AQ-DETR: Low-Bit Quantized Detection Transformer with Auxiliary Queries

Runqi Wang[1], Huixin Sun[1], Linlin Yang[2*], Shaohui Lin[3], Chuanjian Liu[4], Yan Gao[5], Yao Hu[5], Baochang Zhang[1,6,7]

[1]ASEE, EIE and Hangzhou Research Institute, Beihang University;
[2]State Key Laboratory of Media Convergence and Communication, Communication University of China;
[3]School of Computer Science and Technology, East China Normal University;
[4]Huawei Noah's Ark Lab;
[5]Xiaohongshu Inc;
[6]Zhongguancun Laboratory;
[7]Nanchang Institute of Technology

Abstract

DEtection TRansformer (DETR) and its variants have achieved remarkable performance. However, they are accompanied by a large computation overhead cost, which significantly prevents their applications on resource-limited devices. Prior arts attempt to reduce the computational burden of DETR using low-bit quantization, while these methods sacrifice a severe significant performance on weight-activation-attention low-bit quantization. We observe that the number of matching queries and positive samples affects much on the representation capacity of queries in DETR, while quantifying queries of DETR further reduces its representational capacity, thus leading to a severe performance drop. We introduce a new quantization strategy based on Auxiliary Queries for DETR (AQ-DETR), aiming to enhance the capacity of quantized queries. In addition, a layer-by-layer distillation is proposed to reduce the quantization error between quantized attention and full-precision counterpart. Through our extensive experiments on large-scale open datasets, the performance of the 4-bit quantization of DETR and Deformable DETR models is comparable to full-precision counterparts.

Introduction

Object detection, aiming to detect and locate objects of interest, has been successfully applied to many real-world applications including autonomous driving (Hu et al. 2023) and robotics (Jing et al. 2023). Especially, a recent breakthrough in its network architecture contributes significantly to its success. Among different detection architectures (Redmon and Farhadi 2018; Ren et al. 2015; Carion et al. 2020), DEtection TRansformer (DETR) (Carion et al. 2020) has gained substantial interest recently due to its remarkable accuracy and less reliance on post-processing procedures. However, DETR usually suffers from unacceptable memory and computation consumption during inference. For example, in the DETR model with ResNet-50 backbone (DETR-R50), there are 39.8M parameters utilizing 159MB memory and 86G float-pointing operations (FLOPs) to detect an image of the 1333×800 resolution. This restricts its deployment potential on resource-limited devices.

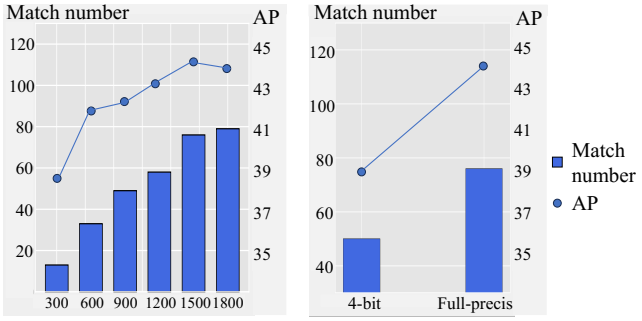

Figure 1: Average Precision (AP) performance and the number of matching queries and positive samples (match number) with respect to (a) the number of queries, and (b) Deformable DETR's quantized encoder from full precision to 4 bits. The bar chart and the line chart show the trend of match number and AP, respectively. We can see that as the number of queries increases from 300 to 1800, the match number increases from around 17 to 79, and AP progressively increase from around 38.5 to 44. When quantizing DETR's encoder from full precision to 4 bits, the match number and AP both decrease significantly.

Network quantization, by representing a network in low-bit formats, provides the potential for DETR to reduce memory and computation consumption. Quantization-aware training (QAT) is a regular training strategy for network quantization. It simultaneously quantizes and fine-tunes the models on the training dataset, and has been demonstrated to be effective for low-bit quantization.

However, when quantizing DETR to 4 bits, a baseline version of QAT (Bhalgat et al. 2020) fails to preserve the effectiveness of DETR, which leads to a significant performance drop. To improve the QAT performance for quantizing DETR, Q-DETR (Xu et al. 2023) is introduced to solve the query information distortion problem. This, however, results in mixed-precision quantization with about 3% performance drop on VOC dataset. Specifically, it only achieves the 4-bit quantization on the attention mechanism but has to keep 8-bit attention activations to prevent performance drop.

*Corresponding Author.

DETR's full model quantization is still a challenge, due to it inevitably requires quantizing queries and the output of encoder, which further reduces the representability of queries and encoded features. The matching of quantized queries to the positive samples becomes more difficult and uncertain, such that the performance suffers from a significant degradation. To verify the reason for the performance drop, we perform an empirical study as shown in Fig. 1. In Fig. 1(a), we study the influence of the total number of queries n_A within the Deformable DETR framework by fixing the number of ground truth as 126. As the number of queries increases, the number of matches between queries and positive samples increases. As a result, the representation capacity of the query feature will be enhanced and thus AP is increasing. Fig. 1(b) shows the full-precision encoder can get more matching between queries and positive samples, leading to a higher AP.

In this paper, we focus on the full quantization of DETR's unique attention mechanism and aim for full 4-bit quantization for DETR. Specifically, as the attention mechanism basically establishes spatial dependencies based on object queries and encoded features, we propose point-to-point solutions, *i.e.*, auxiliary queries for object queries and a layer-by-layer distillation for encoded features, to address the 4-bit quantization on the attention mechanism.

As shown in Fig. 2, a QAT method using Auxiliary Queries is first proposed to improve the capacity of queries for quantizing DETR, termed AQ-DETR. In the training phase, we introduce an additional matching branch that assigns multiple auxiliary queries to each feature. Although these queries and features are quantized, there are more matched queries and features. So the transformer can be fully trained and quantified using bounding box loss. By the way, the test phase still only uses the original queries to reduce the amount of computation. In addition, this paper uses a layer-by-layer distillation to maintain the feature distribution of the encoder, which reduces the quantization error. The layer-by-layer distillation allows the features of the quantized encoder to preserve the feature information of the full-precision encoder, which benefits the matching of queries and features. Extensive experiments show that our method outperforms prior arts by a large margin, even approaching full-precision DETR.

We summarize our contributions below:

1. We introduce an additional matching branch that assigns multiple auxiliary queries to each feature, which significantly enhances the capacity of queries but without any extra cost for the test phase.

2. A layer-by-layer distillation is proposed to maintain the feature distribution of the encoder. Together with our auxiliary queries, we obtain a better object query and positive sample matching and significantly improve the QAT performance for DETR.

3. We successfully achieve full model 4-bit quantization of the DETR-based detector with performance comparable to the full-precision counterparts.

Related Work

Quantization. Quantized neural networks often possess low-bit weights and activations to accelerate model inference and save memory. Existing model quantization strategies can be categorized into Post-Training Quantization (PTQ) and Quantization-Aware Training (QAT).

PTQ is a training-free strategy. It directly maps the original floating-point values of pre-trained models to lower bits. However, PTQ comes with certain trade-offs, primarily related to potential accuracy loss due to quantization-induced errors. Even with techniques like prompt tuning (Lin et al. 2022), the quantization performance of PTQ is inferior to that of QAT when it comes to ultra-low bits (*i.e.*, 4 bits or lower).

Instead, QAT incorporates the quantization process into the training itself, thereby reducing quantization-induced errors. This makes QAT the favored ultra-low bits quantization technique for both CNNs and transformers. For the binarization of CNNs, auxiliary components like single-scaling factors (Rastegari et al. 2016) or modulation filters (Zhang et al. 2021) are typically introduced to optimize jointly with binarized filters to approximate un-binarized ones. Those auxiliary components can alleviate the disturbance caused by the binarized process. For the ultra-low bits quantization of transformer-based architectures, existing works (Li et al. 2022b; Qin et al. 2022; Xu et al. 2023) present various alignment or optimization strategies to eliminate information distortion in the forward and optimization direction mismatch in the backward process. Differently, we shift the focus towards improving the capacity of queries and demonstrate that doing so is also crucial for quantifying DETR.

DETR and its variants. In the context of object detection, DETR (Carion et al. 2020) introduces the first transformer architecture by reformulating object detection as a set prediction problem. It achieves great success due to its end-to-end training manner and the benefits of the transformer.

Most follow-up works emphasize the modification of multi-head attention mechanism (MHA) (Zhu et al. 2020; Gao et al. 2021) or the query strategy (Liu et al. 2022a; Li et al. 2022a; Jia et al. 2023) to mitigate its slow convergence. For MHA, inspired by deformable convolutions, Deformable-DETR (Zhu et al. 2020) constructs a sparse and point-to-point MHA mechanism that only attends to a small set of key sampling points around a reference point. SMCA-DETR (Gao et al. 2021) proposes capturing the global information via a dynamic Gaussian-like spatial prior before the spatially modulated co-attention. The spatial prior can effectively aggregate query-related information from the visual feature map. For the query strategy, DAB-DETR (Liu et al. 2022a) exploits box coordinates as the query for DETR. This new query formulation enhances the query-to-feature similarity and enables the model to modulate the positional attention map based on the box width and height information. DNDETR (Li et al. 2022a) proposes to feed noisy queries into the transformer decoder to recover the original ones. This query denoising procedure stabilizes bipartite graph matching during training and accelerates convergence. H-DETR (Jia et al. 2023) proposes an auxiliary one-to-many matching branch during training. With this branch, H-DETR

enriches the number of queries that are matched with ground truth and gets higher training efficacy.

Apart from the modification of MHA and the query strategy, recent works (Li et al. 2023; Xu et al. 2023) also recognize the heavy nature of DETR and begin to investigate its lightweight version. Lite DETR (Li et al. 2023) develops an efficient encoder block with several deformable self-attention layers to reduce the feature tokens for efficient detection. Q-DETR (Xu et al. 2023) investigates the low bits quantization of DETR and proposes a bi-level optimization framework based on the information bottleneck principle. However, Q-DETR is incapable of quantizing the attention activations to 4 bits or less, resulting in a mixed-precision quantization. In this paper, we focus on the low bits quantization of DETR. Differently, we target at full 4-bit quantization of DETR with performance comparable to that of its full-precision counterpart.

Methodology

This paper focuses on the full 4-bit quantization of DETR (Carion et al. 2020) and its variant, *i.e.*, Deformable DETR (Zhu et al. 2020). This section begins with a quick overview of the two methods. Then our AQ-DETR with an auxiliary query mechanism and a layer-by-layer distillation are introduced.

Model Overview

DETR. DETR consists of a CNN backbone, a transformer encoder, a transformer decoder, and prediction heads. The CNN backbone and the transformer encoder aim to extract a sequence of enhanced pixel embeddings $\mathbf{E} = \{\mathbf{e}_0, \mathbf{e}_1, \cdots, \mathbf{e}_N\}$ from N input images. Given the enhanced pixel embeddings, DETR feeds them with a default group of object query embeddings $\mathbf{O} = \{\mathbf{o}_0, \mathbf{o}_1, \cdots, \mathbf{o}_{n_o}\}$ into the transformer decoder to get the updated object query embeddings. The prediction heads then are applied on the output embeddings of each transformer decoder layer to generate a set of predictions $\mathbf{P} = \{\mathbf{p}_0, \mathbf{p}_1, \cdots, \mathbf{p}_n\}$ independently. For the objective, DETR performs one-to-one bipartite matching between the predictions and the ground-truth bounding boxes and labels $\mathbf{G} = \{\mathbf{g}_0, \mathbf{g}_1, \cdots, \mathbf{g}_m\}$. Specifically, DETR associates each ground truth with the prediction that has the minimal matching cost and applies the corresponding supervision accordingly.

Deformable-DETR. It mainly follows the architecture of DETR, but with three modifications: (1) substituting the original multi-head attention with a multi-scale deformable attention scheme. It makes the attention mechanism sample in a small range and only considers the most valuable points, thereby reducing the calculations and accelerating the convergence rate; (2) implementing multi-scale Deformable Attention, which generates multi-scale feature maps. It not only improves the detection efficiency of small targets, but also effectively controls the number of parameters; (3) replacing the original independent layer-wise prediction scheme with an iterative refinement prediction, and replacing the original image content irrelevantly query with a dynamic query generated by the transformer encoder output.

AQ-DETR

We first build a straightforward solution as a baseline, *i.e.*, weights and activation values of DETR's Backbone, Encoder and Decoder are quantified according to LSQ+ (Bhalgat et al. 2020). Based on the baseline, we introduce the Auxiliary Query module (AQ) and the Layer-by-Layer Distillation module (LLD) to build AQ-DETR. Fig. 2 shows the overview of AQ-DETR, using the format "Q-component" to represent the quantized component. Due to the superior generalization of AQ and LLD, our proposed AQ-DETR is applicable to different variants of DETR, such as Deformable DETR.

Baselines. We follow LSQ+ to introduce a base framework of asymmetric activation quantization and symmetric weight quantization:

$$\begin{aligned} x_q =& \lfloor \mathrm{clip}\{\frac{(x - z_x)}{\alpha_x}, -r_a^d, r_a^u\} \rceil, w_q = \lfloor \mathrm{clip}\{\frac{w}{\alpha_w}, -r_b^d, r_b^u\} \rceil, \\ Q_a(x) =& \alpha_x \circ x_q + z_x, \qquad Q_b(w) = \alpha_w \circ w_q, \end{aligned} \tag{1}$$

where $\mathrm{clip}\{y, r_1, r_2\}$ clips the input y with value bounds r_1 and r_2; the $\lfloor y \rceil$ rounds y to its nearest integer; the $\circ$ denotes the channel-wise multiplication. With a-bit quantization, $r_a^d = 2^{a-1}$ and $r_a^u = 2^{a-1} - 1$ are the discrete bounds. x, x_q, w, w_q and $Q_a(x)$ denote the full-precision activation, the quantized activation, the full-precision weight, the quantized weight and the value of inverse quantization, respectively. α_x and z_x are trainable values of the quantization function.

The fully-connected (FC) layer, multi-head self-attention (MHSA), multi-head cross-attention (MHCA) can all be obtained according to Eq. 1. Specifically, Q-FC is constructed as:

$$\text{Q-FC}(x) = Q_a(x) \cdot Q_b(w) = \alpha_x \alpha_w \circ (x_q \odot w_q + z_x / \alpha_x \circ w_q), \tag{2}$$

where $\cdot$ denotes the matrix multiplication and $\odot$ denotes the bit-wise matrix multiplication operations. The straight-through estimator (STE) (Bengio, Léonard, and Courville 2013) is used to retain the derivation of the gradient in backward propagation. The backward propagation is formulated as:

$$\frac{\partial \mathcal{J}}{\partial x} = \frac{\partial \mathcal{J}}{\partial Q_a(x)} \frac{\partial Q_a(x)}{\partial x} = \begin{cases} \dfrac{\partial \mathcal{J}}{\partial Q_a(x)}, & \text{if } x \in [-r_a^d, r_a^u] \\ 0, & \text{otherwise} \end{cases}, \tag{3}$$

$$\frac{\partial \mathcal{J}}{\partial w} = \frac{\partial \mathcal{J}}{\partial x} \frac{\partial x}{\partial Q_b(w)} \frac{\partial Q_b(w)}{\partial w} = \begin{cases} \dfrac{\partial \mathcal{J}}{\partial x} \dfrac{\partial x}{\partial Q_b(w)}, & \text{if } w \in [-r_b^d, r_b^u] \\ 0, & \text{otherwise} \end{cases}, \tag{4}$$

where $\mathcal{J}$ is the loss function. As Q-MLP is a combination of Q-FC, we can get Q-MLP accordingly.

The computation of attention weight depends on the corresponding query $\mathbf{q}$, key $\mathbf{k}$ and value $\mathbf{v}$, and the quantized computation in one attention head is:

$$\mathbf{q} = \text{Q-FC}_q(\mathbf{x}), \mathbf{k} = \text{Q-FC}_\mathbf{k}(\mathbf{x}), \mathbf{v} = \text{Q-FC}_\mathbf{v}(\mathbf{x}), \tag{5}$$

where $\text{Q-FC}_q, \text{Q-FC}_k, \text{Q-FC}_v$ denote the three quantized linear layers for $\mathbf{q}, \mathbf{k}, \mathbf{v}$, respectively. The input of $\mathbf{q}$ in Encoder is the output of Q-CNN, and the input of $\mathbf{q}$ in Decoder

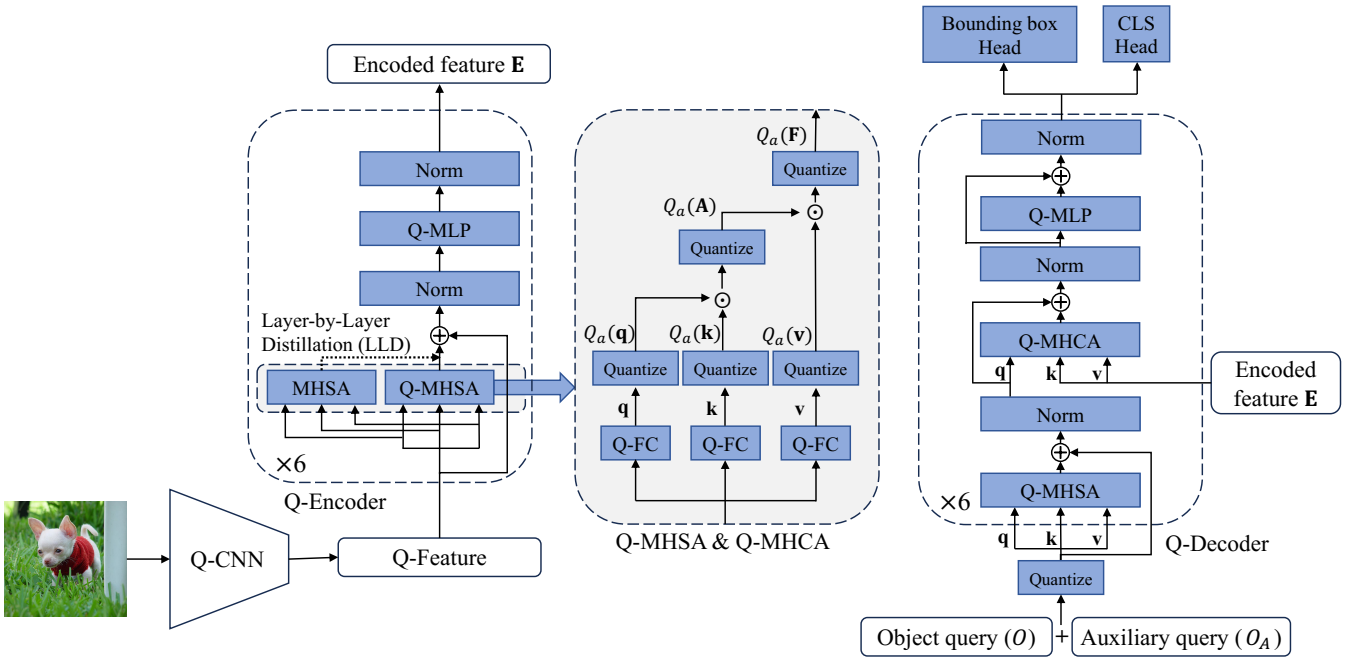

Figure 2: Overview of AQ-DETR. The Q-CNN, Q-Encoder, and Q-Decoder denote the quantized CNN, Encoder and Decoder, respectively. Similarly, Q-MHSA, Q-MHCA, and Q-FC denote the quantized multi-head self-attention, multi-head cross-attention, and full-connected layer. Q-MHSA and Q-MHCA have the same internal structure, but different query inputs. Besides, we introduce the layer-by-layer distillation in Q-Encoder and auxiliary query in Q-Decoder. In each Q-Encoder layer, the Q-MHSA distills the MHSA, thereby reducing the quantization loss of attention. The auxiliary queries are put into Q-Decoder with the object queries.

is object query $\mathbf{O}$. Thus, the attention weight is formulated as:

$$\begin{aligned} \mathbf{A} &= \text{softmax}(\frac{1}{\sqrt{d}}(Q_a(\mathbf{q}) \cdot Q_a(\mathbf{k})^\top)), \\ \mathbf{F} &= Q_a(\mathbf{A}) \cdot Q_a(\mathbf{v}), \end{aligned} \tag{6}$$

where $\mathbf{F}$ is the attention output. In Q-MHSA or Q-MHCA, it will be quantized as $Q_a(\mathbf{F})$ before the output.

Quantization with Auxiliary Queries. Zhu *et al.* (Zhu et al. 2020) observed the mismatching of object queries and positive samples in DETR leads to the low efficiency of training. We further point out that the mismatch issue will be exacerbated by the quantization, directly resulting in significant performance drop. This is the bottleneck of the quantization of DETR and its variants. In this regard, we introduce the Auxiliary Query module to improve the capacity of the object query. As shown in Fig. 3, we maintain object queries $\mathbf{O}$ and auxiliary queries $\mathbf{O_A}=\{\mathbf{o_A}_1, \mathbf{o_A}_2, \cdots, \mathbf{o_A}_{n_A}\}$. The dimensionality of $\mathbf{o_A}_n$ in $\mathbf{O_A}$ is the same as $\mathbf{o}_n$ in $\mathbf{O}$. They are both created by random initiation.

In the training phase, we process both groups of queries $\mathbf{O}$ and $\mathbf{O_A}$ with L transformer decoder layers and perform predictions on the output of each decoder layer respectively. Then, we perform the bipartite matching between the {predictions, ground-truth} pair over each layer, *e.g.*, estimating $\mathcal{L}_{match}(\mathbf{P}^l, \widehat{\mathbf{G}})$, where $\mathbf{P}^l$ represents the predictions outputted by the l-th transformer decoder layer, and $\widehat{\mathbf{G}}$ is the ground truth repeated K times.

$$\widehat{\mathbf{G}}=\{\mathbf{G}^1, \mathbf{G}^2, \cdots, \mathbf{G}^K\}, \tag{7}$$

where $\mathbf{G}^1=\mathbf{G}^2=\cdots=\mathbf{G}^K=\mathbf{G}$, K is the multiple of $\widehat{\mathbf{G}}$ repeated. We choose $\mathcal{L}_{match}(\cdot)$ following DETR (Carion et al. 2020) and Deformable-DETR (Zhu et al. 2020), which consist of a Hungary match, a classification loss, a $\mathcal{L}_1$ regression loss, and a GIoU loss. The use of auxiliary queries helps to resolve the mismatch problem and reduce quantization error. In the test phase, only object queries $\mathbf{O}$ are used, which will not cause any additional burden.

Layer-by-Layer Distillation. In addition to insufficient object query capacity, quantization errors in encoded feature $\mathbf{E}$ also make it difficult for object queries to match positive samples. Especially, the quantization errors in $\mathbf{E}$ can be viewed as an accumulation of distribution disturbance of MHSA in each encoder layer. In this regard, we design a layer-by-layer distillation (LLD) module as shown in Fig. 4. It does not require additional pre-training models and can effectively reduce the quantization errors in $\mathbf{E}$, which improves the matching of object queries and positive samples.

The LLD performs distillation on each encoder layer. Taking one layer as an example, the MHSA and Q-MHSA in the left panel of Fig. 4 share the same set of parameters. The detail of distillation in each encoder layer is shown in the right panel of Fig. 4. $\mathbf{q}$, $\mathbf{k}$, and $\mathbf{v}$ simultaneously perform the general self-attention calculation and the quantized

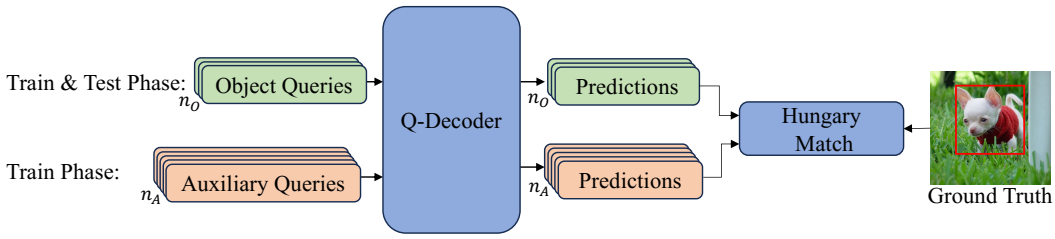

Figure 3: Overview of Auxiliary Queries. In the training phase, object queries and auxiliary queries are put into a decoder together. In the testing phase, only object queries are needed. n_O and n_A denote the number of object queries and auxiliary queries.

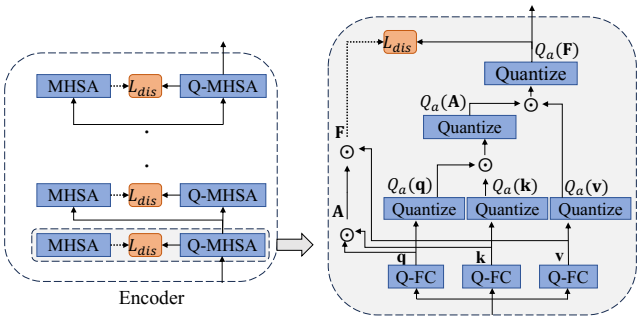

Figure 4: Overview of the Layer-by-Layer Distillation. $\mathcal{L}_{dis}$ denotes the loss function of distillation. The dotted line indicates that back-propagation is cut off. The left figure shows that there are 6 encoder layers, each of which requires distillation. The right figure shows the pipeline of distillation in each encoder layer.

self-attention calculation to obtain the full-precision output $\mathbf{F}$ and the quantized $Q_a(\mathbf{F})$. $Q_a(\mathbf{F})$is updated by distillation loss $\mathcal{L}_{dis}$ and passing it to the next layer.

$$\mathcal{L}_{dis} = \lambda_{dis}\mathcal{E}(\mathbf{F}, Q_a(\mathbf{F})), \tag{8}$$

where $\mathcal{E}(\cdot)$ is the cross-entropy function and λ_{dis} is the scale factor of distillation loss.

Experiment

In this section, we evaluate the performance of our AQ-DETR for DETR (Carion et al. 2020) and Deformable DETR (Zhu et al. 2020) by comparing with different full 4-bit quantized models including quantized DETR by LSQ (Esser et al. 2019), LSQ+ (Bhalgat et al. 2020) and Q-DETR (Xu et al. 2023).

Datasets and Implementation Details

Datasets. We conduct experiments on the PASCAL VOC dataset (Everingham et al. 2010) and the COCO `2017` object detection (Lin et al. 2014). The PASCAL VOC dataset contains natural images from 20 different classes. We use VOC `trainval2012`, and VOC `trainval2007` for training, and VOC `test2007` set for evaluation. The training and the testing set contain approximately 16k and 5K images, respectively. We report COCO-style metrics average precision (AP) for the VOC dataset. For COCO `2017` object detection, we use its standard train and test split, and list the AP for IoUs $\in [0.5 : 0.05 : 0.95]$, designated as AP, using COCO's standard evaluation metric.

Model	Method	#Bits	AP
DETR-R50	Full-precision	32-32-32	59.5
	Percentile	8-8-8	54.7
	VT-PTQ		57.6
	LSQ	4-4-4	36.9
	LSQ+		38.4
	Q-DETR		50.4
	AQ-DETR		**53.7**
Deformable DETR-R50	Full-precision	32-32-32	65.3
	Percentile	8-8-8	59.3
	VT-PTQ		62.9
	LSQ	4-4-4	47.6
	LSQ+		49.5
	Q-DETR		61.1
	AQ-DETR		**63.1**

Table 1: We report AP (%) with state-of-the-art quantization methods on DETR and Deformable DETR using VOC `test2007`. #Bits (W-A-Attention) denotes the bit-width of weights, activations, and attention activations.

Implementation Details. AQ-DETR is used to quantize DETR (Carion et al. 2020) and Deformable DETR (Zhu et al. 2020) . We select ResNet-50 (He et al. 2016) as the default CNN backbone of DETR, and modify it with Pre-Activation structures and RPReLU following previous works (Liu et al. 2020, 2022b). PyTorch (Paszke et al. 2017) is used for implementing our baseline and AQ-DETR. We run the experiments on 8 NVIDIA Tesla A100 GPUs with 40 GB memory and use ImageNet ILSVRC12 (Krizhevsky, Sutskever, and Hinton 2012) to pre-train the backbone of a quantized student. The training protocol is the same as the employed frameworks (Carion et al. 2020; Zhu et al. 2020).

For learning, we use Adam (Loshchilov and Hutter 2017) with a batch size of 8 and an initial learning rate of $2e^{-4}$. The scale factor of LLD λ_{dis} is set to 0.1 and $K = 6$ in Eq. 3 as default, and the cross-entropy is chosen as the distillation loss. The quantized DETR is trained for 300 epochs and the learning rate is multiplied by 0.1 at the 200-th epoch. we use 100 object queries and 500 auxiliary queries for training, and 100 object queries for testing in the DETR framework. Following the Deformable DETR, the quantized De-

formable DETR is trained for 12 epochs, and the learning rate is multiplied by 0.1 at the 11-th epoch on both the VOC and COCO datasets. We use 300 object queries and 1500 auxiliary queries for training, and 300 object queries for testing.

Results on PASCAL VOC

We first compare our method with the full 4-bit quantization methods LSQ, LSQ+ and Q-DETR based on the same frameworks for the object detection task with the VOC dataset. We also report the detection performance of the 8-bit post-training quantization networks, such as percentile (Lin et al. 2021), VT-PTQ (Liu et al. 2021). We use the input resolution following (Carion et al. 2020), *i.e.* 1333×800.

We evaluate the proposed AQ-DETR on DETR-R50 models in Tab. 1. For the DETR-R50 model, compared with the 8-bit PTQ method, our 4-bit AQ-DETR achieves a much larger compression ratio than 8-bit VT-PTQ, and with a bit of performance improvement (53.7% *vs.* 54.7%). Also, the proposed method boosts the performance of both frameworks by 5.3%, and 4.0% with the same architecture and bit-width, which significantly validates the effectiveness of our method.

Besides, our method generates convincing results on Deformable DETR. As shown in Tab. 1, the performance of the proposed AQ-DETR with Deformable DETR-R50 outperforms the 4-4-4 bits LSQ+ method by 5.6% on AP, a large margin. Compared with full-precision methods, our method achieves a significantly higher compression rate and comparable performance.

Results on COCO

Due to the substantial diversity and size, the COCO dataset presents a more significant challenge in the object detection task compared with PASCAL VOC. We further make a comparison on the large-scale COCO dataset. We compare our method with the full 4-bit LSQ, LSQ+ and Q-DETR based on the same frameworks. We also report the detection performance of the 8-bit post-training quantization networks, such as percentile (Lin et al. 2021), VT-PTQ (Liu et al. 2021).

We summarize the experimental results on COCO `val2017` of AQ-DETR in Tab. 2. For the DETR-R50 model, compared with the 8-bit PTQ methods, our 4-bit AQ-DETR achieves a much larger acceleration than the 8-bit VT-PTQ but with an acceptable performance gap. Also, the proposed method boosts the performance of 4-4-4 bits Q-DETR by 2.8% AP with the same architecture and bit-width, which is significant on the large-scale COCO dataset. Compared with the real-valued counterparts, the proposed 4-4-4 bits AQ-DETR achieves computation acceleration and storage savings by 7.25 × and 8.01×. The above results are of great significance in the real-time inference of object detection. All of the improvements have beneficial impacts on object detection.

For the Deformable DETR-R50 model, we observe similar performance improvements and compression ratios. For example, the 4-bit AQ-DETR DETR-R50 theoretically accelerates 6.39× with only a 3.6% performance gap compared with the real-valued counterpart, which is significant for real-time DETR methods.

(1) Full-precision model (2) Quantizing backbone (3) Quantizing Encoder
(4) Quantizing MLP in Decoder (5) Quantizing MHSA & MHCA in Decoder

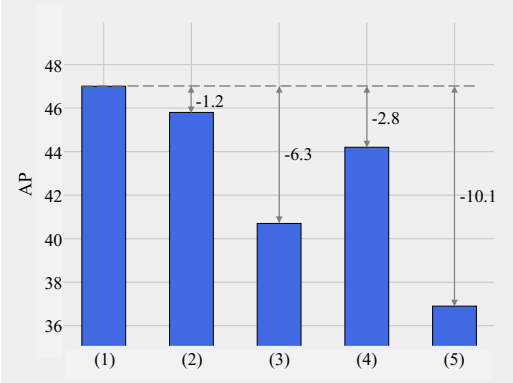

Figure 5: AP performance of Deformable DETR-R50 with different 4-bit quantized modules on COCO.

Ablation Study

In the following ablation, we use the Deformable DETR structure with 4-4-4 bits AQ-DETR.

Effectiveness of Components. We conduct the quantitively ablative experiments by replacing one module of the real-valued DETR baseline with its quantized one and compare the AP drop on the COCO dataset as shown in Fig. 5. We can see that quantizing MHSA & MHCA in Decoder causes the largest degradation, about -10.1 AP.

We show quantitative improvements of components in AQ-DETR in Tab. 3. As shown in Tab. 3, the quantized DETR baseline suffers a severe performance drop on AP (11.8%). AQ and LLD both improve the performance e with respect to the baseline, and their combination further improves the performance considerably. Specifically, compared with LSQ+ alone, AQ, LLD and their combination improve the LSQ+ by 5.8%, 3.3% and 9.5% respectively.

Auxiliary Queries. We study the influence of the total number of auxiliary queries n_A within the Deformable DETR framework by fixing the multiple of ground truth repeated K (See Eq. 7) as 6. Sixteen images are selected as a batch input, with a total of 21 ground truths. As shown in Tab. 4, as the number of auxiliary queries increases, the number of matches between queries and ground truth increases. This indicates the auxiliary queries are beneficial for increasing the number of matches. The optimum performance of AQ-DETR is achieved when $n_A = 1500$, as the number of matches starts to saturate. Despite n_A continues to increase, both the number of matches and AP tend to stabilize. In this case, we choose $n_A = 1500$ as a default value on the COCO object detection task.

Layer-by-Layer Distillation. We test different $\mathcal{E}$ in Eq. 8, such as cross-entropy, KL divergence and cosine similarity. Tab. 5 shows that the cross-entropy loss achieves the best performance while the cosine similarity even having a negative impact.

Model	Method	#Bits	Size (MB)	OPs (G)	AP
DETR-R50	Full-precision	32-32-32	159.3	85.5	42.0
	Percentile	8-8-8	39.8	23.0	38.6
	VT-PTQ				41.2
	LSQ	4-4-4	19.9	11.8	27.9
	LSQ+				31.2
	Q-DETR				37.4
	AQ-DETR				**40.2**
Deformable DETR-R50	Full-precision	32-32-32	193.9	177.0	47.0
	Percentile	8-8-8	48.5	53.6	43.2
	VT-PTQ				45.6
	LSQ	4-4-4	24.9	27.7	31.9
	LSQ+				35.2
	Q-DETR				40.7
	AQ-DETR				**44.1**

Table 2: Comparison with state-of-the-art quantization methods using DETR and Deformable DETR on COCO `val2017`. #Bits (W-A-Attention) denotes the bit-widths of weights, activations, and attention activations.

Model	Method	#Bits	AP
Deformable DETR-R50	Real-valued	32-32-32	47.0
	LSQ+	4-4-4	35.2
	+AQ	4-4-4	41.0
	+LLD	4-4-4	38.5
	+AQ+LLD (AQ-DETR)	4-4-4	**44.1**

Table 3: Evaluating the components of AQ-DETR with Deformable DETR-R50 on the COCO dataset. #Bits (W-A-Attention) denotes the bit-width of weights, activations, and attention activations. AQ denotes the Auxiliary Queries module, while LLD denotes the Layer-by-Layer Distillation module.

n_A	300	600	900	1200	1500	1800
Match number	17	33	49	58	76	79
AP	38.5	41.7	42.3	43.1	44.1	43.8

Table 4: Comparison of different auxiliary queries n_A on COCO. The match number denotes the number of matching queries and positive samples.

We also show the comparison of different distillation strategies in Tab. 6. If we exclude the distillation in our method (denoted by None), the AP decreases from 44.1% to 41.0%. We further explore One layer distillation, *i.e.*, only conducting distillation with the latest encoder layer. We can see that our LLD outperforms one layer distillation. We believe it is because the distillation of intermediate features progressively contributes to the enhancement of quantized features (Zheng et al. 2022).

Additionally, we apply a pre-trained encoder, which is trained on COCO dataset, as the teacher model to distillate the quantized encoder. This causes AP performance drops compared to our LLD. We suppose the reason is that the distribution of the pre-trained encoder's output is far from the 4-bit quantized encoder, leading to a difficult optimization for the quantized encoder. At the same time, the pre-trained encoder does not distill the intermediate features, leading to the lack of intermediate supervisory information.

Loss	Cross entropy	KL divergence	Cosine similarity
AP	**44.1**	43.3	39.7

Table 5: AP comparison of different loss functions for $\mathcal{L}_{dis}$ on COCO.

None	One layer distillation	Pre-trained teacher	LLD
41.0	41.9	42.4	**44.1**

Table 6: AP comparison of different distillation strategies on COCO.

Conclusion

This paper proposes a novel method, AQ-DETR, for training quantized DETR with auxiliary queries and a layer-by-layer distillation. For auxiliary queries, we introduce an additional matching branch that assigns multiple auxiliary queries to each feature, which significantly enhances the capacity of queries but without extra cost for the test phase. For the layer-by-layer distillation, it is proposed to maintain the feature distribution of the encoder. Together with our auxiliary queries, we increase the matching of object queries and positive samples, and significantly improve the QAT performance for DETR. As a result, AQ-DETR achieves full 4-bit quantization of DETR with comparable performance with respect to its full-precision model. Moreover, extensive experiments show that AQ-DETR surpasses state-of-the-art quantization methods for DETR quantization. Besides Deformable DETR, we will evaluate AQ-DETR on more DETR variants in the future.

Acknowledgements

This research was supported by Zhejiang Provincial Natural Science Foundation of China under Grant No. LD24F020007, Beijing Natural Science Foundation L223024, National Natural Science Foundation of China under Grant 62076016, National Key Research and Development Program of China (Grant No. 2023YFC3300029), "One Thousand Plan" projects in Jiangxi Province Jxsg2023102268, ATR key laboratory grant 220402, 232-CXCY-A01-08-06-01, National Natural Science Foundation of China (NO. 62102151), the Shanghai Sailing Program (21YF1411200), CCF-Tencent Rhino-Bird Open Research Fund, the Open Research Fund of Key Laboratory of Advanced Theory and Application in Statistics and Data Science, Ministry of Education (KLATASDS2305), and the Fundamental Research Funds for the Central Universities.

References

Bengio, Y.; Léonard, N.; and Courville, A. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*.

Bhalgat, Y.; Lee, J.; Nagel, M.; Blankevoort, T.; and Kwak, N. 2020. Lsq+: Improving low-bit quantization through learnable offsets and better initialization. In *CVPR Workshops*.

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-end object detection with transformers. In *ECCV*.

Esser, S. K.; McKinstry, J. L.; Bablani, D.; Appuswamy, R.; and Modha, D. S. 2019. Learned step size quantization. *arXiv preprint arXiv:1902.08153*.

Everingham, M.; Van Gool, L.; Williams, C. K.; Winn, J.; and Zisserman, A. 2010. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2): 303–338.

Gao, P.; Zheng, M.; Wang, X.; Dai, J.; and Li, H. 2021. Fast convergence of detr with spatially modulated co-attention. In *ICCV*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*.

Hu, Y.; Yang, J.; Chen, L.; Li, K.; Sima, C.; Zhu, X.; Chai, S.; Du, S.; Lin, T.; Wang, W.; et al. 2023. Planning-oriented autonomous driving. In *CVPR*.

Jia, D.; Yuan, Y.; He, H.; Wu, X.; Yu, H.; Lin, W.; Sun, L.; Zhang, C.; and Hu, H. 2023. Detrs with hybrid matching. In *CVPR*.

Jing, Y.; Zhu, X.; Liu, X.; Sima, Q.; Yang, T.; Feng, Y.; and Kong, T. 2023. Exploring Visual Pre-training for Robot Manipulation: Datasets, Models and Methods. *arXiv preprint arXiv:2308.03620*.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *NeurIPS*.

Li, F.; Zeng, A.; Liu, S.; Zhang, H.; Li, H.; Zhang, L.; and Ni, L. M. 2023. Lite DETR: An interleaved multi-scale encoder for efficient detr. In *CVPR*.

Li, F.; Zhang, H.; Liu, S.; Guo, J.; Ni, L. M.; and Zhang, L. 2022a. Dn-detr: Accelerate detr training by introducing query denoising. In *CVPR*.

Li, Y.; Xu, S.; Zhang, B.; Cao, X.; Gao, P.; and Guo, G. 2022b. Q-vit: Accurate and fully quantized low-bit vision transformer. In *NeurIPS*.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *ECCV*.

Lin, Y.; Zhang, T.; Sun, P.; Li, Z.; and Zhou, S. 2021. FQ-ViT: Post-Training Quantization for Fully Quantized Vision Transformer. In *IJCAI*.

Lin, Y.; Zhang, T.; Sun, P.; Li, Z.; and Zhou, S. 2022. Fq-vit: Post-training quantization for fully quantized vision transformer. *IJCAI*.

Liu, S.; Li, F.; Zhang, H.; Yang, X.; Qi, X.; Su, H.; Zhu, J.; and Zhang, L. 2022a. Dab-detr: Dynamic anchor boxes are better queries for detr. *arXiv preprint arXiv:2201.12329*.

Liu, Z.; Cheng, K.-T.; Huang, D.; Xing, E. P.; and Shen, Z. 2022b. Nonuniform-to-Uniform Quantization: Towards Accurate Quantization via Generalized Straight-Through Estimation. In *CVPR*.

Liu, Z.; Shen, Z.; Savvides, M.; and Cheng, K.-T. 2020. ReActNet: Towards Precise Binary Neural Network with Generalized Activation Functions. In *ECCV*.

Liu, Z.; Wang, Y.; Han, K.; Zhang, W.; Ma, S.; and Gao, W. 2021. Post-training quantization for vision transformer. In *NeurIPS*.

Loshchilov, I.; and Hutter, F. 2017. Decoupled weight decay regularization. In *ICLR*.

Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; and Lerer, A. 2017. Automatic differentiation in pytorch. In *NeurIPS Workshops*.

Qin, H.; Ding, Y.; Zhang, M.; Yan, Q.; Liu, A.; Dang, Q.; Liu, Z.; and Liu, X. 2022. Bibert: Accurate fully binarized bert. In *ICLR*.

Rastegari, M.; Ordonez, V.; Redmon, J.; and Farhadi, A. 2016. Xnor-net: Imagenet classification using binary convolutional neural networks. In *ECCV*.

Redmon, J.; and Farhadi, A. 2018. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*.

Xu, S.; Li, Y.; Lin, M.; Gao, P.; Guo, G.; Lü, J.; and Zhang, B. 2023. Q-DETR: An Efficient Low-Bit Quantized Detection Transformer. In *CVPR*.

Zhang, B.; Wang, R.; Wang, X.; Han, J.; and Ji, R. 2021. Modulated convolutional networks. *IEEE Transactions on Neural Networks and Learning Systems*.

Zheng, H.; Wang, R.; Liu, J.; and Kanezaki, A. 2022. Cross-level distillation and feature denoising for cross-domain few-shot classification. In *ICLR*.

Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; and Dai, J. 2020. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*.