Theoretically Achieving Continuous Representation of Oriented Bounding Boxes

Supplementary Material

8. Futher Explain of Metrics for Continuity

Within Sec. 3.1, we introduced diverse metrics for evaluating methodological continuity. These metrics encompass Target Rotation Continuity, Target Aspect Ratio Continuity, Loss Rotation Continuity, Loss Aspect Ratio Continuity, Decoding Completeness, and Decoding Robustness. While formal definitions were provided in Sec. 3.1, this section delves deeper into their conceptual underpinnings.

Target Rotation Continuity: This metric assesses whether each OBB is encoded into a sole prediction target, and if slight rotations induce gradual changes in the prediction target. Notably, PSC [46] demonstrates target rotation continuity by utilizing phase-shifting coding, ensuring continuous encoding despite OBB orientation changes. Conversely, Gliding Vertex [35] exhibits notable deviations in target rotation continuity when minor rotations affect nearly horizontal OBBs, leading to abrupt changes in the prediction target.

Target Aspect ratio Continuity: Here, the focus lies on determining if every OBB is encoded into a single prediction target and whether slight aspect ratio adjustments cause sudden changes in the prediction target. For instance, Gliding Vertex [35] maintains target aspect ratio continuity. However, PSC [46] struggles to sustain aspect ratio continuity, particularly when dealing with square-shaped OBBs.

Loss Continuity: Loss Continuity encompasses two distinct components: Loss Rotation Continuity and Loss Aspect Ratio Continuity. This metric evaluates whether minor rotation or aspect ratio changes result in abrupt fluctuations in the loss value. While some methods might falter in maintaining target rotation or aspect ratio continuity, they compensate by refining the loss function to ensure loss continuity. Notably, employing an L1 Loss function aids in preserving loss continuity for methods demonstrating target continuity.

Decoding Completeness: This criterion mandates precise representation of all OBBs. Methods rooted in CSL paradigms, such as those discussed in [29, 37, 39], often fall short in achieving Decoding Completeness due to discretized angle predictions, leading to imprecise orientation estimations of OBBs within finite angle classifications. Notably, we consider methods based on Gaussian distribution (such as GWD [40], KLD [41], and KFIoU [44]) satisfying Decoding Completeness because squares in various orientations can be possibly precisely decoded theoretically. However, in actual implementation, squares in only one orientation can be precisely decoded.

Decoding Robustness: Decoding Robustness demands that decoded OBBs remain resilient to slight errors in their representation. An example illustrating dissatisfaction with decoding robustness is GWD [40], which assigns square-like OBBs in different orientations to the same Gaussian distribution, leading to imprecise predictions for square-like targets. Especially, DHRec [22] encodes two symmetrical tilted slender OBBs into comparable representations, particularly when the aspect ratio is significantly large. Although this is uncommon, it suggests a propensity for the algorithm to confuse slender OBBs that are oriented in differing directions. Consequently, this observation leads us to conclude that DHRec does not fulfill the criteria for Decoding Robustness.

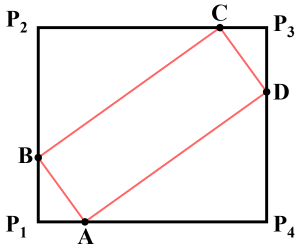


Figure 4. **An OBB with Its outer HBB.**

Target Rotation Continuity, Target Aspect ratio Continuity, Loss Rotation Continuity, and Loss Aspect Ratio Continuity are collectively referred to as Encoding Continuity, while Decoding Completeness and Decoding Robustness are collectively referred to as Decoding Continuity.

9. Details of COBB

9.1. Derivation of Four OBBs with Identical Outer HBB and r_s

We establish the outer HBB (x_c, y_c, w, h) of an OBB along with the sliding ratio r_s in Sec. 3.2. Herein, we elaborate on the existence of precisely four OBBs sharing the same outer HBB and r_s.

Given a generic OBB characterized by the outer HBB x_c, y_c, w, h, and sliding ratio r_s, we employ the principles of similar triangles to derive the equation:

$$||P_1B|| \cdot ||P_2B|| = ||P_1A|| \cdot ||P_2C||. \qquad (14)$$

Here, $||P_1B||$, $||P_2B||$,$||P_1A||$, and $||P_2C||$ are line segments depicted In Fig. 4. When $w \geq h$, $||P_1B|| \cdot ||P_2B|| = h^2 r_s(1 - r_s)$. Conversely, when $w < h$, $||P_1A|| \cdot ||P_2C|| =$

$w^2 r_s(1-r_s)$. Assuming $w \geq h$, we deduce:

$$\begin{cases} ||P_1B|| \cdot (h - ||P_1B||) = h^2 r_s(1-r_s), \\ ||P_1A|| \cdot (w - ||P_1A||) = h^2 r_s(1-r_s). \end{cases} \tag{15}$$

Solving these equations yields:

$$\begin{cases} ||P_1B|| = \frac{1 \pm (1-2r_s)}{2} h, \\ ||P_1A|| = \frac{1 \pm \sqrt{1-4 \cdot \frac{h^2}{w^2} \cdot r_s(1-r_s)}}{2} w. \end{cases} \tag{16}$$

These solutions delineate four distinct groups, corresponding to four unique OBBs. The detailed process for constructing these OBBs is elucidated in Sec. 9.2.

9.2. OBB Recovery from Nine Parameters

This section elucidates the method for computing the OBB from its outer HBB, sliding ratio r_s, and IoU scores s_0, s_1, s_2, and s_3.

Building upon the methodology outlined in Sec. 9.1 for computing two OBB vertices using Eq. 16, the four OBBs can be derived as follows:

$$\begin{aligned} OBB_0 =& [(x_c - x_s, y_c - 0.5h), (x_c + 0.5w, y_c + y_s), \\ & (x_c + x_s, y_c + 0.5h), (x_c - 0.5w, y_c - y_s)], \\ OBB_1 =& [(x_c + x_s, y_c - 0.5h), (x_c + 0.5w, y_c + y_s), \\ & (x_c - x_s, y_c + 0.5h), (x_c - 0.5w, y_c - y_s)], \\ OBB_2 =& [(x_c - x_s, y_c - 0.5h), (x_c + 0.5w, y_c - y_s), \\ & (x_c + x_s, y_c + 0.5h), (x_c - 0.5w, y_c + y_s)], \\ OBB_3 =& [(x_c + x_s, y_c - 0.5h), (x_c + 0.5w, y_c - y_s), \\ & (x_c - x_s, y_c + 0.5h), (x_c - 0.5w, y_c + y_s)]. \end{aligned} \tag{17}$$

Here, y_s and x_s are computed as:

$$\begin{aligned} y_s &= \begin{cases} \frac{1-2r_s}{2} h & w \geq h, \\ \frac{\sqrt{1-4\frac{w^2}{h^2} r_s(1-r_s)}}{2} h & w < h, \end{cases} \\ x_s &= \begin{cases} \frac{\sqrt{1-4\frac{h^2}{w^2} r_s(1-r_s)}}{2} w & w \geq h, \\ \frac{1-2r_s}{2} w & w < h. \end{cases} \end{aligned} \tag{18}$$

The OBB associated with the highest IoU score is selected as the recovered result.

9.3. COBB Implementation for Oriented Proposals

Many models utilize the discrepancy between ground truth and assigned proposals as an n-dimensional regression target. In Sec. 3.3, we introduced employing COBB to compute the bias between the ground truth OBB and a horizontal proposal region. Here, we extend this method to oriented proposal regions.

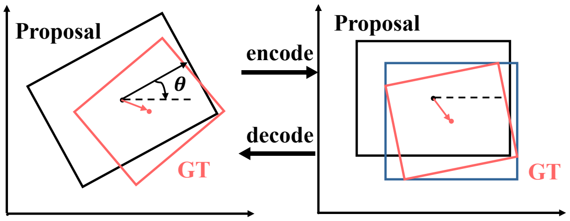


Figure 5. **COBB for Oriented Proposals.**

As depicted in Fig. 5, for an oriented proposal with center (x_c, y_c) and rotation angle θ (clockwise), the regression target computation involves the following steps:

- Rotate the oriented proposal and ground truth OBB by θ counterclockwise around the center (x_c, y_c).
- Calculate the bias between the rotated proposal region and ground truth, leveraging the method introduced in Sec. 3.3 for horizontally oriented regions.

For predicting the OBB from an oriented proposal and predicted vector, the process involves:

- Rotate the oriented proposal and ground truth OBB by θ counterclockwise around the center (x_c, y_c).
- Compute based on the rotated horizontally oriented proposal and the predicted bias to obtain an OBB.
- Rotate the OBB calculated in the previous step by θ clockwise to derive the predicted OBB.

This methodology suits Oriented Object Detection (OOD) models using Rotated RoI Align for feature extraction from oriented regions [2, 33]. However, if a model utilizes a different method for feature extraction from oriented regions, the proposed regression target generation method may not be applicable.

9.4. Relationship between r_s and r_a

We assert that the sliding ratio r_s provides an approximation for r_a, the acreage ratio of the OBB concerning its outer HBB in Sec. 3.2. In this section, we establish the relationship between r_s and r_a and subsequently demonstrate why only a pair of symmetrical OBBs share identical (x_c, y_c, w, h, r_a).

In Sec. 9.1, we established that only four OBBs share the same (x_c, y_c, w, h, r_s). For any OBB, we express r_s in terms of r_a and derive the unified equation:

$$4r_s(1-r_s) = \frac{r_{wh}^2 + 1 - \sqrt{(r_{wh}^2+1)^2 - 16r_{wh}^2 r_a(1-r_a)}}{2r_{wh}^2}. \tag{19}$$

Here, $r_{wh} = \min(\frac{w}{h}, \frac{h}{w})$ denotes the aspect ratio of the outer HBB. Eq. 19 indicates that $r_s(1-r_s)$ monotonically increases with respect to $r_a(1-r_a)$. Since four OBBs share identical x_c, y_c, w, h, r_s, they also share the same $r_a(1-r_a)$. Given that r_a for OBB_0 and OBB_3 is below 0.5, while for the other two, it exceeds 0.5, we conclude that only a symmetrical OBB pair shares identical (x_c, y_c, w, h, r_a).

9.5. Details of Computing IoU scores

Directly computing IoU scores often involves generating four OBBs based on (x_c, y_c, w, h, r_s) and then evaluating the IoU between these OBBs and the ground truth. However, this direct calculation can be intricate and time-consuming. Our approach involves first computing the IoUs between the four OBBs sharing identical (x_c, y_c, w, h, r_s) based on the five parameters.

The simplest method for computing IoU scores is directly generating the four OBBs according to (x_c, y_c, w, h, r_s), and then calculating the IoU between the ground truth and the four OBBs. However, the calculation of IoU between OBBs is complex and time-consuming. We notice that the IoU between the four OBBs sharing the same (x_c, y_c, w, h, r_s) can be directly computed from the five parameters. As a result, we can compute these IoUs first, and then select the IoU scores based on the type of the ground truth OBB.

Assuming $w \geq h$, let's define intermediate variables:

$$\begin{aligned}
r_{sx} &= \frac{1 - \sqrt{1 - 4 \cdot \frac{h^2}{w^2} \cdot r_s(1 - r_s)}}{2}, r_{sy} = r_s, \\
l_1 &= \sqrt{(r_{sx}w)^2 + (r_{sy}h)^2}, \\
l_2 &= \sqrt{[(1 - r_{sx})w]^2 + [(1 - r_{sy})h]^2}, \\
l_3 &= \sqrt{(r_{sx}w)^2 + [(1 - r_{sy})h]^2}, \\
l_4 &= \sqrt{[(1 - r_{sx})w]^2 + (r_{sy}h)^2},
\end{aligned} \tag{20}$$

where l_1, l_2, l_3, and l_4 are correspond to potential side lengths of the OBBs.

IoU between OBB_0 and OBB_1, $IoU_{0,1}$ is:

$$\begin{aligned}
I_{0,1} &= [1 - \frac{(1 - 2r_{sx})r_{sx}w^2}{(1 - r_{sy})h^2}]l_1 l_2, \\
IoU_{0,1} &= \frac{I_{0,1}}{l_1 l_2 + l_3 l_4 - I_{0,1}}.
\end{aligned} \tag{21}$$

IoU between OBB_0 and OBB_2, $IoU_{0,2}$ is:

$$\begin{aligned}
I_{0,2} &= (1 - \frac{(1 - 2r_{sy})r_{sy}h^2}{(1 - r_{sx})w^2})l_1 l_2, \\
IoU_{0,2} &= \frac{I_{0,2}}{l_1 l_2 + l_3 l_4 - I_{0,2}}.
\end{aligned} \tag{22}$$

IoU between OBB_0 and OBB_3, $IoU_{0,3}$ is:

$$\begin{aligned}
I_{0,3} &= \frac{(r_{sx} + r_{sy} - 2r_{sx}r_{sy})^2}{(1 - r_{sx})(1 - r_{sy})} \times \frac{wh}{2}, \\
IoU_{0,3} &= \begin{cases} \frac{I_{0,3}}{2l_1 l_2 - I_{0,3}} & I_{0,3} \neq 0, \\ 0 & I_{0,3} = 0. \end{cases}
\end{aligned} \tag{23}$$

IoU between OBB_1 and OBB_2, $IoU_{1,2}$ is:

$$\begin{aligned}
h_1 &= \frac{1}{2}w - \frac{\frac{1}{2} - r_{sy}}{1 - r_{sy}} r_{sx} w, \\
h_2 &= \frac{1}{2}h - \frac{\frac{1}{2} - r_{sx}}{1 - r_{sx}} r_{sy} h, \\
\tan\alpha &= \frac{\frac{\frac{1}{2} - r_{sx}}{1 - r_{sx}} l_4}{\frac{1}{2(1 - r_{sy})} l_3}, \quad \tan\beta = \frac{\frac{\frac{1}{2} - r_{sy}}{1 - r_{sy}} l_3}{\frac{1}{2(1 - r_{sx})} l_4}, \\
I_{1,2} &= 2\frac{\tan\alpha \tan\beta}{\tan\alpha + \tan\beta}(h_1^2 + h_2^2) + 2h_1 h_2, \\
IoU_{1,2} &= \begin{cases} \frac{I_{1,2}}{2l_3 l_4 - I_{1,2}} & \tan\alpha\tan\beta \neq 0, \\ \frac{2h_1 h_2}{2l_3 l_4 - 2h_1 h_2} & \tan\alpha\tan\beta = 0. \end{cases}
\end{aligned} \tag{24}$$

These IoUs constitute the IoU matrix $M(w, h, r_s)$ as:

$$M(w, h, r_s) = \begin{bmatrix} 1 & IoU_{0,1} & IoU_{0,2} & IoU_{0,3} \\ IoU_{0,1} & 1 & IoU_{1,2} & IoU_{0,2} \\ IoU_{0,2} & IoU_{1,2} & 1 & IoU_{0,1} \\ IoU_{0,3} & IoU_{0,2} & IoU_{0,1} & 1 \end{bmatrix}. \tag{25}$$

This matrix ensures continuity for $w > 0, h > 0, r_s \in [0, 0.5]$. Each element in row i and column j of $M(w, h, r_s)$ represents the IoU between OBB_i and OBB_j.

Given a ground truth OBB, we identify its corresponding type among these four OBBs and extract the corresponding row from $M(w, h, r_s)$ as its IoU scores.

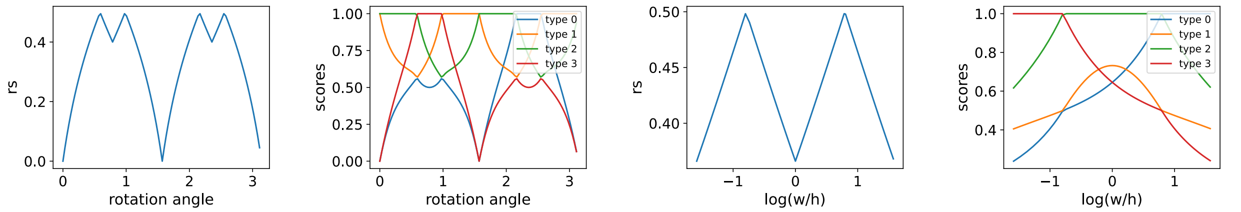
9.6. Proof of Encoding Continuity of COBB

In this section, we demonstrate the continuous evolution of the nine parameters in COBB as the OBB transforms.

The outer HBB and the area of an OBB change continuously during shape transformations, ensuring the continuity of x_c, y_c, w, h, and r_a. Eq. 19 substantiates the continuity of r_s concerning r_a, thereby ensuring the continuity of r_s as well.

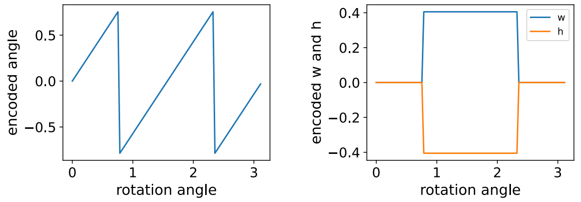
To establish the continuity of IoU scores, we consider an OBB perturbed from OBB X, denoted as Y. The similarity between x_c, y_c, w, h, and r_s of X and Y emerges from their continuous evolution during shape transformations. Eq. 17 confirms the analogous construction of four OBBs using the parameters of X and Y. IoU scores represent the overlaps between X and the four OBBs sharing the same x_c, y_c, w, h, and r_s as X. As both X and the associated four OBBs remain analogous before and after perturbation, minor disturbances on X do not significantly alter its IoU scores.

Fig. 6 and Fig. 7 provide comparative insights into the regression targets of COBB and the traditional Acute-angle Representation. COBB consistently exhibits continuous encoding results, while the Acute-angle Representation displays distinct shifts at rotation angles of $\pi/4$ and $3\pi/4$.



(a) r_s **when Rotation** (b) **IoU scores when Rotation** (c) r_s **when AR changes** (d) **IoU scores when AR changes**

Figure 6. **COBB Parameters during OBB Transformation**. AR represents the Aspect Ratio.



(a) θ **when Rotation** (b) w **and** h **when Rotation**

Figure 7. **Traditional Acute-angle Representation Parameters during OBB Rotation**. The traditional Acute-angle Representation encodes an OBB using the center (x_c, y_c), width w, height h, and rotation angle θ, which is constrained within the range $[-\frac{\pi}{4}, \frac{\pi}{4})$.

In summary, slight disturbances to an OBB minimally affect its x_c, y_c, w, h, r_s, and IoU scores, affirming the stability of these parameters under minor perturbations.

9.7. Proof of Decoding Continuity of COBB

Decoding Completeness is inherently fulfilled within the COBB decoding process. Here, we aim to establish the Decoding Robustness of COBB.

Slight disturbances in the parameters of COBB can be categorized as perturbations in the outer HBB, r_s, and IoU scores. As per Eq. 17, minor perturbations in the outer HBB and r_s won't significantly alter the decoded OBB if IoU scores remain constant. If a perturbation doesn't affect the classification corresponding to the highest IoU score, it won't impact the decoded OBB. Even if a perturbation shifts the highest score from classification i to j, where $s_i = 1$ (indicating the correct classification), slight disturbances maintain the relative values of s_i and s_j, making s_j close to 1. As s_j represents the IoU between the correct OBB and the decoded OBB_j (from Eq. 17), choosing OBB_j as the decoded result won't introduce significant decoding errors.

9.8. Comparison of Different Regression Targets

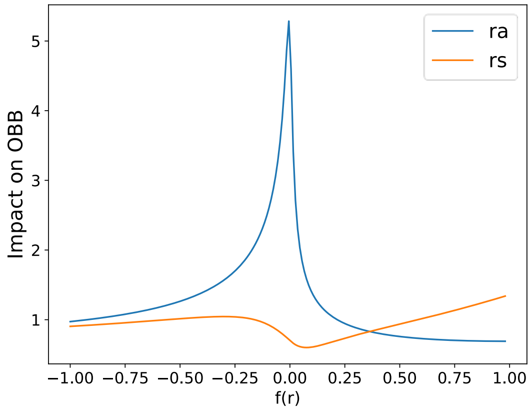


Figure 8. **Impact of Slight Disturbance on** r_s **and** r_a**:** We assessed the effects on the decoded OBB when r_s and r_a undergo slight disturbances. To mitigate the sensitivity of OBBs with large aspect ratios to small perturbations in r_s or r_a, we consider $f(r_s) = r_{ln}$ (as previously defined) and $f_{ln}(r_a) = 1 + \log_2 r_a$ as the specific perturbed values.

Table 8. **NAE of parameters.** α is the parameter for orientation determination in Gliding Vertex. In our method and Gliding Vertex **xy** denotes the center, **wh** represents width and height, r_s and **IoU scores** are parameters of our method, and α denotes the additional parameters for orientation determination of Gliding Vertex.

Method	xy	wh	r_s	IoU scores	α
Ours.	0.18%	0.31%	0.28%	0.41%	-
Glidng Vertex	0.17%	0.31%	-	-	0.75%

To mitigate the sensitivity of objects with large aspect ratios to predicted results, we introduce r_{ln} as the regression target, which is defined by Eq. 11. It is an approximation to $f_{ln}(r_a) = 1 + \log_2(r_a)$. The direct use of $f_{ln}(r_a)$ is feasible by recovering r_s with Eq. 19. Notably, only two OBBs share the same x_c, y_c, w, h, and r_{ln}; hence, we need to derive r_s from r_{ln} first and then select one of the four OBBs that share the same r_s as the decoded result, preventing Decoding Ambiguity.

In previous experiments in Tab. 6, we established the superiority of r_{ln} over $f_{ln}(r_a)$. One reason is that the process of recovering r_s from r_a can introduce precision errors. This section presents another reason: the sensitivity of the decoded OBB to slight disturbances in $f_{ln}(r_a)$.

Fig. 8 illustrates the impact of slight disturbances on r_{ln} and $f_{ln}(r_a)$. The impact is quantified as follows:

$$\Delta OBB = (1 - IoU[f_{Dec}(r), f_{Dec}(r+\epsilon)])\epsilon^{-1}. \quad (26)$$

Table 9. **mAP_{50} of Models in JDet Benchmark on DOTA-v1.0.**

Model	PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SBF	RA	HA	SP	HC	mAP_{50}
CSL	89.36	77.74	37.66	66.73	77.85	59.27	77.75	90.86	79.74	82.23	55.20	63.07	51.55	66.03	44.93	67.99
RsDet	89.47	78.02	38.39	63.01	77.28	60.95	76.41	90.77	81.80	83.47	54.26	61.98	51.68	66.55	52.18	68.41
RetinaNet	89.56	79.70	38.12	67.58	76.21	59.47	76.80	90.70	83.56	81.07	53.88	64.73	53.36	65.99	41.97	68.18
RetinaNet+KLD	89.42	76.51	39.36	65.20	77.83	63.19	82.48	90.90	79.20	83.46	54.39	63.83	53.26	67.67	44.53	68.75
RetinaNet+KFIoU	89.38	81.05	39.12	68.39	77.37	62.58	77.86	90.87	82.43	82.32	56.00	65.60	53.50	67.08	41.36	68.99
RetinaNet+GWD	89.10	78.04	39.07	69.21	77.27	62.05	81.05	**90.90**	84.31	83.02	57.24	61.96	53.79	64.40	43.83	69.02
FCOS	88.07	78.80	44.85	65.58	74.88	68.17	77.84	90.90	79.99	83.60	57.42	65.30	62.96	68.59	48.55	70.37
ATSS	88.44	78.77	49.14	67.17	77.63	74.97	85.56	90.78	83.39	83.82	59.17	61.39	65.35	65.48	55.50	72.44
S^2A-Net	89.21	81.04	50.97	71.35	78.21	77.42	87.05	90.88	82.51	85.00	63.35	64.52	66.45	67.67	53.60	73.95
Faster R-CNN	89.46	83.89	49.64	69.59	77.57	73.23	86.52	90.90	79.33	85.74	58.84	60.49	65.78	68.64	55.55	73.01
Gliding.	89.34	83.68	50.15	69.97	78.20	72.51	87.17	90.90	79.94	85.46	57.07	62.57	66.93	66.12	59.61	73.31
RoI Trans.	89.21	83.88	53.01	**72.97**	77.86	78.08	88.01	90.86	86.94	85.84	63.50	61.53	75.77	**70.33**	56.02	75.59
O-RCNN.	**89.72**	84.41	52.94	71.80	**78.71**	77.51	**88.15**	90.90	85.90	84.90	61.58	63.93	74.23	69.94	51.99	75.11
ReDet	88.90	82.28	52.42	72.76	77.63	82.52	88.12	90.88	86.51	85.81	**67.34**	65.33	76.07	68.58	60.54	76.38
Ours (RoI Trans. based)	89.52	**84.98**	**54.99**	72.16	77.71	82.81	88.10	90.81	85.45	85.62	63.89	**66.15**	**76.64**	70.13	59.05	**76.53**
Ours (ReDet based)	89.71	84.82	53.27	71.40	77.02	**83.80**	88.07	90.85	**87.11**	**86.20**	66.44	63.72	76.15	67.99	**61.26**	76.52

Table 10. **mAP of COBB on DOTA-v1.0 under Multi-scale Data Augmentation.**

Models	mAP_{50}	mAP_{75}	$\text{mAP}_{50:95}$
Oriented R-CNN [33]	78.73	55.07	50.57
+COBB-sig	79.09	55.61	**50.80**
+COBB-ln	**79.23**	**56.15**	50.55

Table 11. **Experiments on Optimizing IoU Enhancement Factor λ.** We utilize IoU^{λ} as a specific regression target for IoU scores to widen the distinction between scores for the ground truth and other classifications. These experiments were conducted using Rotated Faster R-CNN + COBB-sig.

λ	mAP_{50}	mAP_{75}	$\text{mAP}_{50:95}$
1	73.41	**44.21**	43.10
2	**74.00**	44.03	**43.29**
4	73.63	43.85	42.83

Table 12. **Experiments on SOTA baselines on DOTA-v1.0.**

Method	mAP_{50}	mAP_{75}	$\text{mAP}_{50:95}$
SES	75.72	48.86	46.19
SES + Ours.	76.43(+0.71)	49.28(+0.42)	46.59(+0.40)
LSKNet-t	76.68	49.28	46.15
LSKNet-t + Ours.	77.29(+0.61)	50.91(+1.63)	47.62(+1.47)

Here, r is r_{ln} or $f_{ln}(r_a)$, f_{Dec} decodes r with a nearly square outer HBB into an OBB, and ϵ is a small value close to 0. Notably, when $f_{ln}(r_a)$ approaches zero, the decoded OBB demonstrates significant sensitivity to variations in $f_{ln}(r_a)$, whereas the impact of slight disturbances on r_{ln} remains relatively stable.

9.9. Models' Ability to Well Estimate Parameters

We contend that the parameters in our COBB are easily estimable due to their continuity. To elucidate this assertion, we introduce the Normalized Average Error (NAE) as a metric for assessing the difficulty of parameter estimation. Given the i-th prediction of the parameter as P_i and its corresponding ground truth as T_i, the NAE is defined as $\text{NAE} = \frac{1}{N}\sum_{i=1}^{N}\frac{(P_i - T_i)^2}{(\max(T)-\min(T))^2}$, where N represents the number of predictions, and $\max(T)$ and $\min(T)$ denote the maximum and minimum values of ground truth values.

We posit that parameters with small NAE values are more ready to estimate. The NAE values of our method and Gliding Vertex are documented in Tab. 8. Without significantly influencing the prediction difficulty of the other parameters, r_s and IoU scores in our method can be better estimated than αs in Gliding Vertex.

10. Additional Experiments

10.1. Training Settings

Experiments were conducted using Jittor [10] on a single RTX 3090 running on Linux. The models utilized ResNet-50 [9] with FPN [14] to extract multi-level feature maps. During training, an SGD optimizer was employed, with a learning rate of 0.005 for two-stage models and 0.01 for one-stage models.

For dataset-specific training:

- DOTA, FAIR1M, and DIOR datasets were trained for 12 epochs, while HRSC2016 was trained for 36 epochs.

- For images in the DOTA and FAIR1M datasets larger than $1,024 \times 1,024$, they were split into multiple $1,024 \times 1,024$ tiles with a 200-pixel overlap.
- Data augmentation included random horizontal and vertical flips, each with a 50% probability.

10.2. More Results on JDet Benchmark

Tab. 9 provides a detailed breakdown of the results presented in Tab. 3. These comprehensive experimental findings underscore the superior performance achieved by our proposed method within our benchmark evaluation.

Analysis of the results indicates a pronounced advantage of our method in detecting objects characterized by a long aspect ratio, exemplified by categories such as Bridge (BR), Large Vehicle (LV), and Harbor (HA). This advantage is attributable to the inherent continuity embedded within our methodology, which mitigates potential confusion and interference during the training process arising from sudden changes in regression targets as OBBs approach a horizontal orientation. Notably, the models implemented in our benchmark employ the Acute-angle Representation as a default approach. The conspicuous discontinuity inherent in the Acute-angle Representation becomes especially evident when objects exhibit a considerable aspect ratio. Consequently, the discernible advantage exhibited by our proposed method in these scenarios underscores the efficacy of its continuous nature.

10.3. Futher Ablation Study for COBB

COBB Performance under Multi-scale Data Augmentation. Tab. 10 showcases the performance of COBB on DOTA-v1.0 under Multi-scale data augmentation. Multi-scale augmentation involved resizing training images to 0.5, 1.0, and 1.5 times their original dimensions, and these variations were incorporated into training and testing. For large images split into tiles, the width of the overlapping area was adjusted to 500 pixels. The results affirm the efficacy of COBB under this data augmentation technique.

Optimizing the IoU Enhancement Factor λ. We alter the IoU scores by exponentiating them to the power of λ to diminish the impact of incorrect categories. Tab. 11 displays experiments demonstrating that $\lambda = 2$ achieves optimal performance. Notably, $\lambda = 1$ exhibits advantages in mAP$_{75}$. Throughout this article, COBB-sig employs $\lambda = 2$, while COBB-ln uses $\lambda = 1$.

10.4. Experiments on Latest Techniques

To verify the advantage of our method on SOTA methods, we added our method to the latest SOTAs, including SES [36] and LSKNet [12]. Experiments in Tab. 12 demonstrate that our method achieves a relatively large mAP$_{50}$ improvement, 0.71% and 0.61%, over SES and LSKNet, respectively.