AO2-DETR: Arbitrary-Oriented Object Detection Transformer

Linhui Dai, Hong Liu, *Member, IEEE*, Hao Tang, Zhiwei Wu, and Pinhao Song

***Abstract*— Arbitrary-oriented object detection (AOOD) is a challenging task to detect objects in the wild with arbitrary orientations and cluttered arrangements. Existing approaches are mainly based on anchor-based boxes or dense points, which rely on complicated hand-designed processing steps and inductive bias, such as anchor generation, transformation, and non-maximum suppression reasoning. Recently, the emerging transformer-based approaches view object detection as a direct set prediction problem that effectively removes the need for hand-designed components and inductive biases. In this paper, we propose an Arbitrary-Oriented Object DEtection TRansformer framework, termed AO2-DETR, which comprises three dedicated components. More precisely, an oriented proposal generation mechanism is proposed to explicitly generate oriented proposals, which provides better positional priors for pooling features to modulate the cross-attention in the transformer decoder. An adaptive oriented proposal refinement module is introduced to extract rotation-invariant region features and eliminate the misalignment between region features and objects. And a rotation-aware set matching loss is used to ensure the one-to-one matching process for direct set prediction without duplicate predictions. Our method considerably simplifies the overall pipeline and presents a new AOOD paradigm. Comprehensive experiments on several challenging datasets show that our method achieves superior performance on the AOOD task.**

***Index Terms*— Oriented object detection, detection transformer, oriented proposals, feature refinement.**

I. Introduction

ARBITRARY-ORIENTED object detection (AOOD) is a recently-emerged challenging problem in computer vision, which plays an important role in the field of aerial images [1], [2], smart retail [3], and scene text [4]. Unlike generic object detection in nature images, oriented object detection has fundamental difficulties, including often distributed with arbitrary orientation, densely packed, or has highly complex backgrounds. Many recent oriented detection models have employed convolutional neural networks (CNNs) to achieve promising results. These methods could be roughly categorized into two types: anchor-based methods [1], [5], [6], [7] and anchor-free methods [2], [8]. The anchor-based methods need to design the size and preset angle of the anchors manually. For instance, Ma et al. [4] introduce a rotation region proposal network to generate rotated proposals, which places 54 anchors with different angles, scales and aspect ratios. However, abundant anchors cause redundant computation and memory load. To address this issue, RoI Transformer [9] learns spatial transformations from horizontal Region of Interests (HRoIs) to rotated RoIs (RRoIs), as shown in Fig. 1a. Oriented R-CNN [10] generates oriented proposals by directly learning midpoint offset representation. Meanwhile, R^3Det [7] as a one-stage approach generates oriented proposals directly and uses a feature refinement module to realize feature reconstruction and alignment, as shown in Fig. 1b. Nevertheless, these methods still require manual preset boxes and complex hyperparameters to achieve promising results. Therefore, several anchor-free methods [2], [8], [11], [12], [13] are proposed in the AOOD task. For example, CFA [2] models the object layout as a convex-hull, then refines the predicted convex-hull and makes it adapt to densely packed objects, which consists of two stages: convex-hull generation and adaptation. The anchor-free methods directly treat grid points in the feature map as object candidates and largely simplify the detection pipeline.

However, both rotated box candidates and point candidates have a common problem: each object will produce redundant and approximately duplicate predictions. In addition, it is necessary to carry out complicated hand-designed processing steps and inductive biases, e.g., anchor generation, anchor transformation, and non-maximum suppression (NMS) reasoning. Recently, transformer-based detectors [14], [15], [16], [17] have been promoted as being dynamic, attentive, and can directly output predictions without complicated hand-designed processing steps and inductive biases. Set-to-set encoder-decoder models have emerged as a competitive way to model generic object detection. The self-attention and cross-attention operations in transformers are designed to be permutation-invariant. They can enable adaptive receptive fields for oriented objects, making them a natural candidate for processing rotated and irregularly placed objects. Motivated by this observation, we ask the following question: can we leverage transformers to learn an oriented object detector without relying on hand-designed inductive biases?

Manuscript received 24 May 2022; revised 15 September 2022 and 18 October 2022; accepted 6 November 2022. Date of publication 17 November 2022; date of current version 5 May 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62073004 and in part by the Science and Technology Plan of Shenzhen under Grant JCYJ20200109140410340. This article was recommended by Associate Editor Y. Yang. *(Corresponding author: Hong Liu.)*

Linhui Dai, Hong Liu, and Pinhao Song are with the Key Laboratory of Machine Perception, Shenzhen Graduate School, Peking University, Beijing 100871, China (e-mail: dailinhui@pku.edu.cn; hongliu@pku.edu.cn; pinhaosong@pku.edu.cn).

Hao Tang is with the Computer Vision Laboratory, ETH Zürich, 8800 Zürich, Switzerland (e-mail: hao.tang@vision.ee.ethz.ch).

Zhiwei Wu is with the School of Software Engineering, South China University of Technology, Guangzhou, Guangdong 511400, China (e-mail: zhiwei.w@qq.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSVT.2022.3222906.

Digital Object Identifier 10.1109/TCSVT.2022.3222906

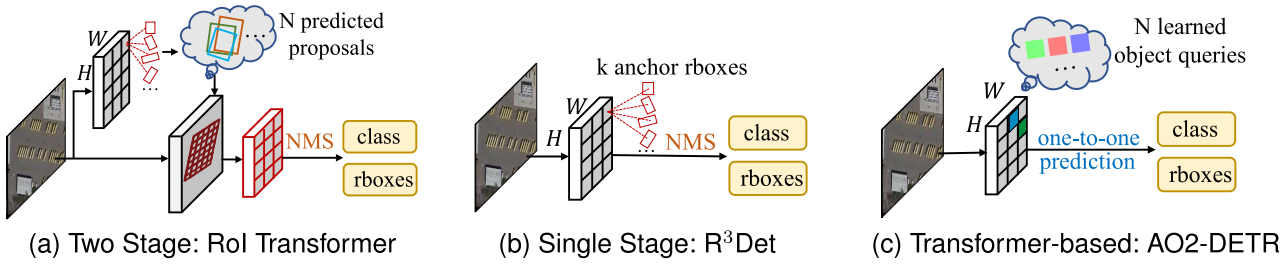


Fig. 1. Comparisons of different oriented object detection pipelines. (a) In two-stage detectors, a small set of N candidates are selected from dense object candidates by rotated region proposal networks (RRPN), and then extract image features with corresponding regions by rotated roi pooling operation, e.g. RoI Transformer [9]. (b) In single-stage detectors, $H \times W \times k$ candidates enumerate on all image grids, e.g. R^3Det [7]. (c) The proposed transformer-based oriented object detector: AO2-DETR, directly outputs the predicted oriented boxes, without prior boxes or complex pre/post-processing steps.

Intuitively, directly extending the transformer-based detectors into the AOOD task by only adding additional angle prediction values will suffer from several issues: (1) Misalignment: the learned positional embeddings in the transformer-based detectors are horizontal. When they are decoded with the feed-forward network (FFN), it will typically lead to misalignment between region features and rotated objects. (2) Cluttered features: due to the various orientation and dense distribution of objects in aerial images, the learned horizontal proposals (object queries) may contain more background areas or multiple objects, resulting in cluttered extracted features. (3) Limited matching: the regular operations in transformer-based detectors have limited generalization to rotation and scale variations. Due to the highly diverse directions of objects, it is often intractable to acquire accurate matching with all objects by using object queries with the horizontal direction.

In this paper, we aim to alleviate the above issues for the challenging transformer-based oriented detection problem. We propose a transformer-based arbitrary-oriented object detection framework called AO2-DETR (see Fig. 1c). The proposed framework has three dedicated components to address oriented object detection settings: an oriented proposal generation (OPG) mechanism, an adaptive oriented proposal refinement (OPR) module, and a rotation-aware set matching loss. Specifically, **for the problem of misalignment and cluttered features**, we explore a novel oriented proposal generation mechanism for generating oriented region proposals as object queries, which are fed into the decoder as initial rotated boxes for adaptive oriented proposal refinement. The oriented proposals present a novel query formulation and provide a better positional prior for pooling features by predicting the orientation of each bounding box in addition to center and size. Next, we present the adaptive oriented proposal refinement module to alleviate the misalignment between the axis-aligned features and the arbitrary oriented objects. The initial position information is encoded by feature interpolation, and then the learned oriented proposals are adaptively adjusted by the refinement convolutional network. Consequently, **for the problem of limited matching**, we introduce a rotation-aware set matching loss, which allows AO2-DETR to infer the oriented bounding boxes directly without prior boxes or complex pre/post-processing steps.

In conclusion, the main contributions of this paper can be summarized as follows:

- We propose a transformer-based arbitrary-oriented object detector AO2-DETR, which eliminates the need for multiple anchors and complex pre/post-processing. We hope that our method can open up the possibility of developing various paradigms for the AOOD task.
- We design an oriented proposal generation (OPG) mechanism, which guides the network to generate oriented proposals with direction information. The oriented proposals can be used to solve the misalignment problem and provide a better positional prior for pooling features to modulate the cross-attention.
- We introduce a novel adaptive oriented proposal refinement (OPR) module into the transformer architecture. The OPR module dynamically adjusts the oriented proposals according to the learned context information by a feature alignment module and a larger receptive field, which can significantly reduce the gap between the oriented proposals and the ground-truth. In addition, we add a rotation-aware set matching loss in the one-to-one matching process to ensure the correct match between the predicted boxes and ground truth.
- The extensive experiments on four public datasets: DOTA-v1.0 [18], DOTA-v1.5 [18], SKU110K-R [19], and HRSC2016 [20] demonstrate the effectiveness of the proposed model. AO2-DETR achieves the state-of-the-art performance among anchor-free and single-stage methods on these four datasets. Our code will be released at https://github.com/Ixiaohuihuihui/AO2-DETR.

II. Related Works

A. Anchors in Oriented Object Detection

Oriented object detection is a well-studied research area. The existing CNN-based oriented object detectors can be divided into two categories: anchor boxes [1], [4], [6], [7], [21] and dense points [2], [8], [11], [22], [23]. A classical solution for the AOOD task is to set rotated anchors [4], [24], such as rotated RPN [4], in which the anchors with different angles, scales, and aspect ratios are placed on each location. These densely rotated anchors lead to extensive computations and memory costs. To address this issue,

RoI-Trans [9] models the geometry transformation and solves the problem of misalignment between Region of Interests (RoIs) and objects. Oriented R-CNN [10] designs an oriented RPN to generate oriented proposals directly. Some methods are devoted to improving object representations. Yang and Yan [25] propose CSL to address the boundary problem by transforming angular prediction from a regression problem to a classification task. Gliding Vertex [26] glides the vertex of the horizontal bounding box on each corresponding side to accurately describe a multi-oriented object instead of directly regressing the four vertices. These well-designed methods have shown promising performance. However, it still produces many rotated anchors and redundant detection boxes.

Meanwhile, the keypoint-based AOOD methods have attracted extensive academic attention. These approaches generate the oriented bounding boxes by a set of keypoints belonging to the objects. DARDet [8] proposes a dense anchor-free rotated object detector and an alignment convolution module to extract aligned features. FCOSR [11] develops an ellipse center sampling method for oriented bounding boxes to define the sampling region. CFA [2] presents the convex-hull representation to learn the irregular shapes and layouts, which intend to alleviate the feature aliasing. Generally, these methods require excessive modifications to horizontal anchor-free detectors, which are prone to feature misalignment problems.

To achieve better detection accuracy, many detectors [5], [7], [10], [27], [28], [29] usually tend to design different feature refinement module. References [27], [28], and [29] are proposed for generic object detection and often lose their performance when detecting objects that are oriented and densely packed in aerial images. And the existing refinement modules [5], [7], [10] operate on the preset anchor boxes, and then classify and refine the anchor boxes.

As the hand-craft anchor boxes need to be carefully tuned to achieve good performance, while our method tends not to use the anchor boxes. Unlike the above methods, we directly predict the absolute position of each object in the image and remove the need for manual preset boxes and complicated hand-designed components.

B. Label Assignment Strategy for Oriented Object Detection

The label assignment is a core issue that mainly seeks to define positive/negative training samples independently for each ground-truth object. Anchor-based detectors [1], [30] usually adopt IoU as the assigning criterion. For instance, RPN in Faster R-CNN uses 0.7 and 0.3 as the positive and negative thresholds, respectively. This strategy introduces many hyperparameters that depend on the datasets. It means that one needs to spend much effort adjusting the hyperparameters when the dataset is changed. While the anchor-free detectors directly assign anchor points around the center of objects as positive samples or view each object as a single or a set of keypoints. GGHL [31] proposes a Gaussian OLA strategy to reflect the shape and direction of the object and refine the positive candidate locations. CFA [2] categorizes convex-hulls into positives or negatives according to the CIoU between the convex-hulls and the ground-truth boxes. The remarkable property of non-end-to-end detectors is a one-to-many positive sample assignment. During the training stage, for a ground-truth box, any samples whose confidence threshold is higher than the preset threshold are assigned as the positive samples. It always results in multiple samples in the feature maps being selected as positive samples. As a result, these detectors produce redundant predictions in the inference stage.

On the contrary, transformer-based detectors apply one-to-one assignments during the training stage [32]. Our method follows this assignment strategy. For one ground-truth box, only one sample with the minimum matching cost is assigned as the positive sample, and the others are all negative samples. The positive sample is usually selected by bipartite matching to avoid sample conflict. In order to apply the bipartite matching loss in AOOD, we introduce a rotation-aware matching loss to ensure that the entire label assignment process is one-to-one.

C. Transformer Network and Its Application

The transformer is firstly proposed for sequence transduction in [14]. The core mechanism of transformer is self-attention which makes it particularly suitable for long-range modeling information contained in all the input tokens. Recently, Carion et al. [15] present the DETR, which is the first method with an end-to-end optimization objective for set prediction. The series of related works [16], [17], [33], [34], [35] prove that transformers could achieve state-of-the-art performance in image classification and detection. Deformable DETR [16] is proposed to combine with sampling deformable points of value to the query and uses multiple level features to solve the slowly converging speed of the transformer detector. In Anchor DETR [17], the object queries are based on the anchor points, while Conditional DETR [33] encodes the reference point as the query position embedding.

Transformers are well suited for operating on the points since they are naturally permutation invariant and can enable adaptive receptive fields for oriented objects. We have been inspired to explore the encoder-decoder paradigm for the AOOD task. The self-attention is effective for global-dependency modeling, and it is likely to be valuable for rigid rotating and arbitrary placed objects. Our work is inspired by the recent Deformable DETR and anchor DETR for object detection. Different from them, the proposed AO2-DETR is an arbitrary-oriented end-to-end transformer-based detector, which can be trained from scratch and has significant design differences such as oriented proposal and adaptive refinement module. Overall, the proposed novel designs offer more flexibility with broad context modeling and fewer inductive biases for the AOOD task.

III. METHOD

A. Overview

The framework of the proposed method is shown in Fig. 2, which is mainly composed of six components: (1) a CNN backbone, (2) a deformable encoder, (3) an oriented proposal

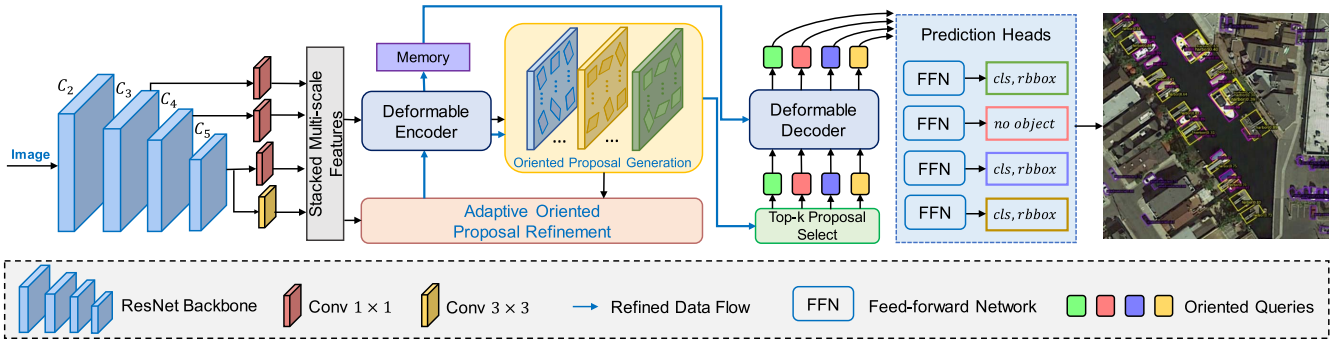


Fig. 2. Illustration of our proposed framework. AO2-DETR adapts the standard Deformable DETR for the AOOD task by introducing: (i) an oriented proposal generation mechanism to generate oriented proposals as object queries, which provides a better positional prior for pooling features. (ii) an adaptive oriented proposal refinement module to adjust the oriented proposals according to the learned context information, and (iii) a rotation-aware set matching loss to ensure the one-to-one matching process in AO2-DETR. The feed-forward network predicts either a detection (class and bounding box) or a "no object" class.

generation mechanism to generate oriented region proposals, (4) an adaptive oriented proposal refinement module to reconstruct the feature map and refine the oriented proposals adaptively, (5) a deformable decoder, (6) and a rotation-aware set matching loss to ensure the correct one-to-one matching process.

AO2-DETR takes an image as input and predicts the positions of objects in the form of oriented bounding boxes (x, y, w, h, θ) (denoted by OpenCV representation). Given an image, a CNN backbone is firstly used to extract a compact multi-scale feature map. The image features from the CNN backbone are passed through the transformer encoder, together with spatial position encoding that are added to queries and keys at every multi-scale deformable attention module.

Then, the oriented proposal generation mechanism receives the encoder memory to generate oriented region proposals. With these oriented region proposals, we can mitigate the problem of misalignment and cluttered features, thus providing a better positional prior for pooling features to modulate the cross-attention. To extract rotation-invariant region features and eliminate the misalignment between region features and oriented proposals, an adaptive oriented proposal refinement module is proposed to reconstruct feature map and refine the initial oriented proposals. The blue line in Fig. 2 denotes the refined data flow. Next, we select the top-k scores refined oriented proposals as object queries, which will be fed into the deformable decoder as object queries.

Consequently, the top-k object queries are transformed into output embeddings by the deformable decoder through multiple multi-head self-attention and multi-scale deformable attention modules. They are then independently decoded into box coordinates and class labels by an FFN, we can obtain the final set of predictions (class and bounding box) or a "no object" class. And the rotation-aware matching loss is proposed to ensure the correct one-to-one matching in the training phase.

B. Backbone and Transformer Encoder

Starting from the initial image $x_{\text{img}} \in \mathbb{R}^{3\times H_0\times W_0}$ (with 3 color channels), a conventional CNN backbone generates multi-scale feature maps $\{x^l\}_{l=1}^{L}$ $(L = 4)$ at different resolutions from the output feature maps of stage $C_3 - C_5$. The input of encoder is multi-scale feature maps $\{x^l\}_{l=1}^{L-1}$ $(L = 4)$ which are extracted from the output feature maps of stages C_3 through C_5 in ResNet (transformed by a 1×1 convolution with stride 1) and positional embeddings. The lowest resolution feature map x^L is obtained via a 3×3 convolution with stride 2 on the final C_5 stage, denoted as C_6. All the multi-scale feature maps are of 256 channels.

The transformer encoder is employed to model the discriminative contextual information among all the pixel levels. The key and query elements are pixels from the multi-scale feature maps. For each query pixel, the reference point is itself. To identify the feature level of each query pixel and the positional embedding, we add a scale-level embedding to the feature representation. Unlike the positional embedding with fixed encodings, the scale-level embeddings are randomly initialized and jointly trained with the network. Each encoder layer has a standard architecture consisting of a multi-scale deformable attention module and a fully connected FFN.

C. Oriented Proposals Generation Mechanism

In the original DETR, object queries in the decoder are irrelevant to the current image. The object queries are a set of learned embeddings. However, each learned embedding does not have an explicit physical meaning, and we can not explain where it will focus on, which is the reason for its slow training convergence. To address this issue, Deformable DETR made some improvements by generating region proposals in the first stage and then providing them into the decoder as object queries in the second stage. However, due to the highly diverse directions of objects in the aerial images, it is intractable to acquire accurate object area by using these horizontal region proposals as object queries. As a result, it usually turns out to be difficult to train a detector for extracting object features and identifying the accurate localization. To address this issue, a novel oriented proposal generation (OPG) mechanism is proposed to produce more accurate oriented proposals by learning the angle of each proposal in addition to the center

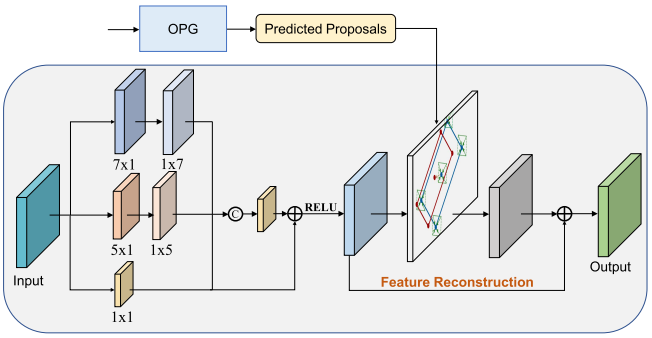
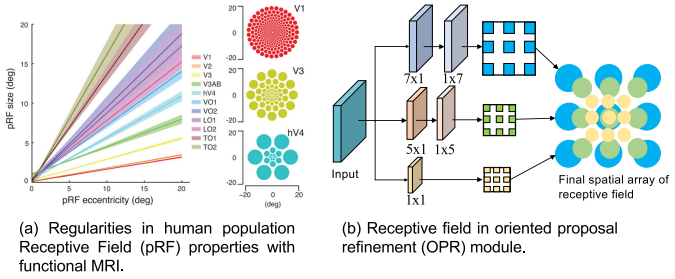


Fig. 3. Illustration of adaptive oriented proposal refinement (OPR) mechanism. It mainly contains two parts: a three-way convolutional layer and feature alignment module.



(a) Regularities in human population Receptive Field (pRF) properties with functional MRI.

(b) Receptive field in oriented proposal refinement (OPR) module.

Fig. 4. (a): Left: pRF size as a function of eccentricity in some human retinotopic maps, the pRF size increases with eccentricity in each map. Right: The spatial array of the pRFs is based on the parameters in the left panel. The radius of each circle is the apparent RF size at the appropriate eccentricity. Reproduced from [36]. (b): The final spatial array of receptive field, which is similar to the spatial array of pRF in hV4.

and size. The generated oriented proposals will be served as object queries in the deformable decoder, which provide a better positional prior for pooling features to modulate the cross-attention, as shown in Fig. 2. Specifically, let i index a pixel from feature level $l_i \in \{1, 2, \ldots, L\}$ with normalized coordinates $(p_{ix}, p_{iy}) \in [0, 1]^2$, an initial rotated box is firstly generated as $p_i = (p_{ix}, p_{iy}, p_{iw}, p_{ih}, p_{i\theta})$. Then we use the prediction results Δr_i obtained from the image features encoded by the deformable encoder and the initial rotated box p_i to obtain the final oriented proposal $\hat{b}_i$ for each pixel i. Here, $\hat{b}_{ij} = (x_{ij}, y_{ij})$ denotes the four vertices of $\hat{b}_i$, $j \in \{1, 2, 3, 4\}$, x_{ij} and y_{ij} represents the x and y coordinates of point $\hat{b}_{ij}$, respectively. Then, $\hat{b}_{ij}$ is formulated as:

$$\begin{aligned}\hat{b}_{ij} = &(\sigma\left(\sigma^{-1}\left(\phi_{x_j}\left(p_i\right)\right) + \phi_{x_j}\left(\Delta r_i\right)\right), \\ &\sigma\left(\sigma^{-1}\left(\phi_{y_j}\left(p_i\right)\right) + \phi_{y_j}\left(\Delta r_i\right)\right)), \end{aligned} \tag{1}$$

$$\begin{aligned}\phi_{x_j}(p_i) = &\, p_{ix} + \frac{1}{2}\left(\cos(p_{i\theta}) \times \cos\left(\left\lfloor \frac{j-1}{2} \right\rfloor \pi\right) \times p_{iw}\right. \\ &\left. - \sin(p_{i\theta}) \times \cos\left(\left\lceil \frac{j+1}{2} \right\rceil \pi\right) \times p_{ih}\right), \end{aligned} \tag{2}$$

$$\begin{aligned}\phi_{y_j}(p_i) = &\, p_{iy} + \frac{1}{2}\left(\sin(p_{i\theta}) \times \cos\left(\left\lfloor \frac{j-1}{2} \right\rfloor \pi\right) \times p_{iw}\right. \\ &\left. + \cos(p_{i\theta}) \times \cos\left(\left\lceil \frac{j+1}{2} \right\rceil \pi\right) \times p_{ih}\right), \end{aligned} \tag{3}$$

where p_{iw} and p_{ih} are both set as $2^{l_i - 1}s$, $s = 0.05$, $p_{i\theta}$ is set as 0. $\phi_{x_j}(p_i)$ and $\phi_{y_j}(p_i)$ represent the calculation process of the four vertices of p_i. Here, $\Delta r_{i\{x,y,w,h,\theta\}} \in \mathbb{R}$ are predicted by the bounding box regression branch. The calculation process of $\phi_{x_j}(\Delta r_i)$ and $\phi_{y_j}(\Delta r_i)$ are the same as $\phi_{x_j}(p_i)$ and $\phi_{y_j}(p_i)$, respectively. And σ and σ^{-1} denote the sigmoid and the inverse sigmoid function, respectively. The usage of σ and σ^{-1} is to ensure $\hat{b}_{ij}$ is of normalized coordinates, as $\hat{b}_{ij} \in [0, 1]$.

D. Adaptive Oriented Proposal Refinement Module

Many objects in aerial images are usually distributed with large-scale variations and arbitrary orientations. The convolution features are usually axis-aligned with the fixed receptive field, which will lead to the misalignment between the extracted convolution features and oriented objects and will affect the final detection performance. Therefore, it is crucial to extract rotation-invariant region features and eliminate the misalignment between region features and objects, especially for dense regions. We introduce the oriented proposal refinement (OPR) module to align the convolutional features and oriented proposals. The difference between the proposed OPR module and the previous methods is that the OPR module can be applied to transformer-based detectors and does not require pre-defined anchor boxes and prior knowledge related to dataset.

The structure of the OPR module is shown in Fig. 3. The inputs are the multi-scale feature map of the backbone and the initial oriented proposal of the deformable encoder. The output is a refined feature map. To effectively excavate the contextual information, we use the relevant knowledge of the human visual perception system [36], as illustrated in Fig. 4. The human visual perception cortex can highlight the importance of the region nearer to the center and elevate the insensitivity to small spatial shifts. We construct the receptive field block module by combining multiple branches with different kernels and dilated convolution layers to simulate the ratio between the size and eccentricity of the population receptive field. To be specific, the feature map is added by three-way convolution (conv 1×1, conv 5×1, and conv 7×1) to obtain a large kernel receptive fields. Here, $F \in \mathbb{R}^{C \times 1 \times 1}$ represents the feature vector of the point on the feature map. The whole process can be expressed as follows:

$$X_{out} = \tau\left(Br_1 \oplus \epsilon(Br_1 \odot Br_2 \odot Br_3)\right), \tag{4}$$

where X_{out} represents the output feature, Br_1, Br_2, and Br_3 denote the output of the three branches "conv 1×1", "conv 1×5", and "conv 1×7", respectively. Here, $\oplus$ represents the operation of feature addition, $\odot$ represents the operation of feature concatenation, ϵ denotes the process of adjusting the number of channels through 1×1 convolution, τ is the activation function of ReLU. After the above steps, the OPR module can highlight the relationship between the size and eccentricity of different receptive fields, and force the network to learn discriminative information. Then the new feature map is sent into the feature alignment module.

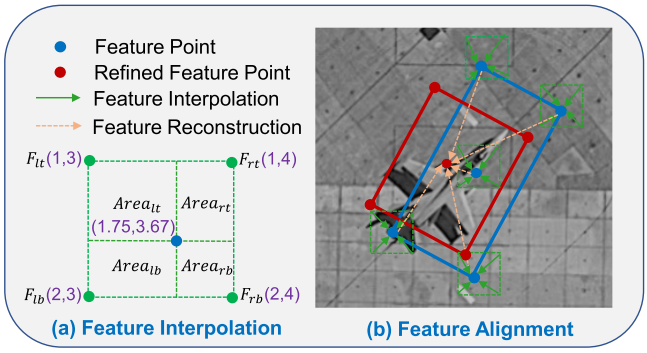


Fig. 5. Illustration of feature alignment module. We aim to re-encode the position information of the initial bounding box (blue rectangle) to the refined bounding box (red rectangle). Purple numbers are simple examples of feature points. We adopt the bilinear feature interpolation method as R^3Det [7].

The process of feature alignment is shown in Fig. 5. We extend the feature alignment module in [7] into our method. The inputs of this module are the feature map X_{out} and the oriented proposals $\hat{b}_i$ (in Equation 1). Specifically, we re-encode the position information of the initial oriented proposal (blue rectangle) to the corresponding feature points (red point), thereby reconstructing the entire feature map by a pixel-wise manner to achieve the alignment of the features. We also adopt the bilinear feature interpolation to obtain the feature information corresponding to the initial oriented proposals. The formulation of feature interpolation is as follows:

$$\begin{aligned} F &= X_{out}, \\ F' &= F_{lt} * A_{rb} + F_{rt} * A_{lb} + F_{rb} * A_{lt} + F_{lb} * A_{rt}, \end{aligned} \tag{5}$$

where $A_{\{lt,rt,lb,rb\}}$ denotes the $Area_{\{lt,rt,lb,rb\}}$ and $F_{\{lt,rt,lb,rb\}}$ denotes corresponding feature vectors on the feature map X_{out}, they are computed according to the coordinates of the oriented proposals $\hat{b}_i$, as shown in Fig. 5(a). A more accurate feature vector is obtained by bilinear interpolation. After traversing the feature points, we can obtain the reconstructed feature map F'. Then, the reconstructed feature map F' is added to the original feature map X_{out} to get the final refined feature map. The OPR module can capture the arbitrary geometric structure of an oriented proposal and its surrounding context information, which is essential for reducing the misalignment between the predicted oriented proposals and the ground truth one. The reconstructed feature map will be sent to the deformable encoder and OPG module to generate refined oriented proposals. This process is represented by the blue line in Fig. 2. Finally, the top-k scores refined oriented proposals will be selected as object queries, where the positional embeddings of object queries are set as positional embeddings of oriented proposal coordinates. Then these object queries are sent into the deformable decoder to output the final set of predictions in parallel.

E. Deformable Transformer Decoder

There are cross-attention and self-attention modules in the decoder. In the cross-attention modules, object queries extract features from the feature maps, where the key elements are of the output feature maps from the OPR module. Following [16], we only replace each cross-attention module to be the multi-scale deformable attention module. The deformable decoder is performed across multi-scale feature maps, encoding richer context over a larger receptive field and enabling the network to learn oriented receptive fields. Thus, we can enable the network to handle object detection with small objects and variable orientations. The output embeddings of the deformable decoder will then be fed into two branches: bounding box regression and classification. The classification F_{cls} is a single layer FFN, while the regression branch F_{reg} is a 3-layer FFN. Consequently, the model globally reasons about all objects and is able to use the whole image as context.

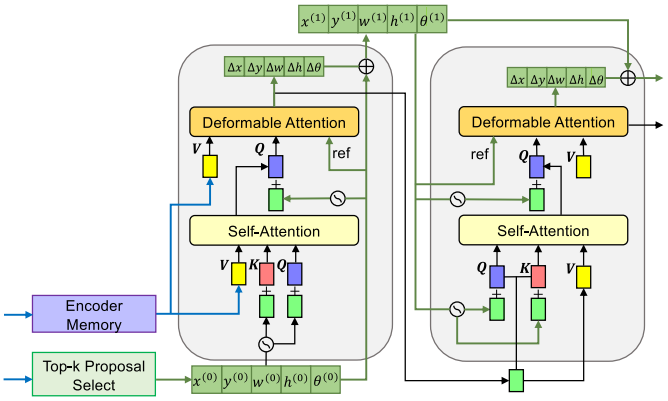


Fig. 6. Illustration of the deformable decoder of the proposed method AO2-DETR. Given the encoder memory and top-k scoring proposals, we perform the deformable cross-attention operation at each reference proposal in the decoder. The object queries are updated layer-by-layer to gradually get close to the ground-truth objects, which provides a better positional prior for pooling features to modulate the cross-attention. The blue line denotes the refined information flow of the OPR module. The green line indicates the information flow of reference proposals.

Unlike the original deformable attention module in the decoder of Deformable DETR, the multi-scale deformable attention module in the proposed method attends to a small fixed number of key sampling points around an oriented region proposal, rather than the horizontal proposal. The deformable decoder of our method is shown in Fig. 6. Let $\hat{c}_q$ be the oriented region proposal for each query element q. Then the multi-scale deformable attention module is applied as:

$$\begin{aligned} &\mathrm{MSDeformAttn}\left(z_q, \hat{\boldsymbol{c}}_q, \left\{\boldsymbol{x}^l\right\}_{l=1}^{L}\right) \\ &= \sum_{m=1}^{M} \boldsymbol{W}_m \left[\sum_{l=1}^{L}\sum_{k=1}^{K} A_{mlqk} \cdot \boldsymbol{W}'_m \boldsymbol{x}^l \left(f_l\left(\hat{\boldsymbol{c}}_q\right) + \Delta \boldsymbol{p}_{mlqk}\right)\right], \end{aligned} \tag{6}$$

where m indices the attention head and K is the total number of sampled keys, l indexes the input feature level. $f_l\left(\hat{\boldsymbol{c}}_q\right)$ rescales the normalized coordinates $\hat{\boldsymbol{c}}_q$ to the input feature map of the l-th level. Δp_{mlqk} and A_{mlqk} denote the sampling offset and attention weight of the k^{th} sampling point in the l^{th} feature level and the m^{th} attention head, respectively.

$\boldsymbol{W}_m$ and W'_m are the projection matrices for multi-head attention. The initialization process for multi-scale deformable attention is the same as the Deformable DETR. By using the refined oriented proposals, the learned decoder attention will have a strong correlation with the predicted bounding boxes, which can avoid learning messy information, including background or other objects, especially for densely placed objects, and it can also accelerate the training convergence.

The detection head predicts the relative offsets, $\hat{b}_q$ is the predicted oriented boxes, the four vertices of $\hat{b}_q$ which is calculated by:

$$\hat{b}_{qj} = \{\sigma\left(\phi_{x_j}\left(r_q\right) + \phi_{x_j}\left(\sigma^{-1}\left(\hat{c}_q\right)\right)\right), \sigma\left(\phi_{y_j}\left(r_q\right) + \phi_{y_j}\left(\sigma^{-1}\left(\hat{c}_q\right)\right)\right)\}, \tag{7}$$

where $j \in \{1, 2, 3, 4\}$, $r_q \in \mathbb{R}$ are predicted by the detection head. Then we express $\hat{b}_q$ as $\{\hat{b}_{qx}, \hat{b}_{qy}, \hat{b}_{qw}, \hat{b}_{qh}, \hat{b}_{q\theta}\}$ by the OpenCV representation of oriented bounding boxes. Crucially there is no down-sampling in spatial resolution but global context modeling at every layer of the transformer decoder, thus offering an entirely new perspective to the oriented object detection. The proposed method simultaneously predicts a set of oriented boxes with no particular ordering.

In addition, inspired by the iterative refinement developed in Deformable DETR, we also adopt a simple and effective iterative bounding box refinement mechanism to improve detection performance. Here, each decoder layer refines the oriented bounding boxes according to the predictions from the previous layer. Suppose there are D number of decoder layers (e.g., $D = 6$), given a normalized oriented box $\hat{b}_q^{d-1}$ predicted by the $(d-1)^{th}$ decoder layer, the d^{th} decoder layer refines the box as:

$$\hat{b}_{qj}^{d} = \{\sigma\left(\phi_{x_j}\left(r_q^d\right) + \phi_{x_j}\left(\sigma^{-1}\left(\hat{b}_q^{d-1}\right)\right)\right), \sigma\left(\phi_{y_j}\left(r_q^d\right) + \phi_{y_j}\left(\sigma^{-1}\left(\hat{b}_q^{d-1}\right)\right)\right)\}, \tag{8}$$

where $d \in \{1, 2, \ldots, D\}$, $\hat{b}_{qj}^d$ is the j^{th} vertex coordinates of $\hat{b}_q$ at the d-th decoder layer, $r_q^d \in \mathbb{R}$ are predicted at the d-th decoder layer. Prediction heads for different decoder layers do not share parameters. In the iterative bounding box refinement module, for the d^{th} decoder layer, we sample key elements respective to the box $\hat{b}_q^{d-1}$ predicted from the $(d-1)^{th}$ decoder layer. For Equation 6 in the cross-attention module of the d^{th} decoder layer, $\hat{b}_{q\{x,y,w,h,\theta\}}^{d-1}$ serves as the new reference oriented proposal.

F. Set Matching and Loss Function

AO2-DETR infers a fixed-sized sequence of N predictions. One of the main challenges is to score oriented predicted objects with respect to the ground truth. To obtain the box predictions, we apply a 3-layer FFN with ReLU activation function and a linear projection layer to the output embeddings of the deformable decoder. Let $\hat{y} = \left\{\hat{y}_i\right\}_{i=1}^{N}$ denote the predicted oriented boxes, and y the ground truth set of objects. Assuming N is larger than the number of objects in the image, we consider y also as a set of size N padded with $\emptyset$ (no object). In order to find a bipartite graph matching between these two sets, we search for a permutation of N elements $\sigma \in O_n$ with the lowest cost:

$$\hat{\sigma} = \arg\min_{\sigma \in O_n} \sum_{i}^{N} \mathcal{L}_{\text{match}}\left(y_i, \hat{y}_{\sigma(i)}\right), \tag{9}$$

where $\mathcal{L}_{\text{match}}\left(y_i, \hat{y}_{\sigma(i)}\right)$ is a pair-wise matching cost between ground truth y_i and a prediction with index $\sigma(i)$. The optimal assignment can be computed efficiently by the Hungarian algorithm [37].

The matching loss takes the class predictions and the similarity of predicted and ground truth boxes into account. Each element i of the ground truth set can be seen as $y_i = (c_i, b_i)$ where c_i is the target class label (which may be $\emptyset$) and $b_i \in [0, 1]^5$ is a vector that defines ground truth box center coordinates and its height, width relative to the image size and angle. The long side is height, and the angle range is [0, pi /2]. For the predictions with index $\sigma(i)$, we denote the probability of class c_i as $\hat{p}_{\sigma(i)}(c_i)$ and the predicted oriented bounding box as $\hat{b}_{\sigma(i)}$.

To ensure the correct match between the predicted oriented boxes and ground truth, we add a rotation-aware set matching loss $\mathcal{L}_{\text{riou}}$ in the one-to-one matching process. With the above notation, we define the $\mathcal{L}_{\text{match}}$ as follows:

$$\begin{aligned}\mathcal{L}_{\text{match}}(y_i, \hat{y}_{\sigma(i)}) = {} & \lambda_{\text{cls}} \cdot -\log \hat{p}_{\sigma(i)}(c_i) \\ & + \lambda_{\text{L1}} \cdot \mathcal{L}_{\text{box}}\left(b_i, \hat{b}_{\sigma}(i)\right) \\ & + \lambda_{\text{riou}} \cdot \mathcal{L}_{\text{riou}}\left(b_i, \hat{b}_{\sigma}(i)\right),\end{aligned} \tag{10}$$

where $c_i \neq \emptyset$. The second and third parts of the matching cost are used to score the bounding boxes. For $\mathcal{L}_{\text{box}}$, we use a linear combination of the SmoothL1 Loss. Here, λ_{cls} and λ_{L1} are the weights of Focal and SmoothL1 set matching loss, respectively. For the rotation-aware loss $\mathcal{L}_{riou}$, we just simply extended the rotated iou loss [38] into the Hungarian matching loss. We first compute the coordinates of four vertices $b_i = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)\}$, $\hat{b}_{\sigma(i)} = \{(x'_1, y'_1), (x'_2, y'_2), (x'_3, y'_3), (x'_4, y'_4)\}$, the computation process is formulated as:

$$\begin{aligned}\text{Area}_{b_i} &= \mathbf{a} \times \mathbf{b}, \\ \text{Area}_{\hat{b}_{\sigma(i)}} &= \mathbf{a}' \times \mathbf{b}', \\ \mathbf{a} &= \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}, \\ \mathbf{b} &= \sqrt{(x_2 - x_3)^2 + (y_2 - y_3)^2}, \\ \mathbf{a}' &= \sqrt{\left(x'_2 - x'_1\right)^2 + \left(y'_2 - y'_1\right)^2}, \\ \mathbf{b}' &= \sqrt{\left(x'_2 - x'_3\right)^2 + \left(y'_2 - y'_3\right)^2},\end{aligned} \tag{11}$$

then determine the vertices of overlap area if they have, and sort these polygon vertices in anticlockwise order to compute the intersection area $\text{Area}_{\text{overlap}} = \max\left(x_2, x'_2\right) - \min\left(x_1, x'_1\right) \times \left(\max\left(y_1, y'_1\right) - \min\left(y_2, y'_2\right)\right)$, the $\mathcal{L}_{riou}$ is computed as:

$$\text{IoU} = \frac{\text{Area}_{\text{overlap}}}{\text{Area}_{b_i} + \text{Area}_{\hat{b}_{\sigma(i)}} - \text{Area}_{\text{overlap}}},$$

$$L_{riou} = 1 - \text{IoU}, \tag{12}$$

where λ_{riou} is the weight rotated iou set matching loss. These losses are normalized by the number of objects inside the batch, which helps our model to avoid complex post-processing steps.

Loss Function: L1 loss is used as the regression loss. Rotated IoU loss [38] is used for the IoU loss computation of two rotated 2D boxes and Focal loss for the classification loss.

IV. Experiments

A. Datasets

To demonstrate the effectiveness for the proposed method, we conduct experiments on four oriented datasets, DOTA-v1.0 [18], DOTA-v1.5 [18], SKU110K-R [3], and HRSC2016 [20] datasets.

DOTA is one of the largest dataset for oriented object detection with two released versions: DOTA-v1.0 and DOTA-v1.5. **DOTA-v1.0** contains 2,806 large aerial images, and the image size ranges from around 800×800 to 4000×4000 and 188, 282 instances among 15 common categories: *Plane (PL), Baseball diamond (BD), Bridge (BR), Ground track field (GTF), Small vehicle (SV), Large vehicle (LV), Ship (SH), Tennis court (TC), Basketball court (BC), Storage tank (ST), Soccer-ball field (SBF), Roundabout (RA), Harbor (HA), Swimming pool (SP), and Helicopter (HC)*. **DOTA-v1.5** is released with a new category, *Container Crane (CC)*. It contains 402,089 annotated object instances within 16 categories.

We use both training and validation sets for training, the test set for testing. We crop the images into 1024×1024 patches with a stride of 824. The random horizontal flipping is adopted to avoid over-fitting during training and no other tricks are utilized. For fair comparisons with other methods, we adopt data augmentation at three scales $\{0.5, 1.0, 1.5\}$ and random rotation from 5 angles $\{30°, 60°, 90°, 120°, 150°\}$. The performance of the test set is evaluated on the official DOTA evaluation server.[1]

SKU110K-R is a challenging dataset for commodity detection. It is an extended version of SKU-110K [19]. The images are collected from supermarket stores around the world and include scale variations, viewing angles, lighting conditions, noise levels, and other sources of variability. The original SKU-110K dataset contains 11,762 images in total (8,233 for training, 588 for validation, and 2,941 for testing). The SKU110K-R dataset performs data augmentation by rotating the image 6 different angles $\{-45°, -30°, -15°, 15°, 30°, 45°\}$ on the original dataset. After the augmentation, the number of training, validation, and testing images are 57,533, 4,116, and 20,587, respectively. Each image contains an average of 154 tightly packed objects, up to 718 objects. The image size ranges from 1840×1840 to 4320×4320. We resize the input image to 800×800 and apply random rotation as DOTA dataset. For the SKU110K-R dataset, the results follow standard COCO-style Average Precision (AP) metrics that include AP_{75} (IoU = 0.75), mAP and AR_{300}.

HRSC2016 contains images of ships at the wharf, which are collected from six famous harbors. It only contains one category "ship". The image size ranges from 300×300 to 1500×900. The HRSC2016 dataset contains 1061 images in total (436 for training, 181 for validation, and 444 for testing). We use both training and validation sets for training and the test set for testing. Random horizontal flipping is applied during training. For the detection accuracy on the HRSC2016, we adopt the mean average precision (mAP) as evaluation criteria, which is consistent with PASCAL VOC2007 and VOC2012.

B. Implementation Details

We implement the proposed method AO2-DETR on MMRotate [43]. In all experiments, we adopt Deformable DETR [16] with ResNet-50 backbone (pre-trained on ImageNet [44]) as the baseline method. Multi-scale feature maps are extracted from $conv_3$ to $conv_5$ of ResNet-50. The transformer encoder-decoder follows the same architecture as in Deformable DETR. The number of object queries is set to 300. We train the network with AdamW [45] for 50 epochs. In the first 40 epochs, the learning rate is $1e-4$ and then $1e-5$ for another 10 epochs. The momentum and weight decay are 0.9 and 0.0001, respectively. Our method is trained on 3 GeForce RTX 3090 GPUs with a total batch size of 4 for training and a single 3090 GPU for inference. The loss weight λ_{cls}, λ_{L1} and λ_{riou} are set as 5, 5, and 8, respectively. In the inference stage, we follow the same scale setting as training. No post-processing is needed for associating objects.

C. State-of-the-Art Comparison

We compare AO2-DETR against some state-of-the-art methods on the different oriented datasets, the results are shown in Table I, II, III, and IV.

1) Results on DOTA-v1.0: Table I shows a comparison of our AO2-DETR with the recently state-of-the-art detectors on the DOTA-v1.0 dataset with respect to oriented bounding box detection. For the accuracy measured by mAP, we achieve 77.73% mAP with single-scale data and 79.22% mAP with multi-scale data. Using the same backbone of ResNet50, AO2-DETR achieves the best result among two-stage, single-stage, and anchor-free methods (e.g., Oriented R-CNN [10], R^3Det [7], KLD [39], CFA [2], SASM [42], and DARDet [8]) using a single model without bells and whistles. Specifically, AO2-DETR outperforms Oriented R-CNN by 1.86% (77.73% vs 75.87%, single-scale), R^3Det by 1.26% (77.73% vs 76.47%), KLD by 0.9% (79.22% vs 78.32%), CFA by 4.17% (79.22% vs 75.05%), SASM by 2.03% (79.22% vs 77.19%), and DARDet by 0.48% (79.22% vs 78.74%), which is a large margin.

Compared with the anchor-based methods, our method is better than most two-stage methods, except for the best two-stage method Oriented R-CNN [10] with multi-scale data. We argue that the superior performance of Oriented R-CNN comes from the oriented region proposal network

[1] https://captain-whu.github.io/DOTA

TABLE I

COMPARISONS WITH STATE-OF-THE-ART METHODS ON DOTA-V1.0 OBB TASK. ∗ INDICATES MULTI-SCALE TRAINING AND TESTING. THE RESULTS WITH RED AND BLUE COLORS INDICATE THE BEST AND SECOND-BEST RESULTS OF EACH COLUMN, RESPECTIVELY

Method	Backbone	PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SBF	RA	HA	SP	HC	mAP
Two-stage:																	
FR-O [18]	resnet101	79.09	69.12	17.17	63.49	34.20	37.16	36.20	89.19	69.60	58.96	49.40	52.52	46.69	44.80	46.30	52.93
RoI-Transformer* [9]	resnet101	88.64	78.52	43.44	75.92	68.81	73.68	83.59	90.74	77.27	81.46	58.39	53.54	62.83	58.93	47.67	69.56
SCRDet [6]*	resnet101	89.98	80.65	52.09	68.36	68.36	60.32	72.41	90.85	87.94	86.86	65.02	66.68	66.25	68.24	65.21	72.61
CSL* [25]	resnet152	90.25	85.53	54.64	75.31	70.44	73.51	77.62	90.84	86.15	86.69	69.60	68.04	73.83	71.10	68.93	76.17
Gliding Vertex* [26]	resnet101	89.64	85.00	52.26	77.34	73.01	73.14	86.82	90.74	79.02	86.81	59.55	70.91	72.94	70.86	57.32	75.02
Oriented R-CNN [10]	resnet50	89.46	82.12	54.78	70.86	78.93	83.00	88.20	90.90	87.50	84.68	63.97	67.69	74.94	68.84	52.28	75.87
Oriented R-CNN* [10]	resnet50	89.84	85.43	61.09	79.82	79.71	85.35	88.82	90.88	86.68	87.73	72.21	70.80	82.42	78.18	74.11	80.87
ReDet [1]	ReR50-ReFPN	88.79	82.64	53.97	74.00	78.13	84.06	88.04	90.89	87.78	85.75	61.76	60.39	75.96	68.07	63.59	76.25
Single-stage:																	
S^2A-Net [5]	resnet50	89.11	82.84	48.37	71.11	78.11	78.39	87.25	90.83	84.90	85.64	60.36	62.60	65.26	69.13	57.94	74.12
R^3Det [7]	resnet50	89.29	75.21	45.41	69.24	75.54	72.89	79.29	90.89	81.02	83.25	58.81	63.15	63.43	62.21	37.41	69.80
R^3Det [7]	resnet152	89.80	83.77	48.11	66.77	78.76	83.27	87.84	90.82	85.38	85.51	65.67	62.68	67.53	78.56	72.62	76.47
KLD* [39]	resnet50	88.91	85.23	53.64	81.23	78.20	76.99	84.58	89.50	86.84	86.38	71.69	68.06	75.95	72.23	75.42	78.32
DAL [40]	resnet101	88.68	76.55	45.08	66.80	67.00	76.76	79.74	90.84	79.54	78.45	57.71	62.27	69.05	73.14	60.11	71.44
Anchor-free:																	
IE-Net [12]	resnet101	80.20	64.54	39.82	32.07	49.71	65.01	52.58	81.45	44.66	78.51	46.54	56.73	64.40	64.24	36.75	57.14
Rotated RepPoints [22]	resnet50	83.36	63.71	36.27	51.58	71.06	50.35	72.42	90.10	70.22	81.98	47.46	59.50	50.65	55.51	3.07	59.15
DRN* [3]	hourglass104	89.45	83.16	48.98	62.24	70.63	74.25	83.99	90.73	84.60	85.35	55.76	60.79	71.56	68.82	63.92	72.95
DARDet [8]	resnet50	88.89	84.31	55.32	75.49	80.33	81.69	88.24	90.88	83.62	87.46	59.85	65.60	76.86	80.46	65.17	77.61
DARDet* [8]	resnet50	89.08	84.30	56.64	77.83	81.10	83.39	88.46	90.88	85.44	87.56	62.77	66.23	77.97	82.03	67.40	78.74
DAFNe* [13]	resnet101	89.40	86.27	53.70	60.51	82.04	81.17	88.66	90.37	83.81	87.27	53.93	69.38	75.61	81.26	70.86	76.95
Oriented RepPoints [41]	resnet50	87.02	83.17	54.13	71.16	80.18	78.40	87.28	90.90	85.97	86.25	59.90	70.49	73.53	72.27	58.97	75.97
SASM [42]	resnet50	86.42	78.97	52.47	69.84	77.30	75.99	86.72	90.89	82.63	85.66	60.13	68.25	73.98	72.22	62.37	74.92
SASM* [42]	resnext101	88.41	83.32	54.00	74.34	80.87	84.10	88.04	90.74	82.85	86.26	63.96	66.78	78.40	73.84	61.97	77.19
CFA [2]	resnet50	88.04	82.14	53.90	73.69	79.94	78.87	87.16	90.87	81.90	85.63	56.14	64.40	70.31	70.63	38.05	73.45
CFA [2]	resnet101	89.26	81.72	51.81	67.17	79.99	78.25	84.46	90.77	83.40	85.54	54.86	67.75	73.04	70.24	64.96	75.05
Ours:																	
AO2-DETR	resnet50	89.27	84.97	56.67	74.89	78.87	82.73	87.35	90.50	84.68	85.41	61.97	69.96	74.68	72.39	71.62	77.73
AO2-DETR*	resnet50	89.95	84.52	56.90	74.83	80.86	83.47	88.47	90.87	86.12	88.55	63.21	65.09	79.09	82.88	73.46	79.22

TABLE II

PERFORMANCE COMPARISONS ON DOTA-V1.5 TEST SET. ∗ INDICATES MULTI-SCALE TRAINING AND TESTING. THE RESULTS WITH RED AND BLUE COLORS INDICATE THE BEST AND SECOND-BEST RESULTS OF EACH COLUMN, RESPECTIVELY

Method	PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SBF	RA	HA	SP	HC	CC	mAP
RetinaNet-O [46]	71.43	77.64	42.12	64.65	44.53	56.79	73.31	90.84	76.02	59.96	46.95	69.24	59.65	64.52	48.06	0.83	59.16
FR-O [18]	71.89	77.64	44.45	59.87	51.28	68.98	79.37	90.78	77.38	67.50	47.75	69.72	61.22	65.28	60.47	1.54	62.00
Mask R-CNN-O [47]	76.84	73.51	49.90	57.80	51.31	71.34	79.75	90.46	74.21	66.07	46.21	70.61	63.07	64.46	57.81	9.42	62.67
HTC-O [48]	77.80	73.67	51.40	63.99	51.54	73.31	80.31	90.48	75.12	67.34	48.51	70.63	64.84	64.48	55.87	5.15	63.40
Ours:																	
AO2-DETR	79.55	78.14	42.41	61.23	55.34	74.50	79.57	90.64	74.76	77.58	53.56	66.91	58.56	73.11	69.64	24.71	66.26
AO2-DETR*	87.13	85.43	65.87	74.69	77.46	84.13	86.19	90.23	81.14	86.56	56.04	70.48	75.47	78.30	72.66	42.62	75.89

TABLE III

EVALUATION RESULTS ON SKU110K-R USING THE COCO-STYLE METRIC. THE RESULTS WITH RED AND BLUE COLORS INDICATE THE BEST AND SECOND-BEST RESULTS, RESPECTIVELY

Method	mAP	AP_{75}	AR_{300}
YoloV3-Rotate [49]	49.1	51.1	58.2
CenterNet-4point [50]	34.3	19.6	42.2
CenterNet [50]	54.7	61.1	62.2
DRN [3]	55.9	63.1	63.3
CFA [2]	57.0	63.5	63.9
AO2-DETR (Ours)	58.0	64.2	64.8

TABLE IV

EVALUATION RESULTS ON HRSC2016. ⊛ INDICATES VOC2012 METRICS, WHILE OTHER METHODS ARE EVALUATED UNDER VOC2007 METRICS. † INDICATES THAT THE RE-IMPLEMENTATION BY USING RESNET50 AS THE BACKBONE. THE RESULTS WITH RED AND BLUE COLORS INDICATE THE BEST AND SECOND-BEST RESULTS, RESPECTIVELY

Method	R^3Det† [7]	CenterMap [51]	S^2A-Net [5]	CFA† [2]
mAP	86.20 / 89.01⊛	92.80⊛	90.17 / 95.01⊛	87.10 / 91.60⊛
Method	ReDet [1]	Oriented R-CNN [10]	SASM† [42]	Ours
mAP	90.46 / 97.63⊛	90.40 / 96.50⊛	87.90 / 91.80⊛	88.12 / 97.47⊛

and a midpoint offset representation to represent oriented objects. Different from existing methods, we aim to design a conceptually simple and a new paradigm framework for the AOOD task. It does not require hand-designed components, complex pre/post-processing steps and inductive biases.

2) Results on DOTA-v1.5: Compared with DOTA-v1.0, DOTA-v1.5 contains more extremely small objects. We summarize the results on DOTA-v1.5 in Table II. Compared to state-of-the-art methods, AO2-DETR can achieve 66.26% mAP with single-scale data and 75.89% mAP with multi-scale data, outperforms RetinaNet OBB [46], Faster R-CNN OBB [18], Mask R-CNN OBB [47], and HTC [48] by a large margin. These experiments validate that an encoder-decoder detection model based on the standard Transformer can also achieve good results in small object detection, on the basis of not using FPN.

3) Results on SKU110K-R: The comparison results on SKU110K-R are shown in Table III. AO2-DETR achieves 58.0% AP and improves the state-of-the-art anchor-free methods by 1.0% (58.0% vs 57.0%). We also report the results of AP_{75} and AR_{300}. Most of the images in this dataset are taken with handheld cameras, the commodity is placed in a messy way, and the angle changes are relatively large. The results on the SKU110K-R dataset show that our method

TABLE V

ABLATION STUDIES OF PROPOSED MODULES IN AO2-DETR. DOTA-V1.0 IS USED IN THIS EXPERIMENT. "RAL" MEANS THE ROTATION-AWARE SET MATCHING LOSS. THE BOLD RESULTS INDICATE THE BEST PERFORMANCE

	OPG	OPR	IBR	RAL	PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SBF	RA	HA	SP	HC	mAP
Deformable DETR-O [16]					75.87	56.79	17.38	39.14	36.31	13.79	19.39	87.65	66.54	48.81	18.99	47.59	29.30	52.73	17.58	41.86
AO2-DETR				✓	85.85	71.62	42.30	58.56	55.74	70.56	75.34	89.65	71.60	76.31	50.39	66.97	53.19	67.88	52.62	65.90
AO2-DETR	✓			✓	89.13	78.23	53.27	69.08	54.60	78.52	79.25	90.42	75.63	77.04	56.69	70.61	65.15	68.43	63.46	71.30
AO2-DETR	✓	✓		✓	89.16	82.69	55.14	73.25	78.09	80.56	85.46	89.65	83.41	82.52	60.82	69.68	69.78	73.88	69.39	76.23
AO2-DETR	✓	✓	✓	✓	89.27	84.97	56.67	74.89	78.87	82.73	87.35	90.50	84.68	85.41	61.97	69.96	74.68	72.39	71.62	**77.73**

TABLE VI

SPEED VERSUS ACCURACY ON THE DOTA-V1.0 AND HRSC2016 DATASETS. WE ADOPT THE MAP AS THE METRIC OF SKU110K-R, VOC2012 AS THE METRIC OF HRSC2016. THE RESULTS WITH RED AND BLUE COLORS INDICATE THE BEST AND SECOND-BEST RESULTS OF EACH COLUMN, RESPECTIVELY

Method	Framework	DOTA-v1.0		HRSC2016	
		FPS	mAP	FPS	mAP
Oriented R-CNN [10]	Two-stage	13.8	75.87	16.0	96.50
ReDet [1]	Two-stage	12.4	76.25	13.2	97.63
R^3Det [7]	One-stage	13.2	69.80	14.7	89.01
CFA [2]	One-stage	15.0	73.45	18.6	91.60
SASM [42]	One-stage	15.6	74.92	18.2	91.80
AO2-DETR (ours)	One-stage	14.0	77.73	16.3	97.47

has strong applicability to this scenario and expands the use of the transformer-based method in the field of intelligent supermarkets.

4) Results on HRSC2016: The HRSC2016 dataset only contains one category "ship", which some of them have large aspect ratios and various orientations. The results are shown in Table IV. It can be seen that our AO2-DETR achieves competitive performance consistently, without the use of a more complicated architecture. Specifically, AO2-DETR achieves 88.12% and 97.47% under PASCAL VOC 2007 and VOC 2012 metrics, respectively.

5) Speed ersus Accuracy: Under the same setting, we compare the inference speed of different methods on the DOTA-v1.0 and HRSC2016 datasets, shown in Table VI. All methods adopt ResNet50 as the backbone and are implemented on the MMRotate [43]. The hardware platform of testing is a GeForce RTX 3090 with a batch size of 1. We adopt single-scale training and testing in this experiment. As shown in Table VI, our method has higher detection accuracy than other methods on the DOTA-v1.0 dataset. Although the performance of our method is slightly lower than ReDet [1] on the HRSC2016 dataset, the proposed method has a faster running speed. The speed of AO2-DETR is almost close to anchor-free detectors, which shows the effectiveness of our method in the inference stage.

D. Ablation Studies

In this section, we conduct a series of ablation experiments on the DOTA-v1.0 test set, SKU110K-R, and HRSC2016 datasets to evaluate the effectiveness of our proposed method. To detect oriented objects, we improve the Deformable DETR by adding the predicted value of the angle and term the detector "Deformable-O". Table V shows the impact of progressively integrating the proposed components into the baseline framework for the transformer-based AOOD methods.

TABLE VII

ABLATION STUDIES OF PROPOSED MODULES IN AO2-DETR. SKU110K-R AND HRSC2016 DATASETS ARE USED IN THIS EXPERIMENT

	OPG	OPR	IBR	RAL	SKU110K-R	HRSC2016
Deformable DETR-O [16]					42.8	70.13
AO2-DETR				✓	50.3	85.71
AO2-DETR	✓			✓	54.6	93.65
AO2-DETR	✓	✓		✓	57.3	95.34
AO2-DETR	✓	✓	✓	✓	**58.0**	**97.47**

1) Oriented Proposal Generation Mechanism: To explore the contribution of the oriented proposals generation (OPG) mechanism, we derive two settings: with/without oriented proposals generation. The results in Table V and Table VII clearly show that the oriented proposals generation is necessary for boosting performance. On the three datasets, OPG improves the performance by 5.4%, 4.3%, and 7.94%, respectively. The significant performance improvement indicates that the OPG mechanism can better locate objects and avoid the extracted features from being interfered with by other objects or backgrounds.

For a better understanding of the role of OPG mechanism, we visualize the adaptation process of sampling points (pink filled circle) and oriented proposals (green rectangle) of the last layer in decoder from 20^{th}, 40^{th}, and 50^{th} epoch, as shown in Fig. 7. For readability, we combine the sampling points and oriented proposals from feature maps of different resolutions into one image. It can be seen that the sampling points of the 20^{th} epoch are still scattered, but the sampling points of the 40^{th} epoch are concentrated in the center of the objects. At the 50^{th} epoch, the sampling points and generated oriented proposals can well cover the objects. In particular, these green rectangles are only proposals. They will be selected as object queries and sent to the deformable decoder layer to generate the final predictions. The final detection results are displayed in the last column. The visualization results show that the proposed component OPG presents a robust performance in both arbitrary orientation and densely packed scenarios.

2) Adaptive Oriented Proposal Refinement Module: To investigate the importance of the adaptive oriented proposal refinement (OPR) module, we perform a study of models with or without the OPR module. As shown in Table V, the proposed oriented proposal refinement significantly improves performance by 4.93% (76.23% vs 71.30%). In addition, we also explore the importance of iterative bounding box refinement (IBR) module. As shown in Table V, the performance

TABLE VIII

ABLATION STUDIES OF SET MATCHING COST. DOTA-V1.0 IS USED IN THIS EXPERIMENT. THE BOLD RESULTS INDICATE THE BEST PERFORMANCE. THE NUMBERS WITH BLUE COLOR INDICATES THE PERFORMANCE GAIN

	Focal Loss Cost	L1 Loss Cost	Rotated IoU Loss Cost	GWD Loss Cost	KLD Loss Cost	mAP
AO2-DETR	✓	✓	✗	✗	✗	56.65
AO2-DETR	✓	✓	✓	✗	✗	**77.73** (+21.08)
AO2-DETR	✓	✗	✓	✗	✗	72.68 (+16.03)
AO2-DETR	✓	✓	✗	✓	✗	58.13 (+1.48)
AO2-DETR	✓	✓	✗	✗	✓	59.76 (+3.11)

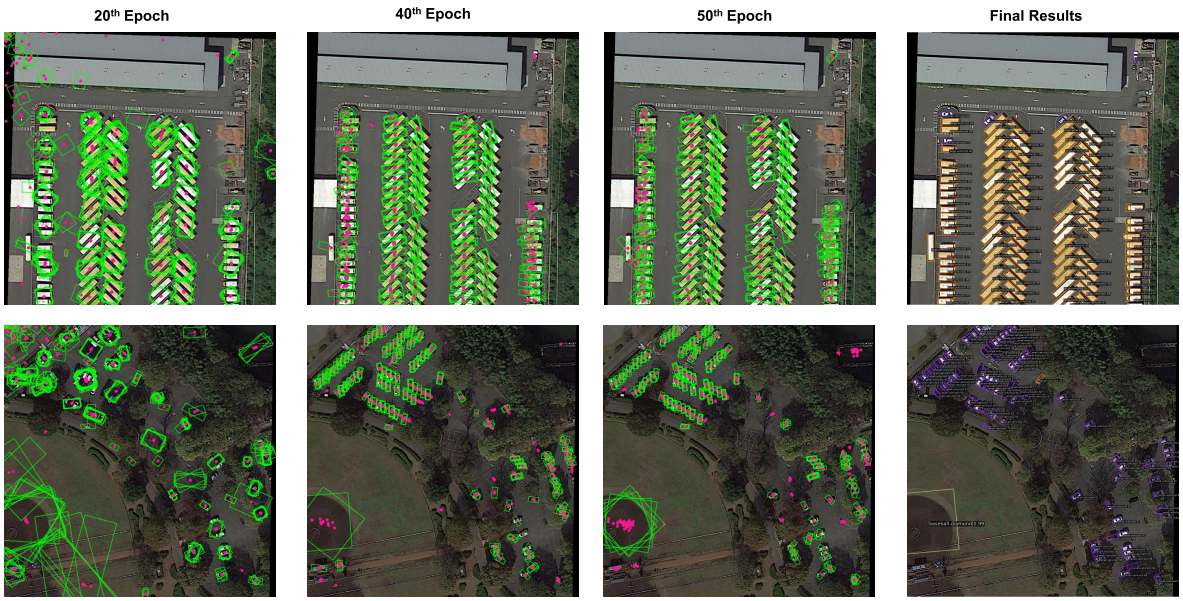


Fig. 7. Visualization of oriented proposal generation (OPG) mechanism. The learned oriented proposals are generated by OPG on the DOTA dataset. The top-300 proposals (green rectangle) per image are displayed. Each sampling point is marked as a pink filled circle. We can see that the sampling points are concentrated on the object, which is the main focus of deformable attention module.

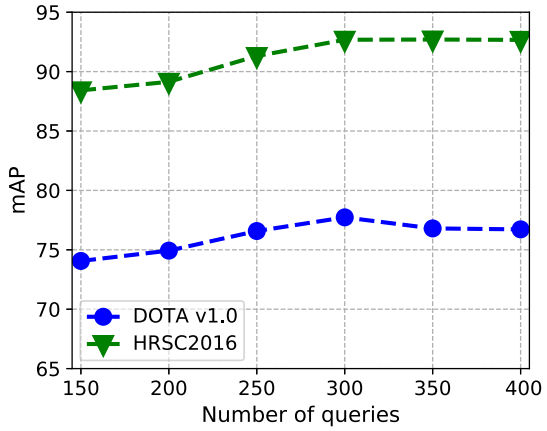


Fig. 8. Ablation results from adapting the number of queries on DOTA-v1.0 and HRSC2016 datasets. We compare the mAP with varying numbers of queries from 150 to 400.

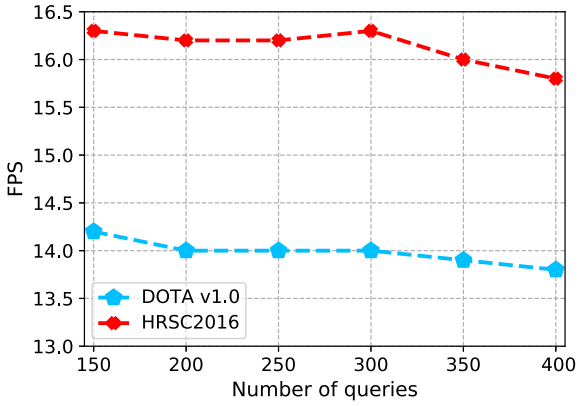


Fig. 9. Ablation results from adapting the number of queries on DOTA-v1.0 and HRSC2016 datasets. We compare the FPS with varying numbers of queries from 150 to 400.

of without iterative refinement module is reduced from 77.73% to 76.23%, which shows that the iterative refinement module can enhance the relationship between transformer decoder and better model context information. Furthermore, we can see that the performance gains of OPR and IBR are effective in some categories, especially in the small vehicle (SV) (24.27%), ship (SH) (8.10%), basketball court (BC) (9.05%), and storage tank (ST) (8.37%). Small vehicles and ships are usually placed densely. And the color of basketball court and storage tank is similar to the background. The background information will interfere with object information, making it difficult to distinguish boundaries. In addition, we also conduct more experiments to investigate the impact of the

Fig. 10. Qualitative results on DOTA-v1.0 [18] testing set using AO2-DETR with ResNet50 backbone. DOTA-v1.0 contains 15 common categories, such as large-vehicle, small-vehicle, plane, swimming-pool, ship, tennis-court, etc. The confidence threshold is set to 0.3 when visualizing these results. One color stands for one object class. Best viewed in color and with zoom.

OPR and IBR modules on the SKU110K-R and HRSC2016 datasets, shown in Table VII. The results show that the proposed OPR and IBR modules can effectively mitigate the problems of misalignment and cluttered features, and can better model contextual information, especially for dense regions.

3) Set Matching Loss: For classification and bounding box distance loss, we follow the default settings of the Deformable DETR, i.e., Focal loss and L1 loss. The effect of set matching loss are shown in Table V and Table VII. Using rotation-aware set matching loss (RAL) can significantly improve the performance. The transformer-based methods apply one-to-one assignments during the training stage, only one sample with the minimum matching cost is assigned as the positive sample, and the others are all negative samples. If we cannot ensure the accurate label assignment process, the initial training epoch will be completely chaotic and difficult to converge. For rotation-aware set matching loss, we have tried different set matching losses and different combinations, such as GWD [52], KLD [39], and Rotated IoU [38]. The experimental results in Table VIII prove that Rotated IoU performs best. With the rotation-aware set matching loss, the performance has been greatly improved, from 56.65% to 77.73%. We can ensure the correct one-to-one matching process in AO2-DETR by adding the rotation-aware set matching loss, which solves the problem of limited matching. Other existing loss works also can be easily extended to AO2-DETR, but exploring the importance of losses is not the focus of this paper.

4) Number of Queries: In this experiment, we compare changing the number of queries at the test stage to different models trained with varying numbers of queries, as shown in Fig. 8. Taking the DOTA-v1.0 dataset as an example, we can observe that with the gradual increase in the number of queries, the performance has been improved. When the number of queries is set to 150, the mAP is only 74.06%, but when the number of queries is increased to 300, the performance can achieve 77.73% (improved by 3.67%). This indicates that the number of queries is sufficient to cover the objects well. When we increase the number of queries to 350 and 400, there is a slight decline in performance. We suspect that redundant object queries will interfere with some dense objects, resulting in performance fluctuations.

In addition, we compare the FPS with varying numbers of queries from 150 to 400, as shown in Fig. 9. We can see that the number of queries has a smaller impact on FPS, and when the number of queries increases from 150 to 400, the FPS drops from 14.2 to 13.8 for the DOTA-v1.0 dataset, and 16.3 to 15.8 for the HRSC2016 dataset. To achieve a balance of speed and performance, the number of queries is set to 300. For the DOTA-v1.0 dataset, the FPS is 14.0, and for the HRSC2016 dataset, the FPS is 16.3.

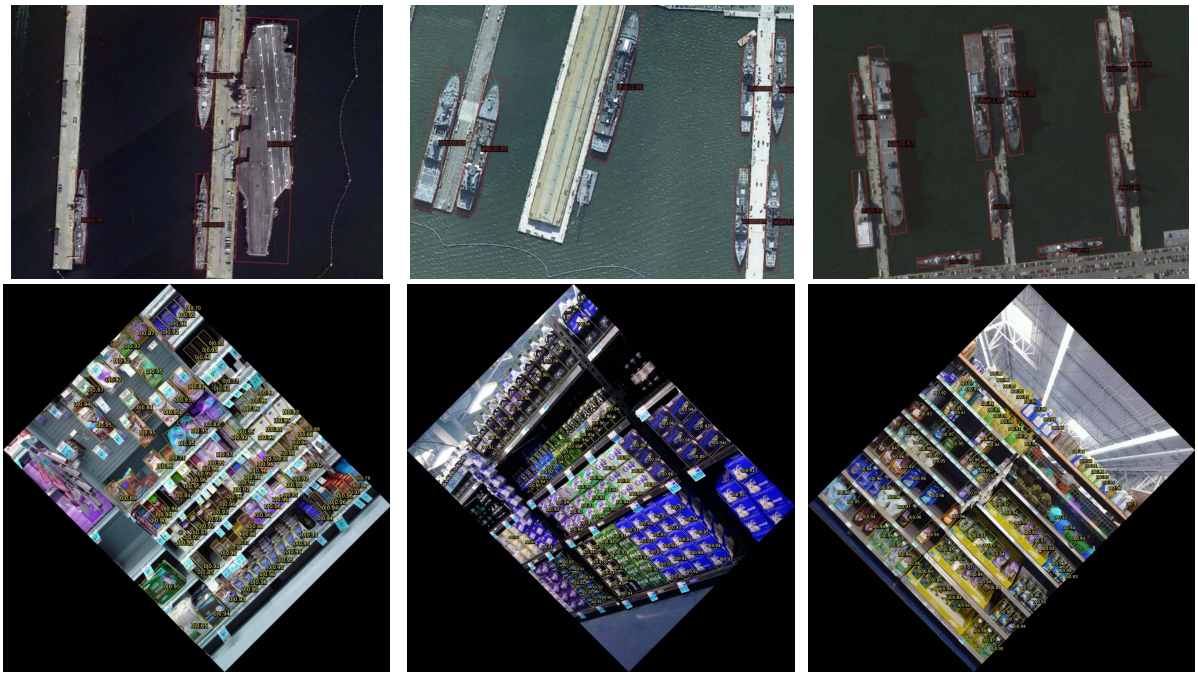


Fig. 11. Detection results on HRSC2016 [20] (the first row) and SKU110K-R [3] (the second row). Best viewed in color and with zoom.

With these ablation studies, we conclude that in the AO2-DETR design: oriented proposal generation, adaptive oriented proposal refinement, iterative bounding box refinement, and rotation-aware set matching loss all play important roles in the final performance.

E. Qualitative Results

Fig. 10 and Fig. 11 show the qualitative results of sample images from the DOTA, HRSC2016, and SKU110K-R datasets. We can notice that the proposed method can deal with various challenges in the AOOD task detection, including multi-oriented objects, small objects, large aspect ratio objects, and densely-packed objects. For example, as shown in Fig. 10, AO2-DETR can properly detect objects of various sizes and arbitrary orientations within the multi-category classification problem; as shown in Fig. 11, AO2-DETR is outstanding in the detection of large aspect ratio objects (the first row) and densely arranged oriented objects (the second row). However, there are still some failure cases, especially when the objects are placed heavily occluded, the sizes of objects are extremely small, and the color of objects is similar to the background. Future transform-based AOOD methods can focus on addressing these difficult cases.

V. Conclusion

In this paper, we propose an end-to-end transformer-based detector AO2-DETR for arbitrary-oriented object detection. The proposed AO2-DETR comprises dedicated components to address AOOD challenges, including an oriented proposal generation mechanism, an adaptive oriented proposal refinement module, and a rotation aware set matching loss in order to accurately detect oriented objects in images. The encoder-decoder architecture transforms the oriented proposals (served as object queries) into each corresponding object, which eliminates the need for hand-designed components and complex pre/post-processing. Our approach achieves state-of-the-art performance compared to recently anchor-free and single-stage methods on the oriented datasets (DOTA, SKU110K-R and HRSC2016 datasets). We validate that the transformer can enable adaptive receptive fields for oriented objects, thus it can deal with oriented and irregular placed objects naturally. Furthermore, we hope that this encoder-decoder paradigm will promote future works in oriented object detection.

Limitations: Compared with other CNN-based methods, the main limitation of our method lie in the longer training convergence time. It is widely known that the superior performance of transformers requires relatively larger computation cost. The future work of transformer-based AOOD methods can be devoted to solving these challenges.

References

[1] J. Han, J. Ding, N. Xue, and G.-S. Xia, "ReDet: A rotation-equivariant detector for aerial object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 2785–2794.

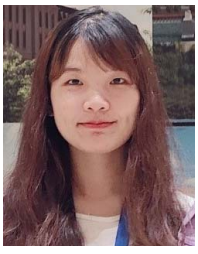
[2] Z. Guo, X. Zhang, C. Liu, X. Ji, J. Jiao, and Q. Ye, "Convex-hull feature adaptation for oriented and densely packed object detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 8, pp. 5252–5265, Aug. 2022.

[3] X. Pan et al., "Dynamic refinement network for oriented and densely packed object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 1–8.

[4] J. Ma et al., "Arbitrary-oriented scene text detection via rotation proposals," *IEEE Trans. Multimedia*, vol. 20, no. 11, pp. 3111–3122, Mar. 2018.

[5] J. Han, J. Ding, J. Li, and G.-S. Xia, "Align deep features for oriented object detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–11, 2022.

[6] X. Yang et al., "SCRDet: Towards more robust detection for small, cluttered and rotated objects," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8232–8241.
[7] X. Yang, J. Yan, Z. Feng, and T. He, "R^{3}Det: Refined single-stage detector with feature refinement for rotating object," in *Proc. AAAI Conf. Artif. Intell.*, May 2021, vol. 35, no. 4, pp. 3163–3171.
[8] F. Zhang, X. Wang, S. Zhou, and Y. Wang, "DARDet: A dense anchor-free rotated object detector in aerial images," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.
[9] J. Ding, N. Xue, Y. Long, G.-S. Xia, and Q. Lu, "Learning RoI transformer for oriented object detection in aerial images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2844–2853.
[10] X. Xie, G. Cheng, J. Wang, X. Yao, and J. Han, "Oriented R-CNN for object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 3520–3529.
[11] Z. Li, B. Hou, Z. Wu, L. Jiao, B. Ren, and C. Yang, "FCOSR: A simple anchor-free rotated detector for aerial object detection," 2021, *arXiv:2111.10780*.
[12] Y. Lin, P. Feng, J. Guan, W. Wang, and J. Chambers, "IENet: Interacting embranchment one stage anchor free detector for orientation aerial object detection," 2019, *arXiv:1912.00969*.
[13] S. Lang, F. Ventola, and K. Kersting, "DAFNe: A one-stage anchor-free approach for oriented object detection," 2021, *arXiv:2109.06148*.
[14] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
[15] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 213–229.
[16] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable transformers for end-to-end object detection," in *Proc. Int. Conf. Learn. Represent.*, 2020.
[17] Y. Wang, X. Zhang, T. Yang, and J. Sun, "Anchor DETR: Query design for transformer-based object detection," 2021, *arXiv:2109.07107*.
[18] G.-S. Xia et al., "DOTA: A large-scale dataset for object detection in aerial images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018.
[19] E. Goldman, R. Herzig, A. Eisenschtat, J. Goldberger, and T. Hassner, "Precise detection in densely packed scenes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5222–5231.
[20] Z. Liu, L. Yuan, L. Weng, and Y. Yang, "A high resolution optical satellite image dataset for ship recognition and some new baselines," in *Proc. Int. Conf. pattern Recognit. Appl.*, vol. 2, 2017, pp. 324–331.
[21] Z. Zhang, W. Guo, S. Zhu, and W. Yu, "Toward arbitrary-oriented ship detection with rotated region proposal and discrimination networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 11, pp. 1745–1749, Nov. 2018.
[22] Z. Yang, S. Liu, H. Hu, L. Wang, and S. Lin, "RepPoints: Point set representation for object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9657–9666.
[23] J. Wang, L. Yang, and F. Li, "Predicting arbitrary-oriented objects as points in remote sensing images," *Remote Sens.*, vol. 13, no. 18, p. 3731, Sep. 2021.
[24] Z. Liu, H. Wang, H. Weng, and L. Yang, "Ship rotated bounding box space for ship extraction from high-resolution optical satellite images with complex backgrounds," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 8, pp. 1074–1078, Aug. 2016.
[25] X. Yang and J. Yan, "Arbitrary-oriented object detection with circular smooth label," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 677–694.
[26] Y. Xu et al., "Gliding vertex on the horizontal bounding box for multi-oriented object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 4, pp. 1452–1459, Apr. 2020.
[27] S. Zhang et al., "RefineDet++: Single-shot refinement neural network for object detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 2, pp. 674–687, Feb. 2021.
[28] X. Chen, H. Li, Q. Wu, K. N. Ngan, and L. Xu, "High-quality R-CNN object detection using multi-path detection calibration network," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 2, pp. 715–727, Feb. 2021.
[29] X. Chen, J. Yu, S. Kong, Z. Wu, and L. Wen, "Joint anchor-feature refinement for real-time accurate object detection in images and videos," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 2, pp. 594–607, Feb. 2021.
[30] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
[31] Z. Huang, W. Li, X.-G. Xia, and R. Tao, "A general Gaussian heatmap label assignment for arbitrary-oriented object detection," 2021, *arXiv:2109.12848*.
[32] P. Sun et al., "What makes for end-to-end object detection?" in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 9934–9944.
[33] D. Meng et al., "Conditional DETR for fast training convergence," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 3651–3660.
[34] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," 2021, *arXiv:2103.14030*.
[35] S. Liu et al., "DAB-DETR: Dynamic anchor boxes are better queries for DETR," in *Proc. Int. Conf. Learn. Represent.*, 2022. [Online]. Available: https://openreview.net/forum?id=oMI9PjOb9Jl
[36] B. A. Wandell and J. Winawer, "Computational neuroimaging and population receptive fields," *Trends Cogn. Sci.*, vol. 19, no. 6, pp. 349–357, Jun. 2015.
[37] H. W. Kuhn, "The Hungarian method for the assignment problem," *Nav. Res. Logistics Quart.*, vol. 2, nos. 1–2, pp. 83–97, 1955.
[38] D. Zhou et al., "IoU loss for 2D/3D object detection," in *Proc. Int. Conf. 3D Vis. (DV)*, Sep. 2019, pp. 85–94.
[39] X. Yang et al., "Learning high-precision bounding box for rotated object detection via Kullback–Leibler divergence," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021.
[40] Q. Ming, Z. Zhou, L. Miao, H. Zhang, and L. Li, "Dynamic anchor learning for arbitrary-oriented object detection," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 3, pp. 2355–2363.
[41] W. Li, Y. Chen, K. Hu, and J. Zhu, "Oriented RepPoints for aerial object detection," 2021, *arXiv:2105.11111*.
[42] L. Hou, K. Lu, J. Xue, and Y. Li, "Shape-adaptive selection and measurement for oriented object detection," in *Proc. AAAI Conf. Artif. Intell.*, 2022.
[43] Y. Zhou et al. (2022). *MMRotate: A Rotated Object Detection Benchmark Using PyTorch*. [Online]. Available: https://github.com/open-mmlab/mmrotate
[44] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
[45] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2017, *arXiv:1711.05101*.
[46] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318–327, Feb. 2020.
[47] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jun. 2017, pp. 2961–2969.
[48] K. Chen et al., "Hybrid task cascade for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 4974–4983.
[49] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.
[50] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," 2019, *arXiv:1904.07850*.
[51] J. Wang, W. Yang, H.-C. Li, H. Zhang, and G.-S. Xia, "Learning center probability map for detecting objects in aerial images," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 5, pp. 4307–4323, May 2021.
[52] X. Yang, J. Yan, Q. Ming, W. Wang, X. Zhang, and Q. Tian, "Rethinking rotated object detection with Gaussian Wasserstein distance loss," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 11830–11841.



Linhui Dai received the B.E. degree in information system and information management from the School of Electronics Engineering and Computer Science, Peking University, Beijing, China, in 2018, where she is currently pursuing the Ph.D. degree. Her current research interests include oriented object detection, open world object detection, and salient object detection.

Hong Liu (Member, IEEE) received the Ph.D. degree in mechanical electronics and automation in 1996. He serves as a Full Professor with the School of Electrical Engineering and Computer Science, Peking University (PKU), China. He has been selected as a Chinese Innovation Leading Talent supported by the National High-Level Talents Special Support Plan since 2013. He has published more than 200 articles. He was a recipient of the Chinese National Aerospace Award, the Wu Wenjun Award on Artificial Intelligence, the Excellence Teaching Award, and the Candidates of Top Ten Outstanding Professors in PKU. He has published many papers in international journals and conferences, including IEEE TRANSACTIONS ON MULTIMEDIA (TMM), IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (TCSVT), IEEE TRANSACTIONS ON CYBERNETICS (TCYB), IEEE/ACM TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING (TASLP), IEEE TRANSACTIONS ON ROBOTICS (TRO), *Pattern Recognition* (PR), IJCAI, ICCV, CVPR, ICRA, and IROS. He has served as a Keynote Speaker, Co-Chair, Session Chair, or PC Member for many important international conferences, such as IEEE/RSJ IROS, IEEE ROBIO, IEEE SMC, and IIHMSP.

Hao Tang received the master's degree from the School of Electronics and Computer Engineering, Peking University, China, and the Ph.D. degree from the Multimedia and Human Understanding Group, University of Trento, Italy. He was a Visiting Scholar at the Department of Engineering Science, University of Oxford. He is currently a Post-Doctoral Researcher with the Computer Vision Laboratory, ETH Zürich, Switzerland. His research interests include deep learning, machine learning, and their applications to computer vision.

Zhiwei Wu received the B.S. degree from the School of Software Engineering, South China University of Technology, Guangdong, China, where he is currently pursuing the master's degree. His research interests include multi-modal named entity recognition and relation extraction.

Pinhao Song received the B.E. degree in mechanical engineering from Peking University, Beijing, China, in 2019, where he is currently pursuing the master's degree in computer applied technology. His current research interests include underwater object detection, generic object detection, and domain generalization.