



# GRA: Detecting Oriented Objects through Group-wise Rotating and Attention

Jiangshan Wang<sup>1\*</sup>, Yifan Pu<sup>1\*</sup>, Yizeng Han<sup>2</sup>, Jiayi Guo<sup>1</sup>,  
Yiru Wang<sup>3</sup>, Xiu Li<sup>1</sup> , and Gao Huang<sup>1</sup> 

<sup>1</sup> Tsinghua University

<sup>2</sup> DAMO Academy, Alibaba group

<sup>3</sup> ModelTC

{wjs23, puyf23}@mails.tsinghua.edu.cn,  
li.xiu@sz.tsinghua.edu.cn, gaohuang@tsinghua.edu.cn

**Abstract.** Oriented object detection, an emerging task in recent years, aims to identify and locate objects across varied orientations. This requires the detector to accurately capture the orientation information, which varies significantly within and across images. Despite the existing substantial efforts, simultaneously ensuring model effectiveness and parameter efficiency remains challenging in this scenario. In this paper, we propose a lightweight yet effective **Group-wise Rotating and Attention (GRA)** module to replace the convolution operations in backbone networks for oriented object detection. GRA can adaptively capture fine-grained features of objects with diverse orientations, comprising two key components: Group-wise Rotating and Group-wise Attention. Group-wise Rotating first divides the convolution kernel into groups, where each group extracts different object features by rotating at a specific angle according to the object orientation. Subsequently, Group-wise Attention is employed to adaptively enhance the object-related regions in the feature. The collaborative effort of these components enables GRA to effectively capture the various orientation information while maintaining parameter efficiency. Extensive experimental results demonstrate the superiority of our method. For example, GRA achieves a new state-of-the-art (SOTA) on the DOTA-v2.0 benchmark, while saving the parameters by nearly 50% compared to the previous SOTA method. Code is available at <https://github.com/wangjiangshan0725/GRA>.

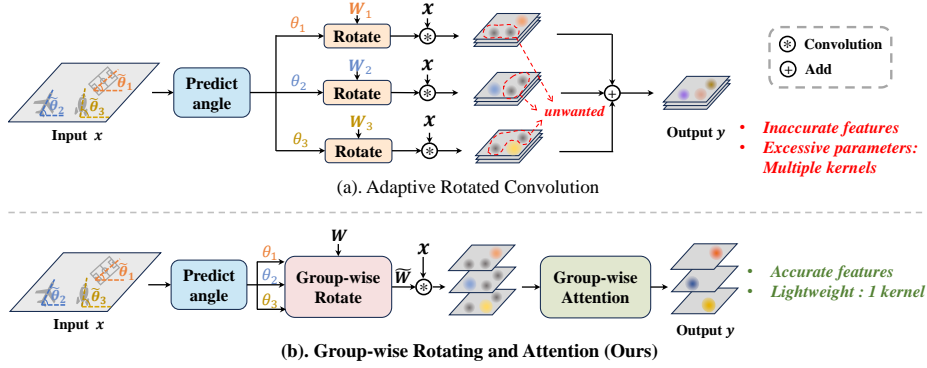
**Keywords:** Oriented Object Detection · Group-wise Rotating · Spatial Attention

## 1 Introduction

Oriented object detection is the task of identifying and locating multiple objects with oriented bounding boxes. Different from conventional object detection

---

\* Equal contribution.  Corresponding author.



**Fig. 1: Comparison between the recent SOTA method ARC [48] and our proposed method GRA.** Our method not only reduces the noise in the features but is also much more lightweight.

which utilizes fixed horizontal bounding boxes, oriented object detection tackles a more intricate challenge, aligning better with real-world scenarios where objects frequently present various orientations. The ability to discern such oriented objects can greatly enhance situational awareness and decision-making processes across numerous industries, thereby having wide-ranging applications in image segmentation [2, 81], autonomous driving [28, 37], text recognition [34, 79], face detection [32, 71], remote sensing [25, 61], exemplar-based generation [7, 8, 43, 55, 63] and Embodied AI [41, 67, 70], *etc.*

Extensive efforts have been made to enhance the performance of oriented object detection from various perspectives, including the oriented bounding box representation [11, 27, 35, 69, 77], loss functions [1, 50, 74, 76, 78, 80], network architecture [12, 26, 72, 73, 75], and label assignment strategies [44, 45]. Recent studies [13, 68] indicate that the key to identifying oriented objects lies in accurately extracting their orientation information. This inspires researchers to start developing object-wise orientation-aware detection backbones.

To this end, the recent SOTA method [48] first proposes an Adaptive Rotated Convolution (ARC) module [48] to replace the convolution operations in existing detection backbones. ARC employs a set of  $m$  convolutional kernels, each maintaining a specific rotation angle, to separately extract the orientation information of different objects. Then the features extracted by each kernel are aggregated through a weighted sum as the output of ARC. Despite the performance improvement, ARC still encounters two primary challenges. Firstly, the utilization of  $m$  convolutional kernels results in  $(m - 1)$  times more parameters, significantly constraining its deployment on resource-constrained devices such as remote sensing equipment. Secondly, we argue that the straightforward output feature aggregation would diminish the representation capacity of the orientation information, resulting in inaccurate detection. Specifically, we discover that a specific convolution kernel (e.g.  $W_1$  in Fig. 1a) primarily captures the fea-

tures of objects aligned with its rotational angles (e.g. the **basketball court** in Fig. 1a). Consequently, features of objects at other orientations (e.g. the **plane** and the **rocket** in Fig. 1a) processed by this kernel would be marred by undesired noises. The weighted sum of these features will mix the satisfying features and undesired noises, leading to imprecise final predictions.

To overcome these challenges, we propose the lightweight yet efficient **Group-wise Rotating and Attention (GRA)** module (Fig. 1b) to capture more fine-grained orientation information. In particular, the GRA module comprises two key components: Group-wise Rotating and Group-wise Attention. The Group-wise Rotating component introduces a novel mechanism for rotation at the group level, which substantially reducing the parameter count compared to the SOTA method [48]. Specifically, the convolution kernel  $\mathbf{W} \in \mathbb{R}^{C_{\text{out}} \times C_{\text{in}} \times k \times k}$  is divided into  $n$  groups along the  $C_{\text{out}}$  channel. Each group of kernels is adaptively rotated based on the input feature before the convolution operation. Subsequent to Group-wise Rotating, Group-wise Attention partitions the output feature into  $n$  groups similarly and applies the spatial attention mechanism. This process enables the adaptive enhancement of desirable feature regions while mitigating extraneous noise. The collaborative endeavor of these components enables extracting refined and accurate orientation information without significantly increasing the parameters of the network.

The proposed GRA module is lightweight and flexible, which is able to be seamlessly integrated into any convolution neural network to achieve improved performance. Extensive experiments on oriented object detection benchmarks including DOTA-v1.0 [61], DOTA-v2.0 [6] and HRSC2016 [40] demonstrate that GRA not only achieves a significant reduction (near 50%) in the number of parameters compared with SOTA method [48] but also yields better detection performances, achieving a new SOTA result on DOTA-v2.0 dataset.

## 2 Related work

**Oriented Object Detection.** Oriented object detection advances the conventional horizontal object detection paradigm by employing oriented bounding boxes, making it a more precise task. Numerous studies have been dedicated to devising specialized detectors for rotated objects, including various components such as enhancing features in the detector neck [72, 73, 75], the oriented region proposal network [3, 66], the mechanism for extracting rotated regions of interest (RoI) [5, 66], the design of the detector head [12, 26], and the utilization of advanced label assignment strategies [44, 45]. Another research direction [11, 27, 35, 69] focuses on developing more adaptable representations of objects. Simultaneously, there has been extensive research into devising suitable loss functions for various oriented object representations [74, 76, 78]. Besides oriented object detection, there are also some works focusing on detecting other kinds of objects [21–23] or even video frames [64], which are more challenging than conventional object detection.

**Design of Backbone for Oriented Object Detection.** Recent endeavors in the field of oriented object detection have increasingly concentrated on optimizing the architecture of backbone networks for oriented object detection. ReDet [13] integrates rotation-equivariant operations within the backbone, yielding features that maintain orientation fidelity. However, such operations may not fully account for the diversity of object orientations within individual images or across datasets. LSKNet [36] proposes to use spatial attention [60, 62] to adaptively choose a suitable kernel size for the input image while the orientation information is also not taken into account. ARC [48] proposes to extract the orientation information of oriented objects by dynamically rotating the convolution kernels conditioned on different image samples. On the other hand, a large number of additional parameters are introduced and the features of different angles are mixed, introducing the undesired noises.

**Dynamic Neural Networks.** Contrary to static models, which maintain constant computational graphs and parameters throughout inference, dynamic neural networks [15, 49, 58] can modify their architecture or parameters in response to varying inputs. Dynamic networks are typically categorized into three distinct types: sample-wise [9, 14, 18, 30, 46, 57, 59, 65], spatial-wise [16, 17, 19, 31], and temporal-wise [10, 20, 56]. Recently, with the development of query-based Detection Transformers [29, 52, 83], a new kind of query-based dynamic network has begun to flourish [47]. In this work, we introduce a sample-wise dynamic network, which dynamically adjusts the model parameters conditioned on different input images. To be specific, the convolution kernels are rotated in a group-wise manner, which endows convolution parameters with a better ability to capture orientation information in the images.

### 3 Method

In this section, we firstly introduce the current SOTA method [48] and analyze its limitations. Subsequently, we delineate our proposed GRA module, comprising Group-wise Rotating and Group-wise Attention. Within the Group-wise Rotating module, a lightweight network named angle generator is employed to predict  $n$  rotation angles from the input feature map  $\mathbf{x}$ . Then the convolution kernel  $\mathbf{W} \in \mathbb{R}^{C_{\text{out}} \times C_{\text{in}} \times k \times k}$  is partitioned into  $n$  groups at  $C_{\text{out}}$  dimension, each rotating to the corresponding angle predicted by the angle generator. These rotated groups of kernels are subsequently concatenated together to perform the convolution with  $\mathbf{x}$ , obtaining the output feature  $\mathbf{y}$ . Followed by Group-wise Rotating, the Group-wise Attention module is proposed to adaptively emphasize satisfying regions of each feature group in  $\mathbf{y}$  while attenuating extraneous noise. The final output of GRA module  $\tilde{\mathbf{y}}$  is fed into subsequent modules in the network. Compared with the recent SOTA method [48], GRA achieves a substantial reduction of parameters and effectively mitigates the issue of inaccurate features in an elegant and reasonable manner. Finally, we will discuss the remarkable advantages of GRA. An overview of our method is shown in Fig. 4.

### 3.1 Preliminary

There have been several previous works focusing on developing specialized backbones for oriented object detection. Among them, Adaptive Rotated Convolution (ARC) [48] proposes to dynamically adjust the orientation of convolutional kernels based on the objects present within the image. This method improves the extraction of features from oriented objects, thereby achieving state-of-the-art (SOTA) performance across various benchmarks. In an ARC module,  $m$  convolution kernels  $\{\mathbf{W}_i\}$ , where  $\mathbf{W}_i \in \mathbb{R}^{C_{\text{out}} \times C_{\text{in}} \times k \times k}$  and  $i \in \{1, 2, \dots, m\}$ , are applied to capture features of multiple oriented objects. A routing function is employed which takes the feature map  $\mathbf{x}$  as the input, predicting the angle  $\theta_i$  and the weight  $\lambda_i$  for each kernel  $\mathbf{W}_i$ .

$$\{\theta_i\}_{i \in \{1, 2, \dots, m\}}, \{\lambda_i\}_{i \in \{1, 2, \dots, m\}} = \text{Routing}(\mathbf{x}), \quad (1)$$

Subsequently, the kernels are rotated at the corresponding predicted angles, obtaining  $\{\widetilde{\mathbf{W}}_i\}$ , where each  $\widetilde{\mathbf{W}}_i = \text{Rotate}(\mathbf{W}_i, \theta_i)$  and  $i \in \{1, 2, \dots, m\}$ . Each  $\widetilde{\mathbf{W}}_i$  independently performs convolution on the input feature map  $\mathbf{x}$ , yielding  $\mathbf{y}_i$ . These intermediate results are then aggregated through a weighted sum to produce the final output  $\mathbf{y}$ ,

$$\mathbf{y}_i = \text{Conv}(\mathbf{x}, \widetilde{\mathbf{W}}_i), \quad \mathbf{y} = \sum_{i=1}^m \lambda_i \times \mathbf{y}_i. \quad (2)$$

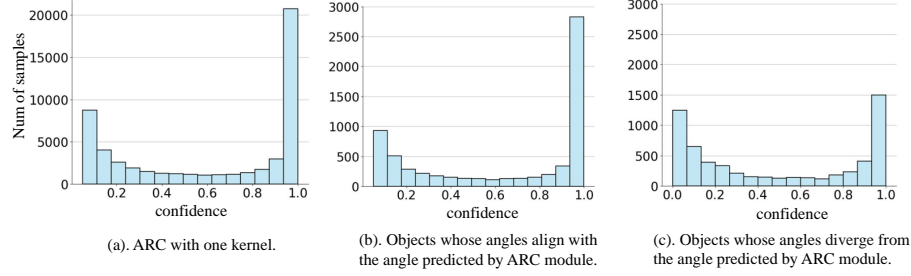
The resultant feature  $\mathbf{y}$  is fed into the following modules. In experiments, the value of  $m$  is generally assigned to a relatively small value (typically set to 4).

### 3.2 Limitations Analysis

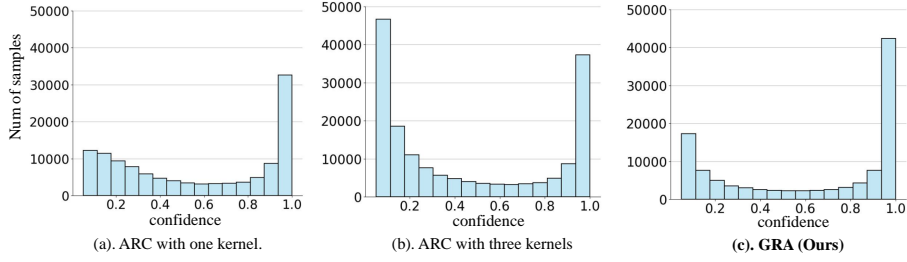
While ARC [48] exhibits superior performance compared to traditional convolution, there still exist two main shortcomings as illustrated following:

**Excessive parameters.** Within each ARC module,  $m$  convolution kernels  $\mathbf{W}_i$  are employed, multiplying the number of parameters by  $m$  compared to a standard convolution layer. The escalation in parameters is proportional to the factor  $m$ , thereby amplifying the complexity of the network. For instance, a standard ResNet50 comprises approximately 23.5M parameters; However, integrating ARC with  $m = 4$  inflates the total parameters to the overwhelming 57.2M. Such expansion poses a considerable challenge for deployment in resource-limited settings such as remote sensing devices, where storage space is at a premium and cannot support models with extensive parameters.

**Inaccurate features.** Within the ARC framework, features obtained from convolution kernels rotated at various angles are aggregated through a weighted sum. While this mechanism aims to detect objects at different rotational orientations within images, it introduces inaccurate information and reduces the representational capacity of the output feature  $\mathbf{y}$ . To illustrate this issue, we train an Oriented RCNN [66] with the ARC module (setting  $m = 1$ ) on the training set from DOTA. We then evaluate the performance of the detector on the validation set and visualize the distribution of confidences across all detected objects



**Fig. 2: Distribution of the confidence predicted by ARC ( $m = 1$ ).** The angle predicted by ARC module can affect the final prediction of the objects with different angles in the images. In general, the objects whose orientation is close to the angle predicted by ARC module can be detected with higher confidence compared to the objects whose orientation diverges from the ARC-predicted angle.



**Fig. 3: Comparison of the prediction between ARC and our method.** The weighted sum of ARC can lead to inaccurate features, causing a number of low-confidence detections. On the other hand, our method can detect more objects in the images with higher confidence.

(Fig. 2a). To examine the influence of the ARC module on the detector’s predictions, we specifically consider objects whose ground truth orientations closely align with the angle predicted by the ARC module (i.e. the angle for rotating the kernel), analyzing the confidence of these objects predicted by the detector. For these objects, the detector detects them with a predominant majority demonstrating high confidence levels as depicted in Fig. 2b. Conversely, for objects whose ground truth orientations diverge from the angle predicted by ARC module, the detector struggles to accurately detect them, resulting in numerous low-confidence detections (Fig. 2c). This indicates that the features extracted by the backbone for these objects are inadequate (corresponds to the unwanted features in Fig. 1), thereby hindering the subsequent detector head from accurately identifying these objects. Additional experiments conducted on the test set of DOTA (as shown in Figure 3) further elucidate the impact of integrating the ARC and GRA modules respectively. Integrating ARC with three kernels ( $m = 3$ ) into the backbone enables the detector to identify a greater number of

oriented objects in images, consequently leading to improved performance compared to using ARC with just one kernel ( $m = 1$ ). However, it also results in a notable increase in low-confidence detections (Fig. 3b), some of which should have been detected with high confidence. This observation suggests that a direct weighted sum of features is sub-optimal, potentially compromising the features obtained by various rotated kernels. In contrast, our approach mitigates this issue effectively, resulting in significantly improved performance (Fig. 3c).

### 3.3 Group-wise Rotation

In this section, we will comprehensively introduce the Group-wise Rotation module within GRA, which comprises three integral components: Predicting Rotating Angle, Grouping, and Rotating.

**Predicting Rotating Angle.** To predict the rotation angle corresponding to objects within a feature map  $\mathbf{x} \in \mathbb{R}^{C_{\text{in}} \times H_{\text{in}} \times W_{\text{in}}}$ , we employ a lightweight network named angle generator. Initially, the feature map  $\mathbf{x}$  undergoes processing through a depth-wise convolution layer, followed by ReLU activation and Layer-Norm layers, which aim to extract inherent orientation information from diverse objects. Subsequently, a global pooling layer transforms the feature into a vector with dimension  $C_{\text{in}}$ . This resulting vector then passes through two distinct linear layers, each with  $n$  output channels, producing a series of predicted angles  $\{\theta_j\}_{j \in \{1, 2, \dots, n\}}$  and scale factors  $\{\lambda_j\}_{j \in \{1, 2, \dots, n\}}$ .

These predicted angles and scale factors are utilized for the group-wise rotation of convolution kernels. Notably, the number of angles  $n$  is significantly smaller than the output channels of convolution,  $C_{\text{out}}$ . Consequently, it is convenient to partition the convolution kernels into  $n$  distinct groups at the  $C_{\text{out}}$  channel, with each group subjected to a unique rotation angle.

**Grouping.** The convolution operation utilizes a kernel  $\mathbf{W} \in \mathbb{R}^{C_{\text{out}} \times C_{\text{in}} \times k \times k}$  which takes  $\mathbf{x} \in \mathbb{R}^{C_{\text{in}} \times H_{\text{in}} \times W_{\text{in}}}$  as the input to produce the output feature  $\mathbf{y} \in \mathbb{R}^{C_{\text{out}} \times H_{\text{out}} \times W_{\text{out}}}$ . This operation involves  $C_{\text{out}}$  individual kernels  $\mathbf{w}_i \in \mathbb{R}^{C_{\text{in}} \times k \times k}$ ,  $i \in \{1, 2, \dots, C_{\text{out}}\}$ , convolving with  $\mathbf{x}$  separately. The resulting outputs are then concatenated to obtain the final output  $\mathbf{y}$ . In our method, these  $C_{\text{out}}$  kernels are uniformly divided into  $n$  groups  $\{\mathbf{W}_j\}$ ,

$$\mathbf{W} = \{\mathbf{w}_i, i \in \{1, 2, \dots, C_{\text{out}}\}\} \quad (3)$$

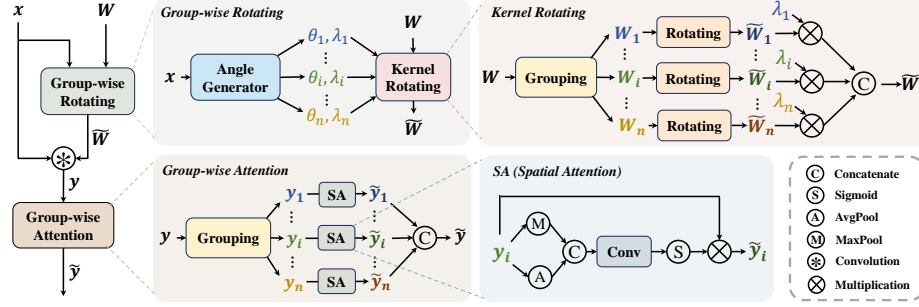
$$= \{\mathbf{W}_j, j \in \{1, 2, \dots, n\}\}. \quad (4)$$

Each group  $\mathbf{W}_j$  contains  $C_{\text{out}}/n$  kernels  $\mathbf{w}_{j,l}$ ,

$$\mathbf{W}_j = \{\mathbf{w}_{j,l}, l \in \{1, 2, \dots, C_{\text{out}}/n\}\}. \quad (5)$$

By employing the grouping strategy, kernels in different groups work independently for the ensuing rotation operation.

**Rotating.** After predicting rotational angles and grouping kernels, each kernel within the  $j$ th group  $\mathbf{W}_j$  undergoes rotation by the corresponding angle  $\theta_j$ , resulting in the rotated kernel group  $\widetilde{\mathbf{W}}_j$ . Additionally, each group of kernels



**Fig. 4: An overview of our proposed GRA method**, which contains two components: Group-wise Rotating and Group-wise Attention. In Group-wise Rotating module, the input kernel  $W \in \mathbb{R}^{C_{out} \times C_{in} \times k \times k}$  is rotated in a group-wise manner, obtaining  $\tilde{W}$ , which performs the convolution with input feature map  $x \in \mathbb{R}^{C_{in} \times H_{in} \times W_{in}}$ . The output  $y \in \mathbb{R}^{C_{out} \times H_{out} \times W_{out}}$  is then fed into the group-wise attention module for denoising and refining, obtaining the final output  $\tilde{y}$ .

is scaled by the corresponding learnable scale factor  $\lambda_j$ , which represents the relative importance of the different groups.

$$\tilde{W}_j = \{\lambda_j \times \text{Rotate}(w_{j,l}, \theta_j)\}, l \in \{1, 2, \dots, C_{out}/n\}. \quad (6)$$

The rotation process is executed using bilinear interpolation as described in [48]. Specifically, considering a  $k \times k$  kernel, the value at each point within this kernel can be seen as the value of a sampled point from the kernel space. During rotating, the values of the original  $k \times k$  kernel are used to span the entire kernel space using bi-linear interpolation. Subsequently, we calculate the new position of each point after rotating by  $\theta$  and obtain its value in the kernel space. The final rotated kernel is obtained by aggregating the different groups of kernels, i.e.,  $\tilde{W} = \{\tilde{W}_j\}$ ,  $j \in \{1, 2, \dots, n\}$ . The implementation of group-wise rotating is simple and efficient, accomplished through a single step of matrix multiplication to transform  $W$  into  $\tilde{W}$ .

Followed Group-wise Rotating, the rotated kernel  $\tilde{W}$  is utilized to perform convolution with the input feature map  $x$ , i.e.,  $y = \text{Conv}(x, \tilde{W})$ . The resulting output  $y \in \mathbb{R}^{C_{out} \times H_{out} \times W_{out}}$  is then passed into the Group-wise Attention module for further processing.

**Differences between Group Convolution.** Although kernels are divided into different groups in our method, it is distinct from Group Convolution. In Group Convolution, the input feature map  $x \in \mathbb{R}^{C_{in} \times H_{in} \times W_{in}}$  is partitioned into  $g$  groups along the  $C_{in}$  dimension. Each group then undergoes convolution with a corresponding set of  $C_{out}/g$  kernels, each with dimensions  $\mathbb{R}^{C_{in}/g \times k \times k}$ . In contrast, our grouping strategy is employed to rotate the  $C_{out}$  kernels. After the rotation, regular convolution is applied to obtain the output.



### 3.4 Group-wise Attention

Following the convolution operation between  $\mathbf{x}$  and  $\widetilde{\mathbf{W}}$ , the resulting feature  $\mathbf{y}$  naturally comprises  $n$  groups, i.e.  $\mathbf{y} = \{\mathbf{y}_j\}, j \in \{1, 2, \dots, n\}$ . These feature groups  $\mathbf{y}_j \in \mathbb{R}^{C_{\text{out}}/n \times H_{\text{out}} \times W_{\text{out}}}$  predominantly capture the relevant characteristics of objects oriented near  $\theta_j$ . However, features corresponding to objects at other angles may contain undesired noise and are sub-optimal. To mitigate this issue, we introduce a group-wise spatial attention mechanism to selectively amplify important regions within each feature group while suppressing the influence of irrelevant areas. To be specific, max and average pooling operations are conducted on each feature group  $\mathbf{y}_j$ , and their outputs are concatenated,

$$\mathbf{S}_j = \text{Concat}[\text{AvgPool}(\mathbf{y}_j), \text{MaxPool}(\mathbf{y}_j)]. \quad (7)$$

This operation highlights the most important regions within each feature group while preserving the overall representation of average characteristics. The pooled feature  $\mathbf{S}_j \in \mathbb{R}^{2 \times H_{\text{out}} \times W_{\text{out}}}$  is passed through a convolution layer  $F$  for channel adjustment. Subsequently, a Sigmoid function  $\sigma$  is applied to map all values to the range  $[0, 1]$ , yielding the attention maps:  $\tilde{\mathbf{S}}_j = \sigma(F(\mathbf{S}_j))$ ,  $\tilde{\mathbf{S}}_j \in \mathbb{R}^{1 \times H_{\text{out}} \times W_{\text{out}}}$ .

The final output feature  $\tilde{\mathbf{y}}_j$  is obtained through element-wise multiplication between  $\mathbf{y}_j$  and  $\tilde{\mathbf{S}}_j$ , i.e.,  $\tilde{\mathbf{y}}_j = \mathbf{y}_j \odot \tilde{\mathbf{S}}_j$ . This group-wise attention mechanism enhances the important regions within the feature, simultaneously attenuating less relevant areas. The resulting final output  $\tilde{\mathbf{y}} = \{\tilde{\mathbf{y}}_j\}$ , where  $j \in \{1, 2, \dots, n\}$ , is then forwarded to subsequent modules in the network for further processing.

### 3.5 Discussion

**Enhanced Detection of Fine-Grained Orientation Details.** Compared with ARC, the proposed grouping mechanism enables the angle generator to predict more angles (with  $n$  set to 16 in our experiments) from the input feature map while introducing minimal extra parameters. This strategy enables a more comprehensive capture of subtle orientation nuances within the input feature map. Therefore, the leverage of the group-wise rotated kernel for convolution enhances the learning of object features across diverse orientations. Moreover, the Group-wise Attention further refines these features, enhancing the precision and quality of the extracted information.

**Relationship between Group-wise Rotating and Group-wise Attention.** The synergy between Group-wise Rotating and Group-wise Attention within the GRA module is essential and irreplaceable. The Group-wise Rotating mechanism, by dynamically adjusting the convolutional kernel to different orientations, enables the network to effectively capture object features across a range of angles. However, as discussed in Section 3.2, this process inadvertently introduces undesired noise into the features. To address this issue, Group-wise Attention is ingeniously employed to adaptively filter out undesirable features while reinforcing the relevant regions of the feature map. Under the joint effect of these two components, our method effectively captures the satisfying features of objects with various orientations within a single image.

## 4 Experiment

### 4.1 Experiment Setup

**Datasets.** We evaluate our methods on three widely-used datasets in oriented object detection, i.e., DOTA-v1.0 [61], DOTA-v2.0 [6] and HRSC2016 [40]. DOTA-v1.0 [61] contains 2806 aerial images and 188,282 objects across various categories, each annotated with precisely oriented bounding boxes. It includes 15 object classes. DOTA-v2.0 [6] contains 11,268 images and 1,793,658 objects. Besides the 15 classes in the DOTA-v1.0 dataset, the DOTA-v2.0 dataset contains 3 additional classes. For these two datasets, we use the official training set and validation set for training. The testing is conducted on the test set. The mean average precision (mAP) on the test set is obtained by submitting the prediction of the model to the official DOTA dataset evaluation server.

HRSC2016 [40] is also a widely-used benchmark for arbitrary-oriented object detection. There are 1061 images with sizes ranging from  $300 \times 300$  to  $1500 \times 900$ . In the experiment, both the training set (436 images) and the validation set (181 images) are utilized for training. The test set is used for testing. We report the COCO [39] style mean average precision (mAP) on the test set to illustrate the effectiveness of our model. During data pre-processing, we maintain the original aspect ratios of images.

**Implementation Details.** GRA is a plug-and-play module and theoretically can be inserted into any convolution network. In the experiment, we replace  $3 \times 3$  convolutions in the last three stages of ResNet, preserving  $1 \times 1$  convolutions due to their inherent rotational invariance. Unless specifically stated, the backbone is pretrained for 100 epochs on ImageNet before training on the aforementioned dataset for oriented object detection.

For the DOTA dataset, the total training epoch is set to 12 unless specifically stated. For HRSC2016 dataset, Rotated RetinaNet [38] is trained for 72 epochs, both S<sup>2</sup>ANet [12] and Oriented R-CNN [66] are trained for 36 epochs. We scale down the learning rate of the backbone during training, making the training procedure of GRA more stable. Most of the experiments in our work are implemented with MMRotate framework [84] except the experiments on Oriented R-CNN, which are implemented with the OBBDetection framework [66].

### 4.2 Main Results

**Effectiveness on various detectors.** We conduct experiments on a variety of popular detectors, including both single-stage methods (Rotated RetinaNet [38], R3Det [72], S<sup>2</sup>ANet [12]) and two-stage approaches (Rotated FasterR-CNN [51], CFA [11], Oriented R-CNN [66]). The results on the DOTA-v1.0 dataset are shown in Tab. 1. In the backbone column, R50 stands for ResNet-50 [24], R50<sub>ARC</sub> is the backbone network that replaces the  $3 \times 3$  convolution in the last three stages of ResNet-50 with the ARC [48] module and R50<sub>GRA</sub> is the backbone network that replaces the  $3 \times 3$  convolution in the last three stages of ResNet-50 with GRA module. Experimental results illustrate the effectiveness and compatibility

**Table 1: GRA outperforms the SOTA method ARC [48] on the DOTA-v1.0 dataset among various oriented object detectors.**

Method	Backbone	Params(M)			mAP(%)
		Backbone	Head	Total	
Rotated RetinaNet [38]	R50	23.51	13.13	36.64	68.42
	R50 <sub>ARC</sub>	57.20	13.13	70.33	71.45
	<b>R50<sub>GRA</sub></b>	23.79	13.13	<b>36.92</b> ( $\downarrow 46\%$ )	<b>72.19</b>
R3Det [72]	R50	23.51	18.62	42.13	69.70
	R50 <sub>ARC</sub>	57.20	18.62	75.82	72.32
	<b>R50<sub>GRA</sub></b>	23.79	18.62	<b>42.41</b> ( $\downarrow 43\%$ )	<b>72.93</b>
S <sup>2</sup> ANet [12]	R50	23.51	15.32	38.83	74.13
	R50 <sub>ARC</sub>	57.20	15.32	72.52	75.49
	<b>R50<sub>GRA</sub></b>	23.79	15.32	<b>39.11</b> ( $\downarrow 45\%$ )	<b>75.98</b>
Rotated Faster RCNN [51]	R50	23.51	17.86	41.37	73.17
	R50 <sub>ARC</sub>	57.20	17.86	75.06	74.77
	<b>R50<sub>GRA</sub></b>	23.79	17.86	<b>41.65</b> ( $\downarrow 43\%$ )	<b>75.22</b>
CFA [11]	R50	23.51	13.33	36.84	69.37
	R50 <sub>ARC</sub>	57.20	13.33	70.53	73.53
	<b>R50<sub>GRA</sub></b>	23.79	13.33	<b>37.12</b> ( $\downarrow 46\%$ )	<b>73.97</b>
Oriented R-CNN [66]	R50	23.51	17.86	41.37	75.81
	R50 <sub>ARC</sub>	57.20	17.86	75.06	77.35
	<b>R50<sub>GRA</sub></b>	23.79	17.86	<b>41.65</b> ( $\downarrow 43\%$ )	<b>77.63</b>

of the proposed method on various frameworks. At the same time, the number of parameters is significantly reduced compared with ARC [48]. Tab. 2 further shows the robustness of our method on the HRSC dataset, evidencing the mAP improvement of 1.27% for Rotated RetinaNet [38], 2.32% for S<sup>2</sup>ANet [12], and 1.90% for Oriented R-CNN [66] compared with the regular ResNet50. On HRSC dataset, our model can also outperform various previous SOTA methods.

**Performance under multi-scale training and testing strategies.** We also assess the performance of GRA under multi-scale training and testing strategies on the DOTA-v1.0 dataset, which is shown in Tab. 3. Under this setting, various data augmentation methods are applied, which is more suitable for larger models to avoid under-fitting during training. On the other hand, our lightweight GRA still illustrates highly competitive performance.

**Performance on DOTA-v2.0 dataset.** Compared with the DOTA-v1.0 dataset, the DOTA-v2.0 dataset contains much more tiny objects, which are more challenging to detect. GRA also illustrates its strong performance as shown in Tab. 4. Using Oriented R-CNN as the detector and training for 40 epochs, we achieve the SOTA mAP of 57.95%, which outperforms all previous methods.

**Performance under different pretraining strategy.** Training a detector typically involves an extensive pre-training phase for its backbone on the ImageNet dataset, followed by further training on task-specific datasets such as DOTA for oriented object detection [53]. Previous methods such as ARC [48] and LSKNet [36] are limited to a from-scratch pre-training approach for the

**Table 2: Results on HRSC2016.** The proposed group-wise rotating and attention mechanism is also effective in some widely used oriented object detectors.

Method	Backbone	Params(M)	AP <sub>50</sub> (%)	AP <sub>75</sub> (%)	mAP(%)
Rotated RetinaNet [38]	R50	36.64	84.20	58.50	52.70
	R50 <sub>ARC</sub>	70.33	85.10	60.20	53.97
	<b>R50<sub>GRA</sub></b>	<b>36.92</b>	<b>85.20</b>	<b>60.34</b>	<b>54.08</b>
S <sup>2</sup> ANet [12]	R50	38.83	89.70	65.30	55.65
	R50 <sub>ARC</sub>	72.52	90.00	67.40	57.77
	<b>R50<sub>GRA</sub></b>	<b>39.11</b>	<b>90.04</b>	<b>68.40</b>	<b>58.03</b>
Oriented R-CNN [66]	R50	41.37	90.40	88.81	70.55
	R50 <sub>ARC</sub>	75.06	90.41	89.02	72.39
	<b>R50<sub>GRA</sub></b>	<b>41.65</b>	<b>90.48</b>	<b>89.23</b>	<b>72.59</b>

backbone, which consumes considerable computational resources and time. In contrast, our GRA framework simply adds several additional modules onto existing foundational models (such as the standard ResNet), without altering their internal structures. This characteristic allows our method to leverage pre-trained weights of standard models that are publicly available, eliminating the need for resource-intensive training from scratch. During the pre-training phase, we load the weights of a standard pre-trained ResNet model, focusing solely on training the newly integrated GRA modules. Meanwhile, other parts of the backbone (i.e., the standard ResNet components) remain frozen. Experimental results (Tab. 5) demonstrate that directly training a randomly initialized model on the DOTA dataset leads to extremely poor performance (35.51% mAP). However, by loading the publicly available pre-trained weights of ResNet and training only the GRA modules during the pre-training phase, the final trained detector still achieves satisfactory performance (77.39% mAP). We believe that the flexibility of GRA, further underscores its applicability to larger models, enhancing its utility and adaptability in the field of oriented object detection.

### 4.3 Ablation Study

**Number of groups.** An ablation study was conducted to assess the impact of varying the number of groups within the GRA module (Tab. 6). Within a certain range, increasing the number of groups results in the angle generator of the Group-wise Rotating module predicting more angles. Consequently, this enables the network to capture finer details of rotating objects with diverse orientations. Besides, increasing the number of groups will not lead to a significant rise in parameters and FLOPs.

Furthermore, it is observed that with a continuous increase in  $n$ , there is no significant improvement in performance. This phenomenon can be attributed to the situation where, as  $n$  becomes excessively large, each group contains only a few kernels. In such cases, if the differences among the predicted angles for these  $n$  groups are significant, the limited number of kernels within each group may fail to adequately capture the features corresponding to the rotated angle.

**Table 3: Experiment results on the DOTA dataset under multi-scale training and testing.** Although our method cut down the parameters significantly, it still outperforms a number of SOTA methods.

Method	Backbone	Params (M)	mAP (%)
R3Det [72]	R152	76.76	76.47
SASM [26]	RX101	58.01	79.17
S <sup>2</sup> ANet [12]	R50	38.83	79.42
ReDet [13]	ReR50	55.82	80.10
R3Det <sub>GWD</sub> [74]	R152	76.76	80.19
RTMDet [42]	R50	52.33	80.54
R3Det <sub>KLD</sub> [76]	R152	76.76	80.63
AOPG [3]	R50	41.95	80.66
KFIoU [78]	Swin-T	60.52	80.93
RVSA [54]	ViTAE-B	114.37	81.24
YOLO-v8 [33]	YOLOv8x	69.5	81.36
Oriented R-CNN [66]	R50	41.37	80.62
	R50 <sub>ARC</sub>	75.06	81.77
	<b>R50<sub>GRA</sub></b>	<b>41.65</b>	<b>81.93</b>

**Table 4: Experiment results on the DOTA-v2.0 dataset.** Our method illustrates its strong performance in detecting tiny objects, achieving SOTA results on the DOTA-v2.0 dataset. † denotes training for 40 epochs.

Method	Backbone	mAP(%)
SASM [26]	R50	44.53
R3Det [72]	R50	47.26
FR-OBb [51]	R50	47.32
FCOS-O [4]	R50	48.51
Ori-Rep [35]	R50	48.95
ATSS-O [82]	R50	49.57
S <sup>2</sup> ANet [12]	R50	49.86
DCFL [68]	R50	51.57
	R50†	55.08
	ReR101†	57.66
Oriented R-CNN [66]	R50	53.28
	R50 <sub>ARC</sub>	55.91
	<b>R50<sub>GRA</sub></b>	<b>56.63</b>
	<b>R50<sub>GRA</sub> †</b>	<b>57.95</b>

**Table 5: Results under different pre-training strategies.** 🔥 denotes training from scratch. ⚡ denotes loading the public available pre-trained weight and freezing the parameters during training. The Oriented-RCNN is used as the detector, other experiment settings follow Tab. 1.

Pretraining Strategy	No Pretraining	ResNet 🔥 GRA module 🔥	ResNet ⚡ GRA module 🔥
mAP(%)	35.51	77.64	77.39

Consequently, the network might predict similar angles for various groups to ensure that the features associated with the particular angle maintain satisfactory quality. This behavior is equivalent to merging several groups into one.

**Each component of GRA.** We also conduct an ablation study to illustrate the effectiveness of different components of GRA (Tab. 7). Group-wise Rotating module and Group-wise Attention module are both of great importance. To be noticed, in the Group-wise Rotating module, each group of kernel is weighted by a learnable scale factor  $\lambda_i$ , which can measure the importance of different groups. We find that this design can help the model to achieve better performance.

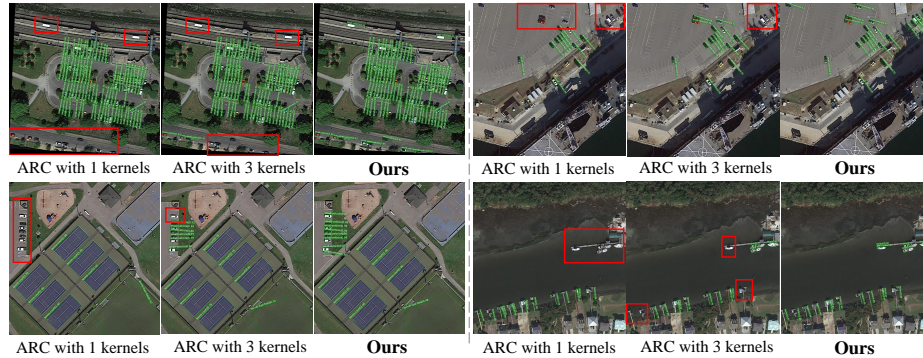
**Visualization.** We provide visualization results to illustrate the effectiveness of our method, which are obtained using the Oriented R-CNN detector on the DOTA-v2.0 dataset under single-scale training. Besides the miss detected object of baseline methods marked in **red boxes**, many detected objects detected by ARC with 3 kernels exhibit lower confidence (best-viewed zoom-in).

**Table 6: Ablation studies on the number of groups  $n$ .** The Oriented-RCNN is used as the detector, other experiment settings follow Tab. 1.

Backbone	$n$	Params (M)	FLOPs (G)	mAP (%)
R50	-	23.51	171.5	75.81
R50 <sub>GRA</sub>	2	23.56	172.4	76.82
R50 <sub>GRA</sub>	8	23.61	172.4	77.27
R50 <sub>GRA</sub>	16	23.67	172.5	77.41
R50 <sub>GRA</sub>	32	23.79	172.7	<b>77.63</b>

**Table 7: Ablation studies on different parts of GRA module.** The Oriented-RCNN is used as the detector, other experiment settings follow Tab. 1.

Group-wise Rotating		Group-wise Attention	mAP (%)
Rotating	Weighted by $\lambda$		
✗	✗	✗	75.81
✓	✗	✗	76.73
✓	✓	✗	77.25
✓	✓	✓	<b>77.63</b>



**Fig. 5: The visualisation of the prediction of ARC [48] and our model.**

**Limitations and future work.** Although GRA is simple and outperforms a large number of methods, it is only used to replace the  $3 \times 3$  convolution kernels in ResNet in our work. Its effectiveness on larger kernels (e.g.,  $7 \times 7$ ) or other convolution-based models (e.g., ConvNeXt) still needs to be explored.

## 5 Conclusion

In this paper, we introduce a lightweight yet effective module named Group-wise Rotating and Attention (GRA), which can adaptively capture the fine-grained features of objects with various orientations. Within the Group-wise Rotating module, convolution kernels are segregated into distinct groups, each rotating independently at various angles to dynamically capture the features of oriented objects. The group-wise attention mechanism is introduced to adaptively focus on important regions in the feature, reducing the undesired noises contained in the feature and enhancing the learning of fine-grained features of oriented objects. Comprehensive experimental results across multiple datasets demonstrate that our approach outperforms various state-of-the-art (SOTA) methods. Furthermore, GRA introduces minimal additional parameters, enhancing its versatility and potential for real-world deployment.

## Acknowledgments

This work was partly supported by Shenzhen Key Laboratory of next generation interactive media innovative technology (No: ZDSYS20210623092001004).

## References

1. Chen, Z., Chen, K., Lin, W., See, J., Yu, H., Ke, Y., Yang, C.: Piou loss: Towards accurate oriented object detection in complex environments. In: ECCV (2020) [2](#)
2. Cheng, B., Choudhuri, A., Misra, I., Kirillov, A., Girdhar, R., Schwing, A.G.: Mask2former for video instance segmentation. arXiv:2112.10764 (2021) [2](#)
3. Cheng, G., Wang, J., Li, K., Xie, X., Lang, C., Yao, Y., Han, J.: Anchor-free oriented proposal generator for object detection. TGARS (2022) [3](#), [13](#)
4. Detector, A.F.O.: Fcos: A simple and strong anchor-free object detector. TPAMI **44**(4) (2022) [13](#)
5. Ding, J., Xue, N., Long, Y., Xia, G.S., Lu, Q.: Learning roi transformer for oriented object detection in aerial images. In: CVPR (2019) [3](#)
6. Ding, J., Xue, N., Xia, G.S., Bai, X., Yang, W., Yang, M.Y., Belongie, S., Luo, J., Datcu, M., Pelillo, M., et al.: Object detection in aerial images: A large-scale benchmark and challenges. TPAMI **44**(11) (2021) [3](#), [10](#)
7. Guo, J., Du, C., Wang, J., Huang, H., Wan, P., Huang, G.: Assessing a single image in reference-guided image synthesis. In: AAAI (2022) [2](#)
8. Guo, J., Manukyan, H., Yang, C., Wang, C., Khachatryan, L., Navasardyan, S., Song, S., Shi, H., Huang, G.: Faceclip: Facial image-to-video translation via a brief text description. TCSVT (2023) [2](#)
9. Guo, J., Wang, C., Wu, Y., Zhang, E., Wang, K., Xu, X., Shi, H., Huang, G., Song, S.: Zero-shot generative model adaptation via image-specific prompt learning. In: CVPR (2023) [4](#)
10. Guo, J., Xu, X., Pu, Y., Ni, Z., Wang, C., Vasu, M., Song, S., Huang, G., Shi, H.: Smooth diffusion: Crafting smooth latent spaces in diffusion models. In: CVPR (2024) [4](#)
11. Guo, Z., Liu, C., Zhang, X., Jiao, J., Ji, X., Ye, Q.: Beyond bounding-box: Convex-hull feature adaptation for oriented and densely packed object detection. In: CVPR (2021) [2](#), [3](#), [10](#), [11](#)
12. Han, J., Ding, J., Li, J., Xia, G.S.: Align deep features for oriented object detection. TGARS (2021) [2](#), [3](#), [10](#), [11](#), [12](#), [13](#)
13. Han, J., Ding, J., Xue, N., Xia, G.S.: Redet: A rotation-equivariant detector for aerial object detection. In: CVPR (2021) [2](#), [4](#), [13](#)
14. Han, Y., Han, D., Liu, Z., Wang, Y., Pan, X., Pu, Y., Deng, C., Feng, J., Song, S., Huang, G.: Dynamic perceiver for efficient visual recognition. In: ICCV (2023) [4](#)
15. Han, Y., Huang, G., Song, S., Yang, L., Wang, H., Wang, Y.: Dynamic neural networks: A survey. TPAMI (2021) [4](#)
16. Han, Y., Huang, G., Song, S., Yang, L., Zhang, Y., Jiang, H.: Spatially adaptive feature refinement for efficient inference. TIP (2021) [4](#)
17. Han, Y., Liu, Z., Yuan, Z., Pu, Y., Wang, C., Song, S., Huang, G.: Latency-aware unified dynamic networks for efficient image recognition. TPAMI (2024) [4](#)
18. Han, Y., Pu, Y., Lai, Z., Wang, C., Song, S., Cao, J., Huang, W., Deng, C., Huang, G.: Learning to weight samples for dynamic early-exiting networks. In: ECCV (2022) [4](#)



19. Han, Y., Yuan, Z., Pu, Y., Xue, C., Song, S., Sun, G., Huang, G.: Latency-aware spatial-wise dynamic networks. In: NeurIPS (2022) [4](#)
20. Hansen, C., Hansen, C., Alstrup, S., Simonsen, J.G., Lioma, C.: Neural speed reading with structural-jump-lstm. In: ICLR (2019) [4](#)
21. He, C., Li, K., Zhang, Y., Tang, L., Zhang, Y., Guo, Z., Li, X.: Camouflaged object detection with feature decomposition and edge reconstruction. In: CVPR (2023) [3](#)
22. He, C., Li, K., Zhang, Y., Xu, G., Tang, L., Zhang, Y., Guo, Z., Li, X.: Weakly-supervised concealed object segmentation with sam-based pseudo labeling and multi-scale feature grouping. NeurIPS **36** (2024) [3](#)
23. He, C., Li, K., Zhang, Y., Zhang, Y., Guo, Z., Li, X., Danelljan, M., Yu, F.: Strategic preys make acute predators: Enhancing camouflaged object detectors by generating camouflaged objects (2024) [3](#)
24. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016) [10](#)
25. Heitz, G., Koller, D.: Learning spatial context: Using stuff to find things. In: ECCV (2008) [2](#)
26. Hou, L., Lu, K., Xue, J., Li, Y.: Shape-adaptive selection and measurement for oriented object detection. In: AAAI (2022) [2](#), [3](#), [13](#)
27. Hou, L., Lu, K., Yang, X., Li, Y., Xue, J.: G-rep: Gaussian representation for arbitrary-oriented object detection. arXiv:2205.11796 (2022) [2](#), [3](#)
28. Hu, Y., Yang, J., Chen, L., Li, K., Sima, C., Zhu, X., Chai, S., Du, S., Lin, T., Wang, W., et al.: Planning-oriented autonomous driving. In: CVPR (2023) [2](#)
29. Hu, Z., Sun, Y., Wang, J., Yang, Y.: Dac-detr: Divide the attention layers and conquer. In: NeurIPS (2023) [4](#)
30. Huang, G., Chen, D., Li, T., Wu, F., Van Der Maaten, L., Weinberger, K.Q.: Multi-scale dense networks for resource efficient image classification. In: ICLR (2018) [4](#)
31. Huang, G., Wang, Y., Lv, K., Jiang, H., Huang, W., Qi, P., Song, S.: Glance and focus networks for dynamic visual recognition. TPAMI (2022) [4](#)
32. Jain, V., Learned-Miller, E.: Fddb: A benchmark for face detection in unconstrained settings. Tech. rep., University of Massachusetts, Amherst (2010) [2](#)
33. Jocher, G., Chaurasia, A., Qiu, J.: Ultralytics YOLO (Jan 2023), <https://github.com/ultralytics/ultralytics> [13](#)
34. Karatzas, D., Shafait, F., Uchida, S., Iwamura, M., i Bigorda, L.G., Mestre, S.R., Mas, J., Mota, D.F., Almazan, J.A., De Las Heras, L.P.: Icdar 2013 robust reading competition. In: ICDAR (2013) [2](#)
35. Li, W., Chen, Y., Hu, K., Zhu, J.: Oriented reppoints for aerial object detection. In: CVPR (2022) [2](#), [3](#), [13](#)
36. Li, Y., Hou, Q., Zheng, Z., Cheng, M.M., Yang, J., Li, X.: Large selective kernel network for remote sensing object detection. arXiv preprint arXiv:2303.09030 (2023) [4](#), [11](#)
37. Li, Z., Wang, W., Li, H., Xie, E., Sima, C., Lu, T., Qiao, Y., Dai, J.: Bevformer: Learning bird’s-eye-view representation from multi-camera images via spatiotemporal transformers. In: ECCV (2022) [2](#)
38. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: CVPR (2017) [10](#), [11](#), [12](#)
39. 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: ECCV (2014) [10](#)
40. Liu, Z., Wang, H., Weng, L., Yang, Y.: Ship rotated bounding box space for ship extraction from high-resolution optical satellite images with complex backgrounds. GRSL (2016) [3](#), [10](#)



41. Lv, K., Yu, M., Pu, Y., Jiang, X., Huang, G., Li, X.: Learning to estimate 3-d states of deformable linear objects from single-frame occluded point clouds. In: ICRA (2023) [2](#)
42. Lyu, C., Zhang, W., Huang, H., Zhou, Y., Wang, Y., Liu, Y., Zhang, S., Chen, K.: RtmDET: An empirical study of designing real-time object detectors. arXiv preprint arXiv:2212.07784 (2022) [13](#)
43. Ma, Y., He, Y., Cun, X., Wang, X., Shan, Y., Li, X., Chen, Q.: Follow your pose: Pose-guided text-to-video generation using pose-free videos. arXiv preprint arXiv:2304.01186 (2023) [2](#)
44. Ming, Q., Miao, L., Zhou, Z., Song, J., Yang, X.: Sparse label assignment for oriented object detection in aerial images. Remote Sensing (2021) [2](#), [3](#)
45. Ming, Q., Zhou, Z., Miao, L., Zhang, H., Li, L.: Dynamic anchor learning for arbitrary-oriented object detection. In: AAAI (2021) [2](#), [3](#)
46. Pu, Y., Han, Y., Wang, Y., Feng, J., Deng, C., Huang, G.: Fine-grained recognition with learnable semantic data augmentation. TIP (2023) [4](#)
47. Pu, Y., Liang, W., Hao, Y., Yuan, Y., Yang, Y., Zhang, C., Hu, H., Huang, G.: Rank-detr for high quality object detection. In: NeurIPS (2023) [4](#)
48. Pu, Y., Wang, Y., Xia, Z., Han, Y., Wang, Y., Gan, W., Wang, Z., Song, S., Huang, G.: Adaptive rotated convolution for rotated object detection. In: ICCV (2023) [2](#), [3](#), [4](#), [5](#), [8](#), [10](#), [11](#), [14](#)
49. Pu, Y., Xia, Z., Guo, J., Han, D., Li, Q., Li, D., , Yuan, Y., Li, J., Han, Y., Song, S., Huang, G., Li, X.: Efficient diffusion transformer with step-wise dynamic attention mediators. In: ECCV (2024) [4](#)
50. Qian, W., Yang, X., Peng, S., Yan, J., Guo, Y.: Learning modulated loss for rotated object detection. In: AAAI (2021) [2](#)
51. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: NeurIPS (2015) [10](#), [11](#), [13](#)
52. Shen, Y., Geng, Z., Yuan, Y., Lin, Y., Liu, Z., Wang, C., Hu, H., Zheng, N., Guo, B.: V-detr: Detr with vertex relative position encoding for 3d object detection. In: ICLR (2024) [4](#)
53. Tang, L., Tian, Z., Li, K., He, C., Zhou, H., Zhao, H., Li, X., Jia, J.: Mind the interference: Retaining pre-trained knowledge in parameter efficient continual learning of vision-language models. arXiv preprint arXiv:2407.05342 (2024) [11](#)
54. Wang, D., Zhang, Q., Xu, Y., Zhang, J., Du, B., Tao, D., Zhang, L.: Advancing plain vision transformer towards remote sensing foundation model. TGARS (2022) [13](#)
55. Wang, J., Ma, Y., Guo, J., Xiao, Y., Huang, G., Li, X.: Cove: Unleashing the diffusion feature correspondence for consistent video editing. arXiv preprint arXiv:2406.08850 (2024) [2](#)
56. Wang, Y., Chen, Z., Jiang, H., Song, S., Han, Y., Huang, G.: Adaptive focus for efficient video recognition. In: ICCV (2021) [4](#)
57. Wang, Y., Guo, J., Wang, J., Wu, C., Song, S., Huang, G.: Erratum to meta-semi: A meta-learning approach for semi-supervised learning. CAAI Artificial Intelligence Research **2** (2023) [4](#)
58. Wang, Y., Han, Y., Wang, C., Song, S., Tian, Q., Huang, G.: Computation-efficient deep learning for computer vision: A survey. arXiv:2308.13998 (2023) [4](#)
59. Wang, Y., Huang, R., Song, S., Huang, Z., Huang, G.: Not all images are worth 16x16 words: Dynamic transformers for efficient image recognition. In: NeurIPS (2021) [4](#)
60. Woo, S., Park, J., Lee, J.Y., Kweon, I.S.: Cbam: Convolutional block attention module. In: ECCV (2018) [4](#)

61. Xia, G.S., Bai, X., Ding, J., Zhu, Z., Belongie, S., Luo, J., Datcu, M., Pelillo, M., Zhang, L.: Dota: A large-scale dataset for object detection in aerial images. In: CVPR (2018) [2](#), [3](#), [10](#)
62. Xia, Z., Pan, X., Song, S., Li, L.E., Huang, G.: Dat++: Spatially dynamic vision transformer with deformable attention. arXiv preprint arXiv:2309.01430 (2023) [4](#)
63. Xiao, J., Li, L., Lv, H., Wang, S., Huang, Q.: R&b: Region and boundary aware zero-shot grounded text-to-image generation. In: ICLR (2024) [2](#)
64. Xiao, Y., Luo, Z., Liu, Y., Ma, Y., Bian, H., Ji, Y., Yang, Y., Li, X.: Bridging the gap: A unified video comprehension framework for moment retrieval and highlight detection. CVPR (2024) [3](#)
65. Xiao, Y., Song, L., Huang, S., Wang, J., Song, S., Ge, Y., Li, X., Shan, Y.: Grootvl: Tree topology is all you need in state space model. arXiv preprint arXiv:2406.02395 (2024) [4](#)
66. Xie, X., Cheng, G., Wang, J., Yao, X., Han, J.: Oriented r-cnn for object detection. In: ICCV (2021) [3](#), [5](#), [10](#), [11](#), [12](#), [13](#)
67. Xie, Y., Lu, M., Peng, R., Lu, P.: Learning agile flights through narrow gaps with varying angles using onboard sensing. RAL (2023) [2](#)
68. Xu, C., Ding, J., Wang, J., Yang, W., Yu, H., Yu, L., Xia, G.S.: Dynamic coarse-to-fine learning for oriented tiny object detection. In: CVPR (2023) [2](#), [13](#)
69. Xu, Y., Fu, M., Wang, Q., Wang, Y., Chen, K., Xia, G.S., Bai, X.: Gliding vertex on the horizontal bounding box for multi-oriented object detection. TPAMI (2020) [2](#), [3](#)
70. Yan, X., Chen, C., Li, X.: Adaptive vision-based control of redundant robots with null-space interaction for human-robot collaboration. In: ICRA (2022) [2](#)
71. Yang, S., Luo, P., Loy, C.C., Tang, X.: Wider face: A face detection benchmark. In: CVPR (2016) [2](#)
72. Yang, X., Yan, J., Feng, Z., He, T.: R3det: Refined single-stage detector with feature refinement for rotating object. In: AAAI (2021) [2](#), [3](#), [10](#), [11](#), [13](#)
73. Yang, X., Yan, J., Liao, W., Yang, X., Tang, J., He, T.: Srdet++: Detecting small, cluttered and rotated objects via instance-level feature denoising and rotation loss smoothing. TPAMI (2022) [2](#), [3](#)
74. Yang, X., Yan, J., Ming, Q., Wang, W., Zhang, X., Tian, Q.: Rethinking rotated object detection with gaussian wasserstein distance loss. In: ICML (2021) [2](#), [3](#), [13](#)
75. Yang, X., Yang, J., Yan, J., Zhang, Y., Zhang, T., Guo, Z., Sun, X., Fu, K.: Srdet: Towards more robust detection for small, cluttered and rotated objects. In: ICCV (2019) [2](#), [3](#)
76. Yang, X., Yang, X., Yang, J., Ming, Q., Wang, W., Tian, Q., Yan, J.: Learning high-precision bounding box for rotated object detection via kullback-leibler divergence. In: NeurIPS (2021) [2](#), [3](#), [13](#)
77. Yang, X., Zhang, G., Yang, X., Zhou, Y., Wang, W., Tang, J., He, T., Yan, J.: Detecting rotated objects as gaussian distributions and its 3-d generalization. TPAMI (2022) [2](#)
78. Yang, X., Zhou, Y., Zhang, G., Yang, J., Wang, W., Yan, J., Zhang, X., Tian, Q.: The kfiou loss for rotated object detection. In: ICLR (2023) [2](#), [3](#), [13](#)
79. Yao, C., Bai, X., Liu, W., Ma, Y., Tu, Z.: Detecting texts of arbitrary orientations in natural images. In: CVPR (2012) [2](#)
80. Yu, Y., Da, F.: On boundary discontinuity in angle regression based arbitrary oriented object detection. TPAMI (2024) [2](#)
81. Yuan, Y., Chen, X., Wang, J.: Object-contextual representations for semantic segmentation. In: ECCV (2020) [2](#)

- 82. Zhang, S., Chi, C., Yao, Y., Lei, Z., Li, S.Z.: Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In: CVPR (2020) [13](#)
- 83. Zhao, C., Sun, Y., Wang, W., Chen, Q., Ding, E., Yang, Y., Wang, J.: Ms-detr: Efficient detr training with mixed supervision. arXiv:2401.03989 (2024) [4](#)
- 84. Zhou, Y., Yang, X., Zhang, G., Wang, J., Liu, Y., Hou, L., Jiang, X., Liu, X., Yan, J., Lyu, C., Zhang, W., Chen, K.: Mmrotate: A rotated object detection benchmark using pytorch. In: ACM MM (2022) [10](#)