Anchor DETR: Query Design for Transformer-Based Object Detection

Yingming Wang, Xiangyu Zhang, Tong Yang, Jian Sun

MEGVII Technology
{wangyingming, zhangxiangyu, yangtong, sunjian}@megvii.com

Abstract

In this paper, we propose a novel query design for the transformer-based object detection. In previous transformer-based detectors, the object queries are a set of learned embeddings. However, each learned embedding does not have an explicit physical meaning and we cannot explain where it will focus on. It is difficult to optimize as the prediction slot of each object query does not have a specific mode. In other words, each object query will not focus on a specific region. To solve these problems, in our query design, object queries are based on anchor points, which are widely used in CNN-based detectors. So each object query focuses on the objects near the anchor point. Moreover, our query design can predict multiple objects at one position to solve the difficulty: "one region, multiple objects". In addition, we design an attention variant, which can reduce the memory cost while achieving similar or better performance than the standard attention in DETR. Thanks to the query design and the attention variant, the proposed detector that we called Anchor DETR, can achieve better performance and run faster than the DETR with $10\times$ fewer training epochs. For example, it achieves 44.2 AP with 19 FPS on the MSCOCO dataset when using the ResNet50-DC5 feature for training 50 epochs. Extensive experiments on the MSCOCO benchmark prove the effectiveness of the proposed methods. Code is available at https://github.com/megvii-research/AnchorDETR.

Introduction

The object detection task is to predict a bounding box and a category for each object of interest in an image. In the last decades, there are many great progresses in object detection based on the CNN (Ren et al. 2015; Cai and Vasconcelos 2018; Redmon et al. 2016; Lin et al. 2017; Zhang et al. 2020; Qiao, Chen, and Yuille 2020; Chen et al. 2021). Recently, Carion et al. (Carion et al. 2020) propose the DETR which is a new paradigm of object detection based on the transformer. It uses a set of learned object queries to reason about the relations of the objects and the global image context to output the final predictions set. However, the learned object query is very hard to explain. It does not have an explicit physical meaning and the corresponding prediction slots of each object query do not have a specific mode. As shown in
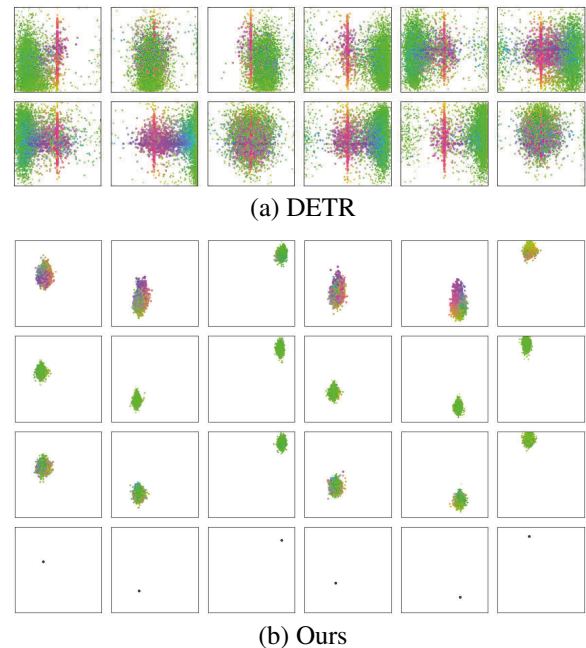
(a) DETR

(b) Ours

Figure 1: Visualization of the prediction slots. Note that the sub-figure (a) comes from the figure in DETR (Carion et al. 2020). Each prediction slot includes all box predictions on the val set of a query. Each of the colored points represents the normalized center position of a prediction. The points are color-coded so that the green color corresponds to small boxes, red to large horizontal boxes, and blue to large vertical boxes. The black points in the last row of sub-figure (b) indicate the anchor points. The prediction slots of ours are more related to a specific position than DETR.

Figure 1(a), the predictions of each object query in DETR are related to different areas and each object query will be in charge of a very large area. This positional ambiguity, i.e., the object query does not focus on a specific region, makes it hard to optimize.

Reviewing the detectors based on CNN, the anchors are highly related to the position and contain interpretable physical meanings. Inspired by this motivation, we propose a novel query design based on the anchor points, i.e., we en-

	Feature	AP	GFLOPs	FPS
DETR	DC5	43.3	187	12
SMCA	multi-level	43.7	152	10
Deformable DETR	multi-level	43.8	173	15
Conditional DETR	DC5	43.8	195	10
Ours	DC5	**44.2**	172	19

Table 1: Comparison with transformer detectors. The results are based on the ResNet-50 backbone and "DC5" means the dilated C5 feature. The DETR is trained with 500 epochs while the others are trained with 50 epochs. We evaluate the FPS of the proposed detector by following the script in DETR and set the batch size to 1 for a fair comparison with others. Note that we follow the DETR to script the model to evaluate the speed.

code the anchor points as the object queries. The object queries are the encodings of anchor points' coordinates so that each object query has an explicit physical meaning. However, this solution will encounter difficulty: there will be multiple objects appearing in one position. In this situation, only one object query in this position cannot predict multiple objects, so the object queries from other positions have to predict these objects collaboratively. It will cause each object query to be in charge of a larger area. Thus, we improve the object query design by adding multiple patterns to each anchor point so that each anchor point can predict multiple objects. As shown in Figure 1(b), all the predictions of the three patterns of each object query are distributed around the corresponding anchor point. In other words, it demonstrates that each object query only focuses on the objects near the corresponding anchor point. So the proposed object query can be easy to explain. As the object query has a specific mode and does not need to predict the object far away from the corresponding position, it can easier to optimize.

Besides the query design, we also design an attention variant that we call Row-Column Decouple Attention (RCDA). It decouples the 2D key feature into the 1D row feature and the 1D column feature, then conducts the row attention and column attention successively. The RCDA can reduce the memory cost while achieving similar or better performance than the standard attention in DETR. We believe it can be a good alternative to the standard attention in DETR.

As shown in Table 1, thanks to the novel query design based on the anchor point and the attention variant, the proposed detector Anchor DETR can achieve better performance and run faster than the original DETR with even $10\times$ fewer training epochs when using the same single-level feature. Compared with other DETR-like detectors with $10\times$ fewer training epochs, the proposed detector achieves the best performance among them. The proposed detector can achieve 44.2 AP with 19 FPS when using a single ResNet50-DC5 (He et al. 2016) feature for training 50 epochs.

The main contributions can be summarized as:

- We propose a novel query design based on anchor points for the transformer-based detectors. Moreover, we attach multiple patterns to each anchor point so that it can pre-

dict multiple objects for each position to deal with the difficulty: "one region, multiple objects". The proposed query based on the anchor point is more explainable and easier to optimize than the learned embedding. Thanks to the effectiveness of the proposed query design, our detector can achieve better performance with $10\times$ fewer training epochs than the DETR.

- We design an attention variant that we called Row-Column Decoupled Attention. It can reduce the memory cost while achieving similar or better performance than the standard attention in DETR, which can be a good alternative to standard attention.

- Extensive experiments are conducted to prove the effectiveness of each component.

Related Work

Anchors in Object Detection

There are two type of anchors used in CNN-based object detectors, i.e. anchor boxes (Ren et al. 2015; Lin et al. 2017) and anchor points (Tian et al. 2019; Zhou, Wang, and Krähenbühl 2019). As the hand-craft anchor boxes need to be carefully tuned to achieve good performance, we may prefer to not use the anchor boxes. We usually treat the anchor-free as the anchor boxes free so that the detectors using anchor point also be treated as anchor-free (Tian et al. 2019; Zhou, Wang, and Krähenbühl 2019). DETR (Carion et al. 2020) adopt neither the anchor boxes nor the anchor points. It directly predicts the absolute position of each object in the image. However, we find that introducing the anchor point into the object query can be better.

Transformer Detector

Carion et al. (Carion et al. 2020) propose the DETR which is based on the transformer (Vaswani et al. 2017) for object detection. The transformer detector will feed the information of value to the query based on the similarity of the query and key. Zhu et al. (Zhu et al. 2021) propose the Deformable DETR that samples information of deformable points around the reference points to the query and applies multiple level features to solve the slowly converge speed of the transformer detector. The reference points are similar to the anchor points but the object queries of it are still the learned embeddings. It predicts the reference points by the object queries rather than encoding the anchor points as the object queries. Gao et al. (Gao et al. 2021) add a Gaussian map in the original attention for each query. Concurrent with us, the Conditional DETR (Meng et al. 2021) encodes the reference point as the query position embedding. But the motivations are different so that it only uses the reference point to generate position embedding as the conditional spatial embedding in the cross attention and the object queries are still the learned embeddings. Besides, it does not involve the multiple predictions of one position and the attention variants.

Efficient Attention

The self-attention of the transformer has high complexity so it cannot well deal with a very large number of queries and

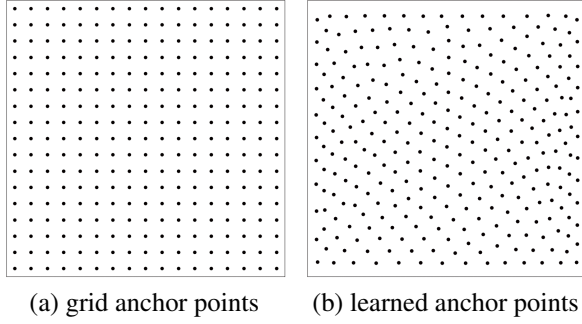| (a) grid anchor points | (b) learned anchor points |

Figure 2: Visualization of the anchor points distribution. Each point represents the normalized position of an anchor.

keys. To solve this problem, many efficient attention modules have been proposed (Shen et al. 2021; Vaswani et al. 2017; Beltagy, Peters, and Cohan 2020; Liu et al. 2021; Ma et al. 2021). One method is to compute the key and value first which can lead to linear complexity of the number of the query or key. The Efficient Attention (Shen et al. 2021) and the Linear Attention (Katharopoulos et al. 2020) follow this idea. Another method is to restrict the attention region of the key for each query instead of the whole region. The Restricted Self-Attention (Vaswani et al. 2017), Deformable Attention (Zhu et al. 2021), Criss-Cross Attention (Huang et al. 2019), and LongFormer (Beltagy, Peters, and Cohan 2020) follow this idea. In this paper, we decouple the key feature to the row feature and the column feature by 1D global average pooling and then perform the row attention and the column attention successively.

Method

Anchor Points

In the CNN-based detectors, anchor points always are the corresponding position of the feature maps. But it can be more flexible in the transformer-based detector. The anchor points can be the learned points, the uniform grid points, or other hand-craft anchor points. We adopt two types of anchor points. One is the grid anchor points and the other is the learned anchor points. As shown in Figure 2(a), the gird anchor points are fixed as the uniform grid points in the image. The learned points are randomly initialized with uniform distribution from 0 to 1 and updated as the learned parameters. With the anchor points, the center position $(\hat{c}x, \hat{c}y)$ of the predicted bounding box will be based on the corresponding anchor point like that in the Deformable DETR (Zhu et al. 2021).

Attention Formulation

The attention of DETR-like transformer can be formulate as:

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V,$$
$$Q = Q_f + Q_p, K = K_f + K_p, V = V_f,$$

(1)

where the d_k is the channel dimension, the subscript f means the feature, and the subscript p means the position

embedding. The Q, K, V are the query, key, value respectively. Note that the Q, K, V will pass through a linear layer respectively and that is omitted in Equation (1) for clarity.

There are two attentions in the DETR decoder. One is self-attention and the other is cross-attention. In the self-attention, the K_f and the V_f are the same as the Q_f while the K_p is the same as the Q_p. The $Q_f \in \mathbb{R}^{N_q \times C}$ is the output of last decoder and the initial $Q_f^{init} \in \mathbb{R}^{N_q \times C}$ for the first decoder can be set to a constant vector or learned embedding. For the query position embedding $Q_p \in \mathbb{R}^{N_q \times C}$, it uses a set of learned embedding in DETR:

$$Q_p = \text{Embedding}(N_q, C). \quad (2)$$

In the cross-attention, the $Q_f \in \mathbb{R}^{N_q \times C}$ is generated from the output of the self-attention in front while the $K_f \in \mathbb{R}^{HW \times C}$ and $V_f \in \mathbb{R}^{HW \times C}$ are the output feature of the encoder. The $K_p \in \mathbb{R}^{HW \times C}$ is the position embedding of K_f. It is generated by the sine-cosine position encoding function (Vaswani et al. 2017; Carion et al. 2020) g_{sin} based on the corresponding key feature position $Pos_k \in \mathbb{R}^{HW \times 2}$:

$$K_p = g_{sin}(Pos_k). \quad (3)$$

Note that the H, W, C is the height, width, channel of the feature and the N_q is the predefined number of the query.

Anchor Points to Object Query

Usually, the Q_p in decoder is regarded as the object query because it is responsible for distinguishing different objects. The object query of learned embedding like Equation (2) is hard to explain as discussed in the Introduction section.

In this paper, we propose to design the object query based on the anchor point Pos_q. The $Pos_q \in \mathbb{R}^{N_A \times 2}$ represents N_A points with their (x, y) positions which range from 0 to 1. Then the object query based on the anchor points can formulate as:

$$Q_p = Encode(Pos_q). \quad (4)$$

It means that we encode the anchor points as the object queries.

So how to design the encoding function? As the object query is designed as the query position embedding as shown in Equation (1), the most natural way is to share the same position encoding function as key:

$$Q_p = g(Pos_q), K_p = g(Pos_k), \quad (5)$$

where g is the position encoding function. The position encoding function could be the g_{sin} or other position encoding functions. In this paper, instead of just using the heuristic g_{sin}, we prefer to use a small MLP network with two linear layers to adapt it additionally.

Multiple Predictions for Each Anchor Point

To deal with the situation that one position may have multiple objects, we further improve the object query to predict multiple objects for each anchor point instead of just one prediction. Reviewing the initial query feature $Q_f^{init} \in \mathbb{R}^{N_q \times C}$, each of the N_q object queries has one pattern $Q_f^i \in \mathbb{R}^{1 \times C}$, where the i is the index of object queries. To predict
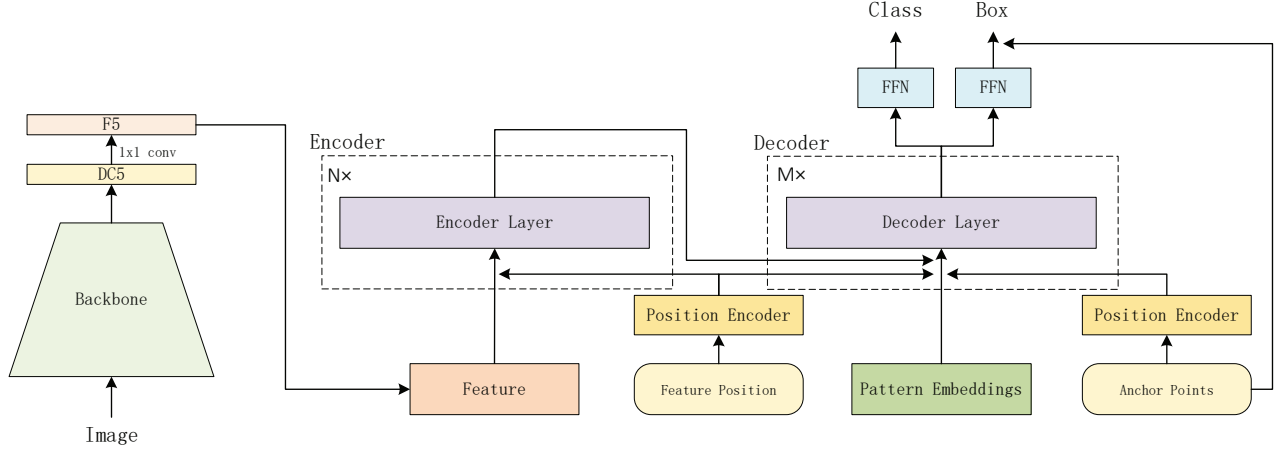
Figure 3: Pipeline of the proposed detector. Note that the encoder layer and decoder layer are in the same structure as DETR except that we replace the self-attention in the encoder layer and the cross-attention in the decoder layer with the proposed Row-Column Decouple Attention.

multiple objects for each anchor point, we can incorporate multiple patterns into each object query. We use a small set pattern embedding $Q_f^i \in \mathbb{R}^{N_p \times C}$:

$$Q_f^i = \text{Embedding}(N_p, C), \qquad (6)$$

to detect objects with different patterns at each position. The N_p is the number of pattern which is very small, e.g. $N_p = 3$. For the property of translation invariance, the patterns are shared for all the object queries. Thus we can get the initial $Q_f^{init} \in \mathbb{R}^{N_p N_A \times C}$ and $Q_p \in \mathbb{R}^{N_p N_A \times C}$ by sharing the $Q_f^i \in \mathbb{R}^{N_p \times C}$ to each of the $Q_p \in \mathbb{R}^{N_A \times C}$. Here the N_q is equal to $N_p \times N_A$. Then we can define the proposed Pattern-Position query design as:

$$Q = Q_f^{init} + Q_p. \qquad (7)$$

For the following decoder, the Q_f is also generated from the output of last decoder like DETR.

Thanks to the proposed query design, the proposed detector has an interpretable query and achieves better performance than the original DETR with $10\times$ fewer training epochs.

Row-Column Decoupled Attention

The transformer will cost a lot of GPU memory which may limit its use of the high-resolution feature or other extensions. The Deformable Transformer (Zhu et al. 2021) can reduce memory cost but it will lead to random access of memory which may not be friendly to modern accelerators of massive parallelism. There are also some attention modules (Ma et al. 2021; Shen et al. 2021) with linear complexity and will not lead to random access of memory. However, in our experiments, we find that these attention modules cannot well deal with the DETR-like detectors. It may be because the cross-attention in the decoder is much more difficult than the self-attention.

In this paper, we propose the Row-Column Decoupled Attention (RCDA) which can not only decrease the memory burdens but also achieve similar or better performance

than the standard attention in DETR. The main idea of the RCDA is to decouple the 2D key feature $K_f \in \mathbb{R}^{H \times W \times C}$ to 1D row feature $K_{f,x} \in \mathbb{R}^{W \times C}$ and 1D column feature $K_{f,y} \in \mathbb{R}^{H \times C}$, then perform the row attention and the column attention successively. We choose the 1D global average pooling to decouple the key feature by default. Without losing generality, we hypothesize $W \geq H$ and the RCDA can be formulated as:

$$A_x = \text{softmax}(\frac{Q_x K_x^T}{\sqrt{d_k}}), A_x \in \mathbb{R}^{N_q \times W},$$
$$Z = \text{weighted\_sumW}(A_x, V), Z \in \mathbb{R}^{N_q \times H \times C},$$
$$A_y = \text{softmax}(\frac{Q_y K_y^T}{\sqrt{d_k}}), A_y \in \mathbb{R}^{N_q \times H}, \qquad (8)$$
$$Out = \text{weighted\_sumH}(A_y, Z), Out \in \mathbb{R}^{N_q \times C},$$

where

$$Q_x = Q_f + Q_{p,x}, \quad Q_y = Q_f + Q_{p,y},$$
$$Q_{p,x} = g_{1D}(Pos_{q,x}), \quad Q_{p,y} = g_{1D}(Pos_{q,y}),$$
$$K_x = K_{f,x} + K_{p,x}, \quad K_y = K_{f,y} + K_{p,y}, \qquad (9)$$
$$K_{p,x} = g_{1D}(Pos_{k,x}), \quad K_{p,y} = g_{1D}(Pos_{k,y}),$$
$$V = V_f, \quad V \in \mathbb{R}^{H \times W \times C}.$$

The weighted_sumW and weighted_sumH operations conduct the weighted sum along the width dimension and the height dimension respectively. The $Pos_{q,x} \in \mathbb{R}^{N_q \times 1}$ is the corresponding row position of $Q_f \in \mathbb{R}^{N_q \times C}$ and the $Pos_{q,y} \in \mathbb{R}^{N_q \times 1}$, $Pos_{k,x} \in \mathbb{R}^{W \times 1}$, $Pos_{k,y} \in \mathbb{R}^{H \times 1}$ are in the similar manner. The g_{1D} is the 1D position encoding function which is similar to g and it will encode a 1D coordinate to a vector with C channels.

Now we analyze why it can save the memory. We hypothesize the head number M of multi-head attention to 1 for clarity in the previous formulation without losing generality, but we should consider the head number M for memory analysis. The major memory cost for the attention in

	Anchor-Free	NMS-Free	RAM-Free	Epochs	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
RetinaNet			✓	36	38.7	58.0	41.5	23.3	42.3	50.3
FCOS	✓		✓	36	41.4	60.1	44.9	25.6	44.9	53.1
POTO	✓	✓	✓	36	41.4	59.5	45.6	26.1	44.9	52.0
Faster RCNN				36	40.2	61.0	43.8	24.2	43.5	52.0
Cascade RCNN				36	44.3	62.2	48.0	26.6	47.7	57.7
Sparse RCNN	✓	✓		36	**44.5**	63.4	48.2	26.9	47.2	59.5
DETR-C5	✓	✓	✓	500	42.0	62.4	44.2	20.5	45.8	61.1
DETR-DC5	✓	✓	✓	500	43.3	63.1	45.9	22.5	47.3	61.1
SMCA	✓	✓	✓	50	43.7	63.6	47.2	24.2	47.0	60.4
Deformable DETR	✓	✓		50	43.8	62.6	47.7	26.4	47.1	58.0
Ours-C5	✓	✓	✓	50	42.1	63.1	44.9	22.3	46.2	60.0
Ours-DC5	✓	✓	✓	50	**44.2**	64.7	47.5	24.7	48.2	60.6
DETR-C5-R101	✓	✓	✓	500	43.5	63.8	46.4	21.9	48.0	61.8
DETR-DC5-R101	✓	✓	✓	500	44.9	64.7	47.7	23.7	49.5	62.3
Ours-C5-R101	✓	✓	✓	50	43.5	64.3	46.6	23.2	47.7	61.4
Ours-DC5-R101	✓	✓	✓	50	**45.1**	65.7	48.8	25.8	49.4	61.6

Table 2: Comparison with other detectors. The results are based on the ResNet-50 backbone if without specific and R101 means the ResNet-101 backbone. "C5", "DC5" indicate the detectors using a single C5 or DC5 feature respectively. The other detectors without specific use multiple-level features. The RAM means the random access of memory which usually is not friendly to hardware in practice.

RCDA	anchors	patterns	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
			39.3	59.4	41.8	20.7	42.6	55.0
	✓	✓	**44.2**	65.3	47.2	24.4	47.8	61.8
✓			40.3	61.6	42.5	21.7	44.1	56.3
✓		✓	40.3	60.8	43.0	21.1	44.2	57.0
✓	✓		42.6	63.6	45.5	23.2	46.4	58.3
✓	✓	✓	**44.2**	64.7	47.5	24.7	48.2	60.6

Table 3: Effectiveness of each component. The RCDA means to replace the standard attention with the RCDA. The "anchors" means using the anchor points to generate the object queries. The "patterns" means assigning multiple patterns to each object query so that each position has multiple predictions.

DETR is the attention weight map $A \in \mathbb{R}^{N_q \times H \times W \times M}$. The attention weight map of RCDA is $A_x \in \mathbb{R}^{N_q \times W \times M}$ and $A_y \in \mathbb{R}^{N_q \times H \times M}$ whose memory cost is much smaller than the A. However, the major memory cost in the RCDA is the temporary result Z. So we should compare the memory cost of $A \in \mathbb{R}^{N_q \times H \times W \times M}$ and $Z \in \mathbb{R}^{N_q \times H \times C}$. The ratio for saving memory of the RCDA is:

$$r = (N_q \times H \times W \times M)/(N_q \times H \times C) \\ = W \times M/C \quad (10)$$

where the default setting is $M = 8, C = 256$. So the memory cost is roughly the same when the large side W is equal to 32 which is a typical value for the C5 feature in object detection. When using the high-resolution feature it can save the memory, e.g saving roughly 2x memory for the C4 or DC5 feature and saving 4x memory for the C3 feature.

Experiment

Implementation Details

We conduct the experiments on the MS COCO (Lin et al. 2014) benchmark. All models are trained on the train2017 split and evaluated on the val2017. The models are trained on 8 GPUs with 1 image per GPU. We train our model on the training set for 50 epochs with the AdamW optimizer (Loshchilov and Hutter 2019) setting the initial learning rate to 10^{-5} for the backbone and 10^{-4} for the others. The learning rate will be decayed by a factor of 0.1 at the 40th epoch. We set the weight decay to 10^{-4} and the dropout rate to 0 (i.e. remove dropout). The head for the attention is 8, the attention feature channel is 256 and the hidden dimension of the feed-forward network is 1024. We choose the number of anchor points to 300 and the number of patterns to 3 by default. We use a set of learned points as anchor points by default. The number of encoder layers and decoder layers is 6 like DETR. We use the focal loss (Lin et al. 2017) as the classification loss following the Deformable DETR.

Main Results

As shown in Table 2, we compare the proposed detector with RetinaNet (Lin et al. 2017), FCOS (Tian et al. 2019), POTO (Wang et al. 2020), Faster RCNN (Ren et al. 2015), Cascased RCNN (Cai and Vasconcelos 2018), Sparse RCNN (Sun et al. 2020), DETR (Carion et al. 2020),

anchor points	patterns	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
100	1	40.8	61.5	43.6	21.0	44.8	57.5
100	3	43.4	63.7	46.5	23.8	47.1	60.9
300	1	42.6	63.6	45.5	23.2	46.4	58.3
300	3	**44.2**	64.7	47.5	24.7	48.2	60.6
900	1	42.9	63.8	46.1	24.4	46.7	58.9

Table 4: Comparison for multiple predictions of one position. We show the performance with the different number of anchor points and patterns. Note that it degenerates to the single prediction of one anchor point when the number of the pattern is 1.
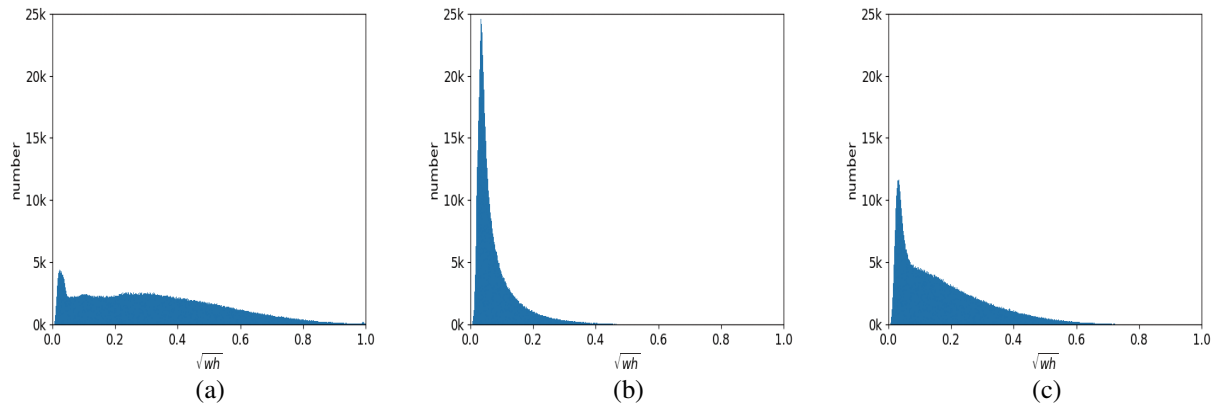

(a)　　　　　　　　　(b)　　　　　　　　　(c)

Figure 4: The histograms of each pattern. We show the histograms of predicted box sizes for different patterns. The abscissa represents the square root of the area of the predicted box with normalized width "w" and height "h". The large boxes usually appear in pattern (a), the pattern (b) focuses on the small objects, and the pattern (c) is in between.

	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
grid	44.1	64.5	47.6	24.8	48.0	60.9
learned	**44.2**	64.7	47.5	24.7	48.2	60.6

Table 5: Comparison of different types of anchor points. The "learned" means that it uses 300 learned anchor points. The "grid" means that it uses the hand-craft grid anchor points with the number of 17×17 which is near to 300.

SMCA (Gao et al. 2021), and Deformable DETR (Zhu et al. 2021). The "C5" and "DC5" mean that the detector uses a single C5 or dilated C5 feature while the other detectors use multiple-level features. Using multiple-level features usually will have a better performance but cost more resources. Surprisingly, our detector with a single DC5 feature can achieve better performance than the Deformable DETR and SCMA which use multiple-level features. The proposed detector can achieve better performance than DETR with 10× fewer training epochs. It proves the proposed query design is very effective.

We also show the property of anchor-free, NMS-free, and RAM-free for each detector in Table 2. The anchor-free and NMS-free indicate the detector does not need the hand-craft anchor box and non-maximum suppression. The RAM-free means the detector will not involve any random access of memory which can be very friendly to modern accelerators

in practice. The two-stage detectors always are not RAM-free because the region of interest (RoI) is stochastic to the hardware and the operation of the RoI-Align (He et al. 2017)/RoI-Pooling (Girshick 2015) will involve the random access of memory. The Deformable DETR is similar to the two-stage detector as the position of the sample point is stochastic to the hardware so that it is not RAM-free. On the contrary, the proposed detector inherits the properties of anchor-free, NMS-free, and RAM-free from the DETR with an improvement of performance and fewer training epochs.

Ablation Study

Effectiveness of each component　Table 3 shows the effectiveness of each component that we proposed. The proposed query design based on anchor points that can predict multiple objects for each position improves the performance from 39.3 AP to 44.2 AP. The improvements are 4.9 AP which proves the query design with a specific focused area can be much easier to optimize. For the attention variant RCDA, it achieves similar performance as the standard attention when adopting the proposed query design. It proves that the RCDA will not degrade the performance with lower memory cost. Besides, by applying the original query design as the DETR, the RCDA can achieve 1.0 improvements. We think it is slightly easier to optimize as the attention map is smaller than the standard attention. This gain will vanish when adopting the proposed query design as the query de-

Attention	Feature	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l	memory
Luna (Ma et al. 2021)		36.2	58.7	37.1	15.4	39.2	54.7	2.9G
Efficient-att (Shen et al. 2021)	DC5	34.1	56.8	34.4	13.9	36.0	53.9	2.2G
std-att		**44.2**	65.3	47.2	24.4	47.8	61.8	10.5G
RCDA		**44.2**	64.7	47.5	24.7	48.2	60.6	4.4G
std-att	C5	42.2	63.5	44.9	21.9	45.9	61.0	2.7G
RCDA		42.1	63.1	44.9	22.3	46.2	60.0	2.3G

Table 6: Comparison of different attention modules. The "std-att" means the standard attention in DETR. The linear complexity attention modules can reduce memory significantly but the performance will drop. The proposed RCDA can achieve similar performance as the standard attention in DETR while saving the memory.

sign has the same effect that makes it easier to optimize. The object query based on anchor points with a single prediction has 2.3 AP improvements and the multiple predictions for each anchor point can have 1.6 further improvements. Applying the multiple predictions to the original object query in DETR will not improve the performance. It is because the original object query is not highly related to the position thus it cannot get benefits from the "one region, multiple predictions".

Multiple Predictions for Each Anchor Point   Table 4 shows that multiple predictions for each anchor point play an essential role in query design. For a single prediction (1 pattern), we find that 100 anchor points are not good enough and 900 anchor points get similar performance as 300 anchor points, thus we use 300 anchor points by default. The multiple predictions (3 patterns) can outperform the single prediction (1 pattern) by 2.6 and 1.6 AP for 100 and 300 anchor points respectively. With the same number of the predictions, the multiple predictions for each anchor point (300 anchor points, 3 patterns) can outperform the single prediction (900 anchor points, 1 pattern) for 1.3 AP. It indicates that the improvements of multiple predictions do not come from the increased number of predictions. These results prove the effectiveness of multiple predictions for each anchor point.

Anchor Points Types   We have tried two types of anchor points, i.e., the grid anchor points and the learned anchor points. As shown in Figure 2, the learned anchor points distribute uniformly in the image which is similar to the grid anchor points. We hypothesize that this is because the objects distribute everywhere in large COCO data set. We also find the grid anchor points can achieve similar performance as the learned anchor points in Table 5.

Prediction Slots of Object Query   As shown in Figure 1(b), we can observe that the prediction slots of each object query in the proposed detector will focus on the objects near the corresponding anchor point. As there are three patterns for the anchor points, we also show the histogram of each pattern in Figure 4. We find that the patterns are related to the object size as the large boxes usually appear in pattern (a) while pattern (b) focuses on the small objects. But the query patterns do not just depend on the size of the object because the small boxes can also appear in the pattern (a). We think it is because there are more small objects, and mul-

tiple small objects are more likely to appear in one region. So all the patterns will be in charge of the small objects.

Row-Column Decoupled Attention   As shown in Table 6, we compare the Row-Column Decoupled Attention with the standard attention in DETR and some efficient attention modules with linear complexity. The attention modules with linear complexity can significantly reduce the training memory compared to the standard attention module. However, their performance drops significantly. It may be because the cross-attention in the DETR-like detectors is much more difficult than the self-attention. On the contrary, the Row-Column Decoupled Attention can achieve similar performance. As previously discussed, the Row-Column Decoupled Attention can significantly reduce memory when using the high-resolution feature and get roughly the same memory cost when using the C5 feature. For example, RCDA reduces the training memory from 10.5G to 4.4G and gets the same performance as the standard attention when using the DC5 feature. In conclusion, the Row-Column Decoupled Attention can be efficient in memory while preserving competitive performance so that it can be a good alternative to the standard attention in DETR.

Conclusion

In this paper, we propose a detector based on the transformer. We propose a novel query design based on the anchor point that has explicit physical meaning. The corresponding prediction slots can have a specific mode, i.e. the predictions are near the anchor point, so that it is easier to optimize. Moreover, we incorporate multiple patterns to each anchor point to solve the difficulty: "one region, multiple objects". We also propose an attention variant named the Row-Column Decoupled Attention. The Row-Column Decouple Attention can reduce the memory cost while achieving similar or better performance than the standard attention in DETR. The proposed detector can run faster and achieve better performance with $10\times$ fewer training epochs than the DETR. The proposed detector is an end-to-end detector with the property of anchor-free, NMS-free, and RAM-free. We hope the proposed detector can be useful in practice and be a strong simple baseline for research.

Acknowledgments

References

Beltagy, I.; Peters, M. E.; and Cohan, A. 2020. Longformer: The long-document transformer. *ArXiv preprint*, abs/2004.05150.

Cai, Z.; and Vasconcelos, N. 2018. Cascade R-CNN: Delving Into High Quality Object Detection. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 6154–6162. IEEE Computer Society.

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-end object detection with transformers. In *European Conference on Computer Vision*, 213–229. Springer.

Chen, Q.; Wang, Y.; Yang, T.; Zhang, X.; Cheng, J.; and Sun, J. 2021. You only look one-level feature. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13039–13048.

Gao, P.; Zheng, M.; Wang, X.; Dai, J.; and Li, H. 2021. Fast Convergence of DETR with Spatially Modulated Co-Attention. *ArXiv preprint*, abs/2101.07448.

Girshick, R. B. 2015. Fast R-CNN. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, 1440–1448. IEEE Computer Society.

He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. B. 2017. Mask R-CNN. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, 2980–2988. IEEE Computer Society.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 770–778. IEEE Computer Society.

Huang, Z.; Wang, X.; Huang, L.; Huang, C.; Wei, Y.; and Liu, W. 2019. CCNet: Criss-Cross Attention for Semantic Segmentation. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, 603–612. IEEE.

Katharopoulos, A.; Vyas, A.; Pappas, N.; and Fleuret, F. 2020. Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, 5156–5165. PMLR.

Lin, T.; Goyal, P.; Girshick, R. B.; He, K.; and Dollár, P. 2017. Focal Loss for Dense Object Detection. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, 2999–3007. IEEE Computer Society.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, 740–755. Springer.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. *ArXiv preprint*, abs/2103.14030.

Loshchilov, I.; and Hutter, F. 2019. Decoupled Weight Decay Regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Ma, X.; Kong, X.; Wang, S.; Zhou, C.; May, J.; Ma, H.; and Zettlemoyer, L. 2021. Luna: Linear Unified Nested Attention. arXiv:2106.01540.

Meng, D.; Chen, X.; Fan, Z.; Zeng, G.; Li, H.; Yuan, Y.; Sun, L.; and Wang, J. 2021. Conditional DETR for Fast Training Convergence. arXiv:2108.06152.

Qiao, S.; Chen, L.-C.; and Yuille, A. 2020. DetectoRS: Detecting Objects with Recursive Feature Pyramid and Switchable Atrous Convolution. *ArXiv preprint*, abs/2006.02334.

Redmon, J.; Divvala, S. K.; Girshick, R. B.; and Farhadi, A. 2016. You Only Look Once: Unified, Real-Time Object Detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 779–788. IEEE Computer Society.

Ren, S.; He, K.; Girshick, R. B.; and Sun, J. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In Cortes, C.; Lawrence, N. D.; Lee, D. D.; Sugiyama, M.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, 91–99.

Shen, Z.; Zhang, M.; Zhao, H.; Yi, S.; and Li, H. 2021. Efficient attention: Attention with linear complexities. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 3531–3539.

Sun, P.; Zhang, R.; Jiang, Y.; Kong, T.; Xu, C.; Zhan, W.; Tomizuka, M.; Li, L.; Yuan, Z.; Wang, C.; et al. 2020. Sparse r-cnn: End-to-end object detection with learnable proposals. *ArXiv preprint*, abs/2011.12450.

Tian, Z.; Shen, C.; Chen, H.; and He, T. 2019. FCOS: Fully Convolutional One-Stage Object Detection. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, 9626–9635. IEEE.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In Guyon, I.; von Luxburg, U.; Bengio, S.; Wallach, H. M.; Fergus, R.; Vishwanathan, S. V. N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 5998–6008.

Wang, J.; Song, L.; Li, Z.; Sun, H.; Sun, J.; and Zheng, N. 2020. End-to-end object detection with fully convolutional network. *ArXiv preprint*, abs/2012.03544.

Zhang, S.; Chi, C.; Yao, Y.; Lei, Z.; and Li, S. Z. 2020. Bridging the Gap Between Anchor-Based and Anchor-Free Detection via Adaptive Training Sample Selection. In *2020*

IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020, 9756–9765. IEEE.

Zhou, X.; Wang, D.; and Krähenbühl, P. 2019. Objects as Points. volume abs/1904.07850.

Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; and Dai, J. 2021. Deformable DETR: Deformable Transformers for End-to-End Object Detection. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.