This ACCV 2024 paper, provided here by the Computer Vision Foundation, is the author-created version.
The content of this paper is identical to the content of the officially published ACCV 2024
LNCS version of the paper as available on SpringerLink: https://link.springer.com/conference/accv

QR-DETR: Query Routing for Detection Transformer

Tharsan Senthivel[1,2] and Ngoc-Son Vu[2]

[1] PMU, Pari Mutuel Urbain, France
[2] ETIS - CY Cergy Paris University, ENSEA, CNRS, France

Abstract. Detection Transformer (DETR) predicts object bounding boxes and classes from learned object queries. However, DETR exhibits three major flaws: (1) Only a *subset of object queries contribute to the final predictions*, leading to inefficient utilization of computational resources. (2) The self-attention and cross-attention layers *indiscriminately mix information across object queries* without any guidance, potentially hindering effective learning of object representations. (3) At each decoder stack layers, a query are processed either *positively*, refining its bounding box and class attributes correctly, or *negatively*, shifting to predict a different object or increasing its bounding box erroneously. This suggest that query informativeness is non-uniform, and enabling inter-query communication could impede the learning of specialized representations for individual queries. To address these concerns, we propose a learnable query routing method that introduces a routing model to identify the object queries requiring processing at each transformer decoder layer. Selected queries pass through the full decoder, while others exit early, and all are scattered back after processing. This prevents indiscriminate information sharing. Extensive COCO experiments show consistent mAP improvements across various DETR models.

1 Introduction

Object detection is a fundamental task in computer vision that involves recognizing objects and precisely localizing them in an image using bounding boxes. Recently, DETR (DEtection TRansformer) [2] has been introduced to perform end-to-end object detection. By utilizing a transformer encoder-decoder architecture, DETR leverages a set of learnable object queries to perform detection, thereby eliminating hand-crafted components like non-maximum suppression (NMS). This approach takes advantage of the powerful learning capabilities of transformers. However, DETR suffers from high computational complexity and requires long training schedules.

A key component in DETR are the object queries $\mathbf{Q}$, which are learnable embeddings that are processed through the transformer layers to provide the final predictions. These queries communicate and share information across the different stacks of the transformer decoder. However, the object queries often share **redundant information**, which hinders their evolution [14, 39, 43]. Rather

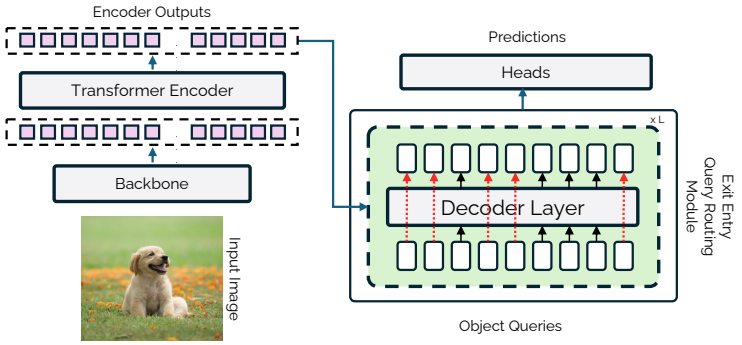


Fig. 1: QR-DETR is a plug-and-play module for the DETR decoder. It uses a backbone to extract features and a transformer encoder for augmented image representation. The decoder's object queries interact with these encoded features. The novelty lies in the entry-exit query routing (EEQR) module at each decoder block, which selectively permits (black) or blocks (red) object queries from passing through specific blocks.

than progressively refining the captured content as it passes through the decoder layers, the interaction between object queries can have two outcomes: a query may refine its bounding boxes, improving the Intersection over Union (IoU) with correct class predictions, or its confidence may decrease in subsequent layers, negatively affecting IoU or class predictions [3]. This is mainly caused by the attention layers, which facilitate information flow between object queries. While self-attention (SA) layers contribute to the removal of duplicate queries, cross-attention (CA) layers are responsible for generating duplicate predictions [15]. This raises the question of the utility of a query if its confidence can fluctuate significantly in this manner.

Manually pruning low average precision (AP) queries and selecting the highest AP queries from each decoder layer can maintain or even improve overall AP performance [46]. Previous methods, such as query recollection [3], aim to utilize information across multiple transformer decoder stages. However, they introduce queries from various layers, potentially leading to noise or misalignment issues. These approaches deviate from the natural flow of object queries being updated at each decoder layer, primarily due to a lack of understanding of the query representation. Motivated by this ambiguity, a pertinent research question arises: **Can selective query skipping of certain transformer decoder layers be performed without compromising prediction performance?**

To address this question, we propose QR-DETR, a plug-and-play query routing strategy for most DETR-based detectors. As illustrated in Fig. 1, we introduce a lightweight wrapper around decoder blocks that employs a learned gating mechanism to determine, for each individual object query, whether it should bypass the current transformer layer or proceed through it. This selective layer skipping reduces the number of supervision signals a query encounters, allowing the model to dynamically adapt the computation for different queries. Specifically, QR-DETR introduces three types of decoder block wrappers that route

queries across different decoder layers, reducing computational complexity. This selective query routing enhances the efficiency and effectiveness of the original DETR framework, improving detection performance.

We summarize our contributions as follows:

- We investigate the utility of all queries in a DETR model and show that not all queries are necessary.
- We design specific plug-and-play routing layers that effectively redirect queries through a decoder stack when needed.
- We conduct experiments on different DETR model across various training settings to verify the effectiveness of the proposed method.

2 Related Works

We introduce routing mechanisms to modify object queries in DETR. This section reviews relevant works: recent DETR improvements, object query modifications, and routing models like Mixture of Experts (MoE).

2.1 Detection Transformer (DETR) and variants

The original DETR [2] introduced an end-to-end transformer-based [42] object detection model that eliminated the need for handcrafted components like non-maximum suppression (such as NMS). However, a significant limitation of DETR was its prolonged training schedule, requiring 500 epochs for convergence. This slow convergence was attributed to multiple factors. First, the lack of explicit positional information necessitates that the model jointly learns positional representations along with the object content. To alleviate the issue of position of queries, Deformable-DETR [52] aimed to address the issue of object query positioning by incorporating deformable attention mechanisms bringing a small set of key sampling points. Further improvements were made by providing anchor points to the object queries, enhancing the model's performance [4, 24–26, 30, 33, 38, 45], while highly reducing the necessitated training time. Another challenge in DETR is the instability of the bipartite matching process, which involves assigning object queries to ground truth labels for optimization. Several works aimed to stabilize this bipartite matching process. DN-DETR [22] introduced denoising query groups, which was further extended by DINO-DETR [48] with the incorporation of contrastive denoising query groups. Other approaches simply added duplicate groups of queries [5, 6, 9, 10, 12, 19, 50, 53]. These methods focused on increasing the number of positive queries to improve the matching process and overall model performance. This work is the first work to introduce query routing strategy in DETRs.

2.2 Object Query Modification

The cross-attention and self-attention mechanisms serve distinct purposes but are crucial components in the DETR architecture. Self-attention enables query-

to-query interactions, facilitating information sharing, while cross-attention aggregates features from the encoded input features. Focus-DETR [51] and Sparse DETR [36] introduce changes to the encoder token representations and initialize the object queries based on these modified tokens. SAM-DETR [47] incorporates a query alignment module between the self-attention and cross-attention layers to efficiently integrate information into the queries. To tackle redundancy, Salience-DETR [14] employs hierarchical query filtering to mitigate redundancy in encoding and selection, whereas DOQ-DETR [39] introduces a redundancy reduction loss to diversify the object queries, alleviating redundancy among them. SQR-DETR [3] proposes a selective query recollection strategy that cumulatively collects selected object queries to mitigate the potential impact of oversupervision in later stage queries. Similarly, DAC-DETR [15] aims to improve training efficacy by introducing a Divide-And-Conquer approach, which separates the cross-attention to avoid competing objectives. Each of these methods integrate informations within the object queries, adding supplementary signals. In contrast to existing methods that aim to integrate new information into the object queries, we propose to not update the object queries independently of their content.

2.3 Routing Model

Mixture of Experts (MoE) [17] approach introduces a set of learnable expert models, where a gating mechanism selectively routes each input to a subset of experts. The gating mechanism uses a softmax function to predict, per input, the probability of each expert being optimal, computing the output from only the top-k highest-probability experts [11]. MOE are extensively used from language models [7, 8, 20, 28, 31, 40, 49] notably thanks to their capacities to accelerate Transformer [42] based architecture. Although Mixture-of-Experts (MoE) layers have accelerated Transformers for visual tasks like classification [16, 29, 37, 44], their application to object detection has been limited. [18] introduces a Dataset-Aware MoE (DAMEX) layer that specializes experts across datasets while sharing information, enabling MoE training on a mixture of object detection datasets. MOE has also been applied to leverage deep ensembling, as in [32], which proposes a Mixture-of-Calibrated Experts (MoCAE) approach that performs deep ensembling over multiple object detection models to enhance the overall performance. In contrast to introducing and leveraging multiple experts replacing the FFN module of a decoder transformer stack, our approach introduces a simple routing model only used to selectively bypass or pass through certain transformer layers for a selected pool of object queries.

3 Effects of Object Query Variation on DETR Performance

To explore the significance of each query and assess how varying object query numbers affect DETR performance, we conduct an initial analysis using a trained Deformable-DETR model. After reviewing DETR's architecture, we analyze how

object queries are processed through the decoder layers, undergoing intercommunication and extracting information from the encoded feature map $\mathbf{F}$.

Notation. We denote the object queries at stack l of the transformer decoder as $\mathbf{Q}^l_{\texttt{X}}$, where the subscript $\texttt{X}$ corresponds to either the $\texttt{SA}$ (self-attention) or $\texttt{CA}$ (cross-attention) operation. We denote $\texttt{Que}$, $\texttt{Key}$, and $\texttt{Val}$ as the query, key, and value linear projections, respectively. For clarity, we omit the layer normalization and dropout layers from the equations.

3.1 Preliminary: Architecture of DETR

DETR processes the input image $I \in \mathbb{R}^{H_0 \times W_0 \times C}$ as follows: A backbone network encodes I into a feature map, which is flattened into a sequence and processed by transformer encoder layers to obtain enhanced pixel embedding sequence $\mathbf{F} \in \mathbb{R}^{HW \times d}$, where C represent the channel numbers, d the embedding dimension, while H, W, H_0, and W_0 denote the spatial shapes, of the input image and encoded features. In the transformer decoder, a set of N_q object queries $\mathbf{Q} = \{q_0, q_1, \ldots, q_{N_q}\} \in \mathbb{R}^{N_q \times d}$, where N_q is typically 300, sequentially undergo self-attention and cross-attention. The self-attention allows the object queries to interact and share information. The cross-attention enables each queries to aggregate relevant information from the encoded feature map $\mathbf{F}$. This process is repeated though multiple decoder stack. Subsequently, multi-layer perceptron (MLP) heads process each query from the last stack, providing a set of bounding box coordinates and class predictions for the detected objects. Finally, these predictions are matched to ground-truth objects via Hungarian matching, and the queries with minimal cost are supervised using bounding box (L_1 and L_{GIoU}) and focal loss for class prediction.

3.2 Query Importance Across Various Decoder Operators

a. Self-Attention (SA) layers. The query representation $\mathbf{Q}^l$ are processed across the network layers. The updated query representation $\mathbf{Q}^l_{SA}$ is computed as:

$$\mathbf{Q}^l_{\texttt{SA}} = \mathbf{Q}^l + \texttt{SA}(\texttt{Que}(\mathbf{Q}^l), \texttt{Key}(\mathbf{Q}^l), \texttt{Val}(\mathbf{Q}^l)) \tag{1}$$

Remark. SA layers facilitate inter-query communication, enabling the sharing of informative features. Within this layer, due to query communication, each query disperses to specific locations. This spatial dispersion minimizes interference from other queries, crucially aiding in duplicate detection removal [15]. However, it is noteworthy that irrelevant queries, those lacking accurate object detection or classification predictions, may propagate their uninformative representations to relevant queries during this communication process [39].

b. Cross-Attention (CA) layers communicate object queries with the encoded feature map $\mathbf{F}$, and integrates features from the input data to enhance object

detection effectiveness. The output queries $\mathbf{Q}^l_{CA}$ are computed as:

$$\mathbf{Q}^l_{\texttt{CA}} = \mathbf{Q}^l_{\texttt{SA}} + \texttt{CA}(\texttt{Que}(\mathbf{Q}^l_{\texttt{SA}}), \texttt{Key}(\mathbf{F}), \texttt{Val}(\mathbf{F})) \tag{2}$$

Remark. The output queries $\mathbf{Q}^l_{\texttt{CA}}$ are obtained by combining the self-attended queries $\mathbf{Q}^l_{\texttt{SA}}$ with the cross-attention features derived from the input embedding F, enabling the incorporation of relevant information from the input data into the object queries. In this layer, object queries interact with the feature map to retrieve position-specific information [25,52]. However, due to Hungarian matching, query-to-ground truth assignments vary across epochs, leading to different queries encoding the same content over time.

c. Feed Forward Networks (FFN). Finally, all $\mathbf{Q}^l_{\texttt{CA}}$ are fed into a FFN and processed to form the queries for next stack:

$$\mathbf{Q}^{l+1} = \text{FFN}(\mathbf{Q}^l_{\texttt{CA}}) \tag{3}$$

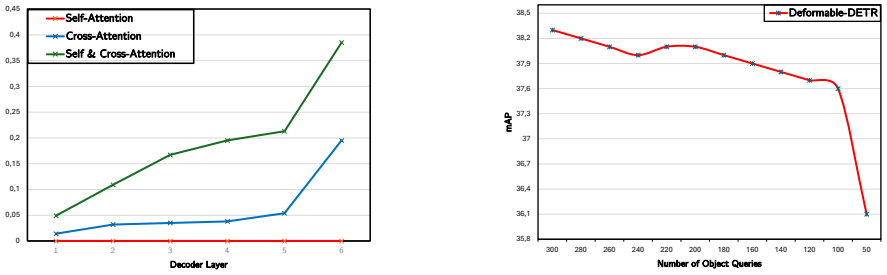
The FFN layer independently processes each query, without inducing specific query communication in this layer.

3.3 Impact of Varying Object Query Numbers

Visual inspection of detected objects indicates that certain object queries experience a deterioration in their alignment with assigned ground truth instances across transformer layers, likely due to the propagation of redundant or conflicting features through SA and CA mechanisms [51]. To assess if object queries can be skipped from the SA and CA layers without impacting performance, and to determine the number that can be skipped, we progressively compute the mean Average Precision (mAP) on a trained Deformable DETR model with varying numbers of queries. Specifically, we evaluate three configurations: ① only SA layers, ② only CA layers, and ③ both SA and CA layers, initially using 300 queries. We then gradually reduce the number of queries and observe the mAP

a. Effect of different components on AP. As can be seen in Fig. 2a, using only the SA layers results in null AP performance. This is expected, as in the SA layers, object queries interact solely with each other to share information. The SA mechanism helps establish and communicate initial content representations, but without interacting with image features, the queries fail to align with the image, leading to zero AP performance. Conversely, CA layers allow queries to refine predictions by accessing image features, enabling precise information extraction. This highlights the importance of CA. The best results are achieved by combining both SA and CA layers, as they collaboratively refine the object representations.

b. Effect of number of queries on AP. As shown in Fig. 2b, reducing the number of object queries results in a moderate decline in AP performance, contrary to the anticipated substantial drop. Reducing the object query count of a trained Deformable-DETR model from 300 to 50 results in a maximum AP drop of



(a) Impact on AP when using only SA, CA, and both in Deformable DETR.

(b) Effect of reducing object query count on mAP in a pre-trained Deformable-DETR.

Fig. 2: Impact of removing SA and CA layers and number of object queries on mAP.

2.1. However, decreasing the query count from 300 to 150 leads to minimal AP degradation (-0.5 AP). We hypothesize that this moderate degradation in AP is due to the redundancy in learned features across object queries [39], resulting in similar representations among queries, sufficient for final prediction. This highlights that not all object queries are essential for the effective functioning of the DETR model, motivating the development of our QR-DETR.

4 QR-DETR

Based on the analyses in the previous section, we hypothesize that object queries can be partitioned into relevant and irrelevant subsets:

$$\mathbf{Q} = [\mathbf{Q}_r, \mathbf{Q}_{ir}] \in \mathbb{R}^{N_q \times d} \tag{4}$$

where the subscript r and ir corresponds to relevant or irrelevant. Unfortunately, there is no well-defined metric to reliably identify optimal object queries, as they encode a combination of positional and content information. Providing a learnable confidence score and threshold for each query is inherently unstable, as the optimal threshold may vary across queries.

To address this issue, we propose an entry-exit query strategy called QR-DETR, which employs a routing mechanism that allows the model to learn and determine which object queries to direct through or out of the stack. This process is conducted end-to-end, without any constraints. This data-driven strategy enables the model to determine the most suitable object queries based on their overall contribution. The flexibility introduced by this approach addresses the absence of a proper metric for evaluating individual object query quality. By allowing the model to freely select object queries, it can potentially identify the most effective representations. In this way, these queries can be refined by passing through the layer or retained as they are by passing out of the layer.

Our QR-DETR comprises three wrappers around decoder blocks, selectively processing or bypassing queries through each block based on a router (Fig. 3).

To address the lack of metrics for evaluating query usefulness, we propose incorporating routers to dynamically filter redundant or uninformative queries, improving efficiency and performance.

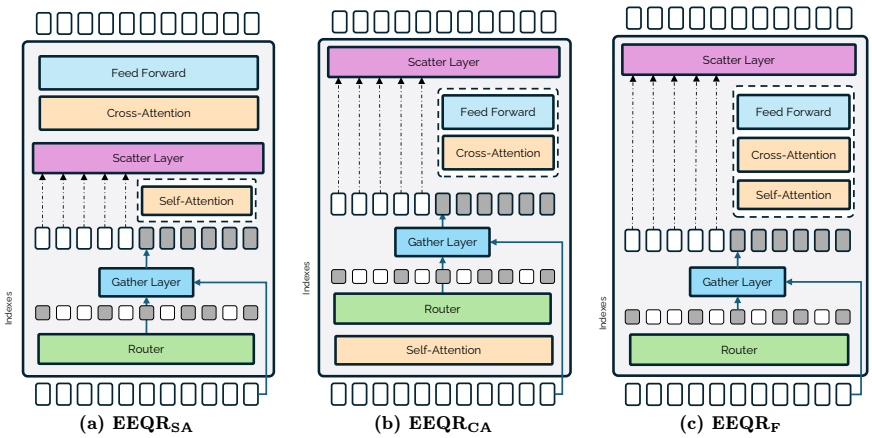


Fig. 3: Entry-Exit Query Routing mechanism for redirecting queries in Transformer decoder. (a) **EEQR$_{SA}$** bypasses the SA layer. (b) **EEQR$_{CA}$** bypasses the CA layer. (c) **EEQR$_{F}$** skips the entire decoder layer. Bypassed layers are highlighted in white.

4.1 Three Components of Entry Exit Query Routing (EEQR) Module

Our algorithm can be seamlessly integrated into various DETR-based architectures. It employs a routing mechanism through strategically chosen layers and encapsulates any decoder stack within the DETR decoder.

a. Router layer computes a routing score for each query to determine whether it should be propagated to the subsequent layer. Our gating mechanism first computes this routing score to decide whether an object query passes through a layer. We accomplish this by projecting the object queries into a 2D space using a learnable projection matrix $\mathbf{W}_R \in \mathbb{R}^{d\times 2}$:

$$\text{score} = \texttt{Linear}(\mathbf{Q}) = \mathbf{Q}\mathbf{W}_R \in \mathbb{R}^{N_q\times 2} \tag{5}$$

This matrix is the only additional parameters we introduce into DETR-based models. The score values are binarized to identify the indices of queries that should bypass or pass through a given transformer layer:

$$\sigma_{\text{EE}} = \texttt{max}(\texttt{sigmoid}(\text{score})) \in \mathbb{R}^{N_q} \tag{6}$$

Consequently, σ_{EE}, where 'EE' stands for 'entry exit', is partitioned into two disjoint subsets. We denote σ_{s} and σ_{p} as the disjoint subsets of σ_{EE}, where σ_{s}

contains the indices of queries that pass through the current layer, and σ_{p} contains the indices of queries that bypass it. These values are pivotal for effectively routing the queries through the decoder layer.

Optimizing Router Layer. To address gradient flow inhibition caused by the max operation in the Router layer, we implement exit-heads [21,34] similar to the final prediction heads. These exit-heads project score values and receive supervision through the same loss function used for prediction heads, enabling targeted training of this layer.

In the following, we denote $\mathbf{Q}_s$ and $\mathbf{Q}_p$ as the queries to be sent and bypassed through the layer.

b. Gather layer, denoted as `Gth(.)`, collects the selected queries based on the routing scores and feeds them through the subsequent transformer layers.

$$\begin{aligned}\mathbf{Q}_s = \texttt{Gth}(\mathbf{Q}, \sigma_s) &= \mathbf{Q}(\{\sigma_s\}) \\ &= (q_{\sigma_s(0)}, \ldots, q_{\sigma_s(N_{q_s})}) \in \mathbb{R}^{N_s \times d}\end{aligned} \tag{7}$$

where $N_s << N_q$ represents the number of queries selected to be sent through the specific layer. Conversely, the $\mathbf{Q}_p$ queries will bypass this layer.

c. Scatter layer, denoted as `Sct(.)`, redistributes the processed queries back to their original positions, thereby preserving the positional information encoded in the object query positional embeddings.

$$\begin{aligned}\mathbf{Q} &= \texttt{Sct}(\mathbf{Q}_s, \mathbf{Q}_p) \\ &= (q_{\sigma_s(0)}, \ldots, q_{\sigma_s(N_{q_s})}) \in \mathbb{R}^{N_s \times d}\end{aligned} \tag{8}$$

This modular approach allows the model to focus on the most informative object queries while preserving the spatial relationships learned by the positional embeddings. This ensures the spatial disposition of the queries is maintained, eliminating the need for additional alignment in the decoder layer.

4.2 Three Variants of EEQR Modules

Building on the three previously presented layers, we developed three distinct EEQR modules, each incorporating our Gather-and-Scatter strategy to bypass specific blocks within each decoder layer. We will now present the three modules in more detail, as illustrated in Fig. 3. For clarity, we omit the superscript l indicating the stack index for the object queries $\mathbf{Q}$, as all elements discussed here belong to the same decoder stack.

a. EEQR$_{\texttt{SA}}$. SA layers facilitate query communication and duplicate prediction removal but also result in redundant information sharing. To address this, we first propose a routing module that bypasses the SA layer then passes through the CA layer, as illustrated in Fig. 3a. This approach allows updating the object queries directly from the feature maps without sharing information across queries from different levels. Specifically, the computations are performed in the following sequence:

$$\mathbf{Q}_{\texttt{SA}} = \texttt{SA}(\texttt{Gth}(\mathbf{Q}, \sigma_s)), \quad \mathbf{Q}_{\texttt{CA}} = \texttt{Sct}(\mathbf{Q}_{\texttt{SA}}, \mathbf{Q}_P), \quad \mathbf{Q} = \texttt{FFN}(\mathbf{Q}_{\texttt{CA}}) \tag{9}$$

b. EEQR$_{\mathtt{CA}}$. The CA layers enable object queries to capture information from the encoded feature maps, thereby updating their content. However, while all object queries are updated, only a subset is supervised, either by the final loss or the auxiliary losses computed within each stack. Consequently, some queries are subject to over-supervision, being supervised multiple times across different stacks. As illustrated in Fig. 3b, our proposed routing module allows queries to bypass the CA layer, preventing them from updating their content based on the feature maps. The computations are:

$$\mathbf{Q}_{\mathtt{SA}} = \mathtt{SA}(\mathbf{Q}), \quad \mathbf{Q}_{\mathtt{CA}} = \mathtt{Gth}(\mathbf{Q}_{\mathtt{SA}}, \sigma_s), \quad \mathbf{Q} = \mathrm{FFN}(\mathtt{Sct}(\mathbf{Q}_{\mathtt{CA}}, \mathbf{Q}_P)) \tag{10}$$

c. EEQR$_{\mathtt{F}}$. As noted in [3], object queries can shift their associations across transformer layers, underscoring the importance of the decoder stack for consistent object query updates from both self-attention and cross-attention modules. Consequently, not all layers are necessary, and our routing module selectively bypasses entire decoder stacks. Following this, as illustrated in Fig. 3c, the **EEQR**$_F$ router design is as follows:

$$\mathbf{Q}_{\mathtt{SA}} = \mathtt{SA}(\mathtt{Gth}(\mathbf{Q}, \sigma_s)), \quad \mathbf{Q}_{\mathtt{CA}} = \mathtt{CA}(\mathbf{Q}_{\mathtt{SA}}), \quad \mathbf{Q} = \mathrm{FFN}(\mathtt{Sct}(\mathbf{Q}_{\mathtt{CA}}, \mathbf{Q}_P))$$

In summary, each wrapper module selectively routes object queries to bypass certain transformer layers. QR-DETR adds computational overhead, mainly from the routing mechanism's gathering and scattering layers.

5 Experiments

We first detail our training settings, then present QR-DETR's consistent AP improvements on COCO over various baselines. Ablation studies and qualitative/quantitative results are also provided for deeper performance insights.

5.1 Settings

Dataset. Similar to other DETR methods [15, 25, 30, 45], we evaluate our approach on the challenging COCO 2017 [23] object detection benchmark, which consists of 118K training and 5K validation images with diverse object instances of varying sizes.

Technical Details. We follow the original DETR training protocol and utilize the same training losses (focal for classification and L_1 and GIoU for regression). Following common practice, the backbone network is initialized with ImageNet-pretrained ResNet-50 weights and the Transformer parameters are initialized using the Xavier Initialization scheme. We use the AdamW [27] optimizer with different learning rates: 10^{-5} for the backbone network and 10^{-4} for the Transformer encoder and decoder. The data augmentation scheme is the same as in DETR: we use scale augmentation, resizing the input images such that the shortest side is at least 480 and at most 800 pixels while the longest at most 1333. The

models are trained on 8 Nvidia V100 GPUs, with a batch size of 8 for training. The models are trained for 12 epochs, with a learning rate decay of 10^{-1} at the 11^{th} epoch.

Evaluations. We use standard object detection metrics on the COCO dataset, reporting mAP as the main performance metric, averaging bounding box AP across multiple IoU thresholds. We also assess our models' computational complexity in terms of Multiply-ACcumulate operations (MACs) to evaluate efficiency.

5.2 Main Results

To demonstrate the advantage of our algorithm, namely a plug-in method designed for seamless integration with DETR-based approaches, we evaluated the performance of EEQR in conjunction with three standard DETR variants, deformable DETR, conditional DETR, and DAB DETR.

Methods	Epoch	w/QR	AP	$AP_{0.50}$	$AP_{0.75}$	AP_S	AP_M	AP_L
Faster RCNN-FPN [35]	108		42.0	62.5	45.9	25.2	45.6	54.6
DETR [2]	500		42.0	62.4	44.2	20.5	45.8	61.1
Deformable-DETR [52]	50		39.4	59.6	42.3	20.6	43.0	55.5
SAM-DETR [47]	50		39.8	61.8	41.6	20.5	43.4	59.6
Align-DETR [1]	50		46.0	64.9	49.5	25.2	50.5	64.7
AdaMixer [13]	12		44.1	63.1	47.8	29.5	47.0	58.8
SQR-DETR [3]	12		39.9	58.4	43.7	23.8	43.2	53.3
StageInteractor [41]	12		46.3	64.3	50.6	29.8	49.6	60.8
Deformable-DETR [52]	12		38.3	58.0	41.3	23.6	41.7	51.4
QR-Deformable-DETR	12	✓	**43.1** (+4.8)	62.2	46.4	26.5	46.4	56.9
Conditional-DETR [30]	12		36.6	57.3	40.1	19.9	39.4	53.4
QR-Conditional-DETR	12	✓	**39.9** (+3.3)	60.9	41.4	20.3	45.2	56.2
DAB-DETR [25]	12		42.2	63.1	45.7	21.6	45.2	60.3
QR-DAB-DETR	12	✓	**43.7** (+1.5)	63.2	47.3	27.7	46.9	57.1

Table 1: Performance on COCO 2017. We showcase the performance of our pipeline on various DETR models comprising Deformable, Conditional and DAB with $N_q = 300$.

Comparison with SOTA. Table 1 compares the performance of Deformable, Conditional and DAB-DETR models in conjunction our EEQR wrapper to the baselines. Our method outperforms many existing baseline methods, proving its integrability and consistently improving the mAP across various DETR variants. Under same training conditions, our proposed plug-and-play training scheme consistently improves performance, achieving up to +4.8 AP gain over Deformable-DETR and +3.3 AP gain over Conditional-DETR. We also note that our proposed methods, trained for only 12 epochs, surpass the performance of existing approaches such as SAM-DETR and Deformable-DETR, which require

training for 50 epochs. Specifically, QR-Deformable DETR, our EEQR-enhanced Deformable-DETR variant, achieves an impressive AP of 43.1, outperforming the baseline Deformable-DETR (39.4 AP) trained for 50 epochs. We also compare with SQR-DETR which provides query recollection. Notably, under similar settings, we outperform SQR-DETR, with the simple addition of our router (+3.2 AP). The results demonstrate the effectiveness of our proposed approach in boosting the performance of DETR-based object detectors.

Query Routing. Table 2 presents the average number of queries passing through the layers. Additionally, Fig. 4 shows the query distributions within the proposed query routing module. For EEQR_{F} and EEQR_{CA}, skewed Gaussian distributions are observed: EEQR_{F} tends to route all update queries through the entire layer, whereas EEQR_{CA} tends to bypass the layer for queries not requiring updates. In contrast, EEQR_{SA} follows a more normal distribution, highlighting the importance of information sharing through the SA layer. As indicated in Table 3, this results in a quicker model despite the higher MACs.

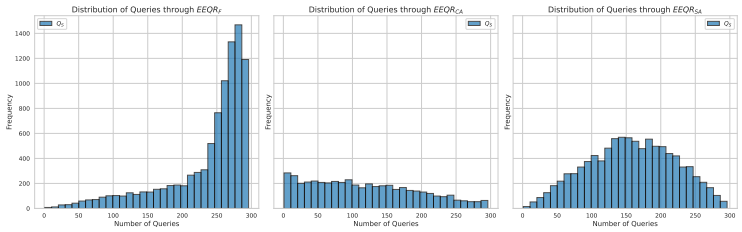


Fig. 4: Distribution of queries passing through each layer Q_p. Bypassed queries are omitted for clarity due to symmetry.

Number of Parameters. Table 2 also presents the number of parameters. While we do not list the parameter counts for all models in the table, we specify those for the latest DAB-DETR, as our proposed method consistently introduces the same number of parameters across various DETR variants. As shown in Equation 5, only our Router layer incorporates learnable parameters, resulting in a slight overall increase in the parameter count.

Layers	Avg # Q_s	N_{Param}
Baseline	300	45M
EEQR_{SA}	63	
EEQR_{CA}	183	46M
EEQR_{F}	142	

Table 2: Average number of queries bypassing layers for each decoder scheme. The number of parameter is rounded up.

Layers	MAC	Processing Time
Baseline	306.2G	0.14s
EEQR	403.4G	0.12s

Table 3: Despite higher MAC and more parameters than the baseline, our model achieves faster inference by skipping unnecessary computations through query routing on COCO 2017.

5.3 Visual Results

We quantitatively evaluate our model on the COCO 2017 validation dataset (Fig. 5), showing that it consistently outperforms the baseline.

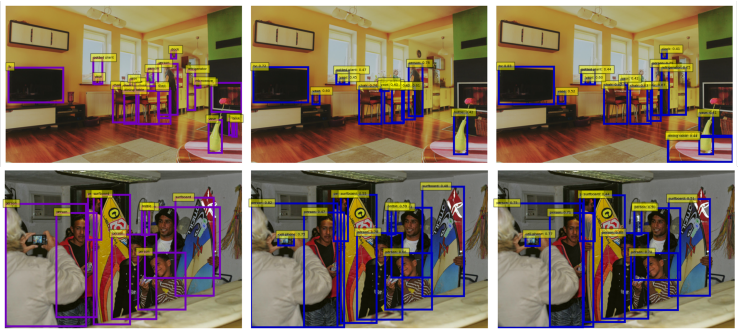


Fig. 5: Detection example on the COCO 2017 dataset. From left to right: ground truth labels, baseline Deformable DETR, and QR-Deformable-DETR.

5.4 Ablation

In this section, we perform ablation studies on Deformable-DETR using a 12-epoch training schedule. We train separate models that exclusively include the proposed layers and evaluate their performance against the baseline model.

a. How to choose the router scheme? We evaluate the importance of the routing scheme employed by training our model using only the **EEQR**$_{\mathtt{SA}}$, **EEQR**$_{\mathtt{CA}}$, **EEQR**$_{\mathtt{F}}$, and a combination of these decoder layers (for simplicity, we denote the combination strategy as **EEQR**, corresponding to the alternance (**EEQR**$_{\mathtt{F}}$ $\rightarrow$ **EEQR**$_{\mathtt{SA}}$ $\rightarrow$ **EEQR**$_{\mathtt{CA}}$), as shown in Tables 3, 4). We compare the performance of these variants against the baseline in Table 4. Our analysis focuses on the contribution of each layer, as discussed earlier, emphasizing how each layer integrates information into object queries differently. Thus, maintaining diversity among EEQR layers is crucial for achieving optimal results.
b. Impact of the Scattering Strategy. We present the impact of two strategies for gathering the output queries from the transformer decoder in Table. 5: 1) *concatenating* all queries after passing through the decoder, and 2) *scattering* the queries back to their initial positions before entering the next decoder stack. The latter strategy achieves 20 mAP higher performance, suggesting that the relative positional information among queries is crucial. We hypothesize that due to the positional embeddings added to each object query, their relative positions are encoded, and simply concatenating the queries breaks this positional encoding, requiring the transformer decoder to perform additional alignment.

Router Scheme	AP	$AP_{0.50}$	$AP_{0.75}$	AP_s	AP_m	AP_l
Baseline	38.3	58.0	41.3	23.6	41.7	51.4
$EEQR_{SA}$	40.6	58.7	44.1	25.3	43.8	53.1
$EEQR_{CA}$	32.5	50.3	35.5	18.5	35.0	43.7
$EEQR_{F}$	23.1	36.5	24.8	15.0	25.6	28.5
EEQR	43.1	62.2	46.4	26.5	46.4	56.9

Table 4: Comparison of object detection performance using different routing schemes for object queries in the EEQR approach. The combination EEQR scheme, achieves the highest AP of 43.1, compared to the baseline. $EEQR_{SA}$ also outperforms the baseline, whereas $EEQR_{CA}$ and $EEQR_{F}$ exhibit lower performance.

c. Impact of Processing All Queries through the FFN. We explore two approaches for the FFN layer: processing either the full query set or only the queries modified by the SA and CA modules. We observe superior performance by processing only the modified queries through the FFN layer. This can be attributed to the different levels of information encoded in the object queries.

Strategy	mAP
Concatenation	19.1
Scattering	43.1

Table 5: We analyze the impact of different strategies for regrouping queries after the routing operation, comparing simple concatenation with a scattering approach. As expected, the scattering strategy proves to be the superior choice.

6 Conclusion

We introduce QR-DETR, a novel query routing approach for transformer-based object detection featuring an Entry-Exit Query Routing (EEQR) module. The EEQR learns to selectively guide object queries through the transformer decoder layers based on a routing policy. This allows the model to prioritize computation on the most informative queries while bypassing less relevant ones early, thereby enhancing computational efficiency. Extensive experiments on the COCO benchmark demonstrate that QR-DETR consistently surpasses baseline DETR models in terms of both accuracy and computational efficiency. The scattering strategy, which preserves spatial relationships among routed queries, proves crucial, and ablation studies confirm the effectiveness of the EEQR module, particularly its variants.

Acknowledgments. This work was performed using HPC resources from GENCI-IDRIS (Grant 2022-[101934])

References

1. Cai, Z., Liu, S., Wang, G., Ge, Z., Zhang, X., Huang, D.: Align-detr: Improving detr with simple iou-aware bce loss. arXiv preprint arXiv:2304.07527 (2023) 11
2. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European conference on computer vision. pp. 213–229. Springer (2020) 1, 3, 11
3. Chen, F., Zhang, H., Hu, K., Huang, Y.k., Zhu, C., Savvides, M.: Enhanced training of query-based object detection via selective query recollection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 23756–23765 (2023) 2, 4, 10, 11
4. Chen, L., Yang, T., Zhang, X., Zhang, W., Sun, J.: Points as queries: Weakly semi-supervised object detection by points. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8823–8832 (2021) 3
5. Chen, Q., Chen, X., Wang, J., Zhang, S., Yao, K., Feng, H., Han, J., Ding, E., Zeng, G., Wang, J.: Group detr: Fast detr training with group-wise one-to-many assignment. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6633–6642 (2023) 3
6. Chen, Q., Wang, J., Han, C., Zhang, S., Li, Z., Chen, X., Chen, J., Wang, X., Han, S., Zhang, G., et al.: Group detr v2: Strong object detector with encoder-decoder pretraining. arXiv preprint arXiv:2211.03594 (2022) 3
7. Csordás, R., Irie, K., Schmidhuber, J., Potts, C., Manning, C.D.: Moeut: Mixture-of-experts universal transformers. arXiv preprint arXiv:2405.16039 (2024) 4
8. Csordás, R., Piękos, P., Irie, K.: Switchhead: Accelerating transformers with mixture-of-experts attention. arXiv preprint arXiv:2312.07987 (2023) 4
9. Dai, X., Chen, Y., Yang, J., Zhang, P., Yuan, L., Zhang, L.: Dynamic DETR: End-to-End Object Detection with Dynamic Attention. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 2968–2977. IEEE, Montreal, QC, Canada (Oct 2021). https://doi.org/10.1109/ICCV48922.2021.00298, https://ieeexplore.ieee.org/document/9709981/ 3
10. Fang, R., Gao, P., Zhou, A., Cai, Y., Liu, S., Dai, J., Li, H.: Feataug-detr: Enriching one-to-many matching for detrs with feature augmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence (2024) 3
11. Fedus, W., Zoph, B., Shazeer, N.: Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. Journal of Machine Learning Research **23**(120), 1–39 (2022) 4
12. Gao, P., Zheng, M., Wang, X., Dai, J., Li, H.: Fast convergence of detr with spatially modulated co-attention. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 3621–3630 (2021) 3
13. Gao, Z., Wang, L., Han, B., Guo, S.: Adamixer: A fast-converging query-based object detector. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5364–5373 (2022) 11
14. Hou, X., Liu, M., Zhang, S., Wei, P., Chen, B.: Salience detr: Enhancing detection transformer with hierarchical salience filtering refinement. In: CVPR (2024) 1, 4
15. Hu, Z., Sun, Y., Wang, J., Yang, Y.: Dac-detr: Divide the attention layers and conquer. Advances in Neural Information Processing Systems **36** (2024) 2, 4, 5, 10
16. Hwang, C., Cui, W., Xiong, Y., Yang, Z., Liu, Z., Hu, H., Wang, Z., Salas, R., Jose, J., Ram, P., et al.: Tutel: Adaptive mixture-of-experts at scale. Proceedings of Machine Learning and Systems **5** (2023) 4

17. Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E.: Adaptive mixtures of local experts. Neural computation **3**(1), 79–87 (1991) 4
18. Jain, Y., Behl, H., Kira, Z., Vineet, V.: Damex: Dataset-aware mixture-of-experts for visual understanding of mixture-of-datasets. Advances in Neural Information Processing Systems **36** (2024) 4
19. Jia, D., Yuan, Y., He, H., Wu, X., Yu, H., Lin, W., Sun, L., Zhang, C., Hu, H.: Detrs with hybrid matching. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19702–19712 (2023) 3
20. Kong, C., Luo, A., Xia, S., Yu, Y., Li, H., Kot, A.C.: Moe-ffd: Mixture of experts for generalized and parameter-efficient face forgery detection. arXiv preprint arXiv:2404.08452 (2024) 4
21. Kouris, A., Venieris, S.I., Laskaridis, S., Lane, N.: Multi-exit semantic segmentation networks. In: European Conference on Computer Vision. pp. 330–349. Springer (2022) 9
22. Li, F., Zhang, H., Liu, S., Guo, J., Ni, L.M., Zhang, L.: Dn-detr: Accelerate detr training by introducing query denoising. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13619–13627 (2022) 3
23. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. pp. 740–755. Springer (2014) 10
24. Lin, Y., Yuan, Y., Zhang, Z., Li, C., Zheng, N., Hu, H.: Detr does not need multi-scale or locality design. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6545–6554 (2023) 3
25. Liu, S., Li, F., Zhang, H., Yang, X., Qi, X., Su, H., Zhu, J., Zhang, L.: DAB-DETR: Dynamic Anchor Boxes are Better Queries for DETR. In: ICLR (2022), https://openreview.net/forum?id=oMI9PjOb9Jl 3, 6, 10, 11
26. Liu, Y., Zhang, Y., Wang, Y., Zhang, Y., Tian, J., Shi, Z., Fan, J., He, Z.: Sap-detr: bridging the gap between salient points and queries-based transformer detector for fast model convergency. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15539–15547 (2023) 3
27. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: International Conference on Learning Representations (2018) 10
28. Lou, Y., Xue, F., Zheng, Z., You, Y.: Cross-token modeling with conditional computation. arXiv preprint arXiv:2109.02008 (2021) 4
29. Ma, J., Huang, P.Y., Xie, S., Li, S.W., Zettlemoyer, L., Chang, S.F., Yih, W.T., Xu, H.: Mode: Clip data experts via clustering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 26354–26363 (2024) 4
30. Meng, D., Chen, X., Fan, Z., Zeng, G., Li, H., Yuan, Y., Sun, L., Wang, J.: Conditional DETR for Fast Training Convergence. In: ICCV (2021) 3, 10, 11
31. Nie, X., Miao, X., Cao, S., Ma, L., Liu, Q., Xue, J., Miao, Y., Liu, Y., Yang, Z., Cui, B.: Evomoe: An evolutional mixture-of-experts training framework via dense-to-sparse gate. arXiv preprint arXiv:2112.14397 (2021) 4
32. Oksuz, K., Kuzucu, S., Joy, T., Dokania, P.K.: Mocae: Mixture of calibrated experts significantly improves object detection. arXiv preprint arXiv:2309.14976 (2023) 4
33. Pu, Y., Liang, W., Hao, Y., Yuan, Y., Yang, Y., Zhang, C., Hu, H., Huang, G.: Rank-detr for high quality object detection. Advances in Neural Information Processing Systems **36** (2024) 3

34. florence regol, Chataoui, J., Coates, M.: Jointly-learned exit and inference for a dynamic neural network. In: The Twelfth International Conference on Learning Representations (2024), https://openreview.net/forum?id=jX2DT7qDam 9
35. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems **28** (2015) 11
36. Roh, B., Shin, J., Shin, W., Kim, S.: Sparse DETR: Efficient End-to-End Object Detection with Learnable Sparsity. Tech. Rep. arXiv:2111.14330, arXiv (Mar 2022). https://doi.org/10.48550/arXiv.2111.14330, http://arxiv.org/abs/2111.14330 4
37. Ruiz, C.R., Puigcerver, J., Mustafa, B., Neumann, M., Jenatton, R., Pinto, A.S., Keysers, D., Houlsby, N.: Scaling vision with sparse mixture of experts. In: Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems (2021), https://openreview.net/forum?id=FrIDgjDOH1u 4
38. Senthivel, T., Vu, N.S.: Subgroups for detection transformer. In: 2024 IEEE International Conference on Image Processing (ICIP). pp. 2194–2200 (2024). https://doi.org/10.1109/ICIP51287.2024.10648285 3
39. Senthivel, T., Vu, N.S., Borzic, B.: Detection Transformer with Diversified Object Queries. In: 2023 IEEE International Conference on Image Processing (ICIP). pp. 2515–2519. IEEE (2023) 1, 4, 5, 7
40. Shen, T., Ott, M., Auli, M., Ranzato, M.: Mixture models for diverse machine translation: Tricks of the trade. In: International conference on machine learning. pp. 5719–5728. PMLR (2019) 4
41. Teng, Y., Liu, H., Guo, S., Wang, L.: Stageinteractor: Query-based object detector with cross-stage interaction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6577–6588 (2023) 11
42. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems. pp. 5998–6008 (2017) 3, 4
43. Wang, T., Yuan, L., Chen, Y., Feng, J., Yan, S.: PnP-DETR: Towards Efficient Visual Analysis with Transformers. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 4641–4650. IEEE, Montreal, QC, Canada (Oct 2021). https://doi.org/10.1109/ICCV48922.2021.00462, https://ieeexplore.ieee.org/document/9710805/ 1
44. Yang, Y., Jiang, P.T., Hou, Q., Zhang, H., Chen, J., Li, B.: Multi-task dense prediction via mixture of low-rank experts. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 27927–27937 (2024) 4
45. Yao, Z., Ai, J., Li, B., Zhang, C.: Efficient detr: improving end-to-end object detector with dense prior. arXiv preprint arXiv:2104.01318 (2021) 3, 10
46. Zeng, W., Jin, S., Liu, W., Qian, C., Luo, P., Ouyang, W., Wang, X.: Not All Tokens Are Equal: Human-Centric Visual Analysis via Token Clustering Transformer. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (2022) 2
47. Zhang, G., Luo, Z., Yu, Y., Cui, K., Lu, S.: Accelerating DETR Convergence via Semantic-Aligned Matching. Tech. Rep. arXiv:2203.06883, arXiv (Mar 2022). https://doi.org/10.48550/arXiv.2203.06883, http://arxiv.org/abs/2203.06883 4, 11

48. Zhang, H., Li, F., Liu, S., Zhang, L., Su, H., Zhu, J., Ni, L.M., Shum, H.Y.: DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection. In: ICLR (2022), _eprint: 2203.03605 3
49. Zhang, X., Shen, Y., Huang, Z., Zhou, J., Rong, W., Xiong, Z.: Mixture of attention heads: Selecting attention heads per token. In: Goldberg, Y., Kozareva, Z., Zhang, Y. (eds.) Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing. pp. 4150–4162. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates (Dec 2022). https://doi.org/10.18653/v1/2022.emnlp-main.278, https://aclanthology.org/2022.emnlp-main.278 4
50. Zhao, C., Sun, Y., Wang, W., Chen, Q., Ding, E., Yang, Y., Wang, J.: Ms-detr: Efficient detr training with mixed supervision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17027–17036 (2024) 3
51. Zheng, D., Dong, W., Hu, H., Chen, X., Wang, Y.: Less is more: Focus attention for efficient detr. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6674–6683 (2023) 4, 6
52. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable DETR: Deformable Transformers for End-to-End Object Detection. In: International Conference on Learning Representations (2020) 3, 6, 11
53. Zong, Z., Song, G., Liu, Y.: Detrs with collaborative hybrid assignments training. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 6748–6758 (2023) 3