2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Task-Aware and Structure-Knowledge-guided Quantization for End-to-End YOLO Object Detection

Minghua Zhu, Liangwei Li, Shunan Zhou, Jingfei Jiang*, and Jinwei Xu

***Abstract*— The YOLO series of models are pivotal for real-time object detection, yet their deployment on resource-constrained edge devices necessitates effective model compression. Post-Training Quantization (PTQ) offers a promising, low-cost solution, but existing methods, primarily designed for classification tasks, often lead to significant performance degradation when applied to YOLO models. In this paper, we systematically analyze the key challenges in quantizing YOLO architectures. We identify three primary obstacles: (1) the high sensitivity of detection tasks to quantization errors, exacerbated by the non-linear IoU metric; (2) the pronounced long-tail distribution of activations, particularly with the SiLU function, which complicates low-bit quantization; and (3) the structural heterogeneity of the multi-scale, multi-task detection head, which renders conventional block-wise quantization strategies ineffective. To address these challenges, we propose a novel framework, Task-Aware and Structure-Knowledge-guided Quantization (TASKQ). Our framework introduces three key components: a sparse quantization strategy to mitigate the impact of long-tailed activations, a Detection-aware Task Regularization (DTR) mechanism that incorporates IoU-based loss to guide parameter fine-tuning, and a Scale-and-Task-Aware Head-wise Quantization (STAHQ) scheme that aligns quantization granularity with the head's functional structure. Extensive experiments on various YOLO models demonstrate that TASKQ significantly outperforms existing PTQ methods, especially in low-bit scenarios, establishing a new state-of-the-art for end-to-end YOLO quantization.**

I. Introduction

Object detection [1]–[5] has been widely applied in various domains such as autonomous driving, industrial inspection, transportation, and medical image analysis.Its core task is to achieve efficient and accurate localization and classification of objects, a process where model performance is critical for the system's real-time responsiveness and reliability.The YOLO series has emerged as the predominant choice for these applications, primarily due to its end-to-end, one-stage detection paradigm that offers an effective balance between inference speed and accuracy.However, with the proliferation of edge devices and mobile terminals, models face stringent constraints in terms of computation, memory, and power consumption, making model compression techniques [6] crucial for ensuring efficient deployment and wide applicability of object detection models.

Among various compression techniques [7], [8], model quantization is highly effective for reducing computational overhead [9], which is broadly categorized into Quantization-Aware Training (QAT) and Post-Training Quantization (PTQ). Notably, PTQ has attracted widespread attention in practice since it does not require labeled data or additional retraining.

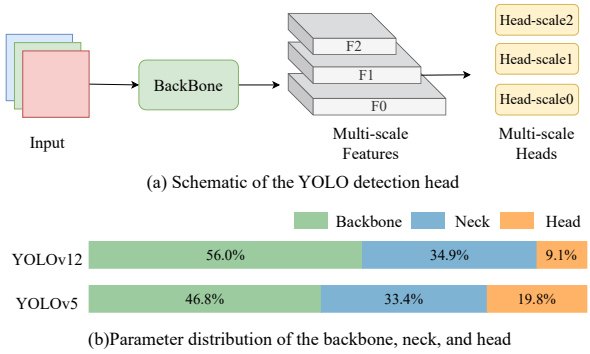


Fig. 1. Details of the YOLO model architecture. (a) Schematic of the YOLO detection head, illustrating its multi-scale parallel branch design for classification and regression. (b) Parameter distribution of the backbone, neck, and head in YOLOv5 and YOLOv12, highlighting the significant parameter proportion of the detection head.

Despite the success of PTQ in image classification tasks, directly applying these methods to detection models such as YOLO often results in significant performance degradation. This is mainly because most existing approaches focus on the quantization of the backbone and neck while neglecting the detection head, which plays a pivotal role in multi-task prediction and bounding box regression. Unlike the simple heads of classification models, the YOLO detection head is fundamentally more complex. It features a multi-scale architecture with parallel branches for classification and regression, handling high structural and task heterogeneity (Fig. 1(a)). Moreover, it constitutes a substantial portion of the model's parameters, accounting for 19.8% in YOLOv5 and 9.1% in YOLOv12 (Fig. 1(b)). These characteristics make the detection head highly sensitive to quantization errors, and methods that overlook this critical component inevitably fail to maintain performance.

Furthermore, in exploring efficient low-bit quantization of YOLO models, we identify two additional key challenges: (1) due to task heterogeneity and the central role of intersection over union (IoU) in bounding box overlap computation, detection models are significantly more sensitive to quantization errors than classification models; and (2) the activation distribution of YOLO models demonstrates a pronounced long-tail property, and the SiLU activation

The authors are with the National Key Laboratory of Parallel and Distributed Computing, College of Computer Science and Technology, National University of Defense Technology, Changsha, China.(*Corresponding author)

function exacerbates this issue, further complicating low-bit quantization. Based on these observations, we propose a PTQ framework specifically tailored for YOLO detection models, termed **T**ask-**A**ware and **S**tructure-**K**nowledge-guided **Q**uantization (**TASKQ**). The framework incorporates sparse quantization to alleviate the adverse effects of long-tail property, and introduces a **Detection-aware Task Regularization** (DTR) mechanism to better learn quantization parameters for regression tasks while preserving feature representation. Additionally, we introduce a **Scale-and-Task-Aware Head-wise Quantization** (STAHQ) strategy designed for the heterogeneous detection head. The code is available at `https://github.com/MingleZhu7/taskq`.

The main contributions of this paper are summarized as follows:

- We systematically analyze the challenges of low-bit quantization in YOLO detection models, highlighting that the heterogeneous detection head, long-tailed activation distributions, and high sensitivity of detection tasks to quantization errors are the primary difficulties.
- We propose TASKQ, a novel PTQ framework for YOLO detection models, which integrates three key components: Sparse Quantization, Detection-aware Task Regularization (DTR), and Scale-and-Task-Aware Head-wise Quantization (STAHQ).
- We conduct extensive evaluations on multiple YOLO variants. Experimental results demonstrate that TASKQ achieves superior performance compared with existing methods, especially under low-bit quantization scenarios.

II. RELATED WORK

A. YOLO for Object Detection

The You Only Look Once (YOLO) [5] series has become a mainstream approach in object detection, primarily due to its end-to-end, one-stage paradigm that provides an excellent trade-off between speed and accuracy. The series has continuously evolved: YOLOv5 [10] is widely adopted for its engineering-friendly and lightweight design; YOLOv7 [11] and YOLOv8 [12] introduce further optimizations in head architecture, feature fusion, and label assignment strategies; and the latest YOLOv12 [13] incorporates regional attention mechanisms to enhance small object detection while maintaining real-time performance.

Given their high inference efficiency and widespread deployment demand on edge devices, YOLO models have become a primary focus for low-bit quantization research. However, their complex multi-task detection heads and structural heterogeneity present significant challenges for quantization.

B. Post-Training Quantization

Current quantization methods are broadly categorized into two main categories: Quantization-Aware Training (QAT) and Post-Training Quantization (PTQ). To recover performance, QAT [14]–[17] relies on a full retraining process, which is resource-intensive. In contrast, PTQ requires only a small unlabeled calibration set and offers a significantly lower quantization cost.

Recently, several PTQ approaches have achieved remarkable success on classification tasks by optimizing rounding values. Although these methods introduce a fine-tuning step during quantization, they fundamentally differ from QAT. While QAT adjusts model weights using the entire labeled training dataset, these PTQ techniques optimize quantization parameters using only a small amount of unlabeled data. For instance, AdaRound [18] first proposed an adaptive rounding strategy to optimize rounding values by minimizing the layer-wise L_2 loss. BRECQ [19] introduced a second-order Taylor expansion of the Hessian matrix to guide the quantization loss of each block. Subsequently, QDrop [20] integrated random activation dropping into the fine-tuning process. MRECG [21] introduced the concept of module capacity, jointly optimizing modules with large capacity disparities to smooth loss oscillations and reduce the final reconstruction error. More recently, PD-Quant [22] utilized a prediction difference loss (PD-loss) as an effective approximation of the task loss.

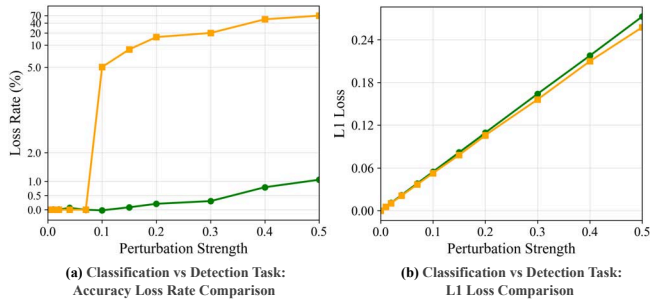
However, these works, primarily developed for classification models, are rarely evaluated on detection models or are limited to quantizing only the backbone and neck. Directly applying these techniques to YOLO-based object detection models often leads to significant performance degradation. For the quantization of detection models, DetPTQ [23] investigated the impact of the L_p metric and guided the selection of p using an object detection output loss. Reg-PTQ [24] employed a log-affine transformation to better characterize non-uniformly distributed parameters. Nevertheless, these methods often focus on minimizing Lp-norm errors, which fails to capture the crucial geometric constraints of the IoU metric. Furthermore, their design is typically centered on the backbone and neck, failing to adequately address the unique characteristics of the detection head.As a consequence, preserving detection performance often requires keeping the detection head at higher precision, which introduces additional computational overhead compared to low-bit quantization.Our goal is to bridge this gap by achieving an effective balance between low-bit quantization of the detection head and the preservation of overall detection performance.

In contrast, we propose a framework tailored to the heterogeneous YOLO detection head and incorporate an IoU-aware loss function, which to our knowledge is the first effective low-bit PTQ scheme for YOLO detection models.

III. MOTIVATION

A. Quantization Sensitivity in Object Detection

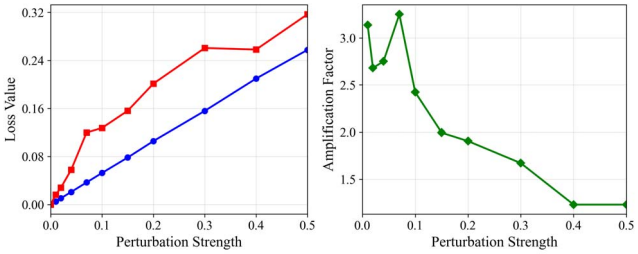
To compare the sensitivity of classification and detection models to quantization-induced errors, we employ YOLOv5s as a unified backbone for both tasks. The two models share identical feature extraction networks, differing only in the task-specific head: the detection model employs the standard YOLO detection head, whereas the classification model replaces it with a classification head.Both models are



(a) Classification vs Detection Task: Accuracy Loss Rate Comparison

(b) Classification vs Detection Task: L1 Loss Comparison

Fig. 2. Comparison of sensitivity to quantization error. The plot shows performance degradation for the classification task (green line) and the regression task (orange line) as the perturbation strength (σ) increases. The regression task exhibits significantly higher sensitivity, with its performance (mAP) dropping sharply compared to that of classification task (Accuracy).



(a) L1 Loss vs Delta IoU (Absolute Values)

(b) Error Amplification: Delta IoU / L1 Loss

Fig. 3. Comparison of error trends and the amplification effect between L1 Loss and ΔIoU. (a) Variation of L1 Loss (blue line) and ΔIoU (red line) under increasing perturbation strength. It is observable that ΔIoU consistently and significantly exceeds the corresponding L1 Loss. (b) The ratio of ΔIoU to L1 Loss, which quantifies the error amplification.

pretrained on their respective datasets, i.e., ImageNet [25] for classification and COCO [26] for detection. To simulate quantization-induced perturbations, we inject Gaussian noise $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$ into the input activation $\mathbf{a}$ of the final layer, resulting in the perturbed activation $\mathbf{a}' = \mathbf{a} + \boldsymbol{\varepsilon}$.

For classification, the logit is computed as $\ell = \mathbf{w}^\top \mathbf{a}$, and the perturbation leads to a shift $\Delta\ell = \mathbf{w}^\top \boldsymbol{\varepsilon}$, with variance $\mathrm{Var}(\Delta\ell) = \sigma^2 \|\mathbf{w}\|_2^2$. Using $|\boldsymbol{\varepsilon}| \approx \sigma\sqrt{d}$ and the Cauchy–Schwarz inequality, the perturbation magnitude is bounded by $|\Delta\ell| \lesssim |\mathbf{w}|\sigma\sqrt{d}$, indicating that the logit shift scales linearly with the weight norm.

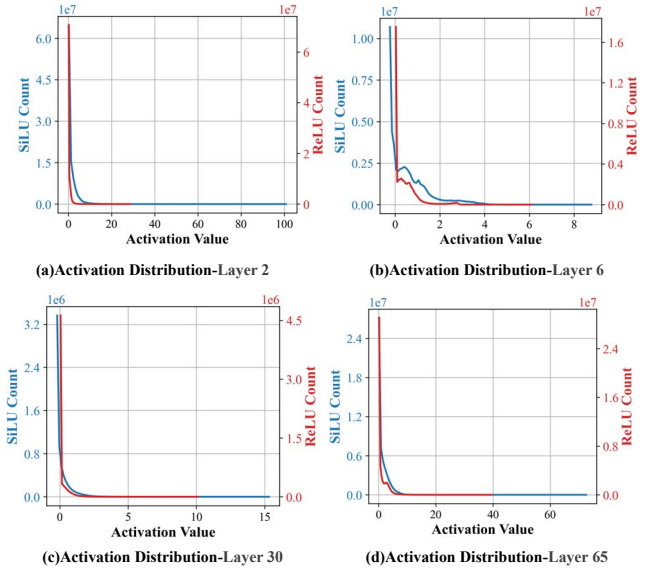
For the detection task, the output consists of a class logit and four regression coordinates. Under a perturbation $\boldsymbol{\varepsilon}$, similar to the classification task, the resulting shifts in the logit ($\Delta\ell_{\text{det}}$) and each bounding box coordinate (Δb_j) are both linear:

$$\Delta\ell_{\text{det}} = \mathbf{w}_{\text{cls}}^\top \boldsymbol{\varepsilon}, \quad \Delta b_j = \mathbf{w}_{\text{reg},j}^\top \boldsymbol{\varepsilon},\ j \in \{1,2,3,4\}. \tag{1}$$

Fig. 2 illustrates the performance degradation trends for both classification and detection tasks under perturbation. The experimental results reveal a significant disparity in sensitivity to error under the same perturbation strength. Although the L1 error for both tasks increases approximately linearly, the classification accuracy drops by only about 1%, whereas the mAP for the detection task plummets by 70%.

This disparity stems primarily from the decision-making mechanism of classification models. A classification model's output remains unchanged as long as the perturbation magnitude does not exceed the decision margin, i.e., $|\Delta\ell| < |\ell - \tau|$, where τ is typically defined as the difference between the maximum and second-maximum logits ($\tau = \min(\ell_{\max} - \ell_{\text{submax}})$). In contrast, a detection task's performance, measured by mAP, is directly impacted by any numerical deviation in the regression outputs, rendering it far more sensitive to perturbations.

Furthermore, beyond the differences in their decision-making mechanisms, detection models involve a post-processing step that calculates the geometric overlap between predicted and ground truth boxes, known as Intersection over Union (IoU). Due to the non-linear operations, the IoU metric is inherently more sensitive to small perturbations in box position and scale compared to a direct L1 difference between coordinates. Experimental results, as shown in Fig. 3, further confirm this: under the same perturbation magnitude applied to the input, the resulting error in the L1 space of the predicted boxes is limited, but this error becomes significantly amplified when mapped to the IoU space. This non-linear amplification effect reveals the weaker robustness of detection tasks to perturbations during post-processing, which explains their heightened susceptibility to quantization errors compared to classification tasks.



(a)Activation Distribution-Layer 2

(b)Activation Distribution-Layer 6

(c)Activation Distribution-Layer 30

(d)Activation Distribution-Layer 65

Fig. 4. Comparison of activation distributions between SiLU (blue line) and ReLU (red line) across different layers of the YOLOv5s model. The plots illustrate the activation density distributions for (a) Layer 2, (b) Layer 6, (c) Layer 30, and (d) Layer 65.

B. Activation Distribution with Long-Tail Property

In our quantization experiments on the YOLOv5 model, we observed that the SiLU activation function significantly

exacerbates the long-tail property of the activation distribution. To investigate the difference between ReLU and SiLU, we retrained the YOLOv5 model with the ReLU activation function on the same dataset for comparison. As illustrated in Fig. 4, SiLU retains a substantial number of low-frequency, high-magnitude activation values, which significantly expands the overall dynamic range compared to ReLU. Such a long-tail property compels the quantizer to accommodate a wider range, causing a large number of small-magnitude activations to be mapped to the same quantized integer. This leads to a loss of fine-grained information and a notable increase in quantization error, particularly in low-bit activation quantization.

IV. METHOD

A. Overall Framework

Motivated by the preceding theoretical analysis and experimental observations, we propose **T**ask-**A**ware and **S**tructure-**K**nowledge-guided **Q**uantization (**TASKQ**), a PTQ framework tailored for YOLO models. As illustrated in Fig. 5, TASKQ integrates three key components.

First, to address the adverse impact of long-tail activation distributions, we employ a **sparse quantization** strategy. As shown in Fig. 6, this method dynamically thresholds the tail of the activation distribution, partitioning it into a high-density part for low-bit quantization and a low-density, high-magnitude part that is processed with high-precision sparse convolution. Subsequently, the results from both pathways are aggregated, thereby preserving the mathematical equivalence of the convolution operation. This dual-path design effectively isolates outliers and significantly enhances representation precision for skewed distributions without incurring substantial computational overhead.

The other two components are **D**etection-aware **T**ask **R**egularization (DTR) and **S**cale-and-**T**ask-**A**ware **H**ead-wise **Q**uantization (STAHQ), which will be detailed in the subsequent sections.

B. Detection-aware Task Regularization

Recent methods like BRECQ [19] have advanced low-bit PTQ by introducing a Hessian-guided loss. This loss, derived from a second-order Taylor expansion, approximates the final task loss with respect to the quantization perturbation, as formulated below:

$$\mathcal{L}_{\mathbf{H}^{\mathbf{z}^{(b)}}} = \left[\Delta\mathbf{z}^{(b)\top}\mathbf{H}^{\mathbf{z}^{(b)}}\Delta\mathbf{z}^{(b)}\right] \tag{2}$$

where $\mathbf{z}^{(b)}$ denotes the output of module b, $\Delta\mathbf{z}^{(b)}$ is the difference in the output before and after quantization, and $\mathbf{H}^{\mathbf{z}^{(b)}}$ corresponds to the Hessian matrix of the task loss with respect to $\mathbf{z}^{(b)}$. Due to the prohibitive size of the Hessian matrix, the diagonal of the Fisher Information Matrix (FIM) is used to approximate its computation:

$$\mathbf{H}^{\mathbf{z}^{(b)}} \approx \operatorname{diag}\left(\left(\frac{\partial L}{\partial \mathbf{z}_1^{(b)}}\right)^2, \ldots, \left(\frac{\partial L}{\partial \mathbf{z}_a^{(b)}}\right)^2\right) \tag{3}$$

However, this diagonal FIM approximation is often inadequate for object detection. As works like Reg-PTQ [24] have shown, the Hessian's structure in regression tasks is highly dependent on error propagation paths, which a simple diagonal approximation fails to capture. This mismatch hinders performance recovery, leaving the accurate modeling of task-level quantization impact as a critical unresolved issue.

Algorithm 1 Optimizing the Target b-th Module with DTR and Sparse Quantization

Input: The pre-trained full-precision model $\mathcal{M}_{fp}$; The mixed-precision model $\mathcal{M}_q$ (modules $1 \ldots b-1$ are optimized, $b+1 \ldots N$ remain FP); The target b-th modules: frozen $M_{fp}^{(b)}$ and learnable $M_q^{(b)}$; The calibration dataset $\mathcal{D}_{cali}$.

Output: Optimized parameters for $M_q^{(b)}$

$(\mathbf{O}_{fp}, \mathbf{O}_q) \leftarrow \mathcal{M}_{fp}(\mathcal{D}_{cali}), \mathcal{M}_q(\mathcal{D}_{cali})$ ▷ forward & capture inputs $\mathbf{z}^{in}$ via hooks
$B, \hat{B} \leftarrow \text{NMS}(\mathbf{O}_{fp}), \text{NMS}(\mathbf{O}_q)$ ▷ obtain detection boxes
$\mathcal{L}_{det} \leftarrow \text{Eq.4}(B, \hat{B})$ ▷ compute detection loss
$\mathbf{z}_{fp} \leftarrow \mathbf{z}_{fp}^{in},\ \mathbf{z}_q \leftarrow \mathbf{z}_q^{in}$ ▷ initialize module inputs
for layer l_{fp}, l_q in $M_{fp}^{(b)}, M_q^{(b)}$ **do**
  $\mathbf{z}_{fp} \leftarrow l_{fp}(\mathbf{z}_{fp})$
  $\mathbf{z}_q \leftarrow l_q.\text{sparse\_inference}(\mathbf{z}_q)$ ▷ perform sparse inference
end for
$\mathcal{L}_{rec} \leftarrow \text{Eq.2}(\mathbf{z}_{fp}, \mathbf{z}_q)$ ▷ compute reconstruction loss
Update $M_q^{(b)}$ via $\nabla(\mathcal{L}_{rec} + \mathcal{L}_{det})$ ▷ optimize using Eq.8

Our analysis in Section III-A reveals that the IoU transformation mechanism is a key factor influencing quantization sensitivity in regression tasks. Based on this, we propose a regularization strategy guided by the detection head's task loss to steer the gradient learning of the local reconstruction error. Specifically, we define a head-level task loss, $\mathcal{L}_{\text{det}}$, to measure the structural difference in prediction results between the full-precision and quantized models. This loss function consists of three main parts: a bounding box regression loss L_{box}, a classification logits loss L_{cls}, and an objectness confidence loss L_{obj}. The overall expression is as follows:

$$\mathcal{L}_{\text{det}} = \frac{1}{n}\sum_{i=1}^{n}\left[\lambda_{\text{box}}L_{\text{box}} + \lambda_{\text{cls}}L_{\text{cls}} + \lambda_{\text{obj}}L_{\text{obj}}\right], \tag{4}$$

where n is the number of positive detection boxes, and $\lambda_{\text{box}}, \lambda_{\text{cls}}, \lambda_{\text{obj}}$ are the loss weights. Classification and objectness branches are optimized with Cross-Entropy (CE) loss, while the regression branch adopts Complete IoU (CIoU) loss L_{box} to better capture structural differences, formulated as:

$$L_{\text{box}} = 1 - \text{CIoU}(B, \hat{B}) = 1 - \text{IoU}(B, \hat{B}) + \frac{\rho^2(B, \hat{B})}{c^2} + \alpha v, \tag{5}$$

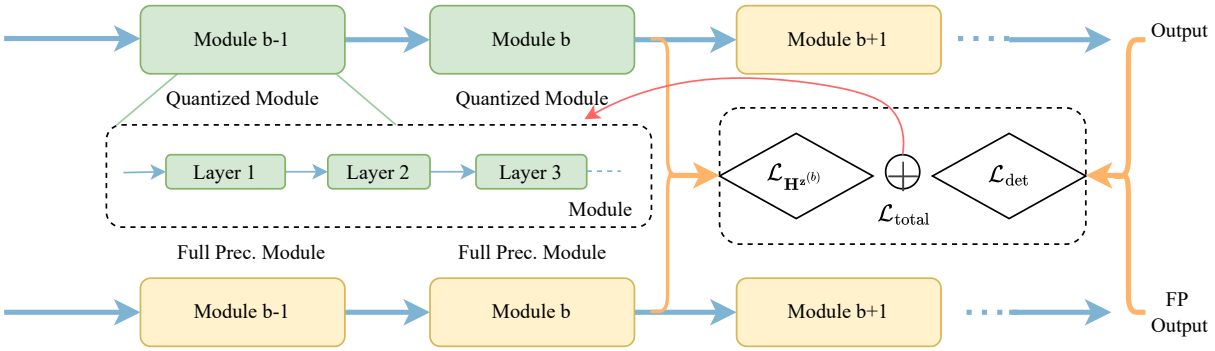


Fig. 5. The overall architecture of TASKQ. The framework utilizes a sparse quantization strategy and a Scale-and-Task-Aware Head-wise Quantization (STAHQ) scheme. Quantization parameters are optimized by minimizing a joint loss function that combines local reconstruction error with a detection-aware task regularization.

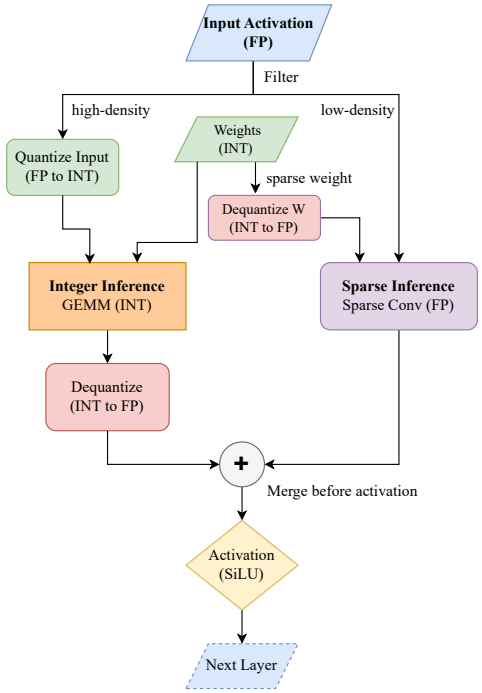


Fig. 6. Sparse quantization inference process.

Here, B and $\hat{B}$ are the bounding boxes from the full-precision and quantized models. ρ represents the Euclidean distance between their centers, while c denotes the diagonal length of the smallest enclosing box covering both. v quantifies the aspect ratio consistency, and α serves as a positive trade-off parameter, defined as follows:

$$v = \frac{4}{\pi^2}\left(\arctan\frac{w}{h} - \arctan\frac{\hat{w}}{\hat{h}}\right)^2, \tag{6}$$

$$\alpha = \frac{v}{(1 - \text{IoU}(B, \hat{B})) + v}, \tag{7}$$

where w, h and $\hat{w}, \hat{h}$ denote the width and height of B and $\hat{B}$.

Consequently, the b-th module is fine-tuned using a joint loss that combines local reconstruction error ($\mathcal{L}_{\mathbf{H}^{\mathbf{z}^{(b)}}}$) and detection-oriented regularization ($\mathcal{L}_{\text{det}}$):

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\mathbf{H}^{\mathbf{z}^{(b)}}} + \mathcal{L}_{\text{det}} \tag{8}$$

The detailed execution procedure is outlined in Algorithm 1. By integrating the sparse inference mechanism into the DTR strategy, the algorithm iteratively optimizes the quantization parameters of the b-th module by minimizing the objective function defined in Eq. 8.

C. Scale-and-Task-Aware Head-wise Quantization

The detection head exhibits strong heterogeneity, characterized by multi-scale structures, parallel output paths, and distinct task branches. This architectural complexity results in unbalanced sensitivity to quantization perturbations, rendering standard block-wise reconstruction ineffective. Empirical evidence corroborates this observation. As shown in Table I,when employing the BRECQ [19] method to quantize the backbone and neck to W4A4 while keeping the detection head at FP16, the YOLOv12s model maintains a robust accuracy of 50.1 mAP@50 on COCO [26]. However, fully quantizing the detection head to W4A4 causes a catastrophic performance drop to 20.5 mAP@50.

To investigate the root cause of this sensitivity, we adopt a layer-wise quantization strategy and sequentially measure each convolutional layer's reconstruction error $\mathcal{L}_H^{(i)}$ in YOLOv12. As shown in Fig. 7(a), periodic fluctuations emerge that directly map to the model's functional architecture: the regression module (CV2) shows a 3-layer period, while the classification module (CV3) shows a 5-layer period, corresponding to their respective single outputs per scale. Errors peak at layers preceding each scale-task output due to their significantly larger activation ranges, indicating that independent layer-wise reconstruction fails to capture their dominant influence and leads to severe interference in final outputs.

In response to these structural limitations, we propose a **S**cale-and-**T**ask-**A**ware **H**ead-wise **Q**uantization (STAHQ)

TABLE I

COMPARISON OF QUANTIZATION SENSITIVITY BETWEEN THE BACKBONE/NECK AND THE DETECTION HEAD ON COCO VAL SET (W4A4).

Backbone & Neck	Detect Head	mAP@50
FP16	FP16	59.4
W4A4	**FP16**	**50.1**
W4A4	**W4A4**	**20.5** (↓ 38.9)

strategy. This approach functionally deconstructs the YOLO detection head according to the scale × task pathway paradigm, partitioning it into six sub-modules (i.e., three scales × two tasks). Building on this, STAHQ treats these functional sub-modules as the fundamental units for quantization. By jointly optimizing the reconstruction error of each sub-module's overall output, it replaces the traditional layer-wise local minimization objective. This establishes a global quantization optimization mechanism with cross-layer information coordination. By enabling error feedback and transfer within sub-modules, the proposed strategy substantially enhances the reconstruction fidelity and robustness of quantized detection heads under heterogeneous architectures.

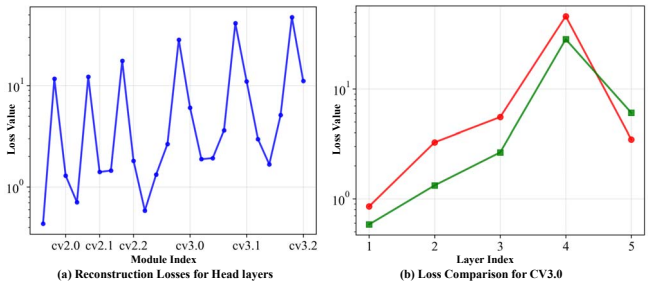


Fig. 7. Comparison of reconstruction loss between layer-wise and STAHQ strategies. (a) Layer-wise reconstruction error across the detection head. (b) Reconstruction loss in a functional sub-module, comparing STAHQ (red) with the layer-wise approach (green).

The results demonstrate that the STAHQ strategy significantly reduces the reconstruction error of critical output layers within functional sub-modules. By enabling cross-layer error coordination, this approach shifts the optimization from a local, layer-wise objective to a holistic, sub-module level, thereby achieving a more balanced and robust reconstruction quality.

V. EXPERIMENTS

A. Experimental Setup

Models and Datasets. We evaluate our method on the COCO [26] dataset using YOLOv5 [10], YOLOv8 [12], and YOLOv12 [13], which differ in backbone and neck designs. This diverse selection verifies the effectiveness and generalization of our approach across varying architectures.
Implementation Details. Our experiments use official pretrained models from Ultralytics [27]. We use a calibration set of 256 randomly sampled COCO images with a batch size of 32. Following standard PTQ practices [18]–[21], all batch normalization layers are fused prior to quantization. We denote bit width as $WwAa$ for weights and activations. A key aspect of our setup is that the model's input layer is kept at 8-bit precision, while all other layers are quantized to the target $WwAa$ bit width. Following the original YOLO loss configuration,we set the hyperparameters for the detection loss in Equation 4 as: $\lambda_{\text{box}} = 7.5$, $\lambda_{\text{cls}} = 0.5$, and $\lambda_{\text{obj}} = 1.0$.

TABLE II

PERFORMANCE COMPARISON (MAP@50) ON THE COCO DATASET

Method	Bits(W/A)	YOLOv5s	YOLOv8s	YOLOv12s
FP	32/32	59.4	61.3	64.6
AdaRound	4/4	17.8	41	26.8
BRECQ	4/4	20.5	41.2	30.5
PD-Quant	4/4	13.2	40.8	25.9
TASKQ (Ours)	4/4	**54.6**	**57.1**	**60.1**
AdaRound	4/6	54.5	57.2	59.7
BRECQ	4/6	54.9	57.3	60.0
PD-Quant	4/6	53.8	56.9	59.2
TASKQ (Ours)	4/6	**58.2**	**60.1**	**63.1**
AdaRound	4/8	56.4	58.9	61.5
BRECQ	4/8	56.5	59.2	61.5
PD-Quant	4/8	55.9	58.9	61.4
TASKQ (Ours)	4/8	**58.3**	**60.2**	**63.1**

B. Object Detection Results on COCO

We compare our TASKQ framework against several classic and recent PTQ methods, including AdaRound [18], BRECQ [19], and PD-Quant [22]. The evaluation is conducted under various bit width settings, including $W4A4$, $W4A6$, and $W4A8$. In the following tables, results from our method are highlighted in **bold**.

As shown in Table II, our method consistently outperforms other learning-based PTQ approaches across different YOLO variants. In the most challenging $W4A4$ low-bit scenario, existing methods suffer severe degradation due to the long-tail distribution of activations, a phenomenon particularly pronounced in YOLOv5 and YOLOv12, which exhibit larger performance declines than YOLOv8. In contrast, TASKQ effectively mitigates this issue, maintaining stable performance with only about a 5% accuracy drop. This demonstrates the robustness of our framework in extreme quantization settings.Besides, since the final output layer of detection models produces many redundant predictions, PD-loss is ineffective, whereas our Detection-aware Task Regularization (DTR) remains effective, as confirmed by the ablation study.

A key observation is that TASKQ shows little difference between $W4A6$ and $W4A8$. This stems from the sparse quantization strategy, which isolates high-magnitude outliers in the activation's long-tail and leaves a compact distribution well-represented with only 6 bits. This saturation at a lower bit-width underscores the efficiency of our approach, enabling near-lossless performance without relying on high-precision activations, which represents a significant advantage for resource-constrained deployments.

C. Ablation Study

The results of our ablation study are presented in Table III. Starting from the BRECQ baseline, we incrementally add our proposed components to validate their individual contributions under the challenging $W4A4$ setting.

TABLE III
ABLATION STUDY OF EACH COMPONENT OF TASKQ

Bit width	Method	YOLOv5s	YOLOv12s
W4A4	BRECQ (Baseline)	20.5	30.5
	+ Sparse Quantization	51.6	57.6
	+ Sparse + DTR	51.9	57.8
	+ Sparse + PD-loss	50.1	56.6
	+ Sparse + STAHQ	54.5	59.9
	TASKQ (Full Method)	**54.6**	**60.1**

First, incorporating Sparse Quantization alone brings a significant performance improvement over the baseline. Building on this, the addition of DTR and STAHQ further boosts the performance by approximately 0.3% and 2.9% on YOLOv5s, respectively. The substantial improvement from STAHQ highlights the critical role of quantization strategies tailored to the detection head's structural heterogeneity. When all three components are integrated to form the full TASKQ framework, performance improves by another 0.1% compared to the "+Sparse+STAHQ" configuration. Although this marginal gain is expected, since STAHQ's structure-aware grouping already provides partial task-level optimization, the results confirm that the three components are complementary and that their synergistic integration yields the best overall performance.

VI. CONCLUSIONS

In this work, we have focused on developing a PTQ framework specifically tailored for YOLO detection models. We identified the fundamental challenges of quantizing YOLO structures and introduced the TASKQ framework to address them. Extensive experiments have demonstrated the effectiveness of our approach across multiple YOLO variants. Beyond methodological contributions, this work provides one of the first systematic attempts toward fully quantized YOLO architectures, offering key insights and practical solutions for enabling end-to-end deployment of lightweight object detectors in edge scenarios.

REFERENCES

[1] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 936–944.

[2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," 2016. [Online]. Available: https://arxiv.org/abs/1506.01497

[3] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, *SSD: Single Shot MultiBox Detector*. Springer International Publishing, 2016, pp. 21–37. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-46448-0_2

[4] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4490–4499.

[5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.

[6] T. Choudhary, V. Mishra, A. Goswami, and J. Sarangapani, "A comprehensive survey on model compression and acceleration," *Artificial Intelligence Review*, vol. 53, no. 7, pp. 5113–5155, 2020.

[7] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," *arXiv preprint arXiv:1510.00149*, 2015.

[8] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[9] R. Krishnamoorthi, "Quantizing deep convolutional networks for efficient inference: A whitepaper," 2018. [Online]. Available: https://arxiv.org/abs/1806.08342

[10] R. Khanam and M. Hussain, "What is yolov5: A deep look into the internal features of the popular object detector," 2024. [Online]. Available: https://arxiv.org/abs/2407.20892

[11] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," 2022. [Online]. Available: https://arxiv.org/abs/2207.02696

[12] M. Yaseen, "What is yolov8: An in-depth exploration of the internal features of the next-generation object detector," 2024. [Online]. Available: https://arxiv.org/abs/2408.15857

[13] Y. Tian, Q. Ye, and D. Doermann, "Yolov12: Attention-centric real-time object detectors," 2025. [Online]. Available: https://arxiv.org/abs/2502.12524

[14] S. K. Esser, J. L. McKinstry, D. Bablani, R. Appuswamy, and D. S. Modha, "Learned step size quantization," 2020. [Online]. Available: https://arxiv.org/abs/1902.08153

[15] Z. Wang, Z. Wu, J. Lu, and J. Zhou, "Bidet: An efficient binarized object detector," 2020. [Online]. Available: https://arxiv.org/abs/2003.03961

[16] Y. Li, S. Xu, B. Zhang, X. Cao, P. Gao, and G. Guo, "Q-vit: Accurate and fully quantized low-bit vision transformer," 2022. [Online]. Available: https://arxiv.org/abs/2210.06707

[17] Y. Bengio, N. Léonard, and A. Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," 2013. [Online]. Available: https://arxiv.org/abs/1308.3432

[18] M. Nagel, R. A. Amjad, M. van Baalen, C. Louizos, and T. Blankevoort, "Up or down? adaptive rounding for post-training quantization," 2020. [Online]. Available: https://arxiv.org/abs/2004.10568

[19] Y. Li, R. Gong, X. Tan, Y. Yang, P. Hu, Q. Zhang, F. Yu, W. Wang, and S. Gu, "Brecq: Pushing the limit of post-training quantization by block reconstruction," 2021. [Online]. Available: https://arxiv.org/abs/2102.05426

[20] X. Wei, R. Gong, Y. Li, X. Liu, and F. Yu, "Qdrop: Randomly dropping quantization for extremely low-bit post-training quantization," 2023. [Online]. Available: https://arxiv.org/abs/2203.05740

[21] Y. Ma, H. Li, X. Zheng, X. Xiao, R. Wang, S. Wen, X. Pan, F. Chao, and R. Ji, "Solving oscillation problem in post-training quantization through a theoretical perspective," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 7950–7959.

[22] J. Liu, L. Niu, Z. Yuan, D. Yang, X. Wang, and W. Liu, "Pd-quant: Post-training quantization based on prediction difference metric," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 24 427–24 437.

[23] L. Niu, J. Liu, Z. Yuan, D. Yang, X. Wang, and W. Liu, "Improving post-training quantization on object detection with task loss-guided lp metric," 2023. [Online]. Available: https://arxiv.org/abs/2304.09785

[24] Y. Ding, W. Feng, C. Chen, J. Guo, and X. Liu, "Reg-ptq: Regression-specialized post-training quantization for fully quantized object detector," in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 16 174–16 184.

[25] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[26] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft coco: Common objects in context," 2015. [Online]. Available: https://arxiv.org/abs/1405.0312

[27] G. Jocher, J. Qiu, and A. Chaurasia, “Ultralytics YOLO,” Jan. 2023. [Online]. Available: https://github.com/ultralytics/ultralytics