained encoder-decoder models for code understanding and generation. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 8696–8708, 2021.

Guangxuan Xiao, Ji Lin, Mickaël Seznec, Hao Wu, Julien Demouth, and Song Han. Smoothquant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, Proceedings of Machine Learning Research, pp. 38087–38099. PMLR, 2023.

Bo Xu, Haotian Wu, Hehai Lin, Weiquan Huang, Beier Zhu, Yao Shu, and Chengwei Qin. Acemerging: Data-free model merging with adaptive covariance estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29472–29482, June 2026.

Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. Adamerging: Adaptive model merging for multi-task learning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, pp. 57755–57775. PMLR / OpenReview.net, 2024.

A Details of the Theoretical Derivations and Conditions

In this section, we provide further details on the theoretical derivations and conditions used in Eq. 8.

Approximation 1: gradient and input decoupling. The first approximation in Eq. 8 assumes weak dependence between $\|\delta_u\|_2^2$ and $x_u x_u^\top$. We provide a sufficient condition for this approximation. Let

$$a_u = \|\delta_u\|_2^2, \qquad \Sigma_u = \mathbb{E}[x_u x_u^\top], \tag{16}$$

and define

$$\Sigma_u(a) = \mathbb{E}[x_u x_u^\top \mid a_u = a]. \tag{17}$$

Conditioning on a_u gives

$$\mathbb{E}[a_u x_u x_u^\top] = \mathbb{E}_a\left[a\Sigma_u(a)\right]. \tag{18}$$

Therefore, the exact error introduced by the decoupling approximation is

$$\mathbb{E}[a_u x_u x_u^\top] - \mathbb{E}[a_u]\Sigma_u = \mathbb{E}_a\left[a\left(\Sigma_u(a) - \Sigma_u\right)\right]. \tag{19}$$

Suppose that

$$\|\Sigma_u(a) - \Sigma_u\|_2 \le \varepsilon_u \tag{20}$$

for different gradient energy levels a. Then

$$\begin{aligned}
\left\|\mathbb{E}[a_u x_u x_u^\top] - \mathbb{E}[a_u]\Sigma_u\right\|_2 &= \left\|\mathbb{E}_a\left[a\left(\Sigma_u(a) - \Sigma_u\right)\right]\right\|_2 && (21)\\
&\le \mathbb{E}_a\left[\|a(\Sigma_u(a) - \Sigma_u)\|_2\right] = \mathbb{E}_a\left[a\|\Sigma_u(a) - \Sigma_u\|_2\right] && (22)\\
&\le \mathbb{E}_a[a\varepsilon_u] = \mathbb{E}[a_u]\varepsilon_u. && (23)
\end{aligned}$$

Hence, the approximation error is small when samples associated with different gradient magnitudes have similar activation second moments. Intuitively, samples producing large gradients may receive greater weights, but they should not be concentrated in an activation subspace substantially different from that of the overall task distribution. This condition is considerably weaker than full independence between δ_u and x_u. Full independence would imply $\Sigma_u(a) = \Sigma_u$ exactly, whereas our condition only requires their difference to be bounded and constrains only the conditional second moment of x_u, rather than its complete conditional distribution.

Approximation 2: input statistics drift. After the above approximation, the expected self terms in Eq. 7 become

$$\sum_{u=1}^{T} \gamma_u^2 \mathbb{E}[\|\delta_u\|_2^2]\Sigma_u = \lambda\bar{\Sigma}, \tag{24}$$

where

$$\lambda = \sum_{u=1}^{T} \gamma_u^2 \mathbb{E}[\|\delta_u\|_2^2], \qquad \bar{\Sigma} = \sum_u \frac{\gamma_u^2 \mathbb{E}[\|\delta_u\|_2^2]}{\lambda}\Sigma_u. \tag{25}$$

Thus, $\bar{\Sigma}$ is a weighted average of the input second moments along the fine-tuning trajectory. Let

$$\Sigma_\star = \mathbb{E}[xx^\top] \tag{26}$$

denote the input second moment of the final fine-tuned model. If

$$\|\Sigma_u - \Sigma_\star\|_2 \le \kappa_u, \tag{27}$$

then

$$\|\bar{\Sigma} - \Sigma_\star\|_2 = \left\|\sum_{u=1}^{T} \frac{\gamma_u^2 \mathbb{E}[\|\delta_u\|_2^2]}{\lambda}(\Sigma_u - \Sigma_\star)\right\|_2 \le \sum_{u=1}^{T} \frac{\gamma_u^2 \mathbb{E}[\|\delta_u\|_2^2]}{\lambda}\kappa_u. \tag{28}$$

Therefore, $\bar{\Sigma}$ remains close to $\mathbb{E}[xx^\top]$ when the input statistics change only mildly during optimization steps with large weights $\frac{\gamma_u^2 \mathbb{E}[\|\delta_u\|_2^2]}{\lambda}$. This gives a sufficient condition for the second approximation in Eq. 8.

Table 7: Full analysis results of applying TIGER-Q to calibration-free baselines with 3-bit quantization, which is corresponding to Table 5. The gray background represents using our TIGER-Q on the calibration-free baselines.

Method	SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD	Avg
ViT-B/32									
RTN	34.05	17.37	80.89	97.22	94.51	94.65	99.52	53.83	71.51
RTN *w.* TIGER-Q	51.83	39.44	87.89	97.78	95.62	96.83	99.57	60.80	78.72
HQQ	41.58	29.87	83.41	97.67	94.84	94.87	99.33	52.87	74.31
HQQ *w.* TIGER-Q	53.69	46.79	87.65	98.37	95.09	96.62	99.51	60.32	79.76
Hadamard+RTN	35.69	36.00	86.92	98.04	96.32	96.97	99.47	56.65	75.76
Hadamard+RTN *w.* TIGER-Q	52.63	48.14	88.98	98.37	96.72	97.55	99.52	62.87	80.60
SINQ	41.32	23.69	83.70	97.63	94.94	95.35	99.52	55.85	74.00
SINQ *w.* TIGER-Q	54.51	40.74	88.35	98.11	95.49	97.11	99.53	63.03	79.61
ViT-L/14									
RTN	73.29	80.16	94.16	99.63	97.68	98.14	99.65	74.68	89.67
RTN *w.* TIGER-Q	76.06	83.71	94.73	99.67	97.76	98.27	99.66	76.76	90.83
HQQ	75.23	82.56	95.56	99.59	97.52	98.33	99.56	78.94	90.91
HQQ *w.* TIGER-Q	77.59	85.56	95.62	99.63	97.77	98.53	99.62	80.53	91.86
Hadamard+RTN	75.60	83.71	95.62	99.81	97.69	98.27	99.67	77.02	90.92
Hadamard+RTN *w.* TIGER-Q	78.73	87.05	96.52	99.78	97.96	98.83	99.69	80.74	92.41
SINQ	67.70	68.91	91.48	98.85	96.77	96.18	99.35	70.48	86.21
SINQ *w.* TIGER-Q	75.72	84.54	94.75	99.44	97.75	98.33	99.68	77.29	90.94

Approximation 3: cross-step term contribution. A complementary intuition comes from the temporal structure of fine-tuning. During the early stage, the model adapts consistently toward the downstream objective, so updates from nearby steps tend to share a coherent task-related direction. For such step pairs, $\delta_u^\top \delta_v$ is likely to be positive, and the corresponding cross terms may reinforce directions repeatedly used during task adaptation. As fine-tuning approaches convergence, the task-directed component of the gradient becomes smaller, while the remaining mini-batch updates become less directionally coherent. The magnitude of each cross term is bounded by $\|x_u(\delta_u^\top \delta_v)x_v^\top\| \le \|x_u\|_2\|x_v\|_2\|\delta_u\|_2\|\delta_v\|_2$. Consequently, cross terms involving a late-stage update are suppressed by its smaller gradient magnitude. Cross terms between late-stage updates may be further reduced because their directions are more dispersed, causing their aggregate contributions to partially cancel. Thus, cross terms do not necessarily destroy the task-relevant subspace, although they weaken the claim that $\tau^\top\tau$ is strictly proportional to $\mathbb{E}[xx^\top]$.

Implication for the task-aware proxy. Combining the above conditions gives

$$\mathbb{E}[\tau^\top\tau] \approx \lambda\mathbb{E}[xx^\top] + \mathbb{E}[R_{\text{cross}}]. \tag{29}$$

When the gradient and input decoupling error, the input statistics drift, and the aggregate cross term are sufficiently small, this relation motivates

$$\mathbb{E}[\tau^\top\tau] \approx \lambda\mathbb{E}[xx^\top]. \tag{30}$$

The observed $\tau^\top\tau$ is a single realization of this quantity. Therefore, we use

$$\tau^\top\tau \approx \lambda\mathbb{E}[xx^\top] \tag{31}$$

as a practical approximation. Note that our goal is not to prove that the two are equal, but rather to show that $\tau^\top\tau$ is an intuitively reasonable and practical proxy. We do not require strict proportionality between the two matrices. It is sufficient for the dominant subspace of $\tau^\top\tau$ to capture the task-relevant directions of $\mathbb{E}[xx^\top]$. The direction alignment analysis in Appendix B.3 provides empirical support for this property.

B Full Results and Further Analysis

B.1 Full Results of the Extensibility Analysis

Table 7 provides the detailed results on each dataset for the experiments summarized in Table 5. As shown in the table, integrating TIGER-Q as a plug-in module brings consistent improvements across

Table 8: Further ablation on row-wise acceptance using ViT-B/32 at 3-bit quantization.

row-wise acceptance	SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD	Avg
✓	54.51	40.74	88.35	98.11	95.49	97.11	99.53	63.03	79.61
×	54.51	40.59	88.35	98.04	95.48	97.11	99.51	63.03	79.57

Table 9: Detailed structural similarity between different Hessian proxies and the true input covariance $E[xx^\top]$ on ViT-B/32 across eight vision tasks. Dominant Eigenspace Overlap is computed using the top 25% eigenvectors. Higher values indicate stronger structural alignment. R denotes a randomly initialized matrix and I denotes the identity matrix.

Proxy	SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD	Avg
Dominant Eigenspace Overlap in Eq. 32									
$R^\top R$	0.250	0.250	0.249	0.250	0.249	0.250	0.249	0.250	0.250
I	0.238	0.244	0.241	0.239	0.249	0.243	0.241	0.240	0.241
$\tau^\top\tau$ (Ours)	**0.500**	**0.483**	**0.536**	**0.489**	**0.484**	**0.527**	**0.508**	**0.497**	**0.502**
Cosine Similarity in Eq. 33									
$R^\top R$	0.117	0.128	0.085	0.071	0.048	0.112	0.149	0.077	0.098
I	0.152	0.166	0.111	0.092	0.063	0.146	0.196	0.099	0.128
$\tau^\top\tau$ (Ours)	**0.339**	**0.382**	**0.372**	**0.287**	**0.245**	**0.388**	**0.360**	**0.337**	**0.339**

nearly all downstream datasets. Our task-aware refinement increases the base accuracy across different baseline methods including RTN, HQQ, Hadamard+RTN, and SINQ. These broad improvements show that TIGER-Q generalizes well, suggesting its further potential to be used as a post-processing plugin for existing calibration-free quantization methods.

B.2 Further analysis of row-wise acceptance

As shown in Table 8, removing the row-wise acceptance mechanism causes slight performance degradations on specific datasets (e.g., Cars and EuroSAT) while leaving others unaffected. Row-wise acceptance prevents accepted combined proposals from increasing the objective in Eq. 11. In the ViT-B/32 ablation, it changes the average score from 79.57 to 79.61.

B.3 Direction Alignment of the Task-Aware Proxy

We further examine how well $\tau^\top\tau$ captures the task-relevant direction of the true input covariance $\mathbb{E}[xx^\top]$ using two complementary metrics. First, we measure the Dominant Eigenspace Overlap using the top k eigenvectors, where k is set to 25% of the matrix dimension. Let $V_{P,k}$ and $V_{\mathbb{E},k}$ denote the top k eigenvectors of a proxy P and $\mathbb{E}[xx^\top]$, respectively. The normalized overlap is defined as

$$\mathrm{Overlap}(P, \mathbb{E}[xx^\top]) = \frac{1}{k}\left\|V_{P,k}^\top V_{\mathbb{E},k}\right\|_F^2. \tag{32}$$

Second, we measure the overall matrix alignment using cosine similarity

$$\mathrm{CosSim}(P, \mathbb{E}[xx^\top]) = \frac{\langle P, \mathbb{E}[xx^\top]\rangle_F}{\|P\|_F \|\mathbb{E}[xx^\top]\|_F}, \tag{33}$$

where $\langle\cdot,\cdot\rangle_F$ denotes the Frobenius inner product. The higher these two metrics are, the more aligned the directions of the two matrices are. As shown in Table 9, $\tau^\top\tau$ achieves an average eigenspace overlap of 0.502, substantially higher than 0.250 for $R^\top R$ and 0.241 for I. It also achieves the highest average cosine similarity of 0.339, compared with 0.098 and 0.128, respectively. These consistent improvements across all eight tasks show that $\tau^\top\tau$ captures both the dominant eigenspace and the overall structure of $\mathbb{E}[xx^\top]$ more effectively than the information free proxies, supporting its use as a practical proxy for task-aware quantization.

B.4 Full Comparison Results

We provide full results of 3-bit Flan-T5, 4-bit RoBERTa, and 4-bit ViT-L/14 in Table 10 and Table 11, respectively. At relatively higher bit-widths, the performance improvements of TIGER-Q

Table 10: Comparison results of Flan-T5 and RoBERTa models on eight datasets from GLUE benchmark under 3-bit and 4-bit quantization, respectively. The best results of calibration-free methods are highlighted in **bold**.

	Method	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST2	STSB	Average
					Flan-T5					
3-Bit	RTN	75.55	81.79	85.04	90.95	84.76	**85.92**	92.88	**87.90**	85.60
	HQQ	74.88	81.16	86.76	90.99	83.54	81.58	92.66	87.26	84.85
	Hadamard+RTN	75.17	81.25	85.53	**91.23**	**85.10**	81.58	**93.80**	87.43	85.14
	SINQ	74.97	81.94	87.50	91.19	84.94	84.83	93.23	87.26	85.73
	TIGER-Q (Ours)	**75.83**	**82.27**	**88.23**	91.03	84.72	85.20	92.89	87.35	**85.94**
					RoBERTa					
4-Bit	RTN	43.58	79.24	58.69	77.70	80.04	58.12	93.69	71.07	70.26
	HQQ	35.12	83.13	76.35	84.99	84.24	62.09	93.34	79.71	74.87
	Hadamard+RTN	53.52	84.62	84.86	90.82	87.26	61.73	89.90	84.73	79.68
	SINQ	55.11	86.47	**87.63**	91.12	**88.73**	**65.70**	**94.72**	86.34	**81.97**
	TIGER-Q (Ours)	**55.17**	**86.55**	87.20	**91.73**	88.51	65.34	94.61	**86.66**	**81.97**

Table 11: Comparison results of ViT-L/14 across 8-task vision benchmarks under 4-bit quantization. † denotes the calibration-based method. The best results of calibration-free methods are highlighted in **bold**.

	Method	SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD	Average
4-Bit	AWQ†	81.80	91.58	97.25	99.81	98.06	99.18	99.67	84.63	93.99
	RTN	81.35	91.28	**97.29**	99.85	**98.11**	99.12	99.71	83.14	93.73
	HQQ	81.34	**91.44**	97.22	99.85	98.02	99.17	**99.73**	83.72	93.81
	Hadamard+RTN	81.52	91.28	97.27	**99.89**	98.08	99.14	99.71	83.67	93.82
	SINQ	81.48	91.41	97.22	99.85	98.08	99.19	99.69	83.62	93.81
	TIGER-Q (Ours)	**81.64**	91.23	97.19	99.85	**98.11**	**99.21**	99.69	**83.72**	**93.83**

tend to saturate. This aligns with our expectations, since the stronger representational capacity of higher bit-widths allows the optimization space of quantization to overlap with the task-sensitive subspace. Although performance gains tend to saturate at higher bit-widths, this does not affect the contributions and conclusions regarding our TIGER-Q. The larger representational capacity at higher bit-widths allows most calibration-free methods to reach performance saturation, the performance gaps among different methods naturally appear narrower. By significantly mitigating the performance degradation at low bit-widths, we believe TIGER-Q takes a positive step toward making calibration-free quantization more effective and reliable.

C Details of Evaluation Metrics.

For the vision tasks, we evaluate the quantized ViT-B/32 and ViT-L/14 models using the standard Top-1 accuracy across all eight image classification datasets. For the generative language tasks, we follow standard evaluation protocols for the respective domains. Specifically, the mathematical reasoning capabilities of the WizardMath models on the GSM8K and MATH benchmarks are measured using the exact match accuracy. The code generation capabilities of the WizardCoder models on the HumanEval and MBPP benchmarks are evaluated using the pass@1 metric. For the discriminative language tasks on the GLUE benchmark, we adopt specific metric settings corresponding to the original evaluation protocols of the baselines. For the encoder model RoBERTa-Base, following the specific evaluation script of Lu et al. (2024), we report the Matthew's Correlation Coefficient (MCC) for CoLA, the average of Accuracy and F1 score for MRPC and QQP, the average of Pearson and Spearman correlation coefficients for STS-B, the average accuracy of the matched and mismatched splits for MNLI, and the accuracy for the remaining tasks (QNLI, RTE, and SST-2). For the encoder-decoder model Flan-T5-Base, following Tang et al. (2024), the downstream tasks are formulated as text-to-text generation. The performance is evaluated using Spearman's rank correlation coefficient for STS-B, and the exact match accuracy between the generated text and the target labels for the remaining tasks.

Algorithm 1: Task-Aware Integer Weight Refinement

Input: Initial integer weights w^q, scales s, zero-points z, full-precision weights w
Output: Refined integer weights w^q
for $m = 1$ **to** M **do**
  $\Delta w^q_m \leftarrow \mathbf{0}$; `// Initialize row sparse proposal`
  `// Evaluate local moves and generate group-wise proposal`
  for $t = 1$ **to** G **do**
    $\forall n \in \text{group}(t) : \Delta w^{q\star}_{m,n} \leftarrow \arg\max_{\delta \in \{\pm 1\}} \{-\Delta \mathcal{J}_{m,n}(\delta) \mid w^q_{m,n} + \delta \in [0, 2^b - 1]\}$;
    $\forall n \in \text{group}(t) : \mathcal{R}_{m,n} \leftarrow \max(0, -\Delta \mathcal{J}_{m,n}(\Delta w^{q\star}_{m,n}))$;
    $n^\star \leftarrow \arg\max_{n \in \text{group}(t)} \mathcal{R}_{m,n}$;
    if $\mathcal{R}_{m,n^\star} > 0$ **then** $\Delta w^q_{m,n^\star} \leftarrow \Delta w^{q\star}_{m,n^\star}$;
  $\Delta \widehat{w}_m \leftarrow [s_{m,1} \Delta w^q_{m,1}, \ldots, s_{m,G} \Delta w^q_{m,G}]^\top$
  Compute exact row-wise objective change $\Delta \mathcal{J}_m$;
  if $\Delta \mathcal{J}_m \geq 0$ **then**
    $n^{\text{best}} \leftarrow \arg\max_n \mathcal{R}_{m,n}$ across the entire row m;
    $\Delta w^q_m \leftarrow \mathbf{0}$;
    if $\mathcal{R}_{m,n^{\text{best}}} > 0$ **then**
      $\Delta w^q_{m,n^{\text{best}}} \leftarrow \Delta w^{q\star}_{m,n^{\text{best}}}$;
  $w^q_m \leftarrow \text{clip}(w^q_m + \Delta w^q_m, 0, 2^b - 1)$;
return w^q;

D Processing Flow of Task-Aware Integer Weight Refinement

The processing flow is shown in the Algorithm 1

E Limitation

While TIGER-Q preserves task-specific capabilities without calibration data, its current optimization scope involves certain practical simplifications. Specifically, our decoupled refinement pipeline focuses on optimizing the group-wise continuous scales and discrete integer weights, leaving the zero-points (z) fixed to their initial values. Furthermore, some advanced quantization formats may incorporate auxiliary parameters, such as additional per-input-channel scales, which are not refined within our framework. Extending the task-aware optimization objective to jointly refine zero-points and these supplementary quantization parameters represents a direction for future work.