AugDETR: Improving Multi-scale Learning for Detection Transformer

Jinpeng Dong, Yutong Lin, Chen Li, Sanping Zhou, Nanning Zheng*

National Key Laboratory of Human-Machine Hybrid Augmented Intelligence,
National Engineering Research Center for Visual Information and Applications, and
Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University
{djp1235a, yutonglin, edward82}@stu.xjtu.edu.cn
{spzhou, nnzheng}@xjtu.edu.cn

Abstract. Current end-to-end detectors typically exploit transformers to detect objects and show promising performance. Among them, Deformable DETR is a representative paradigm that effectively exploits multi-scale features. However, small local receptive fields and limited query-encoder interactions weaken multi-scale learning. In this paper, we analyze local feature enhancement and multi-level encoder exploitation for improved multi-scale learning and construct a novel detection transformer detector named Augmented DETR (AugDETR) to realize them. Specifically, AugDETR consists of two components: Hybrid Attention Encoder and Encoder-Mixing Cross-Attention. Hybrid Attention Encoder enlarges the receptive field of the deformable encoder and introduces global context features to enhance feature representation. Encoder-Mixing Cross-Attention adaptively leverages multi-level encoders based on query features for more discriminative object features and faster convergence. By combining AugDETR with DETR-based detectors such as DINO, AlignDETR, DDQ, our models achieve performance improvements of 1.2, 1.1, and 1.0 AP in the COCO under the ResNet-50-4scale and 12 epochs setting, respectively.

Keywords: Object detection · Detection transformer · Hybrid attention · Multi-level encoder

1 Introduction

Many CNN-based detectors [1,12,20,27,34,37,47] have been proposed to improve detection performance in the past decade. Despite their promising progress, these detectors suffer from many hand-crafted components, such as non-maximum suppression and anchor design. In contrast, CNN-based paradigm is broken by DETR [4], a novel transformer detector. It gets rid of hand-crafted components and implements an end-to-end paradigm. Despite its novel paradigm, DETR suffers from some limitations, including slow convergence and inferior performance. To address these limitations, many methods [2, 6, 14, 24, 25, 29, 42, 50]

* Corresponding author.

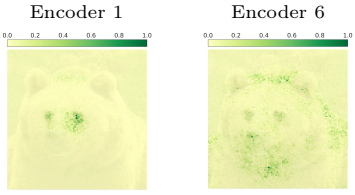
have been proposed from different perspectives, such as introducing spatial priors, hybrid matching training, designing efficient attention modules, assigning sample weights, etc. With these optimizations, DINO [42] achieved a new record on the COCO benchmark [22].

Although DETR-based detectors have achieved promising performance, multi-scale object detection has not received sufficient attention so far, which may limit the performance of detectors. Scale variation of objects remains one of the crucial challenges in object detection. Objects at different scales require inconsistent semantic features, e.g., large objects require features with larger receptive fields. To the best of our knowledge, Deformable-DETR [50] has made some effort to remedy this issue. Deformable-DETR uses multi-scale backbone features to improve multi-scale learning by deformable attention. In addition to the use of multi-scale backbone features, we believe that the multi-scale learning of Deformable-DETR has not been fully exploited. As shown in Table 1, although Deformable DETR is 2.5 AP higher than the DETR in overall performance, the performance on large objects is still 1.5 AP lower than the DETR. We present two techniques that can further improve multi-scale learning as follows:

Table 1: Performance comparison between DETR and Deformable DETR.

Method	Epochs	AP	AP_S	AP_M	AP_L
DETR [4]	500	42.0	20.5	45.8	61.1
Deformable DETR [50]	50	44.5	27.1	47.6	59.6

Fig. 1: Comparison of receptive fields of different encoders.



Local feature enhancement. Deformable-DETR [50] replaces the original self-attention (global dense attention) with deformable attention (sparse attention) to reduce the complexity and slow convergence of dense attention. Deformable attention is implemented by a fixed number of deformable sampling points. Limited by local sampling points, deformable attention is difficult to achieve a larger receptive field compared to self-attention. This is not conducive to the detection of large objects. Further stacking of deformable attention still improves performance, as analyzed in the ablations. In addition, deformable attention loses the ability to perceive global context, while multi-scale object detection in some complex scenes may rely on global context information. Increasing the sampling points degrades the performance of the deformable attention, as analyzed in the ablations. Therefore, we introduce global contextual information into the deformable encoder to enhance multi-scale learning.

Multi-level encoder exploitation. DETR-based detectors typically stack six encoder layers and six decoder layers. Encoder layers enhance the backbone features, and decoder layers extract features from last encoder layer to decode

queries. Previous methods ignore the use of multi-level encoder features. More discriminative object features can be obtained by fully exploiting multi-level encoder features with different receptive fields and feature preferences. We refer to RepLKNet [7] for analysis of receptive field. As shown in Figure 1, the comparison of encoder 1 and 6 in DINO [42] shows that different encoders have varying feature preferences and receptive fields. In addition, the fact that multi-level encoders are all involved in object feature generation can speed up convergence because the supervisory signals are available to multiple levels of encoders, rather than being available only to the last level encoder. Utilizing multi-level encoder features also can introduce redundancy to mitigate the risk of overfitting. Although multi-level encoder features is beneficial for object detection, using multi-level encoder features fixed for all objects is not conducive to multi-scale learning. Since objects at different scales have different feature preferences, adaptive use of multi-level encoder features by learning weights based on query features will facilitate multi-scale learning for DETR-based detectors.

In this paper, we propose AugDETR, a simple yet effective DETR-based detector that uses two different components to improve multi-scale learning. First, Hybrid Attention Encoder is proposed to augment local features in the deformable encoder by partially combining dense attention with deformable attention. Hybrid Attention Encoder can enlarge the receptive field and introduce global context, which can improve detection performance on large objects. Second, Encoder-Mixing Cross-Attention is introduced to better exploit features from different encoder layers by weighted aggregation. Objects at different scales can learn different aggregation weights based on object features to facilitate multi-scale learning. With this design, Encoder-Mixing Cross-Attention can generate more discriminative object features for subsequent classification and localization tasks. We validate the effectiveness of AugDETR on the COCO benchmark. Our method achieves 50.2 AP (+1.2 AP), 51.3 AP (+1.1 AP), and 51.8 AP (+1.0 AP) based on DINO [42], AlignDETR [2], and DDQ [45] with ResNet-50-4scale under 12 epochs settings, respectively.

We summarize our contributions as follows:

- We analyze local feature enhancement and multi-level encoder exploitation techniques for improved multi-scale learning.
- A new detection transformer detector named AugDETR is proposed to achieve the techniques with Hybrid Attention Encoder and Encoder-Mixing Cross-Attention.
- We validate AugDETR equipped with various DETR-based detectors and backbones on the COCO benchmark, and it consistently yields significant performance improvements over DETR-based detectors.

2 Related Work

2.1 Object Detection

Early deep-learning detectors can be roughly classified into anchor-based and anchor-free methods. Anchor-based methods use predefined anchors to locate

objects, including Faster R-CNN [34], SSD [27], RetinaNet [21], and their variants [1, 13, 31, 44]. Anchor-free methods use points near object centers to locate objects, including CenterNet [47], FCOS [37], and others [9, 16, 17, 48, 49].

Recently, DETR [4] proposes a simple transformer pipeline for object detection, effectively alleviating the need for heuristic components. Compared to CNN-based work, DETR is rather novel but still suffers from some limitations, including slow convergence and inferior performance. Many follow-up works focus on alleviating the limitations of DETR from different perspectives. Conditional DETR [29] proposes to decouple the context embedding and position embedding, which helps DETR speed up convergence. Anchor DETR [38] uses the anchor points as the initial queries for fast convergence. DAB DETR [24] uses 4D anchor boxes that are dynamically updated with the decoder layer to represent queries. DN DETR [18] proposes a denoising training mechanism to speed up convergence. DINO [42] further extends DN DETR with contrastive denoising training, mixed query, and look-forward twice techniques. Group DETR [5], HDETR [14], and CO-DETR [51] propose various one-to-one matching and one-to-many matching combination methods to stabilize the matching process of DETR. Align DETR [2], Stable DINO [25], and Rank DETR [32] design different schemes to weight the samples by the IoU-aware classification loss. Plain DETR [23] proposes a box-to-pixel relative position bias and a masked image modeling [39] pre-training scheme to improve the DETR detector. Unlike these methods that design spatial priors, hybrid matching, sample weight assignments, etc., our approach focuses on improving the multi-scale learning of DETR detectors. In addition, we conducted experiments using some SOTA methods as the baseline models. Our techniques are complementary to theirs.

2.2 Multi-Scale Learning for Object Detection

In the early era, image pyramid is a common solution to improve multi-scale learning. Instead of image pyramid, some methods utilize multi-scale features to mitigate scale variation. A representative work among these methods is the feature pyramid network (FPN [20]), which proposes a top-down pathway and lateral connections to construct the feature pyramid. Inspired by FPN, multi-scale feature extraction has been widely studied, such as BiFPN [36], PAFPN [26], NAS-FPN [11], AugFPN [12], and other variants [8,15,33]. Recently, some works have attempted to design multi-scale feature extraction modules in DETR. Deformable DETR [50] proposes deformable attention to utilize multi-scale backbone features. Dynamic DETR [6] proposes a CNN-based dynamic encoder with scale-aware attention, spatial-aware attention, and task-aware attention. SMCA [10] generates the Gaussian-like weight map as the spatial prior and the scale weights as the scale prior to modulated multi-scale features. IMFA [40] designs a new scheme to adaptively sample sparse multi-scale features. Different from them, our approach enhances multi-scale learning by introducing global attention to expand the receptive field and integrate global context. To use global attention effectively, we propose hybrid scale and hybrid layer strategies to construct the hybrid attention encoder. In addition, we adaptively use multi-level

encoder features based on query features to improve multi-scale learning, which is also distinctly different from previous methods.

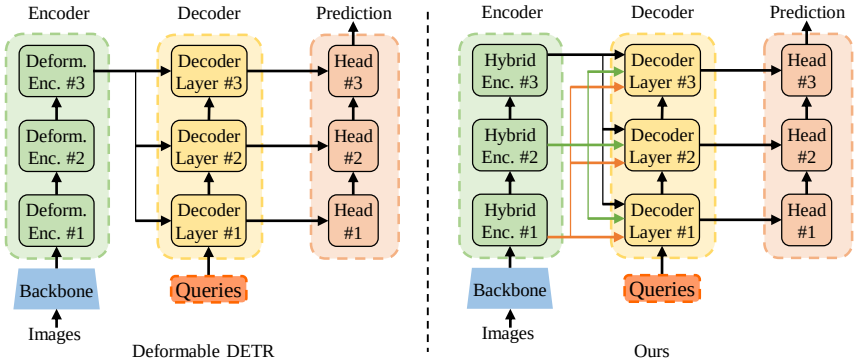


Fig. 2: Left: The pipeline of conventional DETR-based detectors. The backbone network and the stacked encoder layers are used to extract and enhance the image features. Then, the object queries utilize the decoder layers to interact with the enhanced image features. Finally, the interacted object queries are fed into the detection head to obtain the predictions. Only the features from the last encoder layer can be used to interact with the object queries. Right: The pipeline of Our AugDETR. Our AugDETR restructures the conventional encoder-decoder pipeline. The object queries can interact with features from all encoder layers. For clarity, only the 3-layer structure is shown.

3 Method

In section 3.1, we first briefly revisit the detection transformer and then introduce the pipeline of AugDETR. In section 3.2, we describe the details of the proposed Hybrid Attention Encoder. In section 3.3, we introduce the Encoder-Mixing Cross-Attention used in the transformer decoder.

3.1 A Revisit of DETR-based Detector

Since our proposed method is constructed on top of the recent DETR-based object detectors, we first briefly review the pipeline of the DETR-based detectors, taking the Deformable DETR [50] as an example.

The pipeline of Deformable DETR is shown in Figure 2 (left). Deformable DETR consists of 3 subcomponents: a backbone, a encoder, and a decoder. Given an input image $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$, a backbone network generates multi-scale features. We denote the multi-scale features as $\{C_2, C_3, C_4, C_5\}$. The spatial resolutions of these features are usually $1/4^2$, $1/8^2$, $1/16^2$, and $1/32^2$ of the given image. Then these features reduced to the same channel by a linear projection are denoted as $\{P_2, P_3, P_4, P_5\}$. Next, a transformer encoder is used to further enhance these

multi-scale features. The transformer encoder is typically stacked with 6 encoder layers. The encoder layer mainly consists of a self-attention and a feed-forward network. For Deformable DETR, multi-scale deformable attention is employed as self-attention to enhance multi-scale features. The encoded features of the different layers in the encoder are denoted as $\{E_1, E_2, E_3, E_4, E_5, E_6\}$. Finally, a transformer decoder is used to produce object predictions from a set of object queries. Typically, only the last level of encoder features are fed into the decoder to interact with the query. The transformer decoder is also typically stacked with 6 decoder layers for better detection performance. The decoder layer mainly consists of a self-attention, a cross-attention, and a feed-forward network. For deformable DETR, multi-scale deformable attention is used as cross-attention to achieve query-feature interaction. As shown in Figure 2 (right), our pipeline changes the fact that the decoder can utilize only the last level of encoder features, but rather all encoder features can be utilized by the decoder. This design lays the foundation for cross-attention to utilize multi-level encoder features.

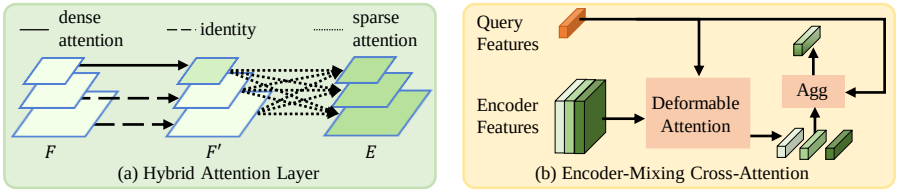


Fig. 3: (a) is the process of hybrid attention in the Hybrid Attention Layer. Dense attention is first applied to only the top features, and then sparse attention is applied to the multi-scale features. (b) is the details of Encoder-Mixing Cross-Attention. Deformable attention extracts multiple object features from multi-level encoder and then the extracted features are aggregated based on the adaptive fusion weights learned from the query features.

3.2 Hybrid Attention Encoder

The encoder layer in the original DETR [4] uses the self-attention as follows:

$$\begin{aligned} &\mathbf{Q} = W_q P_5, \mathbf{K} = W_k P_5, \mathbf{V} = W_v P_5, \\ &\text{SelfAttention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Softmax}(\mathbf{QK})\mathbf{V}, \end{aligned} \tag{1}$$

where P_5 are the last level features of the backbone network; $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ are query, key, and value features, respectively.

The standard self-attention performs dense computation that involves all features. Applying standard self-attention to a large feature map will cause high complexity due to its dense computation, so the multi-scale features of the backbone network are not utilized in the original DETR. In addition, dense compu-

tation will lead to slow convergence. The original DETR required 500 epochs of training to achieve good performance.

To exploit the multi-scale features of the backbone, Deformable DETR [50] proposes deformable attention. The multi-scale deformable self-attention can be simply formulated as:

$$\begin{aligned}
&\mathbf{Q} = \text{Concat}(P_3, P_4, P_5), \mathbf{A} = W_a\mathbf{Q}, \\
&\Delta r = W_p\mathbf{Q}, \mathbf{V} = \text{Samp}(W_v\mathbf{Q}, r + \Delta r), \\
&\text{DeformableAttention}(\mathbf{Q}, \mathbf{V}) = \text{Softmax}(\mathbf{A})\mathbf{V},
\end{aligned} \tag{2}$$

where P_3, P_4, P_5 are multi-scale features of the backbone, $\mathbf{Q}$ are a set of queries, Δr are the offsets of reference points, r are the reference points, $\mathbf{A}$ are the attention weights of sample features, Samp is a function that extracts the features corresponding to the location $(r + \Delta r)$ by bilinear interpolation.

The deformable attention performs sparse computation that involves only a predefined number of surrounding features. Because deformable attention does not produce high complexity when combined with multi-scale features, it can utilize the multi-scale features of the backbone to detect multi-scale objects. In addition, sparse computation will lead to fast convergence. Deformable DETR converges 10 times faster than DETR. However, limited by sparse computation, deformable attention typically does not have a large receptive field and does not use global context information. This is not conducive to detecting large objects and objects in complex scenes.

To address these issues, we propose the Hybrid Attention Encoder that combines standard self-attention and deformable attention in the encoder layer. The hybrid attention layer is shown in Figure 3 (a). Introducing dense attention into the deformable attention encoder can enlarge the receptive field and perceive the global context. This will be more beneficial for multi-scale object detection. However, how to introduce dense attention into deformable attention without increasing too much complexity and slowing convergence is a key challenge. In order not to introduce too much complexity, we propose the hybrid-scale strategy. This strategy means that only the last scale features receive dense attention, not all scale features. With this strategy, our method does not introduce too much computation. Our hybrid attention can be simply formulated as:

$$\begin{aligned}
&\mathbf{Q}' = \mathbf{K}' = \mathbf{V}' = P_5, \\
&P_5' = \text{SelfAttention}(\mathbf{Q}', \mathbf{K}', \mathbf{V}'), \\
&\mathbf{Q} = \text{Concat}(P_3, P_4, P_5'), \mathbf{A} = W_a\mathbf{Q}, \\
&\Delta r = W_p\mathbf{Q}, \mathbf{V} = \text{Samp}(W_v\mathbf{Q}, r + \Delta r), \\
&\text{DeformableAttention}(\mathbf{Q}, \mathbf{V}) = \text{Softmax}(\mathbf{A})\mathbf{V},
\end{aligned} \tag{3}$$

In order not to slow down the convergence, we propose the hybrid-layer strategy. This strategy means that instead of adopting hybrid attention at all encoder layers, it is only adopted at some layers. From the ablation experiments, it can be seen that adopting hybrid attention in all the encoder layers will have no

performance gain due to slow convergence. In our practical use, we introduce hybrid attention only in the first two encoder layers in order to balance complexity and performance. With the above design, we build the Hybrid Attention Encoder that can enlarge the receptive field and utilize the global context.

3.3 Encoder-Mixing Cross-Attention

The cross-attention in the decoder layer of Deformable DETR [50] uses multi-scale deformable attention to achieve query-feature interaction. The multi-scale deformable cross-attention can be simply formulated as:

$$\begin{aligned} &\mathbf{A} = W_a\mathbf{Q}, \Delta r = W_p\mathbf{Q}, \mathbf{V} = \text{Samp}(W_v E_6, r + \Delta r), \\ &\text{DeformableAttention}(\mathbf{Q}, \mathbf{V}) = \text{Softmax}(\mathbf{A})\mathbf{V}, \end{aligned} \tag{4}$$

where $\mathbf{Q}$ are a set of decoder queries, $\mathbf{A}$ is the attention weights, Δr is the offsets of reference points, r is the reference points, E_6 is the multi-scale features of last encoder layer, Samp is the sample function.

DETR-based detectors typically stack six encoder layers to enhance the backbone features. However, only features of the last encoder layer are utilized in cross-attention. Although features of the last encoder layer have larger receptive fields and stronger semantics, it is not optimal for multi-scale objects due to the fact that objects have different favors for features. Features from different encoder layers have different receptive fields, and if they are fully utilized they will be more beneficial for multi-scale object detection.

To address these issues, cross-attention interacting with multi-level encoder can obtain more discriminative features. However, using the same multi-level encoder for all objects is not conducive to multi-scale learning since multi-scale objects have different feature preferences. We propose Encoder-Mixing Cross-Attention (EMCA) to adaptively exploit features of different encoders based on query features. The EMCA is shown in Figure 3 (b). Specifically, we first extract features from each encoder layer by using multi-scale deformable attention. Then, we generate fusion weights for extracted features from each encoder based on the query features. Finally, we adaptively aggregate features from each encoder according to the fusion weights. The EMCA can be simply formulated as:

$$\begin{aligned} &\mathbf{A} = W_a\mathbf{Q}, \Delta r = W_p\mathbf{Q}, \mathbf{w}_e = \sigma(W_e\mathbf{Q}) \\ &\mathbf{V}_l = \text{Samp}(W_{vl}E_l, r + \Delta r), \\ &\text{EMCA}(\mathbf{Q}, \mathbf{E}) = \sum_{l=1}^{L} w_e^l \cdot \text{Softmax}(\mathbf{A})\mathbf{V}_l, \end{aligned} \tag{5}$$

where σ means the Sigmoid, E_l is the features of the l level encoder, $\mathbf{E}$ is a set of features from each level encoders, w_e^l is the weights of the l level encoder, $\mathbf{w}_e$ is a set of weights for all encoders. L is the number of encoder layers.

We use the EMCA to replace all the cross-attention in the decoder layer. With this design, each object can adaptively utilize multi-level encoder features to obtain more discriminative features for improved multi-scale learning.

Table 2: Comparison with the state-of-the-art DETR variants on COCO `val2017` with the ResNet-50-4scale backbone. * indicates the result is from our implemented model.

Model	Backbone	#epochs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Conditional-DETR [29]	R50	108	43.0	64.0	45.7	22.7	46.7	61.5
SAM-DETR [41]	R50	50	39.8	61.8	41.6	20.5	43.4	59.6
Anchor-DETR [38]	R50	50	42.1	63.1	44.9	22.3	46.2	60.0
Dynamic-DETR [6]	R50	12	42.9	61.0	46.3	24.6	44.9	54.4
SMCA-DETR [10]	R50	108	45.6	65.5	49.1	25.9	49.3	62.6
CF-DETR [3]	R50	36	47.8	66.5	52.4	31.2	50.6	62.8
Sparse-DETR [35]	R50	50	46.3	66.0	50.1	29.0	49.5	60.8
BoxeR-2D [30]	R50	50	50.0	67.9	54.7	30.9	52.8	62.6
Deformable-DETR [50]	R50	50	46.9	65.6	51.0	29.6	50.1	61.6
DAB-Deformable-DETR [24]	R50	50	46.8	66.0	50.4	29.1	49.8	62.3
DN-Deformable-DETR [18]	R50	50	48.6	67.4	52.7	31.0	52.0	63.7
$\mathcal{H}$-DETR [14]	R50	12	48.7	66.4	52.9	31.2	51.5	63.5
Co-DETR [51]	R50	12	49.5	67.6	54.3	32.4	52.7	63.7
DINO [42]	R50	12	49.0	66.6	53.5	32.0	52.3	63.0
DINO [42]	R50	24	50.4	68.3	54.8	33.3	53.7	64.8
DDQ* [45]	R50	12	50.8	67.9	56.1	34.6	54.3	65.5
Focus DETR [46]	R50	36	50.4	68.5	55.0	34.0	53.5	64.4
AlignDETR [2]	R50	12	50.2	67.8	54.4	32.9	53.3	65.0
Stable-DINO [25]	R50	12	50.4	67.4	55.0	32.9	54.0	65.5
Rank-DETR [32]	R50	12	50.2	67.7	55.0	34.1	53.6	64.0
DINO+Ours	R50	12	50.2 (+1.2)	67.8	55.0	32.3	53.2	64.7
DINO+Ours	R50	24	51.3 (+0.9)	69.0	56.2	33.5	54.7	66.1
AlignDETR+Ours	R50	12	51.3 (+1.1)	68.8	55.8	33.3	54.8	66.6
DDQ+Ours	R50	12	51.8 (+1.0)	69.2	56.9	34.8	55.0	66.7

4 Experiments

Dataset and Metrics. We conduct all experiments on the challenging COCO 2017 detection dataset. It is split into training set, validation set, and test-dev set with 115K, 5K, and 20K images respectively. We train models on the training set and evaluate performance on the validation set and test-dev set. All evaluated results follow the COCO-style Average Precision (AP) metrics.

Implementation Details. We use DINO [42], Align DETR [2], and DDQ [45] as our baseline methods. All ablation studies are conducted on DINO-4scale with ResNet-50 for 12 epochs. We employ AdamW with 1×10^{-4} weight decay as the optimizer to train our models. The batch size is set as 16 for all experiments. The initial learning rate is set as 1×10^{-4} and it decreases by multiplying 0.1 after the 11th epoch for the 12 epochs setting. We use a random seed 0 in all our experiments. All other hyper-parameters follow the default from the codebase.

4.1 Main Results

In this section, we evaluate AugDETR on the COCO val2017 set and compare it with the state-of-the-art DETR variants. All results with ResNet50 are sum-

Table 3: Comparison with the DETR variants on `val2017` set with the Swin-T.

Method	Backbone	#epochs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
DINO [42]	Swin-T	12	51.3	69.0	56.0	34.5	54.4	66.0
Ours	Swin-T	12	52.3 (+1.0)	70.2	57.0	36.8	55.3	67.2
AlignDETR [2]	Swin-T	12	52.5	70.1	57.2	35.9	55.9	68.4
Ours	Swin-T	12	53.5 (+1.0)	71.3	58.1	36.9	56.6	69.1

marized in Table 2. Combining our components with DINO [42], AugDETR achieves 50.2 AP, which is 1.2 AP higher than the DINO baseline with the same setting. Recent work on replacing the focal loss in DINO with the IoU-aware focal loss [19, 21, 43] has shown good performance. We chose a typical work AlignDETR [2] from these studies as baseline to validate the generality of our approach. Combining our components with AlignDETR, AugDETR achieves 51.3 AP, which is 1.1 AP higher than the AlignDETR baseline. Besides, we conduct experiments in the latest SOTA model DDQ [45]. Combining the our components with DDQ, AugDETR achieves 51.8 AP, which is 1.0 AP higher than the DDQ baseline. To demonstrate the effectiveness of our method under longer training, we conducted experiments under the 2× schedule. Our method achieves 51.3 AP, which is 0.9 AP higher than the baseline DINO. Besides, AugDETR can consistently achieve performance improvements even with more advanced backbone networks. As shown in Table 3, when using Swin-T [28] as the backbone, our method achieves performance improvements of 1.0 AP and 1.0 AP over DINO and AlignDETR, respectively. These results demonstrate the effectiveness and generalizability of our method.

Table 4: Ablations on each component of AugDETR. "HAE" and "EMCA" represent Hybrid Attention Encoder, and Encoder-Mixing Cross-Attention, respectively.

HAE	EMCA	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
		49.0	66.6	53.5	32.0	52.3	63.0
✓		49.6	67.0	54.2	31.8	52.9	63.8
	✓	49.8	67.4	54.6	33.0	52.8	64.7
✓	✓	50.2	67.8	55.0	32.3	53.2	64.7

4.2 Ablations

Ablation studies on the effectiveness of each component. To analyze the contributions of individual components in AugDETR, Hybrid Attention Encoder (HAE), and Encoder-Mixing Cross-Attention (EMCA) are gradually introduced to the baseline. As shown in Table 4, HAE brings 0.6 AP improvement to the baseline model. This benefits from the HAE enlarges the receptive field and introduces the global context features to improve semantic representation. The

improvements of AP_M (+0.6 AP) and AP_L (+0.8 AP) contribute to the final performance improvement. These results are good demonstrations of our motivation and design. The EMCA improves the detection performance from 49.0 AP to 49.8 AP. It can be seen that the results on small, medium, and large objects are all improved, which means that object queries adaptively interacting with multi-level encoder features by cross-attention can extract more discriminative object features for improved multi-scale learning. When both components are included in the baseline model, it achieves 50.2 AP with 1.2 AP performance improvement. The results show that the two components are complementary and address different issues in DETR [4].

Table 5: Comparisons of ours HAE with Deformable Encoder variants.

Method	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	Flops
Deform. Enc. (4 Points)	49.0	66.6	53.5	32.0	52.3	63.0	245G
Deform. Enc. (6 Points)	48.4	65.9	52.7	31.0	51.6	62.9	250G
Deform. Enc. (8 Points)	48.5	66.1	52.8	30.5	51.8	62.8	255G
HAE (Ours)	49.6	67.0	54.2	31.8	52.9	63.8	248G

Comparisons of HAE with Deformable Encoder variants. To make a fair comparison with the deformable encoder [50], we conduct experiments on increasing sampling points in the deformable encoder. The results are shown in Table 5. The performance of baseline with 4 sampling points is 49.0 AP. Performance decreases as sample points in the deformable encoder increases to 6 and 8, which are 48.4 AP and 48.5 AP, respectively. Even increasing the number of sample points in the deformable encoder does not improve performance for large and medium objects. This may be because deformable attention determines sampling weights based on query alone, leading to inaccurate weights as the number of points increases. However, our HAE can achieve performance gains (+0.6 AP) with less computational consumption (248G vs 250G). We think the possible reason is that the weight generation in standard self-attention, which calculates the similarity by query and key, is more suitable for long-distance sampling. These results support the motivation and design of our hybrid attention encoder.

The number of Hybrid Attention Layers. Introducing too much self-attention may lead to slow convergence. We analyze the effects of the number of hybrid attention layers. The results are shown in Table 6. When only a hybrid attention layer is introduced, the model achieves 49.3 AP with 0.3 AP improvement. The improvement is mainly contributed by large and medium objects. Increasing the number of hybrid attention layers by 2 or 3 results in performance gains of 0.6 and 0.7 AP, respectively. It is still the large and medium objects that contribute the most performance gains. As the number of hybrid attention layers increases, the performance gains begin to decrease. In particular, there is no performance gain when increasing to 6 hybrid attention layers. This may be because the negative effect of slow convergence becomes dominant.

Table 6: Ablations on the number of hybrid attention layers in HAE.

Number	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
–	49.0	66.6	53.5	32.0	52.3	63.0
1	49.3	66.6	53.7	31.1	52.7	63.7
2	49.6	67.0	54.2	31.8	52.9	63.8
3	49.7	67.2	54.3	32.3	52.8	64.0
4	49.3	67.0	53.8	31.9	52.7	63.5
5	49.3	66.8	54.0	31.4	52.8	63.1
6	48.8	66.3	53.4	30.7	52.2	63.1

Table 7: Comparisons of ours EMCA with other decoder variants. * indicates that these models scale to the similar complexity as EMCA.

Method	AP	AP_{50}	AP_{75}	AP_{2th}	AP_{6th}	AP_{10th}	Flops
DINO [42]	49.0	66.6	53.5	34.2	43.2	45.7	245G
+Iterative Enc [40]	47.7	65.2	52.1	33.4	42.0	44.6	245G
+Iterative Enc*	47.4	65.1	51.6	33.2	42.0	44.1	284G
+Memory Fusion [25]	49.3	67.0	53.7	35.7	43.7	46.2	253G
+Memory Fusion*	49.3	66.9	53.9	35.7	43.5	46.2	290G
+EMCA (Ours)	49.8	67.4	54.6	36.0	44.5	46.6	283G

Comparisons of EMCA with decoder variants. We compare our EMCA with other decoder variants. As shown in Table 7, using the Iterative Encoder [40] degrades the performance of the baseline. There are 0.3 AP performance gains when the Memory Fusion [25] is introduced, but there is no performance gain from the scaling of the approach. The performance gain of Memory Fusion is 0.5 AP less than our EMCA. These results demonstrate the effectiveness of our method design, which adaptively fuses multi-level encoder features based on query features for multi-scale objects, rather than using the same features for all objects. In addition, to show that our method converges faster, we also show results for the 2nd, 6th, and 10th epochs. It can be seen that our method outperforms the other methods throughout the training process, indicating the faster convergence of our method.

Comparisons of EMCA with DINO variants. For a fair comparison, we compared with more complex baseline models due to the increased computational complexity introduced by EMCA. More complex baseline models are implemented by introducing more encoders and features. The results are shown in Table 8. By increasing the original baseline from 6 encoders to 7 encoders, the model reached 49.3 AP with 268G. When using 8 encoders, the model achieves 49.5 AP with 292G. While performance can be improved by stacking encoders, our EMCA utilizes encoder redundancy information and is a good complement to encoders. When using 10 encoders, the model suffers from overfitting and achieves only 47.8 AP with 339G. However, our EMCA with 10 encoders still has performance gains to reach 50.6 AP. This results show that our EMCA can

Table 8: Comparisons of ours EMCA with DINO variants.

Method	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	Flops
6 encoder (Baseline)	49.0	66.6	53.5	32.0	52.3	63.0	245G
7 encoder	49.3	66.7	53.7	32.0	52.4	63.7	268G
8 encoder	49.5	66.9	54.0	31.9	52.8	63.4	292G
10 encoder	47.8	64.8	52.1	30.6	50.9	61.5	339G
5 scale	49.4	66.9	53.8	32.3	52.5	63.9	721G
6encoder+EMCA (Ours)	49.8	67.4	54.6	33.0	52.8	64.7	283G
8encoder+EMCA (Ours)	50.2	67.8	54.6	32.4	53.3	64.6	346G
10encoder+EMCA (Ours)	50.6	68.1	55.3	33.5	53.9	65.3	409G

mitigate the risk of overfitting. A very large amount of computation (721G) is required when 5 scales of backbone features are used, and then the performance reaches only 49.4 AP. These results show that our EMCA is a good complement to encoders and can mitigate the risk of overfitting.

Quantitative analysis of EMCA. To analyze the weights of EMCA, we count the fusion weights in the last decoder layer from the COCO val2017 set. First, we analyzed the weights of different encoder layers, and the results are shown in Figure 4. It can be seen that our method learns different weights for encoder layers. Specifically, the first encoder layer plays the most important role, the 2nd, 3rd, and 4th encoder layers have decreasing importance, and the 4th, 5th, and 6th encoder layers have similar roles. As shown in Table 9, using the 2nd, 4th, and 6th encoder layers as inputs, our approach drops 0.3 AP and saves 23G of computational overhead. When only the 1st and 6th encoder layers are used, the computation is reduced by 30G while the performance is dropped by 0.2 AP. We call the EMCA with only the 1st and 6th encoder layers as EMCA-Lite. We then analyze the weights of the objects at different scales to illustrate that our method improves multi-scale learning. As shown in Figure 5, objects of different sizes are given different weights, and the larger the scale, the higher the weights. See the Appendix for object scale intervals. These results illustrate the effectiveness of fusion weights based on queries.

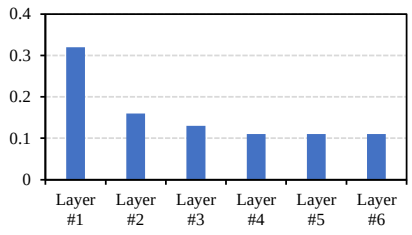


Fig. 4: Statistical fusion weights for different encoders. Layer #6 means the last-level encoder.

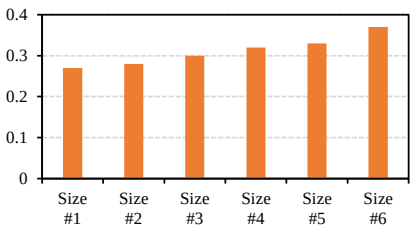


Fig. 5: Statistical fusion weights for different scale objects. Size #1 to size #6 are ordered by scale from small to large.

Table 9: Ablations on the layer of input encoders in EMCA.

Encoder	AP	AP_{50}	AP_{75}	Flops
–	49.0	66.6	53.5	245G
123456	49.8	67.4	54.6	283G
246	49.5	67.2	53.9	260G
26	49.3	67.2	53.8	253G
16	49.6	67.4	54.3	253G

Table 10: Ablations on the fusion methods in EMCA.

Fusion Method	AP	AP_S	AP_M	AP_L
–	49.0	32.0	52.3	63.0
sum	49.3	31.9	52.5	63.9
max	48.6	30.9	51.6	63.2
sigmoid weighted	49.8	33.0	52.8	64.7
softmax weighted	49.5	32.1	52.8	64.5

Fusion Methods in EMCA. The fusion methods we explore include sum fusion, max fusion, and adaptive weighted fusion (sigmoid and softmax). All results are shown in Table 10. By using sum fusion, our method improves the baseline by 0.3 AP. This result shows that multi-level encoder features are richer in semantic information than single-level encoder features. When using max fusion, our method achieves 48.6 AP below the baseline. It can be seen that the performance decreases significantly for small and medium objects, but increases slightly for large objects. This may be because large objects tend to have higher activation so the features of small and medium objects are weakened during max fusion. When using adaptive weighted fusion, our method improves the baseline by 0.8 AP and 0.5 AP, respectively. It can be seen that the performance of small, medium, and large objects is all improved. In particular, the performance on large objects has improved significantly (+1.7 AP) by sigmoid weighted fusion. This may be because the larger the object, the more information is activated at all levels of the encoder, and therefore the more information can be used. In addition, the corresponding query features of large objects are richer so that the learned fusion weights are more accurate. These results suggest that sigmoid weighted fusion is a better fusion method.

5 Conclusion

In this paper, we analyze the limitations of DETR detectors for multi-scale learning and propose local feature enhancement and multi-level encoder exploitation techniques to address the limitations. Based on these, we construct a novel detection transformer detector named Augmented DETR (AugDETR), which consists of two components: Hybrid Attention Encoder and Encoder-Mixing Cross-Attention. HAE enlarges the receptive field and introduces global context to enhance the local feature of the deformable encoder. EMCA uses multi-level encoders to obtain more discriminative features and speed up convergence. Extensive experiments with various DETR-based detectors and backbones validate the effectiveness of our method.

Acknowledgements

This work is supported by National Natural Science Foundation of China (NSFC) under Grant 62088102. This work is partly supported by China Telecom Corporation Ltd. Data&AI Technology Company, Beijing, China and China Telecom - Xi'an Jiaotong University Joint Innovation Institute for Science-Education Integration of Intelligent Cloud Network. We also appreciate Jingwen Fu, He Zhang and Tao Yang for their insightful discussions. We thank all the anonymous reviewers and Area Chairs for their constructive and helpful comments, which have significantly improved the quality of the paper.

References

1. Cai, Z., Vasconcelos, N.: Cascade r-cnn: Delving into high quality object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6154–6162 (2018)
2. 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)
3. Cao, X., Yuan, P., Feng, B., Niu, K.: Cf-detr: Coarse-to-fine transformers for end-to-end object detection. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 185–193 (2022)
4. 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)
5. Chen, Q., Chen, X., Zeng, G., Wang, J.: Group detr: Fast training convergence with decoupled one-to-many label assignment. arXiv preprint arXiv:2207.13085 (2022)
6. Dai, X., Chen, Y., Yang, J., Zhang, P., Yuan, L., Zhang, L.: Dynamic detr: End-to-end object detection with dynamic attention. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2988–2997 (2021)
7. Ding, X., Zhang, X., Han, J., Ding, G.: Scaling up your kernels to 31x31: Revisiting large kernel design in cnns. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11963–11975 (2022)
8. Dong, J., Huang, Y., Zhang, S., Chen, S., Zheng, N.: Construct effective geometry aware feature pyramid network for multi-scale object detection. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 534–541 (2022)
9. Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., Tian, Q.: Centernet: Keypoint triplets for object detection. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 6569–6578 (2019)
10. 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)
11. Ghiasi, G., Lin, T.Y., Le, Q.V.: Nas-fpn: Learning scalable feature pyramid architecture for object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 7036–7045 (2019)
12. Guo, C., Fan, B., Zhang, Q., Xiang, S., Pan, C.: Augfpn: Improving multi-scale feature learning for object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12595–12604 (2020)

13. He, Y., Zhu, C., Wang, J., Savvides, M., Zhang, X.: Bounding box regression with uncertainty for accurate object detection. In: Proceedings of the ieee/cvf conference on computer vision and pattern recognition. pp. 2888–2897 (2019)
14. 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)
15. Kim, S.W., Kook, H.K., Sun, J.Y., Kang, M.C., Ko, S.J.: Parallel feature pyramid network for object detection. In: Proceedings of the European conference on computer vision (ECCV). pp. 234–250 (2018)
16. Kong, T., Sun, F., Liu, H., Jiang, Y., Li, L., Shi, J.: Foveabox: Beyound anchor-based object detection. IEEE Transactions on Image Processing **29**, 7389–7398 (2020)
17. Law, H., Deng, J.: Cornernet: Detecting objects as paired keypoints. In: Proceedings of the European conference on computer vision (ECCV). pp. 734–750 (2018)
18. 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)
19. Li, X., Wang, W., Wu, L., Chen, S., Hu, X., Li, J., Tang, J., Yang, J.: Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. Advances in Neural Information Processing Systems **33**, 21002–21012 (2020)
20. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2117–2125 (2017)
21. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017)
22. 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)
23. 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)
24. 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: International Conference on Learning Representations (2021)
25. Liu, S., Ren, T., Chen, J., Zeng, Z., Zhang, H., Li, F., Li, H., Huang, J., Su, H., Zhu, J., et al.: Detection transformer with stable matching. arXiv preprint arXiv:2304.04742 (2023)
26. Liu, S., Qi, L., Qin, H., Shi, J., Jia, J.: Path aggregation network for instance segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8759–8768 (2018)
27. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14. pp. 21–37. Springer (2016)
28. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 10012–10022 (2021)

29. Meng, D., Chen, X., Fan, Z., Zeng, G., Li, H., Yuan, Y., Sun, L., Wang, J.: Conditional detr for fast training convergence. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3651–3660 (2021)
30. Nguyen, D.K., Ju, J., Booij, O., Oswald, M.R., Snoek, C.G.: Boxer: Box-attention for 2d and 3d transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4773–4782 (2022)
31. Pang, J., Chen, K., Shi, J., Feng, H., Ouyang, W., Lin, D.: Libra r-cnn: Towards balanced learning for object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 821–830 (2019)
32. 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)
33. Qiao, S., Chen, L.C., Yuille, A.: Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10213–10224 (2021)
34. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**(6), 1137–1149 (2016)
35. Roh, B., Shin, J., Shin, W., Kim, S.: Sparse detr: Efficient end-to-end object detection with learnable sparsity. In: International Conference on Learning Representations (2021)
36. Tan, M., Pang, R., Le, Q.V.: Efficientdet: Scalable and efficient object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10781–10790 (2020)
37. Tian, Z., Shen, C., Chen, H., He, T.: Fcos: Fully convolutional one-stage object detection. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9627–9636 (2019)
38. Wang, Y., Zhang, X., Yang, T., Sun, J.: Anchor detr: Query design for transformer-based detector. In: Proceedings of the AAAI conference on artificial intelligence. vol. 36, pp. 2567–2575 (2022)
39. Xie, Z., Zhang, Z., Cao, Y., Lin, Y., Bao, J., Yao, Z., Dai, Q., Hu, H.: Simmim: A simple framework for masked image modeling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9653–9663 (2022)
40. Zhang, G., Luo, Z., Tian, Z., Zhang, J., Zhang, X., Lu, S.: Towards efficient use of multi-scale features in transformer-based object detectors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6206–6216 (2023)
41. Zhang, G., Luo, Z., Yu, Y., Cui, K., Lu, S.: Accelerating detr convergence via semantic-aligned matching. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 949–958 (2022)
42. Zhang, H., Li, F., Liu, S., Zhang, L., Su, H., Zhu, J., Ni, L., Shum, H.Y.: Dino: Detr with improved denoising anchor boxes for end-to-end object detection. In: The Eleventh International Conference on Learning Representations (2022)
43. Zhang, H., Wang, Y., Dayoub, F., Sunderhauf, N.: Varifocalnet: An iou-aware dense object detector. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8514–8523 (2021)
44. Zhang, H., Chang, H., Ma, B., Wang, N., Chen, X.: Dynamic r-cnn: Towards high quality object detection via dynamic training. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16. pp. 260–275. Springer (2020)

45. Zhang, S., Wang, X., Wang, J., Pang, J., Lyu, C., Zhang, W., Luo, P., Chen, K.: Dense distinct query for end-to-end object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7329–7338 (2023)
46. 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)
47. Zhou, X., Wang, D., Krähenbühl, P.: Objects as points. arXiv preprint arXiv:1904.07850 (2019)
48. Zhou, X., Zhuo, J., Krahenbuhl, P.: Bottom-up object detection by grouping extreme and center points. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 850–859 (2019)
49. Zhu, C., He, Y., Savvides, M.: Feature selective anchor-free module for single-shot object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 840–849 (2019)
50. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159 (2020)
51. 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)