Learning High-Precision Bounding Box for Rotated Object Detection via Kullback-Leibler Divergence

Anonymous Author(s)
Affiliation
Address
email

Abstract

Existing rotated object detectors are mostly inherited from the horizontal detection paradigm, as the latter has evolved into a well-developed area. Taking the perspective that horizontal detection is a special case for rotated object detection, in this paper, we are motivated to change the design of rotation regression loss from induction paradigm to deduction methodology. We show that one essential challenge is how to modulate the coupled parameters in the rotation regression loss, as such the estimated parameters can influence to each other during the dynamic joint optimization, in an adaptive and synergetic way. Specifically, we first convert the rotated bounding box into a 2-D Gaussian distribution, and then calculate the Kullback-Leibler Divergence (KLD) between the Gaussian distributions as the regression loss. By analyzing the gradient of each parameter, we show that KLD (and its derivatives) can dynamically adjust the parameter gradients according to the characteristics of the object. Specifically, it will adjust the importance (gradient weight) of the angle parameter according to the aspect ratio. This mechanism can be vital for high-precision detection as a slight angle error would cause a serious accuracy drop for large aspect ratios objects. We further show that the KLD loss can be degenerated into the popular l_n-norm loss for horizontal detection. Experimental results on seven datasets using different detectors show the consistent superiority of our new approach.

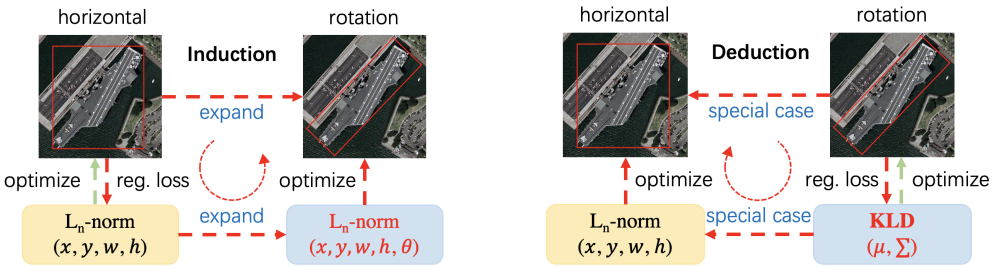
1 Introduction

As a fundamental building block for visual analysis in different kinds of image data including aerial images, scene text, and faces, rotated object detection has recently been developed rapidly [41; 42; 47; 43; 45], which benefit themselves from the well-established horizontal detection approaches [7; 30; 15; 16; 5]. Specifically, many works [6; 8; 27; 24] build themselves upon the previously established horizontal box detection pipeline from an inductive perspective, as shown in Figure 1(a).

In this paper, we take a step back, and aim to develop (from a deductive perspective) a unified regression framework for rotation detection and its special case: horizontal detection. In fact, our new framework enjoys a coherent property that it can be degenerated into the current commonly used regression loss (e.g. l_n-norm) in special cases (horizontal detection), as shown in Figure 1(b).

For a devising a rotation regression loss for high-precision rotation detection, one important observation is that the importance of different parameters to different types of objects can vary. For example, the angle parameter (θ) and the center point parameter (x, y) are important for large aspect ratio objects and small objects, respectively. In another word, it is conjectured that the parameters in the regression loss are coupled, which influence to each other during the learning process. Therefore it calls for more dynamic optimization strategy.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.



(a) Previous methods follow the induction paradigm from special horizontal to general rotated detection.

(b) Our proposed method adopts a deduction methodology from general rotated to special horizontal detection.

Figure 1: Methodological road-map difference between horizontal detection and rotation detection in the previous methods and the proposed method.

Inspired by the above ideas, we first convert the rotated bounding box $\mathcal{B}(x, y, h, w, \theta)$ into a 2-D Gaussian distribution $\mathcal{N}(\mu, \mathbf{\Sigma})$. This conversion can preliminarily couple (h, w, θ) into a new parameter $\mathbf{\Sigma}$. As a standard distance metric, we then use the Kullback-Leibler Divergence (KLD) [14] to calculate the distribution distance between the predicted bounding box and ground truth as the regression loss. We compare KLD with Smooth L1 loss [7] and another distance metric, Gaussian Wasserstein Distance (GWD) [33; 45], and find that KLD has a more complete parameter optimization mechanism. In particular, by analyzing the gradient of the parameters during learning, we show that the optimization of one parameter will be affected by other parameters (as the gradient weight). It means that the model will dynamically adjust the optimization strategy given a specific configuration of an object for detection, which is shown can lead to excellent performance in high-precision detection. As the horizontal bounding box is a special case of the rotated bounding box, we show that KLD can also be degenerated into the l_n-norm loss as commonly used in existing horizontal detection pipeline. The highlights of this paper are four-folds:

1) Differing from the dominant existing practices that build rotation detectors heavily upon the horizontal detectors, we develop new rotation detection loss from scratch and show that it is coherent with existing horizontal detection protocol in its degenerated case for horizontal detection.

2) To achieve a more principled measurement between the prediction and ground truth, instead of computing the difference for each physically-meaningful parameter related to the bounding box which are in different scales and units, we innovatively convert the regression loss of rotation detection into the KLD of two 2-D Gaussian distributions, leading to a clean and coherent regression loss.

3) Through the gradient analysis of each parameter in KLD, we further find that the coupling optimization between the parameters greatly promotes the improvement of high-precision detection, which verify the advantage of our loss design.

4) Extensive experimental results on seven public datasets and two popular detectors show the effectiveness of our approach.

2 Related Work

Horizontal Object Detection. Horizontal object detection which covers most existing detection literature, normally uses a horizontal bounding box to represent the object. The mainstream classical object detection algorithms can be roughly divided according to the following standards: Two- [7; 30; 15; 5] or Single-stage [29; 18; 16] object detection, Anchor-free [32; 55; 48] or Anchor-based [30; 15; 16] object detection and CNN [30; 16; 32] or Transformer-based [3; 57] object detection. Although the pipelines may vary, the mainstream regression loss often uses the popular l_n-norm loss (such as smooth L1 loss) or IoU-based loss (such as GIoU [31], and DIoU [52]). These above-mentioned detectors have also been widely used in other scenarios and have achieved satisfactory performance. However, horizontal detectors do not provide accurate orientation and scale information, which are needed in many important applications such as object detection in aerial images and recognition of sequential characters for multi-oriented scene texts.

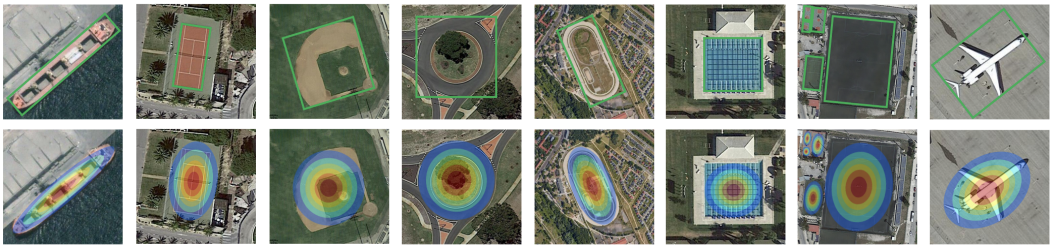


Figure 2: **Top:** rotated bounding box $\mathcal{B}(x, y, h, w, \theta)$. **Bottom:** 2-D Gaussian distribution $\mathcal{N}(\mu, \mathbf{\Sigma})$.

Rotated Object Detection. Recent advances in rotation detection [6; 47; 27; 43; 44; 40] are mainly driven by adapting the horizontal object detectors with rotated bounding boxes to represent multi-oriented objects. To accurately predict the rotated bounding box, most rotation detection methods extend the l_n-norm [12; 20; 2; 6; 24] used in horizontal detection, or construct a derivable approximate IoU loss [47; 4; 45]. From scratch, we try to change the design of rotation regression loss from induction paradigm to deduction methodology, which in fact is a generalization to the horizontal case.

3 Approach

3.1 Design the Rotation Regression Loss by Induction Paradigm

Regression loss is a vital part of most current object detection algorithms. For horizontal bounding box regression, the model mainly outputs four items in terms of location and size:

$$t_x^p = \frac{x_p - x_a}{w_a}, t_y^p = \frac{y_p - y_a}{h_a}, t_w^p = \ln\left(\frac{w_p}{w_a}\right), t_h^p = \ln\left(\frac{h_p}{h_a}\right) \tag{1}$$

to match the four targets from the ground truth

$$t_x^t = \frac{x_t - x_a}{w_a}, t_y^t = \frac{y_t - y_a}{h_a}, t_w^t = \ln\left(\frac{w_t}{w_a}\right), t_h^t = \ln\left(\frac{h_t}{h_a}\right) \tag{2}$$

where x, y, h, w denote the center coordinates, height and width, respectively. Variables x_t, x_a, x_p are for the ground-truth box, anchor box, and predicted box, respectively (likewise for y, w, h).

Extending the above horizontal case, most existing rotation detection algorithms [41; 6; 8; 27; 24] are also based on regression loss which simply involves an extra angle parameter θ:

$$t_\theta^p = f(\theta_p - \theta_a), t_\theta^t = f(\theta_t - \theta_a) \tag{3}$$

where $f(\cdot)$ is used to deal with angular periodicity, such as trigonometric functions, modulo, etc.

The overall regression loss for rotation detection is calculated as follows:

$$L_{reg} = l_n\text{-norm}\left(\Delta t_x, \Delta t_y, \ln \Delta t_w, \ln \Delta t_h, \Delta t_\theta\right) \tag{4}$$

where

$$\begin{aligned} \Delta t_x =& t_x^p - t_x^t = \frac{\Delta x}{w_a}, \quad \Delta t_y = t_y^p - t_y^t = \frac{\Delta y}{h_a} \\ \Delta t_w =& t_w^p - t_w^t = \frac{w_p}{w_t}, \quad \Delta t_h = t_h^p - t_h^t = \frac{h_p}{h_t}, \quad \Delta t_\theta = t_\theta^p - t_\theta^t = \Delta\theta \end{aligned} \tag{5}$$

It can be seen that different parameters are optimized independently, making the loss (or detection accuracy) sensitive to the under-fitting of any of the parameters. This mechanism is fatal to high-precision detection. Taking the left side of Figure 3 as an example, the detection result based on the Smooth L1 loss often shows the deviation of the center point or angle. Moreover, different types of objects have different sensitivity to these five parameters. For example, the angle parameter is very important for detecting objects with large aspect ratios. This requires the user to select an appropriate set of weights given a specific single object sample during the training process, which is nontrivial or even unrealistic.

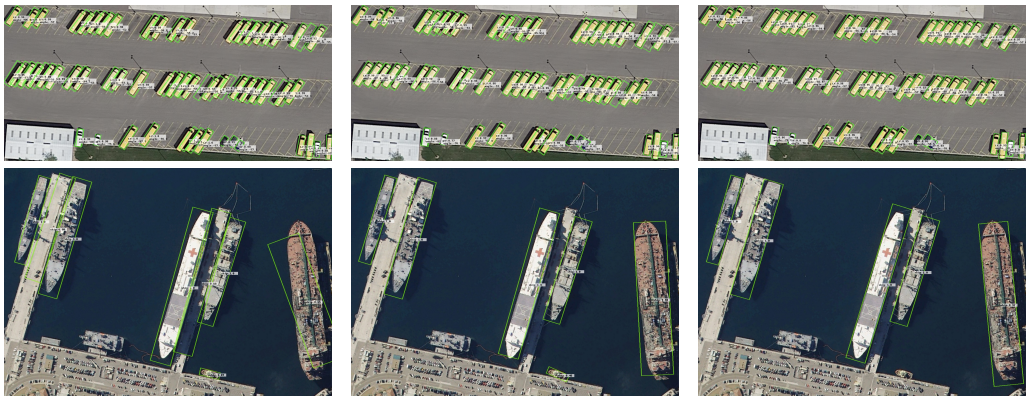


Figure 3: Visual comparison between Smooth L1 loss (left), GWD (middle) and KLD (right). Since the center point parameters in Smooth L1 Loss and GWD are independently optimized, their prediction results are slightly shifted. In contrast, the KLD-based prediction results are closer to the object boundary and show strong robustness in dense scenes. Similarly, the prediction angle of Smooth L1 Loss is not as accurate as KLD.

3.2 Design the Rotation Regression Loss by Deduction Paradigm

Gaussian distribution representation of the arbitrary-oriented box. Intersection over Union (IoU) loss is an effective coupled regression loss, and its multiple variants [31; 52] are widely used in horizontal box detection. However, due to the non-differentiable problem of IoU calculation between rotated boxes, we cannot directly apply it to rotation detection. Considering that IoU is a similarity function, we plan to use a distance function to approximate it. First of all, we convert a arbitrary-oriented bounding box $\mathcal{B}(x, y, h, w, \theta)$ into a 2-D Gaussian distribution $\mathcal{N}(\mu, \mathbf{\Sigma})$, as illustrated in Figure 2. Then calculate the distance between two Gaussian distributions as the final regression loss. Specifically, the conversion relationship is expressed as follows:

$$
\begin{aligned}
\mu =& (x, y)^{\top} \\
\mathbf{\Sigma}^{1/2} =& \mathbf{R}\mathbf{\Lambda}\mathbf{R}^{\top} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} \frac{w}{2} & 0 \\ 0 & \frac{h}{2} \end{pmatrix} \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \\
=& \begin{pmatrix} \frac{w}{2}\cos^2\theta + \frac{h}{2}\sin^2\theta & \frac{w-h}{2}\cos\theta\sin\theta \\ \frac{w-h}{2}\cos\theta\sin\theta & \frac{w}{2}\sin^2\theta + \frac{h}{2}\cos^2\theta \end{pmatrix}
\end{aligned}
\tag{6}
$$

where $\mathbf{R}$ represents the rotation matrix, and $\mathbf{\Lambda}$ represents the diagonal matrix of eigenvalues.

The above-mentioned parameter space conversion realizes the preliminary parameter coupling, (h, w, θ) into $\mathbf{\Sigma}$. On this basis, we further studies how to design high-precision detection regression loss through new parameter space $\mathcal{N}(\mu, \mathbf{\Sigma})$. Our view is that the coupling of parameters is positively correlated with the final high-precision performance. Compared with the currently widely used parameter fully decoupled l_n-norm loss, we will introduce two coupled regression losses.

Gaussian Wasserstein Distance. The Wasserstein distance [33; 45] between two probability measures μ and ν on $\mathbb{R}^n$ expressed as:

$$
\mathbf{D}_w(\mu; \nu) = \inf \mathbb{E}(\|\mathbf{X} - \mathbf{Y}\|_2^2)^{1/2} \tag{7}
$$

where the inferior runs over all random vectors $(\mathbf{X}, \mathbf{Y})$ of $\mathbb{R}^n \times \mathbb{R}^n$ with $\mathbf{X} \sim \mu$ and $\mathbf{Y} \sim \nu$. It turns out that we have the following formula:

$$
\mathbf{D}_w(\mathcal{N}_p, \mathcal{N}_t)^2 = \underbrace{\|\mu_p - \mu_t\|_2^2}_{\text{center distance}} + \underbrace{\mathbf{Tr}(\mathbf{\Sigma}_p + \mathbf{\Sigma}_t - 2(\mathbf{\Sigma}_p^{1/2}\mathbf{\Sigma}_t\mathbf{\Sigma}_p^{1/2})^{1/2})}_{\text{coupling terms about } h_p,\, w_p \text{ and } \theta_p} \tag{8}
$$

Eq. 8 shows that the Gaussian Wasserstein Distance (GWD) is mainly divided into two parts: the distance between the center points (x, y) and the coupling terms about h, w and θ. Accordingly, the regression loss based on GWD can be regarded as a semi-coupled loss. Although the independent optimization of the center point may make the detection result slightly shifted (see Figure 3), GWD can still greatly improve the performance of high-precision rotation detection due to the coupling between part of the parameters.

When all the boxes are horizontal ($\theta = 0°$), Eq. 8 can be further simplified:

$$\begin{aligned}\mathbf{D}_w^h(\mathcal{N}_p, \mathcal{N}_t)^2 =& \|\mu_p - \mu_t\|_2^2 + \|\mathbf{\Sigma}_p^{1/2} - \mathbf{\Sigma}_t^{1/2}\|_F^2 \\ =& (x_p - x_t)^2 + (y_p - y_t)^2 + \left((w_p - w_t)^2 + (h_p - h_t)^2\right)/4 \\ =& l_2\text{-norm}(\Delta x, \Delta y, \Delta w/2, \Delta h/2)\end{aligned} \tag{9}$$

where $\|\cdot\|_F$ is the Frobenius norm. Although Eq. 4 and 9 are not completely consistent, Eq. 9 can still be used as the regression loss of horizontal detection.

Kullback-Leibler Divergence. In order to explore the more appropriate regression loss, we further adopt the Kullback-Leibler divergence (KLD) [14]. Similarly, the KLD between two 2-D Gaussian distributions can be expressed as:

$$\mathbf{D}_{kl}(\mathcal{N}_p||\mathcal{N}_t) = \underbrace{\frac{1}{2}(\mu_p - \mu_t)^\top \mathbf{\Sigma}_t^{-1}(\mu_p - \mu_t)}_{\text{term about } x_p \text{ and } y_p} + \underbrace{\frac{1}{2}\mathbf{Tr}(\mathbf{\Sigma}_t^{-1}\mathbf{\Sigma}_p) + \frac{1}{2}\ln\frac{|\mathbf{\Sigma}_t|}{|\mathbf{\Sigma}_p|}}_{\text{coupling terms about } h_p,\, w_p \text{ and } \theta_p} - 1 \tag{10}$$

or

$$\mathbf{D}_{kl}(\mathcal{N}_t||\mathcal{N}_p) = \underbrace{\frac{1}{2}(\mu_p - \mu_t)^\top \mathbf{\Sigma}_p^{-1}(\mu_p - \mu_t) + \frac{1}{2}\mathbf{Tr}(\mathbf{\Sigma}_p^{-1}\mathbf{\Sigma}_t) + \frac{1}{2}\ln\frac{|\mathbf{\Sigma}_p|}{|\mathbf{\Sigma}_t|}}_{\text{chain coupling of all parameters}} - 1 \tag{11}$$

It can be seen that each item in $\mathbf{D}_{kl}(\mathcal{N}_t||\mathcal{N}_p)$ is composed of partial parameter coupling, which makes all parameters form a chain coupling relationship. In the optimization process of the KLD-based detector, the parameters influence each other and are jointly optimized. In contrast, $\mathbf{D}_{kl}(\mathcal{N}_p||\mathcal{N}_t)$ and GWD are both semi-coupled, but $\mathbf{D}_{kl}(\mathcal{N}_p||\mathcal{N}_t)$ has a better central point optimization mechanism.

Although KLD is asymmetric, we find that the optimization principles of these two forms are similar by analyzing the gradients of various parameters and experimental results. Take the relatively simple $\mathbf{D}_{kl}(\mathcal{N}_p||\mathcal{N}_t)$ as an example, according to Eq. 6, each item of Eq. 10 can be expressed as

$$(\mu_p - \mu_t)^\top \mathbf{\Sigma}_t^{-1}(\mu_p - \mu_t) = \frac{4\left(\Delta x \cos\theta_t + \Delta y \sin\theta_t\right)^2}{w_t^2} + \frac{4\left(\Delta y \cos\theta_t - \Delta x \sin\theta_t\right)^2}{h_t^2} \tag{12}$$

$$\mathbf{Tr}(\mathbf{\Sigma}_t^{-1}\mathbf{\Sigma}_p) = \frac{h_p^2}{w_t^2}\sin^2\Delta\theta + \frac{w_p^2}{h_t^2}\sin^2\Delta\theta + \frac{h_p^2}{h_t^2}\cos^2\Delta\theta + \frac{w_p^2}{w_t^2}\cos^2\Delta\theta \tag{13}$$

$$\ln\frac{|\mathbf{\Sigma}_t|}{|\mathbf{\Sigma}_p|} = \ln\frac{h_t^2}{h_p^2} + \ln\frac{w_t^2}{w_p^2} \tag{14}$$

where $\Delta x = x_p - x_t, \Delta y = y_p - y_t, \Delta\theta = \theta_p - \theta_t$. When $\theta_t = 0°$, we can get

$$\frac{\partial f_{kl}(\mu_p)}{\partial \mu_p} = \left(\frac{4}{w_t^2}\Delta x, \frac{4}{h_t^2}\Delta y\right)^\top \tag{15}$$

The weights $1/w_t^2$ and $1/h_t^2$ will make the model dynamically adjust the optimization of the object position according to the scale. For example, when the object scale is small or an edge is too short, the model will pay more attention to the optimization of the offset of the corresponding direction. For this kind of object, a slight deviation on the corresponding direction will often cause a sharp drop in IoU. In contrast, the gradient of the center point in GWD is $\frac{\partial f_w(\mu_p)}{\partial \mu_p} = (2\Delta x, 2\Delta y)^\top$. Therefore, the detection result of the GWD-based model will show a slight deviation, while the detection result of the KLD-based model is quite accurate, as shown in Figure 3. Without loss of generality, when $\theta_t \neq 0°$, the gradient of the object offset (Δx and Δy) will be dynamically adjusted according to the θ_t for better optimization.

For h_p and w_p, we have

$$\frac{\partial f_{kl}(\mathbf{\Sigma}_p)}{\partial \ln h_p} = \frac{h_p^2}{h_t^2}\cos^2\Delta\theta + \frac{h_p^2}{w_t^2}\sin^2\Delta\theta - 1, \quad \frac{\partial f_{kl}(\mathbf{\Sigma}_p)}{\partial \ln w_p} = \frac{w_p^2}{w_t^2}\cos^2\Delta\theta + \frac{w_p^2}{h_t^2}\sin^2\Delta\theta - 1 \tag{16}$$

On the one hand, the optimization of the h_p and w_p is affected by the $\Delta\theta$. When $\Delta\theta = 0°$, $\frac{\partial f_{kl}(\mathbf{\Sigma}_p)}{\partial \ln h_p} = \frac{h_p^2}{h_t^2} - 1$, $\frac{\partial f_{kl}(\mathbf{\Sigma}_p)}{\partial \ln w_p} = \frac{w_p^2}{w_t^2} - 1$, which means that the smaller targeted height or width leads to heavier penalty on its matching loss. This is desirable, as smaller height or width needs higher

matching precision. On the other hand, the optimization of the $\Delta\theta$ is also affected by the h_p and w_p, as follows:

$$\frac{\partial f_{kl}(\mathbf{\Sigma}_p)}{\partial \theta_p} = \left(\frac{h_p^2 - w_p^2}{w_t^2} + \frac{w_p^2 - h_p^2}{h_t^2} \right) \sin 2\Delta\theta \tag{17}$$

when $w_p = w_t, h_p = h_t$, then $\frac{\partial f_{kl}(\mathbf{\Sigma}_p)}{\partial \theta_p} = \left(\frac{h_t^2}{w_t^2} + \frac{w_t^2}{h_t^2} - 2 \right) \sin 2\Delta\theta \geq \sin 2\Delta\theta$. This shows that the larger the aspect ratio of the object, the model will pay more attention to the optimization of the angle. This is the main reason why the KLD-based model has a huge advantage in high-precision detection indicators in the experiment section. Through the above analysis, we find that when one of the parameters is optimized, the other parameters will be used as its weight to dynamically adjust the optimization rate. In other words, the optimization of parameters is no longer independent, that is, optimizing one parameter will also promote the optimization of other parameters. The optimization of this virtuous circle is the key to KLD as a generalized regression loss.

In the horizontal detection task, combine with Eq. 10 to Eq. 14, we can get

$$\begin{aligned} \mathbf{D}_{kl}^{h}(\mathcal{N}_p||\mathcal{N}_t) =& \frac{1}{2} \left(\frac{w_p^2}{w_t^2} + \frac{h_p^2}{h_t^2} + \frac{4\Delta^2 x}{w_t^2} + \frac{4\Delta^2 y}{h_t^2} + \ln \frac{w_t^2}{w_p^2} + \ln \frac{h_t^2}{h_p^2} - 2 \right) \\ =& 2l_2\text{-norm}(\Delta t_x, \Delta t_y) + l_1\text{-norm}(\ln \Delta t_w, \ln \Delta t_h) + \frac{1}{2} l_2\text{-norm}(\frac{1}{\Delta t_w}, \frac{1}{\Delta t_h}) - 1 \end{aligned} \tag{18}$$

where the first two terms of Eq. 18 are very similar to Eq. 4, and the divisor part of the two terms x and y is the main difference ($\frac{\Delta x}{w_t}$ vs $\frac{\Delta x}{w_a}$).

In addition, $\mathbf{D}_{kl}(\mathcal{N}_t||\mathcal{N}_p)$ has similar properties, refer to the supplementary materials for details. We have also introduced some variants of KLD to further verify the influence of asymmetry on rotation detection can be ignored. The variants mainly including

$$\begin{aligned} \mathbf{D}_{kl_min(max)}(\mathcal{N}_p||\mathcal{N}_t) =& \min(\max)(\mathbf{D}_{kl}(\mathcal{N}_p||\mathcal{N}_t), \mathbf{D}_{kl}(\mathcal{N}_t||\mathcal{N}_p)) \\ \mathbf{D}_{js}(\mathcal{N}_p||\mathcal{N}_t) =& \frac{1}{2} \left(\mathbf{D}_{kl}(\mathcal{N}_t||\frac{\mathcal{N}_p + \mathcal{N}_t}{2}) + \mathbf{D}_{kl}(\mathcal{N}_p||\frac{\mathcal{N}_p + \mathcal{N}_t}{2}) \right) [21] \\ \mathbf{D}_{jef}(\mathcal{N}_p||\mathcal{N}_t) =& \mathbf{D}_{kl}(\mathcal{N}_t||\mathcal{N}_p) + \mathbf{D}_{kl}(\mathcal{N}_p||\mathcal{N}_t))[11] \end{aligned} \tag{19}$$

Rotation Regression Loss. We normalize the distance function as our final regression loss $\mathcal{L}_{reg}$:

$$\mathcal{L}_{reg} = 1 - \frac{1}{\tau + f(\mathbf{D})}, \quad \tau \geq 1 \tag{20}$$

where $f(\cdot)$ denotes a non-linear function to transform the distance $\mathbf{D}$ to make the loss more smooth and expressive. In this paper, we mainly use two nonlinear functions, $sqrt(\mathbf{D})$ and $\ln(\mathbf{D} + 1)$. The hyperparameter τ modulates the entire loss. The multi-task loss is:

$$\mathcal{L} = \frac{\lambda_1}{N_{pos}} \sum_{n=1}^{N_{pos}} \mathcal{L}_{reg}(b_n, gt_n) + \frac{\lambda_2}{N} \sum_{n=1}^{N} \mathcal{L}_{cls}(p_n, t_n) \tag{21}$$

where N_{pos} and N indicate the number of positive and whole anchors, respectively. While b_n denotes the n-th predicted bounding box, gt_n is the n-th target ground-truth. t_n represents the label of n-th object, p_n is the n-th probability distribution of various classes calculated by sigmoid function. The hyper-parameter λ_1, λ_2 control the trade-off and are set to $\{2, 1\}$ by default. The classification loss L_{cls} is set as the focal loss [16].

4 Experiment

4.1 Datasets and Implementation Details

Our experiments are conducted over a variety of datasets, including three large-scale public datasets for aerial images i.e. DOTA [37], UCAS-AOD [56], HRSC2016 [19], as well as scene text dataset ICDAR2015 [13], MLT [25] and MSRA-TD500 [49].

DOTA is one of the largest dataset for oriented object detection in aerial images with three released versions: DOTA-v1.0, DOTA-v1.5 and DOTA-v2.0. DOTA-v1.0 contains 15 common categories, 2,806 images and 188,282 instances. The proportions of the training set, validation set, and testing

Table 1: Ablation study of the hyperparameter τ of $\mathcal{L}_G$ on HRSC2016.

Method	Loss	$\mathbf{D}_{kl}$	$f(\mathbf{D}_{kl})$	$\mathcal{L}_G(f(\mathbf{D}_{kl}),\tau)$			
				$\tau=1$	$\tau=2$	$\tau=3$	$\tau=5$
RetinaNet	$f(\mathbf{D}_{kl}) = sqrt(\mathbf{D}_{kl})$	0.20	82.96	84.85	84.15	75.23	73.32
	$f(\mathbf{D}_{kl}) = \log(\mathbf{D}_{kl}+1)$		83.23	**85.25**	83.63	80.79	73.44

Table 2: Ablation of different KLD-based regression loss form. The based detector is RetinaNet.

Dataset	$\mathbf{D}_{kl}(\mathcal{N}_p\|\|\mathcal{N}_t)$	$\mathbf{D}_{kl}(\mathcal{N}_t\|\|\mathcal{N}_p)$	$\mathbf{D}_{kl_min}(\mathcal{N}_p\|\|\mathcal{N}_t)$	$\mathbf{D}_{kl_max}(\mathcal{N}_p\|\|\mathcal{N}_t)$	$\mathbf{D}_{js}(\mathcal{N}_p\|\|\mathcal{N}_t)$	$\mathbf{D}_{jeffreys}(\mathcal{N}_p\|\|\mathcal{N}_t)$
DOTA-v1.0	70.17	70.64	**70.71**	70.55	69.67	70.56
HRSC2016	82.83	83.82	83.60	82.70	**84.06**	83.66

set in DOTA-v1.0 are 1/2, 1/6, and 1/3, respectively. In contrast, DOTA-v1.5 uses the same images as DOTA-v1.0, but extremely small instances (less than 10 pixels) are also annotated. Moreover, a new category, containing 402,089 instances in total is added in this version. While DOTA-v2.0 contains 18 common categories, 11,268 images and 1,793,658 instances. Compared to DOTA-v1.5, it further includes the new categories. The 11,268 images in DOTA-v2.0 are split into training, validation, test-dev, and test-challenge sets. We divide the images into 600×600 subimages with an overlap of 150 pixels and scale it to 800×800, in line with the cropping protocol in literature.

UCAS-AOD contains 1,510 aerial images of approximately $659 \times 1,280$ pixels, with two categories of 14,596 instances in total. In line with [2; 37], we randomly select 1,110 for training and 400 for testing. HRSC2016 contains images from two scenarios including ships on sea and ships close inshore. The training, validation and test set include 436, 181 and 444 images.

ICDAR2015, MLT and MSRA-TD500 are commonly used for oriented scene text detection and spotting. ICDAR2015 includes 1,000 training images and 500 testing images. ICDAR2017 MLT is a multi-lingual text dataset, which includes 7,200 training images, 1,800 validation images and 9,000 testing images. MSRA-TD500 dataset consists of 300 training images and 200 testing images.

We use Tensorflow [1] to implement the proposed methods on a server with Tesla V100 and 32G memory. The experiments are all initialized by ResNet50 [10] by default unless otherwise specified. Weight decay and momentum are set 0.0001 and 0.9, respectively. We employ MomentumOptimizer over 8 GPUs with a total of 8 images per minibatch (1 image per GPU).

All the used datasets are trained by 20 epochs in total, and the learning rate is reduced tenfold at 12 epochs and 16 epochs, respectively. The initial learning rate is set to 5e-4. The number of image iterations per epoch for DOTA-v1.0, DOTA-v1.5, DOTA-v1.0, UCAS-AOD, HRSC2016, ICDAR2015, MLT and MSRA-TD500 are 54k, 64k, 80k, 5k, 10k, 10k, 10k, 10k and 5k respectively, and doubled if data augmentation or multi-scale training is used.

4.2 Ablation Study and Further Comparison

Regression loss form and hyperparameter. Table 1 compares three forms of KLD-based regression loss on HRSC2016, including $\mathbf{D}_{kl}$, $f(\mathbf{D}_{kl})$ and $\mathcal{L}_{reg}(f(\mathbf{D}_{kl}),\tau)$. Due to extreme sensitivity to large errors, the performance of $\mathbf{D}_{kl}$ is extremely poor, only **0.20%**. Through a simple nonlinear linear transformation, the performance can be increased to **82.96%** and **83.23%** corresponding to $sqrt$ and log. We further perform a detailed hyperparameter experiment on the loss $\mathcal{L}_{reg}$ proposed in this paper, and the performance reaches the optimal when $\tau = 1, f(\mathbf{D}_{kl}) = \log(\mathbf{D}_{kl}+1)$, about **85.25%**. Keeping the same loss pattern, we compare six KLD-based distance functions in Table 2, and conclude that the asymmetry of KLD does not have much impact on performance. In subsequent experiments, we use $\mathcal{L}_{reg}(\log(\mathbf{D}_{kl}(\mathcal{N}_p||\mathcal{N}_t)),1)$ as the basic setting.

High-precision detection experiment. We expect that the designed rotation regression loss can show advantages in high-precision detection. Table 3 shows the comparison of the high-precision detection results of three different regression losses using Smooth L1, GWD and KLD on different datasets and different detectors. For the HRSC206 dataset containing a large number of ship with large aspect ratios, GWD-based RetinaNet has a **11.89%** improvement over Smooth L1 on AP_{75}, KLD even gets a **23.97%** gain. Even with a stronger R^3Det detector, KLD and GWD still increased by **33.96%** and **22.46%** in AP_{75}, and **15.22%** and **9.89%** in $AP_{50:95}$. The same experimental conclusion are also reflected in the other two scene text datasets MASR-TF500 and ICDAR2015, which is KLD > GWD > Smooth L1. In general, the coupling optimization of parameters has a

Table 3: High-precision detection experiment under different regression loss. 'R', 'F' and 'G' indicate random rotation, flipping, and graying, respectively.

Method	Dataset	Data Aug.	Reg. Loss	$\text{Hmean}_{75}/\text{AP}_{50}$	$\text{Hmean}_{75}/\text{AP}_{60}$	$\text{Hmean}_{75}/\text{AP}_{75}$	$\text{Hmean}_{75}/\text{AP}_{85}$	$\text{Hmean}_{50:95}/\text{AP}_{50:95}$
RetinaNet	HRSC2016	R+F+G	Smooth L1	84.28	74.74	48.42	12.56	47.76
			GWD	85.56 (+1.28)	84.04 (+9.30)	60.31 (+11.89)	17.14 (+4.58)	52.89 (+5.13)
			KLD	**87.45** (+3.17)	**86.72** (+11.98)	**72.39** (+23.97)	**27.68** (+15.12)	**57.80** (+10.04)
R^3Det			Smooth L1	88.52	79.01	43.42	4.58	46.18
			GWD	89.43 (+0.91)	88.89 (+9.88)	65.88 (+22.46)	15.02 (+10.44)	56.07 (+9.89)
			KLD	**89.97** (+1.45)	**89.73** (+10.72)	**77.38** (+33.96)	**25.12** (+20.54)	**61.40** (+15.22)
RetinaNet	MSRA-TD500	R+F+G	Smooth L1	70.98	62.42	36.73	12.56	37.89
			GWD	76.76 (+5.78)	68.58 (+6.16)	44.21 (+7.48)	17.75 (+5.19)	43.62 (+5.73)
			KLD	**76.96** (+5.98)	**70.08** (+7.66)	**46.95** (+10.22)	**19.59** (+7.03)	**45.24** (+7.35)
	ICDAR2015	F	Smooth L1	69.78	64.15	36.97	8.71	37.73
			GWD	74.29 (+4.51)	68.34 (+4.19)	43.39 (+6.42)	10.50 (+1.79)	41.68 (+3.95)
			KLD	**75.32** (+5.54)	**69.94** (+5.79)	**44.46** (+7.49)	**10.70** (+1.99)	**42.68** (+4.95)
		R+F	Smooth L1	74.83	69.46	42.02	11.59	41.98
			GWD	76.15 (+1.32)	71.26 (+1.80)	**45.59** (+3.57)	**11.65** (+0.06)	43.58 (+1.60)
			KLD	**77.92** (+3.09)	**72.77** (+3.31)	43.27 (+1.25)	11.09 (-0.50)	**43.65** (+1.67)
R^3Det		F	Smooth L1	74.28	68.12	35.73	8.01	39.10
			GWD	75.59 (+1.31)	68.36 (+0.24)	40.24 (+4.51)	9.15 (+1.14)	40.80 (+1.70)
			KLD	**77.72** (+2.43)	**71.99** (+3.87)	**43.95** (+8.22)	**10.43** (+2.42)	**43.29** (+4.19)
		R+F	Smooth L1	75.53	69.69	37.69	9.03	40.56
			GWD	77.09 (+1.56)	71.52 (+1.83)	41.08 (+3.39)	10.10 (+1.07)	42.17 (+1.61)
			KLD	**79.63** (+4.63)	**73.30** (+3.61)	**43.51** (+5.82)	**10.61** (+1.58)	**43.61** (+3.05)

Table 4: More ablation experiments on other datasets.

Method	Reg. Loss	MLT	UCAS-AOD	DOTA-v1.0	DOTA-v1.5	DOTA-v2.0
RetinaNet	Smooth L1	48.42	94.56	65.73	58.87	44.16
	GWD	54.58 (+6.16)	95.44 (+0.88)	68.93 (+3.20)	60.03 (+1.16)	46.65 (+2.49)
	KLD	**57.59** (+9.17)	**96.14** (+1.58)	**71.28** (+5.55)	**62.50** (+3.63)	**47.69** (+3.53)

Table 5: Accuracy comparison between different rotation detectors on DOTA dataset. † and ‡ represent the large aspect ratio object and the square-like object, respectively. The bold **red** and **blue** fonts indicate the top two performances respectively. D_{oc} and D_{le} represent OpenCV Definition ($\theta \in [-90^{\circ}, 0^{\circ})$) and Long Edge Definition ($\theta \in [-90^{\circ}, 90^{\circ})$) of RBox.

Baseline	Method	Box Def.	v1.0 tranval/test									v1.0 train/val			v1.5	v2.0
			BR†	SV†	LV†	SH†	HA†	ST‡	RA‡	7-AP_{50}	AP_{50}	AP_{50}	AP_{75}	$\text{AP}_{50:95}$	AP_{50}	AP_{50}
RetinaNet	-	D_{oc}	42.17	65.93	51.11	72.61	53.24	78.38	62.00	60.78	65.73	64.70	32.31	34.50	58.87	44.16
	-	D_{le}	38.31	60.48	49.77	68.29	51.28	78.60	60.02	58.11	64.17	62.21	26.06	31.49	56.10	43.06
	IoU-Smooth L1	D_{oc}	44.32	63.03	51.25	72.78	56.21	77.98	63.22	61.26	66.99	64.61	34.17	36.23	59.16	46.31
	Modulated Loss	D_{oc}	42.92	67.92	52.91	72.67	53.64	**80.22**	58.21	61.21	66.05	63.50	33.32	34.61	-	-
	Modulated Loss	Quad.	43.21	70.78	54.70	72.68	**60.99**	79.72	62.08	63.45	67.20	65.15	**40.59**	**39.12**	**61.42**	**46.71**
	CSL	D_{le}	42.25	68.28	54.51	72.85	53.10	75.59	58.99	60.80	67.38	64.40	32.58	35.04	58.55	43.34
	DCL (BCL)	D_{le}	41.40	65.82	56.27	73.80	54.30	79.02	60.25	61.55	67.39	**65.93**	35.66	36.71	59.38	45.46
	GWD	D_{oc}	**44.07**	**71.92**	**62.56**	**77.94**	60.25	79.64	**63.52**	**65.70**	**68.93**	65.44	38.68	38.71	60.03	46.65
	KLD	D_{oc}	**44.00**	**74.45**	**72.48**	**84.30**	**65.54**	**80.03**	**65.05**	**69.41**	**71.28**	**68.14**	**44.48**	**42.15**	**62.50**	**47.69**
R^3Det	-	D_{oc}	44.15	75.09	72.88	**86.04**	56.49	82.53	61.01	68.31	70.66	67.18	38.41	38.46	62.91	48.43
	DCL (BCL)	D_{le}	**46.84**	74.87	74.96	85.70	57.72	**84.06**	**63.77**	69.70	71.21	67.45	35.44	37.54	61.98	48.71
	GWD	D_{oc}	46.73	**75.84**	**78.00**	**86.71**	**62.69**	**83.09**	61.12	**70.60**	**71.56**	**69.28**	**43.35**	**41.56**	**63.22**	**49.25**
	KLD	D_{oc}	**48.34**	**75.09**	**78.88**	86.52	**65.48**	82.08	**61.51**	**71.13**	**71.73**	**68.87**	**44.48**	**42.11**	**65.18**	**50.90**

significant help for high-precision detection, and the higher the degree of coupling, the better the result. For a more intuitive comparison, we visually compare these three regression losses, as shown in Figure 3. Since the center point (x, y) parameters in Smooth L1 Loss and GWD are independently optimized, their prediction results are slightly shifted. In contrast, the KLD-based prediction results are closer to the object boundary and show strong robustness in dense scenes. Similarly, GWD-based or KLD-based model has more accurate angle prediction capabilities than Smooth L1-based model due to their angle parameters (θ) are not independently optimized.

Ablation study on more datasets. In order to make the obtained results more credible, we continue to verify on the other five datasets, as shown in Table 4. The performance improvement of KLD on the three data sets of MLT, UCAS-AOD and DOTA-v1.0 is still considerable, with an increase of **9.17%**, **1.58%**, and **5.55%** respectively. It is worth noting that for DOTA-v1.5 and DOTA-v2.0, which contain a large number of small objects (less than 10 pixels), the regression loss of parameter coupling still outperforms, especially KLD has achieved significant gains of **3.63%** and **3.53%**.

Comparison of peer methods. Table 5 compares the five peer techniques, including IoU-Smooth L1 Loss [47], Modulated loss [28], CSL [43], and DCL [40] on DOTA-v1.0. For fairness, these methods are all implemented on the same baseline method, and are trained and tested under the same environment and hyperparameters. We detail the accuracy of the seven categories, including large aspect ratio (e.g. BR, SV, LV, SH, HA) and square-like object (e.g. ST, RD), which can better reflect the real-world challenges and advantages of our method. Without bells and whistles, the combination of RetinaNet and KLD directly surpasses R^3Det (**71.28%** vs **70.66%** in AP_{50} and **69.41%** vs **68.31%**

Table 6: Performance evaluation of KLD on horizontal detection.

Detector	Reg. Loss	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l	Detector	Reg. Loss	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
RetinaNet	Smooth L1	37.2	56.6	39.7	21.4	41.1	48.0	Faster RCNN	Smooth L1	37.9	58.8	41.0	22.4	41.4	49.1
	GIoU	37.4	56.7	39.7	22.2	41.7	48.1		GIoU	**38.3**	58.7	41.5	22.5	41.7	49.7
	KLD	**38.0**	56.4	40.6	23.3	43.2	49.3		KLD	38.2	58.7	41.7	22.6	41.8	49.3

Table 7: AP on different objects on DOTA-v1.0. Here R-101 denotes ResNet-101 (likewise for R-50, R-152), and RX-101 and H-104 represent ResNeXt101 [38] and Hourglass-104 [26], respectively. MS indicates that multi-scale training or testing is used.

	Method	Backbone	MS	PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SBF	RA	HA	SP	HC	AP_{50}
Two-stage	ICN [2]	R-101	✓	81.40	74.30	47.70	70.30	64.90	67.80	70.00	90.80	79.10	78.20	53.60	62.90	67.00	64.20	50.20	68.20
	RoI-Trans. [6]	R-101	✓	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 [47]	R-101	✓	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
	Gliding Vertex [39]	R-101		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
	Mask OBB [34]	RX-101	✓	89.56	85.95	54.21	72.90	76.52	74.16	85.63	89.85	83.81	86.48	54.89	69.64	73.94	69.06	63.32	75.33
	CenterMap OBB [35]	R-101	✓	89.83	84.41	54.60	70.25	77.66	78.32	87.19	90.66	84.89	85.27	56.46	69.23	74.13	71.56	66.06	76.03
	FPN-CSL [43]	R-152	✓	**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
	RSDet-II [28]	R-152	✓	89.93	84.45	53.77	74.35	71.52	78.31	78.12	**91.14**	87.35	86.93	65.64	65.17	75.35	**79.74**	63.31	76.34
	SCRDet++ [46]	R-101	✓	**90.05**	84.39	55.44	73.99	77.54	71.11	86.05	90.67	87.32	87.08	69.62	68.90	73.74	71.29	65.08	76.81
	ReDet [9]	ReR-50	✓	88.81	82.48	**60.83**	80.82	78.34	**86.06**	88.31	90.87	**88.77**	87.03	68.65	66.90	**79.26**	**79.71**	74.67	80.10
Single-stage	PIoU [4]	DLA-34		80.90	69.70	24.10	60.20	38.30	64.40	64.80	**90.90**	77.20	70.40	46.50	37.10	57.10	61.9	64.00	60.50
	O^2-DNet [36]	H-104	✓	89.31	82.14	47.33	61.21	71.32	74.03	78.62	90.76	82.23	81.36	60.93	60.17	58.21	66.98	61.03	71.04
	DAL [24]	R-101	✓	88.61	79.69	46.27	70.37	65.89	76.10	78.53	90.84	79.98	78.41	58.71	62.02	69.23	71.32	60.65	71.78
	P-RSDet [54]	R-101	✓	88.58	77.83	50.44	69.29	71.10	75.79	78.66	90.88	80.10	81.71	57.92	63.03	66.30	69.77	63.13	72.30
	BBAVectors [50]	R-101	✓	88.35	79.96	50.69	62.18	78.43	78.98	87.94	90.85	83.58	84.35	54.13	60.24	65.22	64.28	55.70	72.32
	DRN [27]	H-104	✓	89.71	82.34	47.22	64.10	76.22	74.43	85.84	90.57	86.18	84.89	57.65	61.93	69.30	69.63	58.48	73.23
	PolarDet [51]	R-101	✓	89.65	**87.07**	48.14	70.97	78.53	80.34	87.45	90.76	85.63	86.87	61.64	70.32	71.92	73.09	67.15	76.64
	RDD [53]	R-101	✓	89.15	83.92	52.51	73.06	77.81	79.00	87.08	90.62	86.72	87.15	63.96	70.29	76.98	75.79	72.15	77.75
	GWD [45]	R-152	✓	89.06	84.32	55.33	77.53	76.95	70.28	83.95	89.75	84.51	86.06	**73.47**	67.77	72.60	75.76	74.17	77.43
	KLD	R-50		88.91	83.71	50.10	68.75	78.20	76.05	84.58	89.41	86.15	85.28	63.15	60.90	75.06	71.51	67.45	75.28
		R-50	✓	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
Refine-stage	CFC-Net [22]	R-101	✓	89.08	80.41	52.41	70.02	76.28	78.11	87.21	90.89	84.47	85.64	60.51	61.52	67.82	68.02	50.09	73.50
	R^3Det [44]	R-152	✓	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
	DAL [24]	R-50	✓	89.69	83.11	55.03	71.00	78.30	81.90	88.46	90.89	84.97	**87.46**	64.41	65.65	76.86	72.09	64.35	76.95
	DCL [40]	R-152	✓	89.26	83.60	53.54	72.76	79.04	82.56	87.31	90.67	86.59	86.98	67.49	66.88	73.29	70.56	69.99	77.37
	RIL [23]	R-50	✓	89.31	80.77	54.07	76.38	**79.81**	81.99	**89.13**	90.72	83.58	87.22	64.42	67.56	78.08	79.17	62.07	77.62
	S^2A-Net [8]	R-101	✓	89.28	84.11	56.95	79.21	**80.18**	82.93	**89.21**	90.86	84.66	**87.61**	71.66	68.23	**78.58**	78.20	65.55	79.15
	R^3Det-KLD	R-50		88.90	84.17	55.80	69.35	78.72	84.08	87.00	89.75	84.32	85.73	64.74	61.80	76.62	78.49	70.89	77.36
		R-50	✓	89.90	84.91	**59.21**	78.74	78.82	83.95	87.41	89.89	86.63	86.69	70.47	70.87	76.96	79.40	**78.62**	80.17
		R-152	✓	89.92	85.13	59.19	**81.33**	78.82	**84.38**	87.50	89.80	87.33	87.00	**72.57**	**71.35**	77.12	79.34	**78.68**	**80.63**

in 7-AP_{50}). Even combined with R^3Det, KLD can still further improve performance of the large aspect ratio object (**2.82%** in 7-AP_{50}) and high-precision detection (**6.07%** in AP_{75} and **3.65%** $AP_{50:95}$). KLD-based method shows the best performer in almost all indicators. Similar conclusions can still be drawn on the more challenging datasets (DOTA-v1.5 and DOTA-v2.0), which contain more data and tiny object (less than 10 pixels).

Horizontal detection verification. As analyzed by Eq. 18, KLD can be degenerated into the commonly used regression loss in horizontal detection task. In Table 6, we compare the commonly used regression loss Smooth L1 and GIoU for horizontal detection with the proposed regression loss KLD on the MS COCO [17] dataset. The results show that our KLD is not worse than other losses on the Faster RCNN and the RetinaNet, and even has an improvement of **0.6%** on RetinaNet. This provides a strong support for our original intention, which is to design a generalized regression loss for horizontal and rotation detection.

4.3 Comparisons with the State-of-the-Art Methods

The evaluation is performed on the DOTA dataset, which contains a considerable number of categories, complexity scenes. Without bells and whistles, our single-scale model RetinaNet-KLD-R50 and R^3Det-KLD-R50 achieve **75.28%** and **77.36%** respectively. They outperform many multi-scale models as shown in Table 7. With large backbone and multi-scale testing, our method further achieves state-of-the-art **80.63%**.

5 Conclusion

Departure from the vast existing literature in object detection, in this paper we have designed the regression loss for rotation detection from scratch and consider the popular horizontal object detection as its degenerated case. Specifically, we calculate the KLD between the Gaussian distributions corresponding to the rotated bounding box as the regression loss, and we find that in the learning procedure guided by the KLD loss, the gradient of the parameters can be dynamically adjusted according to the characteristics of the object which is a desirable property for robust object detection, regardless its rotation, size and aspect ratio etc. Interestingly, we have shown that KLD can be degenerated into the currently commonly used l_n-norm loss in the horizontal detection task. Extensive experimental results across different detectors and datasets show the effectiveness of our approach.

References

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pages 265–283, 2016.

[2] S. M. Azimi, E. Vig, R. Bahmanyar, M. Körner, and P. Reinartz. Towards multi-class object detection in unconstrained remote sensing imagery. In *Asian Conference on Computer Vision*, pages 150–165. Springer, 2018.

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

[4] Z. Chen, K. Chen, W. Lin, J. See, H. Yu, Y. Ke, and C. Yang. Piou loss: Towards accurate oriented object detection in complex environments. *Proceedings of the European Conference on Computer Vision*, 2020.

[5] J. Dai, Y. Li, K. He, and J. Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems*, pages 379–387, 2016.

[6] J. Ding, N. Xue, Y. Long, G.-S. Xia, and Q. Lu. Learning roi transformer for oriented object detection in aerial images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2849–2858, 2019.

[7] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015.

[8] J. Han, J. Ding, J. Li, and G.-S. Xia. Align deep features for oriented object detection. *IEEE Transactions on Geoscience and Remote Sensing*, 2021.

[9] J. Han, J. Ding, N. Xue, and G.-S. Xia. Redet: A rotation-equivariant detector for aerial object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[11] H. Jeffreys. An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 186(1007):453–461, 1946.

[12] Y. Jiang, X. Zhu, X. Wang, S. Yang, W. Li, H. Wang, P. Fu, and Z. Luo. R2cnn: rotational region cnn for orientation robust scene text detection. *arXiv preprint arXiv:1706.09579*, 2017.

[13] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu, et al. Icdar 2015 competition on robust reading. In *2015 13th International Conference on Document Analysis and Recognition*, pages 1156–1160. IEEE, 2015.

[14] S. Kullback and R. A. Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[15] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017.

[16] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2980–2988, 2017.

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

[18] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[19] Z. Liu, L. Yuan, L. Weng, and Y. Yang. A high resolution optical satellite image dataset for ship recognition and some new baselines. In *Proceedings of the International Conference on Pattern Recognition Applications and Methods*, volume 2, pages 324–331, 2017.

[20] J. Ma, W. Shao, H. Ye, L. Wang, H. Wang, Y. Zheng, and X. Xue. Arbitrary-oriented scene text detection via rotation proposals. *IEEE Transactions on Multimedia*, 20(11):3111–3122, 2018.

[21] C. Manning and H. Schutze. *Foundations of statistical natural language processing*. MIT press, 1999.

[22] Q. Ming, L. Miao, Z. Zhou, and Y. Dong. Cfc-net: A critical feature capturing network for arbitrary-oriented object detection in remote sensing images. *arXiv preprint arXiv:2101.06849*, 2021.

[23] Q. Ming, Z. Zhou, L. Miao, X. Yang, and Y. Dong. Optimization for oriented object detection via representation invariance loss. *arXiv preprint arXiv:2103.11636*, 2021.

[24] Q. Ming, Z. Zhou, L. Miao, H. Zhang, and L. Li. Dynamic anchor learning for arbitrary-oriented object detection. *arXiv preprint arXiv:2012.04150*, 2020.

[25] N. Nayef, F. Yin, I. Bizid, H. Choi, Y. Feng, D. Karatzas, Z. Luo, U. Pal, C. Rigaud, J. Chazalon, et al. Icdar2017 robust reading challenge on multi-lingual scene text detection and script identification-rrc-mlt. In *2017 14th IAPR International Conference on Document Analysis and Recognition*, volume 1, pages 1454–1459. IEEE, 2017.

[26] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *Proceedings of the European Conference on Computer Vision*, pages 483–499. Springer, 2016.

[27] X. Pan, Y. Ren, K. Sheng, W. Dong, H. Yuan, X. Guo, C. Ma, and C. Xu. Dynamic refinement network for oriented and densely packed object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11207–11216, 2020.

[28] W. Qian, X. Yang, S. Peng, J. Yan, and Y. Guo. Learning modulated loss for rotated object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

[29] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[30] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[31] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 658–666, 2019.

[32] Z. Tian, C. Shen, H. Chen, and T. He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9627–9636, 2019.

[33] C. Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

[34] J. Wang, J. Ding, H. Guo, W. Cheng, T. Pan, and W. Yang. Mask obb: A semantic attention-based mask oriented bounding box representation for multi-category object detection in aerial images. *Remote Sensing*, 11(24):2930, 2019.

[35] J. Wang, W. Yang, H.-C. Li, H. Zhang, and G.-S. Xia. Learning center probability map for detecting objects in aerial images. *IEEE Transactions on Geoscience and Remote Sensing*, 2020.

[36] H. Wei, Y. Zhang, Z. Chang, H. Li, H. Wang, and X. Sun. Oriented objects as pairs of middle lines. *ISPRS Journal of Photogrammetry and Remote Sensing*, 169:268–279, 2020.

[37] G.-S. Xia, X. Bai, J. Ding, Z. Zhu, S. Belongie, J. Luo, M. Datcu, M. Pelillo, and L. Zhang. Dota: A large-scale dataset for object detection in aerial images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3974–3983, 2018.

[38] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1492–1500, 2017.

[39] Y. Xu, M. Fu, Q. Wang, Y. Wang, K. Chen, G.-S. Xia, and X. Bai. Gliding vertex on the horizontal bounding box for multi-oriented object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[40] X. Yang, L. Hou, Y. Zhou, W. Wang, and J. Yan. Dense label encoding for boundary discontinuity free rotation detection. In *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR)*, 2021.

[41] X. Yang, H. Sun, K. Fu, J. Yang, X. Sun, M. Yan, and Z. Guo. Automatic ship detection in remote sensing images from google earth of complex scenes based on multiscale rotation dense feature pyramid networks. *Remote Sensing*, 10(1):132, 2018.

[42] X. Yang, H. Sun, X. Sun, M. Yan, Z. Guo, and K. Fu. Position detection and direction prediction for arbitrary-oriented ships via multitask rotation region convolutional neural network. *IEEE Access*, 6:50839–50849, 2018.

[43] X. Yang and J. Yan. Arbitrary-oriented object detection with circular smooth label. In *Proceedings of the European Conference on Computer Vision*, pages 677–694. Springer, 2020.

[44] X. Yang, J. Yan, Z. Feng, and T. He. R3det: Refined single-stage detector with feature refinement for rotating object. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

[45] X. Yang, J. Yan, M. Qi, W. Wang, Z. Xiaopeng, and T. Qi. Rethinking rotated object detection with gaussian wasserstein distance loss. In *International Conference on Machine Learning*, 2021.

[46] X. Yang, J. Yan, X. Yang, J. Tang, W. Liao, and T. He. Scrdet++: Detecting small, cluttered and rotated objects via instance-level feature denoising and rotation loss smoothing. *arXiv preprint arXiv:2004.13316*, 2020.

[47] X. Yang, J. Yang, J. Yan, Y. Zhang, T. Zhang, Z. Guo, X. Sun, and K. Fu. Scrdet: Towards more robust detection for small, cluttered and rotated objects. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8232–8241, 2019.

[48] Z. Yang, S. Liu, H. Hu, L. Wang, and S. Lin. Reppoints: Point set representation for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9657–9666, 2019.

[49] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu. Detecting texts of arbitrary orientations in natural images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1083–1090. IEEE, 2012.

[50] J. Yi, P. Wu, B. Liu, Q. Huang, H. Qu, and D. Metaxas. Oriented object detection in aerial images with box boundary-aware vectors. *arXiv preprint arXiv:2008.07043*, 2020.

[51] P. Zhao, Z. Qu, Y. Bu, W. Tan, Y. Ren, and S. Pu. Polardet: A fast, more precise detector for rotated target in aerial images. *arXiv preprint arXiv:2010.08720*, 2020.

[52] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren. Distance-iou loss: Faster and better learning for bounding box regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 12993–13000, 2020.

[53] B. Zhong and K. Ao. Single-stage rotation-decoupled detector for oriented object. *Remote Sensing*, 12(19):3262, 2020.

[54] L. Zhou, H. Wei, H. Li, W. Zhao, Y. Zhang, and Y. Zhang. Arbitrary-oriented object detection in remote sensing images based on polar coordinates. *IEEE Access*, 8:223373–223384, 2020.

[55] X. Zhou, D. Wang, and P. Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019.

[56] H. Zhu, X. Chen, W. Dai, K. Fu, Q. Ye, and J. Jiao. Orientation robust object detection in aerial images using deep convolutional neural network. In *2015 IEEE International Conference on Image Processing*, pages 3735–3739. IEEE, 2015.

[57] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section **??**.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [No]
   (c) Did you discuss any potential negative societal impacts of your work? [No] Our work has no potential negative societal impacts.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [No]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No] The data used in our work is open source and can be used for academic research.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] The data does not contain personally identifiable information or offensive content.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] We do not use crowdsourcing or conducted research with human subjects.
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]