Detecting Rotated Objects as Gaussian Distributions and Its 3-D Generalization

Xue Yang, Gefan Zhang, Xiaojiang Yang, Yue Zhou, Wentao Wang,
Jin Tang, Tao He, Junchi Yan *Senior Member, IEEE*

Abstract—Existing detection methods commonly use a parameterized bounding box (BBox) to model and detect (horizontal) objects and an additional rotation angle parameter is used for rotated objects. We argue that such a mechanism has fundamental limitations in building an effective regression loss for rotation detection, especially for high-precision detection with high IoU (e.g. 0.75). Instead, we propose to model the rotated objects as Gaussian distributions. A direct advantage is that our new regression loss regarding the distance between two Gaussians e.g. Kullback-Leibler Divergence (KLD), can well align the actual detection performance metric, which is not well addressed in existing methods. Moreover, the two bottlenecks i.e. boundary discontinuity and square-like problem also disappear. We also propose an efficient Gaussian metric-based label assignment strategy to further boost the performance. Interestingly, by analyzing the BBox parameters' gradients under our Gaussian-based KLD loss, we show that these parameters are dynamically updated with interpretable physical meaning, which help explain the effectiveness of our approach, especially for high-precision detection. We extend our approach from 2-D to 3-D with a tailored algorithm design to handle the heading estimation, and experimental results on twelve public datasets (2-D/3-D, aerial/text/face images) with various base detectors show its superiority.

Index Terms—Rotation Detection, Gaussian Distributions, Kullback-Leibler Divergence, 3-D Object Detection.

1 Introduction

Rotated objects are ubiquitous for visual detection scenarios, such as aerial images [1], [2], [3], scene text [4], [5], [6], [7], face [8] and 3-D objects [9], [10], retail scenes [11], [12], etc. Compared with the abundant literature on horizontal object detection [13], [14], [15], many oriented detectors build themselves upon the well established horizontal detection pipelines. However, these detectors still encounter challenges in the emerging rotation detection cases, e.g. large aspect ratio objects, dense scenes with rotated objects, especially for high-precision purpose detection, namely the detector is required to achieve high Intersection over Union (IoU) e.g. 0.75 or even higher.

Challenges in rotation regression loss design. Although the dominant line of works [2], [16], [17], [18] take a regression methodology to predict the rotation angle and have achieved state-of-the-art performance, the angle regression model suffer a few issues: i) the inconsistency between final detection metric e.g. mAP and loss function as not well addressed [19], [20], ii) boundary discontinuity [2], [21] when the loss jumps at boundary condition due to periodicity of angle and exchangeability of edges, and iii) square-like problem [22] which refers to the case that the loss still sensitive to the rotation angle, when the objects are approximately in square shape under the long edge definition. See more details in Sec. 3. These issues remain open and there lacks a unified approach. In fact, they can largely hurt the final performance, especially at the boundary condition in the sense of horizontal or vertical rotation due to the periodicity of angles, as shown in Fig. 1.

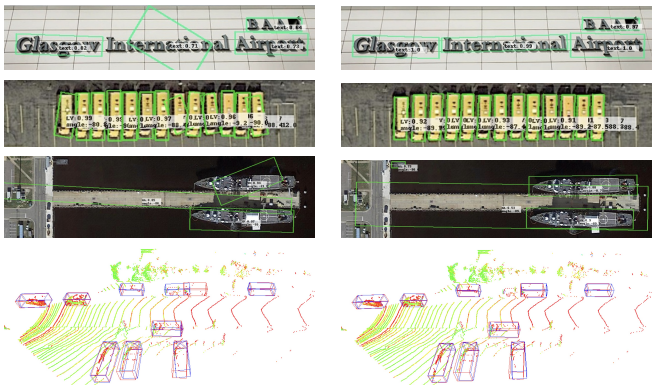


Fig. 1. Detection results comparison (top: 2-D, bottom: 3-D) at the boundary condition (i.e. horizontal or vertical rotation) between Smooth L1 loss-based (left) and the Gaussian-based (right) detectors. See illustration in Fig. 2 for the Gaussian-based bounding box detection.

Challenges in rotation regression loss implementation. To resolve the inconsistency between final detection metric which has been a pronounced challenge in literature, Skew Intersection over Union (SkewIoU) induced loss have been devised [9], [23] which is unfortunately very hard-to-implement due to the need of handling the complicated corner cases of geometry overlapping[1].

X. Yang, G. Zhang, X. Yang, Y. Zhou, W. Wang, J. Yan are with School of Electronic Information and Electrical Engineering, and MoE Key Lab of Artificial Intelligence, Shanghai Jiao Tong University, Shanghai, China. J. Yan is also with Shanghai AI Laboratory, Shanghai, China. J. Tang is with Anhui Province Key Laboratory of Multimodal Cognitive Computation, Hefei, China, and Anhui University, Hefei, China. T. He is with Cowa Robot, Co Ltd, Wuhu, China, and Anhui Province Key Laboratory of Multimodal Cognitive Computation, Hefei, China. G. Zhang is also with Cowa Robot, Co Ltd.
E-mail: {yangxue-2019-sjtu, lizaozhouke, yangxiaojiang, sjtu_zy, wwt117, yanjunchi}@sjtu.edu.cn, tj@ahu.edu.cn, tommie.he@cowarobot.com
Correspondence author: Junchi Yan.

1. See an open-source version with thousands of lines of code for implementing the loss in [23]: https://github.com/open-mmlab/mmcv/pull/1854. While our new loss only costs tens of lines of code.

Challenges in rotation regression loss optimization. Meanwhile, we argue that (and will be verified in our later technical analysis) for devising an effective rotation regression loss for high-precision rotation detection, the importance of different parameters of the bounding box (BBox) to different types of objects can vary. For example, the angle parameter (θ) and the center point parameters (x, y) are important for high aspect ratio objects and small objects, respectively. In other words, the regression loss should be self-modulated during the learning process and calls for a more dynamic optimization strategy.

Seeing the above challenges, we propose to use a Gaussian distribution to model a rotated BBox for 2-D/3-D rotated object detection, as shown in Fig. 2. Specifically, our method approximates the aforementioned metric/loss-consistent yet hard-to-implement SkewIoU loss [9], [23] between two boxes by directly calculating their distance, which can be readily fulfilled (thanks to the Gaussian parameterization) by popular metrics e.g. Gaussian Wasserstein Distance (GWD) [24], Bhattacharyya Distance (BCD) [25] and Kullback-Leibler Divergence (KLD) [26]. Our Gaussian parameterization of BBox also makes our loss immune from both boundary discontinuity [2], [21] and square-like problem [22] as shown on the right of Fig. 1.

In particular, by analyzing the gradient of the parameters during learning, we show that the optimization of one parameter will be affected by the morphological parameters of the object (as the gradient weight). It means that the model will adaptively adjust the optimization strategy given a specific configuration of an object for detection, which can lead to excellent performance in high-precision detection. In addition, KLD and BCD are proven to be scale invariant, which is an important property that Smooth L1 loss and GWD do not possess. As the horizontal BBox is a special case of the rotated BBox, we show that KLD can also be degenerated into the l_n-norm loss as commonly used in horizontal detection pipeline.

The preliminary content has partly appeared in the conference papers: [27] (GWD-based) and [28] (KLD-based)[2]. The contributions of this extended journal version are:

1) We propose to model and detect general 2-D/3-D objects using a Gaussian distribution, which is in contrast to the commonly used BBox parameterization protocol regarding with the shape (and rotation) in existing object detection works. Our work is also beyond the a few works using Gaussian to model for the specific ellipse detection tasks e.g. lesion [29] and knot [30]. In 3-D case, the object heading can be estimated by our devised post-processing algorithm.

2) Our approach naturally address the bottlenecks in existing rotation detection methods: First, we can easily derive a new and easy-to-implement SkewIoU induced regression loss (compared to the plain SkewIoU loss [23]) regarding the distance between two Gaussians e.g. KLD, can well align the actual detection accuracy metric. Second, the boundary discontinuity and square-like problem, naturally disappear regardless how the rotated BBox is defined.

3) We apply three metrics between two Gaussian distributions to establish the regression loss: Gaussian Wasserstein Distance (GWD), Bhattacharyya Distance (BCD) and Kullback-Leibler Divergence (KLD), and perform experiments on twelve public datasets (2-D/3-D, aerial/text/face images) using three popular detectors (RetinaNet [15], R^{3}Det [18], FPN [14]). The results show the effectiveness of our approach, especially for high-precision detection with high IoU (e.g. IoU = 0.75). Source code is made publicly available for both 2-D and 3-D cases (see experiment part).

4) By studying the BBox parameters' gradients with the KLD loss, we show that the regression loss is self-modulated during the learning process and parameters are dynamically optimized. It further explains the effectiveness of our loss.

5) We propose to replace the IoU with Gaussian metric for label assignment to efficiently divide positive and negative samples for better training, so that the label assignment is consistent with the regression loss. We dynamically calculate the threshold via Adaptive Training Sample Selection (ATSS) [31], and achieve further improvements.

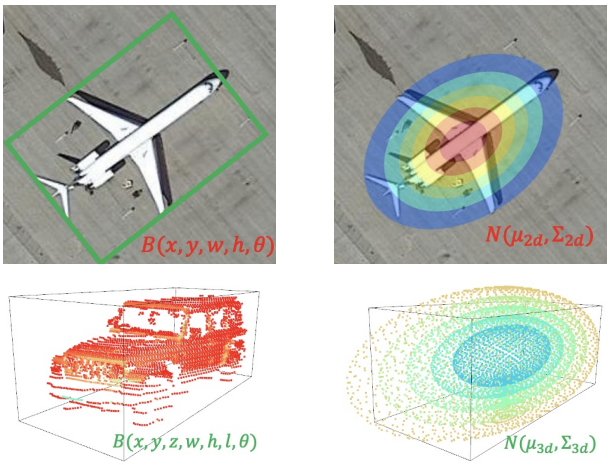


Fig. 2. A schematic diagram of modeling a rotating bounding 2-D (top) and 3-D (bottom) box by a Gaussian distribution instead of the BBox.

2 Related Work

We first review the works on rotated 2-D/3-D object detection, followed by the key challenges analysis in existing rotation detection methods. Readers are referred to [32] for comprehensive literature review on horizontal detection.

As we will show later in the paper, our proposed regression loss is coherent to existing horizontal detection loss, in the sense that it degenerates to the popular l_n-norm loss when the rotation is horizontal.

2. This journal version extends the previous two conference versions in the following aspects: i) The Bhattacharyya Distance is employed and analyzed to further show the advantages of Gaussian modeling and the importance of scale invariance, see Sec. 4.2.3 and Sec. 5.3; ii) We extend the framework based on Gaussian distribution modeling from 2-D to 3-D object detection as specified in Sec. 4.5, Fig. 10-11, and Tab. 7-8; iii) We propose a novel label assignment strategy based on Gaussian metric, combined with the dynamic threshold division by ATSS to further improve the performance, see Sec. 4.3 and Tab. 13. iv) We have added a more robust baseline, which eliminates the boundary discontinuity problem by predicting the two cos and sin components of the angle, see Sec. 4.4 and Tab. 10; v) We verify our approach on additional more challenging datasets, including FDDB, DIOR-R, and DOTA-v1.5/v2.0, see Tab. 5, Tab. 10 and Tab. 11. Among them DOTA-v1.5/v2.0 contain more samples and tiny objects (less than 10 pixels) than DOTA-v1.0; vi) We add comparative experiments between different approximate SkewIoU losses to demonstrate that the proposed technique is an easy-to-implement and better-performing alternative, as shown in Tab. 12.

2.1 Rotated 2-D/3-D object detection

As an emerging direction, advance in this area try to extend classical horizontal detectors to the rotation case by adopting the rotated BBoxes. Compared with [21], [22] that treat the rotation detection tasks as an angle classification problem, regression based detectors are more dominant. For aerial images, ICN [16], ROI-Transformer [17], SCRDet [2], Gliding Vertex [33] and ReDet [34] are representative two-stage methods whose pipeline comprises of object localization and classification, while DRN [12] and RSDet [35] are single-stage methods. To pursue the trade-off of accuracy and speed, single-stage based refined detectors, such as R^3Det [18] and S^2A-Net [36], have been proposed.

For scene text detection, RRPN [6] employs rotated RPN to generate rotated proposals and further perform rotated BBox regression. TextBoxes++ [37] adopts vertex regression on SSD [38]. RRD [7] improves TextBoxes++ by decoupling classification and BBox regression on rotation-invariant and rotation sensitive features, respectively. Most of the above detectors extend the l_n-norm loss from horizontal detector by adding extra parameters.

3-D object detection tasks requires predicting rotated BBoxes in three dimensional space. 3-D object detectors can be classified as camera-based or LiDAR-based according to the sensing modality, and our work mainly focuses on the LiDAR-based methods. Many prior works of LiDAR-based 3-D detectors focus on the design of the feature encoding paradigm from raw points. For instance, PointRCNN [39] uses PointNet++ [40] to encode the per-point features and aggregate points' features by multi-scale set abstraction. VoxelNet [41] partitions the 3-D spaces into rasterized voxels and uses PointNet [42] to encode the voxel features from raw points inside each voxel and thus generates a unified feature representation of the 3-D space. SECOND [43] simplifies VoxelNet and implements computationally efficient sparse convolution operators for its feature encoder. PointPillars [44] partitions the space into a grid of pillars, resulting in single voxel per location in the bird-eye-view feature map, which improves backbone efficiency. Similar to 2-D detection, all those methods adopt the l_n-norms regression loss.

2.2 Inconsistency between Metric and Rotation Loss

It has been shown in classic horizontal detectors that the use of IoU induced loss e.g. GIoU [19], DIoU [20] can ensure the consistency of the final detection metric and loss. However, the efforts [9], [23] of adapting such losses to rotation detection for differentiable learning is nontrivial because the calculation of SkewIoU needs to determine whether the BBox intersect, and how many intersection points, etc. incurring significant engineering as mentioned in Sec. 1.

Efforts have been made to gradient-friendly approximate SkewIoU loss. One representative work is PolarMask [45], whose calculation yet is discrete, which incurs numerical calculation error and the granularity of discretization can greatly affect the final calculation accuracy. PIoU [11] is devised by simply counting the number of pixels. To tackle the uncertainty of convex caused by rotation, [9] proposes a projection operation to estimate the intersection area.

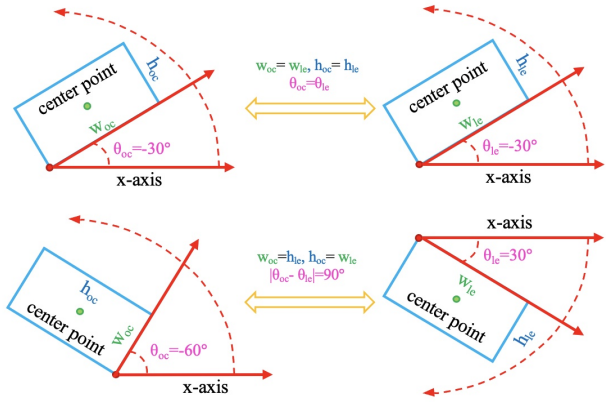


Fig. 3. Two classic definitions of rotated BBoxes. Left: OpenCV Definition D_{oc} [2], [18], Right: Long Edge Definition D_{le} [17], [34].

SCRDet [2] combines SkewIoU and Smooth L1 loss to develop an IoU-Smooth L1 loss, which partly circumvents the need for differentiable SkewIoU loss. Polygon-to-Polygon distance loss [46] is derived from the area sum of triangles specified by the vertexes of one polygon and the edges of the other. KFIoU [47] achieves a trend-level alignment with SkewIoU by Gaussian modeling and Kalman filtering.

2.3 Boundary Discontinuity and Square-like Problems

In general, due to the periodicity of angle parameters and the fundamental limitation of the classic BBox definitions (see Fig. 3), regression-based rotation detectors often suffer from the so-called boundary discontinuity and square-like problem. The first problem refers to the loss jump at the boundary condition, while the latter leads to the sensitivity of loss to the rotation change when the object is approximately square. As will be detailed in Sec. 3, these two issues in depend on the choice of BBox definition.

Existing methods try to solve part of these problems by different means. For instance, SCRDet [2] and RSDet [35] propose IoU-Smooth L1 loss and the so-called modulated loss to smooth the boundary loss jump. CSL [21], [48] transforms angular prediction from a regression task to a classification one. DCL [22] further solves square-like object detection problem under the long edge definition. Instance segmentation-based methods are practical, and relevant methods (e.g. Mask OBB [49]) have been proposed. However, there still exist limitations. First, using rotated boxes as binary masks will introduce background area, which will reduce the classification accuracy of pixels and affect the accuracy of the final prediction box. Secondly, for the top-down methods (e.g. Mask RCNN [50]), dense scenes will limit the detection of horizontal boxes because of the excessive suppression of dense horizontal overlapping BBoxes due to non-maximum suppression (NMS), thereby affecting subsequent segmentation. Aerial images often show large scenes with a large number of dense and small objects, which is not suitable for the bottom-up methods, such as SOLO [51] and CondInst [52], which assign different instances to different channels. This is the main reason why regression-based rotation detection algorithms still dominate in the field of aerial imagery.

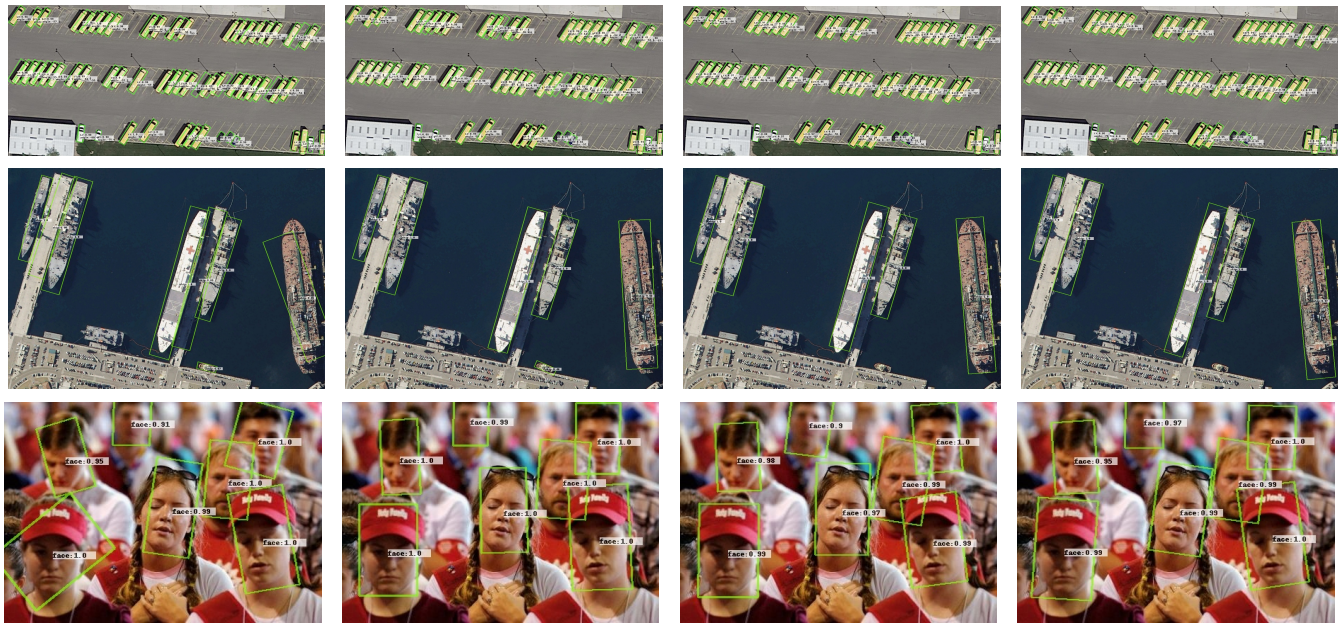


Fig. 4. High-precision detection by Smooth L1 loss, GWD, BCD and KLD (left to right). Datasets: DOTA (top) [53], HRSC2016 (bottom) [54] and FDDB [55]. Since the center point parameters in Smooth L1 Loss and GWD are independently optimized, their prediction results are slightly shifted. In contrast, the 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 Rotated Regression Detector Revisit

To motivate this work, in this section, we introduce and analyze some deficiencies in state-of-the-art rotating detectors, which are mostly based on angle regression.

3.1 Bounding Box Definition Specific Detector Design

Existing rotated object detection are mainly developed by the adaption of horizontal detectors, with classic regression loss [2], [17], [18], [56], e.g. l_n-norms, defined on the parameterization of two BBoxes in 2-D/3-D to maximize the IoU.

Fig. 3 shows two popular definitions for parameterizing rotated BBox: OpenCV protocol denoted by D_{oc} [2], [18], and long edge definition denoted by D_{le} [17], [34]. Note $\theta \in [-90^\circ, 0^\circ)$ of the former denotes the acute or right angle between w_{oc} of BBox and x-axis. While $\theta \in [-90^\circ, 90^\circ)$ of the latter definition is the angle of BBox's long edge w_{le} and x-axis. The two definitions are convertible to each other:

$$D_{le}(w_{le}, h_{le}, \theta_{le}) = \begin{cases} D_{oc}(w_{oc}, h_{oc}, \theta_{oc}), & w_{oc} \geq h_{oc} \\ D_{oc}(h_{oc}, w_{oc}, \theta_{oc} + 90^\circ), & otherwise \end{cases}$$

$$D_{oc}(w_{oc}, h_{oc}, \theta_{oc}) = \begin{cases} D_{le}(w_{le}, h_{le}, \theta_{le}), & \theta_{le} \in [-90^\circ, 0^\circ) \\ D_{le}(h_{le}, w_{le}, \theta_{le} - 90^\circ), & otherwise \end{cases}$$

Unfortunately, regardless the BBox definition, there always exist specific issues to solve, which is coupled with the definition choice (e.g. SCRDet [2], R^{3}Det [18] using OpenCV protocol D_{oc} while CSL [21], DCL [22] adopting D_{le}). Specifically, both definitions would raise the boundary discontinuity issue with a common reason of periodicity of angle (PoA) [21], while the OpenCV protocol suffers an additional cause of the exchangeability of edges (EoE) [21]. While the long edge definition meanwhile incurs the aforementioned square-like problem but not for OpenCV protocol.

Moreover, the effects of different definitions can be entangled with other factors including the network, datasets, and hyperparameters. There lacks an elegant solution to decouple the detector design from the definition choice.

3.2 Regression Loss Design Revisit: From Horizon to Rotation Detection

Regression loss is widely used in visual object detectors. For horizontal BBox regression, the model [13], [14], [15] mainly outputs four items for location and size:

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

to match the four targets from the ground truth:

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

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

Extending the above horizontal case, existing rotation detection models [1], [12], [17], [36], [57] also use regression loss which simply involves an extra angle parameter θ:

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

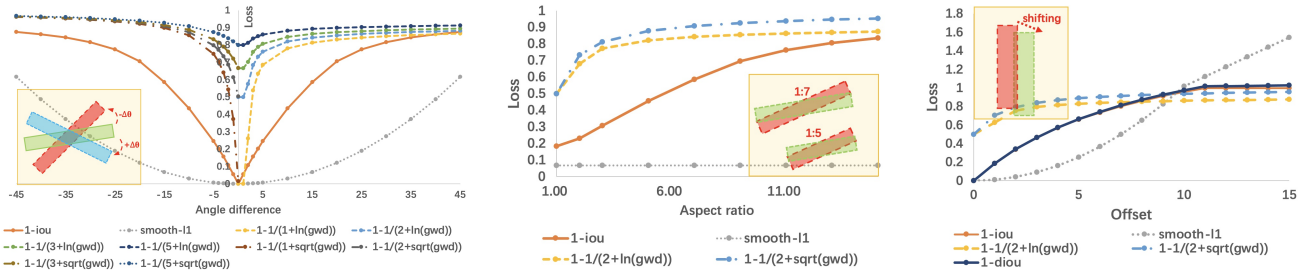
where $f(\cdot)$ is used to deal with angular periodicity, such as trigonometric functions, modulo, etc.

The overall regression loss for rotation detection is:

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

where $\Delta t_x = t_x^p - t_x^t = \frac{\Delta x}{w_a}$, $\Delta t_y = t_y^p - t_y^t = \frac{\Delta y}{h_a}$, $\Delta t_w = t_w^p - t_w^t = \ln(w_p/w_t)$, $\Delta t_h = t_h^p - t_h^t = \ln(h_p/h_t)$, and $\Delta t_\theta = t_\theta^p - t_\theta^t$.

It can be seen that l_n-norm focuses on the difference of individual BBox parameters and the parameters will not be dynamically optimized according to the shape of object, 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 Fig. 4 as an example, the detection result based on the Smooth L1



(a) Angle difference case (b) Aspect ratio case (c) Center shifting case

Fig. 5. Behavior comparison of different losses: Smooth L1 Loss, IoU Loss and Gaussian-based loss (e.g. GWD) in different detection cases.

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 to select an appropriate set of weights given a specific single object sample during the training, which is nontrivial or even unrealistic. Thus, regression loss should be self-modulated during the learning process and calls for a more dynamic optimization strategy. This provides inspiration for the design of regression loss in this paper.

3.3 Major Challenges in Rotation Detection

3.3.1 Inconsistency between Metric and Loss

Intersection over Union (IoU) has been the standard metric for both horizontal detection and rotation detection. However, there is an inconsistency between the metric and regression loss (e.g. l_n-norms), that is, a smaller training loss cannot guarantee a higher performance, which has been extensively discussed in horizontal detection [19], [20]. This misalignment becomes more prominent in rotating object detection due to the introduction of angle parameter in regression based models. To illustrate this, we use Fig. 5 to compare IoU induced loss and Smooth L1 loss [13]:

Case 1: Fig. 5(a) depicts the relation between angle difference and loss functions. Though they all bear monotonicity, only Smooth L1 curve is convex while the others are not.

Case 2: Fig. 5(b) shows the changes of the two loss functions under different aspect ratio conditions. It can be seen that the Smooth L1 loss of the two BBoxe are constant (mainly from the angle difference), but the IoU loss will change drastically as the aspect ratio varies.

Case 3: Fig. 5(c) explores the impact of center point shifting on different loss functions. Similarly, despite the same monotonicity, there is no high degree of consistency.

Seeing the above flaws of classic Smooth L1 loss, IoU-induced losses emerge for horizontal detection e.g. GIoU [19], DIoU [20]. It could help to fill the gap between metric and regression loss for rotation detection. Thanks to the introduction of Gaussian distribution for object modeling, the between-distribution metric becomes a simple yet effective approximate SkewIoU loss. Moreover, we will show later that the Gaussian framework has unique properties to solve boundary discontinuity and square-like problem.

3.3.2 Boundary Discontinuity

As a standing issue for regression-based rotation detectors, the boundary discontinuity [2], [21] in general refers to the sharp loss increase at the boundary induced by the angle and edge parameterization.

Specifically, *Case 1-2* in Fig. 6 describe the boundary discontinuity. Take *Case 2* as an example, we assume that there is a red anchor/proposal $(0, 0, 70, 10, -90°)$ and a green ground truth (GT) $(0, 0, 10, 70, -25°)$ at the boundary position[3], both of which are defined in OpenCV definition D_{oc}. The upper right corner of Fig. 6 shows two ways to regress from anchor/proposal to GT. The way1 achieves the goal by only rotating anchor/proposal by an angle counterclockwise, but a very large Smooth L1 loss occurs at this time due to the periodicity of angle (PoA) and the exchangeability of edges (EoE). As discussed in CSL [21], this is because the result of the blue prediction box $(0, 0, 70, 10, -115°)$ is outside the defined range. As a result, the model has to make predictions in other complex regression forms, such as rotating anchor/proposal by an large angle clockwise to the blue box while scaling w and h (way2 in *Case 2*). A similar problem (only PoA) also occurs in the long edge definition D_{le}, as shown in *Case 1*.

When the predefined anchor/proposal and GT are not in the boundary position, way1 will not produce a large loss. Therefore, there exists inconsistency between the boundary position and the non-boundary position regression, which makes the model very confused about in which way it should perform regression. Since non-boundary cases account for the majority, the regression results of models, especially those with weaker learning capacity, are fragile in boundary cases, as shown in the left of Fig. 1.

3.3.3 Square-Like Problem

In addition, there is also a square-like object detection problem in the D_{le}-based methods e.g. [22]. In fact, D_{le} cannot uniquely define a square BBox. For square-like objects[4], D_{le}-based method will encounter high IoU but high loss value similar to the boundary discontinuity, as shown by the upper part of *Case 3* in Fig. 6. In way1, the red anchor/proposal $(0, 0, 45, 44, 0°)$ rotates a small angle clockwise to get the blue prediction box. The IoU of green GT $(0, 0, 45, 43, -60°)$ and the blue prediction box $(0, 0, 45, 44, 30°)$ is close to 1, but the regression loss is high due to the inconsistency of angle parameters. Therefore, the model will rotate a larger

3. The angle of the BBox is close to the maximum and minimum values of the angle range. For more clearly visualization, the GT has been rendered with a larger angle in Fig. 6.

4. Many object instances are in square shape, e.g. the two categories of storage-tank (ST) and roundabout (RA) in the DOTA dataset.

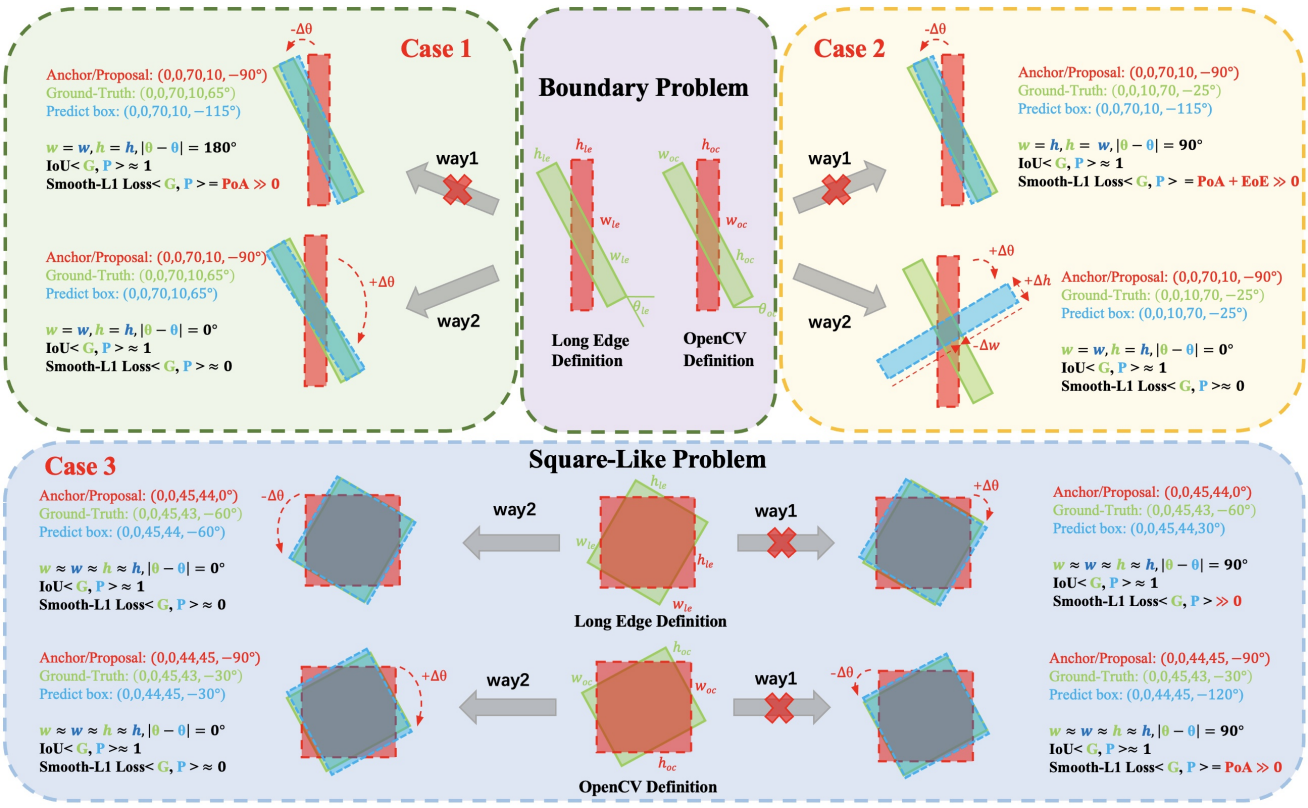


Fig. 6. Boundary discontinuity under two BBox definitions (top), and illustration of the square-like problem (bottom).

angle counterclockwise to make predictions, as described by way2. The reason for the square-like problem in D_{le}-based method is not the above-mentioned PoA and EoE, but the inconsistency of evaluation metric and loss. In contrast, the negative impact of EoE will be weakened when we use D_{oc}-based method to detect square-like objects, as shown in the comparison between *Case 2* and the lower part of *Case 3*. Therefore, there is no square-like problem in the D_{oc}-based method.

Recent methods start to address these issues. SCRDet [2] combines IoU and Smooth L1 loss to propose a IoU-Smooth L1 loss, which does not require the SkewIoU to be gradient backpropagable. It also solves the problem of inconsistency between loss and metric by eliminating the discontinuity of loss at the boundary. However, the gradient direction of IoU-Smooth L1 Loss is still dominated by Smooth L1 loss. RSDet [35] devises modulated loss to smooth the loss mutation at the boundary, but it needs to calculate the loss of as many parameter combinations as possible. CSL [21] transforms angular prediction from a regression problem to a classification problem. CSL needs to carefully design their method according to the BBox definition (D_{le}), and is limited by the classification granularity with theoretical limitation for high-precision angle prediction. On the basis of CSL, DCL [22] further solves the problem of square-like object detection introduced by D_{le}.

4 Proposed Method

4.1 Gaussian Distribution Modeling

In this paper, we adopt Gaussian modeling to construct more accurate rotation regression loss. Most of the IoU-based loss can be considered as a distance function. Inspired by this, we propose a new regression loss based on Gaussian distribution metric. Specifically, we convert a 2-D rotated BBox $\mathcal{B}(x, y, w, h, \theta)$ into a Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_{2d}, \boldsymbol{\Sigma}_{2d})$ (see Fig. 2) by the following formula:

$$
\begin{aligned}
\boldsymbol{\Sigma}_{2d}^{1/2} &= \mathbf{R}\boldsymbol{\Lambda}\mathbf{R}^{\top} \\
&= \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} \frac{w}{2} & 0 \\ 0 & \frac{h}{2} \end{pmatrix} \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \\
&= \begin{pmatrix} \frac{w}{2}\cos^2\theta + \frac{h}{2}\sin^2\theta & \frac{w-h}{2}\cos\theta\sin\theta \\ \frac{w-h}{2}\cos\theta\sin\theta & \frac{w}{2}\sin^2\theta + \frac{h}{2}\cos^2\theta \end{pmatrix} \\
\boldsymbol{\mu}_{2d} &= (x, y)^{\top}
\end{aligned}
\tag{5}
$$

where $\mathbf{R}$ represents the rotation matrix, and $\boldsymbol{\Lambda}$ represents the diagonal matrix of eigenvalues. Gaussian representation is also adopted by [58], but it is not validated on more appropriate downstream tasks (i.e. oriented object detection) and no theoretical analysis is given.

According to Eq. 5, we have the following properties:

- *Property 1:* $\boldsymbol{\Sigma}_{2d}^{1/2}(w, h, \theta) = \boldsymbol{\Sigma}_{2d}^{1/2}(h, w, \theta - \frac{\pi}{2})$;
- *Property 2:* $\boldsymbol{\Sigma}_{2d}^{1/2}(w, h, \theta) = \boldsymbol{\Sigma}_{2d}^{1/2}(w, h, \theta - \pi)$;
- *Property 3:* $\boldsymbol{\Sigma}_{2d}^{1/2}(w, h, \theta) \approx \boldsymbol{\Sigma}_{2d}^{1/2}(w, h, \theta - \frac{\pi}{2}), w \approx h.$

From the two BBox definitions recall that the conversion between two definitions is, the two sides are exchanged and the angle difference is $\frac{\pi}{2}$. Many methods are designated inherently according to the choice of definition in advance to solve some problems, such as D_{le} for EoE and D_{oc} for square-like problem. It is interesting to note that according to *Property 1*, definition D_{oc} and D_{le} are equivalent based

on Gaussian modeling, which makes our method free from the choice of box definitions. This does not mean that the final performance of the two definition methods will be the same. Different factors, e.g. order of edge and angle regression range, will still cause effects. But the method based on Gaussian distribution modeling does not need to bind a certain definition to solve the boundary discontinuity and square-like problem.

Gaussian distribution modeling can also help resolve the boundary discontinuity and square-like problem. The prediction box and GT in way1 of *Case 1* in Fig. 6 satisfy the following relation: $x_p = x_{gt}, y_p = y_{gt}, w_p = h_{gt}, h_p = w_{gt}, \theta_p = \theta_{gt} - \frac{\pi}{2}$. According to *Property 1*, the Gaussian distribution corresponding to these two boxes are the same (in the sense of same mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$), so it naturally eliminates the ambiguity in box representation. Similarly, according to *Properties 2-3*, the GT and prediction box in way1 of *Case 1* and *Case 3* in Fig. 6 are also the same or nearly the same (note the approximate equal symbol for $w \approx h$ for square-like boxes) Gaussian. The Gaussian distribution has degenerated into an isotropic circle, losing the ability to predict the direction, especially the head of the object [48], but this does not prevent getting a high IoU prediction in mAP calculation for most 2-D object detection task. However, this can be a big hassle for 3-D object detection, which will be described in detail in Sec. 4.5.

A drawback of Gaussian model is that it cannot be directly applied to quadrilateral/polygon detection [33], [59], [60] which is an important task in the applications of aerial images and scene text. The difficulty is how to convert point set into a Gaussian. We leave it for future work [61].

We now aim to design an easy-to-implement approximate SkewIoU loss to mitigate the inconsistency between metric and loss and achieve high-precision detection.

4.2 Between-distribution Metric Implementation

4.2.1 Gaussian Wasserstein Distance

The Gaussian Wasserstein Distance (GWD) [24] between two probability measures $\mathbf{X}_p \sim \mathcal{N}_p(\boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p)$ and $\mathbf{X}_t \sim \mathcal{N}_t(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$ can be expressed as:

$$\mathbf{D}_w(\mathcal{N}_p, \mathcal{N}_t)^2 = \|\boldsymbol{\mu}_p - \boldsymbol{\mu}_t\|_2^2 + \mathbf{Tr}\left(\boldsymbol{\Sigma}_p + \boldsymbol{\Sigma}_t - 2(\boldsymbol{\Sigma}_p^{1/2}\boldsymbol{\Sigma}_t\boldsymbol{\Sigma}_p^{1/2})^{1/2}\right) \tag{6}$$

Note the following equation mathematically holds, which indicates GWD is symmetrical:

$$\mathbf{Tr}\left((\boldsymbol{\Sigma}_p^{1/2}\boldsymbol{\Sigma}_t\boldsymbol{\Sigma}_p^{1/2})^{1/2}\right) = \mathbf{Tr}\left((\boldsymbol{\Sigma}_t^{1/2}\boldsymbol{\Sigma}_p\boldsymbol{\Sigma}_t^{1/2})^{1/2}\right) \tag{7}$$

For horizontal detection setting with a constant angle, we have: $\boldsymbol{\Sigma}_p\boldsymbol{\Sigma}_t = \boldsymbol{\Sigma}_t\boldsymbol{\Sigma}_p$, then Eq. 6 becomes:

$$\begin{aligned}\mathbf{D}_w^h(\mathcal{N}_p, \mathcal{N}_t)^2 =& \|\boldsymbol{\mu}_p - \boldsymbol{\mu}_t\|_2^2 + \|\boldsymbol{\Sigma}_p^{1/2} - \boldsymbol{\Sigma}_t^{1/2}\|_F^2 \\ =& (x_p - x_t)^2 + (y_p - y_t)^2 + \frac{(w_p - w_t)^2 + (h_p - h_t)^2}{4} \\ =& l_2\text{-norm}(\Delta x, \Delta y, \frac{\Delta w}{2}, \frac{\Delta h}{2})\end{aligned} \tag{8}$$

where $\|\cdot\|_F$ is the Frobenius norm. Note that both boxes are horizontal here, and Eq. 8 is approximately equivalent to the l_2-norm loss (note the additional denominator of 2 for w and h), which is consistent with the loss commonly used in horizontal detection. As Eq. 8 calculates the Euclidean distance in absolute coordinates, there is still a gap to Eq. 4.

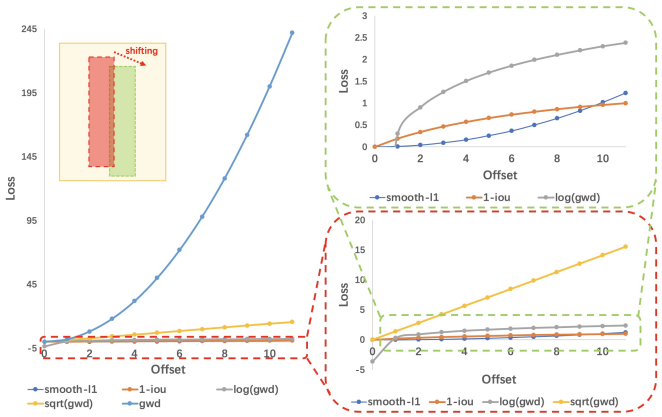


Fig. 7. Illustration of the sensitivity of GWD to large errors. GWD is very sensitive to large errors and requires a suitable transformation for normalization. The x-axis (offset) represents the displacement in pixels

Note that GWD alone can be sensitive to large errors, as shown in the blue curve on the left of Fig. 7. We perform a nonlinear transformation f and then convert GWD into an affinity measure $\frac{1}{\tau + f(\mathbf{D}_w^2)}$ similar to IoU between two BBoxes. Then we follow the standard IoU based loss form in detection literature [19], [20], as written by:

$$L_{gwd} = 1 - \frac{1}{\tau + f(\mathbf{D}_w^2)}, \quad \tau \geq 1 \tag{9}$$

where $f(\cdot)$ denotes a non-linear function to transform the Wasserstein distance $\mathbf{D}_w^2$ to make the loss more smooth and expressive. In this paper, we mainly use two nonlinear functions, $sqrt(\mathbf{D}_w^2)$ and $\ln(\mathbf{D}_w^2 + 1)$. The hyperparameter τ modulates the entire loss.

Fig. 5(a) plots the curve under different combinations of $f(\cdot)$ and τ. Compared with the Smooth L1 loss, the curve of Eq. 9 is more consistent with the IoU loss curve. Also, we can find in Fig. 5(c) that GWD still can measure the distance between two non-overlapping BBoxes (IoU=0), which is exactly the problem that GIoU and DIoU try to solve in horizontal detection. However, the first term of Eq. 6 is the Euclidean distance of the center point between two BBoxes. In addition to making GWD sensitive to large errors, this term also makes the regression loss lose scale invariance. Therefore, we further explore more suitable alternative metrics in subsequent sections.

4.2.2 Kullback-Leibler Divergence

We also adopt the Kullback-Leibler divergence (KLD) [26], which can be written by:

$$\begin{aligned}\mathbf{D}_{kl}(\mathcal{N}_p||\mathcal{N}_t) =& \frac{1}{2}(\boldsymbol{\mu}_p - \boldsymbol{\mu}_t)^\top \boldsymbol{\Sigma}_t^{-1}(\boldsymbol{\mu}_p - \boldsymbol{\mu}_t) \\ &+ \frac{1}{2}\mathbf{Tr}(\boldsymbol{\Sigma}_t^{-1}\boldsymbol{\Sigma}_p) + \frac{1}{2}\ln\frac{|\boldsymbol{\Sigma}_t|}{|\boldsymbol{\Sigma}_p|} - 1\end{aligned} \tag{10}$$

One can also adopt the opposite one:

$$\begin{aligned}&\mathbf{D}_{kl}(\mathcal{N}_t||\mathcal{N}_p) = \\ &\frac{1}{2}(\boldsymbol{\mu}_p - \boldsymbol{\mu}_t)^\top \boldsymbol{\Sigma}_p^{-1}(\boldsymbol{\mu}_p - \boldsymbol{\mu}_t) + \frac{1}{2}\mathbf{Tr}(\boldsymbol{\Sigma}_p^{-1}\boldsymbol{\Sigma}_t) + \frac{1}{2}\ln\frac{|\boldsymbol{\Sigma}_p|}{|\boldsymbol{\Sigma}_t|} - 1\end{aligned} \tag{11}$$

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. 5, each term of Eq. 10 can be expressed as

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

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

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

where $\Delta x = x_p - x_t, \Delta y = y_p - y_t, \Delta\theta = \theta_p - \theta_t$.

Analysis of high-precision detection. Without loss of generality, we set $\theta_t = 0°$, then

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

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 SkewIoU. 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. In contrast, the gradient of the center point in GWD and L_2-norm are $\frac{\partial \mathbf{D}_w(\mu_p)}{\partial \mu_p} = (2\Delta x, 2\Delta y)^\top$ and $\frac{\partial \mathbf{D}_{L_2}(\mu_p)}{\partial \mu_p} = (\frac{2}{w_a^2}\Delta x, \frac{2}{h_a^2}\Delta y)^\top$. The former cannot adjust the dynamic gradient according to the length and width of the object. The latter is based on the length and width of the anchor (w_a, h_a) to adjust the gradient instead of the target object (w_t, h_t), which is almost ineffective for those detectors [2], [18], [36], [57], [59], [62] that use horizontal anchors for rotation detection. More importantly, they are not related to the angle of the target object when $\theta_t \neq 0°$. Therefore, the detection result of the GWD-based and L_n-norm models will show a slight deviation, while the detection result of the KLD-based model is quite accurate, as shown in Fig. 4.

For h_p (similar for w_p), we have

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

On one hand, the optimization of the h_p and w_p by updating their gradients is affected by the $\Delta\theta$. When $\Delta\theta = 0°$, $\frac{\partial \mathbf{D}_{kl}(\boldsymbol{\Sigma}_p)}{\partial \ln h_p} = \frac{h_p^2}{h_t^2} - 1$, $\frac{\partial \mathbf{D}_{kl}(\boldsymbol{\Sigma}_p)}{\partial \ln w_p} = \frac{w_p^2}{w_t^2} - 1$, 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 θ_p is also affected by h_p and w_p:

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

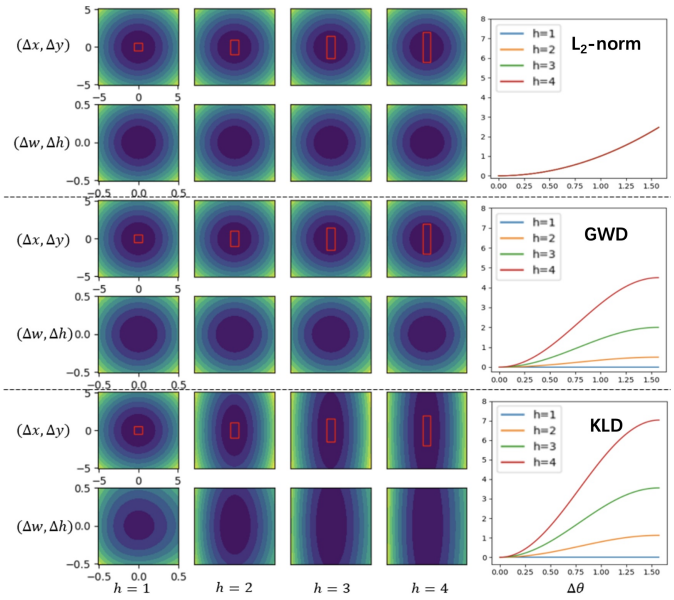


Fig. 8. Behavior of L_2-norm, GWD and KLD versus parameters when the targeted height varies, as generated by a simulation test. Left: the gradient landscape of $(\Delta x, \Delta y, \Delta w, \Delta h)$; Right: gradient curve of $\Delta\theta$.

When $w_p = w_t$, $h_p = h_t$, then we can get $\frac{\partial \mathbf{D}_{kl}(\boldsymbol{\Sigma}_p)}{\partial \theta_p} = \left(\frac{h_t^2}{w_t^2}+\frac{w_t^2}{h_t^2}-2\right)\sin 2\Delta\theta \geq \sin 2\Delta\theta$, the condition for the equality sign is $h_t = w_t$. 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 as a slight angle error would cause a serious accuracy drop for large aspect ratios objects. 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. We believe this largely contributes to the effectiveness of KLD-based regression loss. In addition, $\mathbf{D}_{kl}(\mathcal{N}_t||\mathcal{N}_p)$ has similar properties, refer to appendix in conference version for details.

In general, KLD can be suited to high-precision detection especially for objects with large aspect ratio. For bounding box with larger aspect ratio, KLD gives heavier penalties to matching of shorter edge's length and the center point's position along the shorter edge's direction, as well as the matching of angle. These characteristics are desirable, as when matching bounding box with large aspect ratio, IoU can be intuitively sensitive to the shorter edge's length, the center point's position along the shorter edge's direction and the angle. Specifically, we consider a target box with $x = 0$, $y = 0$, $w = 1$, $\theta = 0$, and set $h = \{1, 2, 3, 4\}$ to control the aspect ratio, and plot KLD versus parameter variation in Fig. 8 where L_2-norm and GWD are also included for comparison. When h increases, KLD is more sensitive to the variation of x, w and θ, meaning it has desirable advantages for objects with large aspect ratio. Comparatively, both L_2-norm and GWD pay no more attention to the matching of x and w when h increases, and L_2-norm is even unchanged

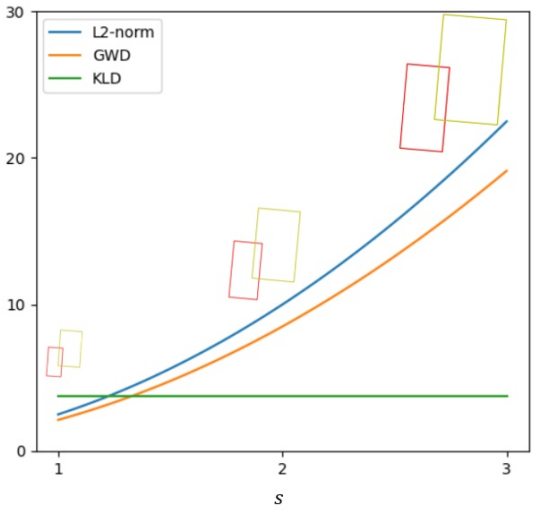


Fig. 9. The loss of L_2-norm, GWD and KLD versus scaling factor, generated by a simulation test. Only the value of KLD is invariant to the scaling factor s. Th bounding boxes denote different scales of objects.

when the difference of angle $\Delta\theta$ is fixed.

Scale invariance. Suppose there are two Gaussian distributions, denoted as $\mathbf{X}_p \sim \mathcal{N}_p(\boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p)$ and $\mathbf{X}_t \sim \mathcal{N}_t(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$. Then, for a full-rank matrix $\mathbf{M}$, $|\mathbf{M}| \neq 0$, we have $\mathbf{X}_{p'} = \mathbf{M}\mathbf{X}_p \sim \mathcal{N}_p(\mathbf{M}\boldsymbol{\mu}_p, \mathbf{M}\boldsymbol{\Sigma}_p\mathbf{M}^\top)$, $\mathbf{X}_{t'} = \mathbf{M}\mathbf{X}_t \sim \mathcal{N}_t(\mathbf{M}\boldsymbol{\mu}_t, \mathbf{M}\boldsymbol{\Sigma}_t\mathbf{M}^\top)$, denoted as $\mathcal{N}_{p'}$ and $\mathcal{N}_{t'}$. The Kullback-Leibler Divergence (KLD) between $\mathcal{N}_{p'}$ and $\mathcal{N}_{t'}$ is:

$$\begin{aligned}\mathbf{D}_{kl}(\mathcal{N}_{p'}||\mathcal{N}_{t'}) =& \frac{1}{2}(\boldsymbol{\mu}_p - \boldsymbol{\mu}_t)^\top \mathbf{M}^\top (\mathbf{M}^\top)^{-1}\boldsymbol{\Sigma}_t^{-1}\mathbf{M}^{-1}\mathbf{M}(\boldsymbol{\mu}_p - \boldsymbol{\mu}_t) \\ &+ \frac{1}{2}\mathbf{Tr}\left((\mathbf{M}^\top)^{-1}\boldsymbol{\Sigma}_t^{-1}\mathbf{M}^{-1}\mathbf{M}\boldsymbol{\Sigma}_p\mathbf{M}^\top\right) + \frac{1}{2}\ln\frac{|\mathbf{M}||\boldsymbol{\Sigma}_t||\mathbf{M}^\top|}{|\mathbf{M}||\boldsymbol{\Sigma}_p||\mathbf{M}^\top|} - 1 \\ =& \mathbf{D}_{kl}(\mathcal{N}_p||\mathcal{N}_t)\end{aligned} \tag{18}$$

where we have $\mathbf{Tr}(\mathbf{AB}) = \mathbf{Tr}(\mathbf{BA})$.

Therefore, KLD can achieve affine invariance. When $\mathbf{M} = k\mathbf{I}$ ($\mathbf{I}$ is identity matrix), its scale invariance holds. To visualize the scale invariance of KLD, we consider the KLD of two given boxes, and investigate the variation of KLD when the two boxes are enlarged with a scaling factor s. As shown in Fig. 9, the value of KLD is invariant to the scaling factor s. Compared with this, the values of L_2-norm and GWD change when s increases.

Horizontal special case. For horizontal detection, combine Eq. 10 to Eq. 14, we have

$$\begin{aligned}\mathbf{D}_{kl}^h(\mathcal{N}_p||\mathcal{N}_t) =& \frac{1}{2}\left(\frac{w_p^2}{w_t^2} + \frac{h_p^2}{h_t^2} + \frac{4\Delta^2 x}{w_t^2} + \frac{4\Delta^2 y}{h_t^2} + \ln\frac{w_t^2}{w_p^2} + \ln\frac{h_t^2}{h_p^2} - 2\right) \\ =& 2l_2\text{-norm}(\Delta t_x, \Delta t_y) + l_1\text{-norm}(\Delta t_w, \Delta t_h) \\ &+ \frac{1}{2}l_2\text{-norm}\left(\frac{1}{\Delta t_w}, \frac{1}{\Delta t_h}\right) - 1\end{aligned} \tag{19}$$

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

Variants of KLD. We introduce two variants [63], [64] to verify the influence of asymmetry on rotation detection:

$$\begin{aligned}\mathbf{D}_{js}(\mathcal{N}_p||\mathcal{N}_t) =& \frac{1}{2}\left(\mathbf{D}_{kl}\left(\mathcal{N}_t||\frac{\mathcal{N}_p + \mathcal{N}_t}{2}\right) + \mathbf{D}_{kl}\left(\mathcal{N}_p||\frac{\mathcal{N}_p + \mathcal{N}_t}{2}\right)\right) \\ \mathbf{D}_{jef}(\mathcal{N}_p||\mathcal{N}_t) =& \mathbf{D}_{kl}(\mathcal{N}_t||\mathcal{N}_p) + \mathbf{D}_{kl}(\mathcal{N}_p||\mathcal{N}_t)\end{aligned} \tag{20}$$

4.2.3 Bhattacharyya Distance

We adopt the Bhattacharyya Distance (BCD) [25], which is specified as follows, where $\boldsymbol{\Sigma} = \frac{1}{2}(\boldsymbol{\Sigma}_p + \boldsymbol{\Sigma}_t)$:

$$\mathbf{D}_{bcd}(\mathcal{N}_p, \mathcal{N}_t) = \frac{1}{8}(\boldsymbol{\mu}_p - \boldsymbol{\mu}_t)^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_p - \boldsymbol{\mu}_t) + \frac{1}{2}\ln\frac{\det(\boldsymbol{\Sigma})}{\sqrt{\det(\boldsymbol{\Sigma}_p\boldsymbol{\Sigma}_t)}} \tag{21}$$

Compared with the above two metrics, BCD is symmetrical and scale invariance, and has a similar parameter optimization mechanism to KLD. Experimental results show that BCD and KLD achieve the similar performance, thus we omit the analysis and verification to BCD.

4.3 Label Assignment based on Gaussian Metric

The label assignment strategy is a key component of object detection [31], [57], which aims to assign targets, foreground or background, to sampled regions in an image. Many current methods use IoU as the basis for label assignment, e.g. the Max-IoU strategy [13], [15]. To align label assignment and regression loss, we propose a new label assignment strategy based on Gaussian metric, that is, Gaussian metric (e.g. KLD) replaces IoU as the basis for sample division. However, setting the threshold is a tricky problem because the Gaussian metric does not have a very intuitive physical meaning like IoU. Inspired by Adaptive Training Sample Selection (ATSS) [31], we adopt a statistical approach to automatically calculate appropriate thresholds. For the i-th GT box (g_i), the dynamic threshold t_{g_i} is:

$$\begin{aligned}t_{g_i} =& m_{g_i} + v_{g_i}, \quad N = kL, \quad \mathcal{G}_{ij} = \frac{1}{\tau + \mathcal{D}_{ij}} \\ m_{g_i} =& \frac{1}{N}\sum_{j=1}^{N}\mathcal{G}_{ij}, \quad v_{g_i} = \sqrt{\frac{1}{N}\sum_{j=1}^{N}(\mathcal{G}_{ij} - m_{g_i})^2}\end{aligned} \tag{22}$$

where k and L represent the number of proposals/anchors closest to the center point of GT box and that of feature pyramid layers in the detector neck. $\mathcal{D}_{ij}$ is the Gaussian metric between the i-th GT box and the j-th proposal/anchor. We set $\tau = 2$ by default to modulate the loss.

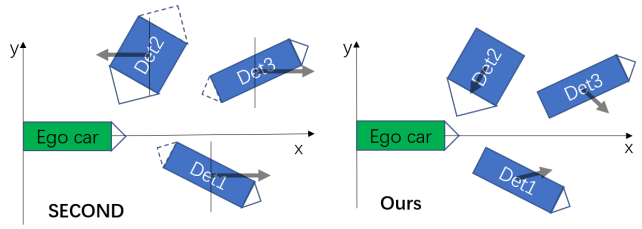
4.4 Overall Loss Function Design

In this section, we take 2-D object detection as the main example. In line with [18], [21], [22], we use the one-stage detector RetinaNet [15] as the baseline. Rotated rectangle is represented by five parameters (x, y, w, h, θ). In our experiments we mainly follow D_{oc}. First of all, we need to clarify that the network has not changed the output of the original regression branch, that is, it is not directly predicting the parameters of the Gaussian distribution. The process of Gaussian distribution metric is as follows: i) predict offset $(t_x^*, t_y^*, t_w^*, t_h^*, t_\theta^*)$; ii) decode prediction box; iii) convert prediction box and target ground-truth into Gaussian distribution; iv) calculate L_{reg} of two Gaussian distributions. Therefore, the inference time remains unchanged.

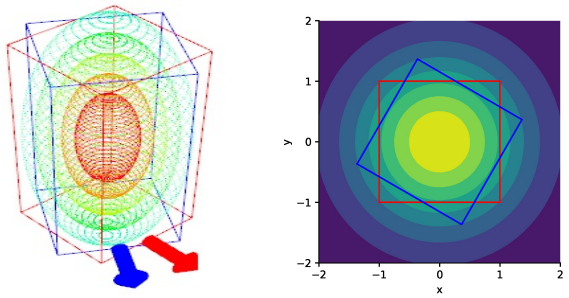
The regression equation of (x, y, w, h) are already listed in Eq. 1 and Eq. 2. As for the regression equation of θ, we use two forms as the baseline to be compared:

- Direct regression, marked as *Reg. ($\Delta\theta$)*. The model directly predicts the angle offset t_θ^*:

$$t_\theta = (\theta - \theta_a)\cdot\pi/180, \quad t_\theta^* = \quad (\theta^* - \theta_a)\cdot\pi/180 \tag{23}$$



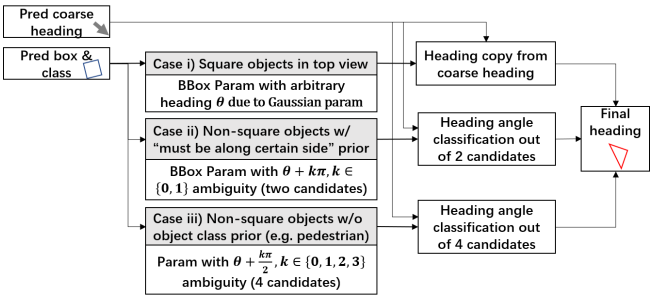
(a) Detection comparison from the top view between existing binary-classification based methods for heading e.g. SECOND [43] and ours.



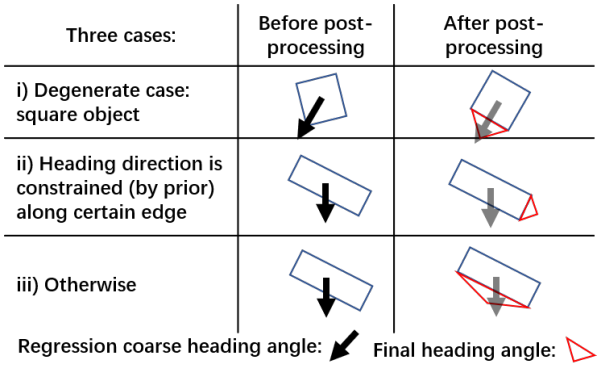
(b) 3-D BBox with square shape in top-view e.g. pedestrian.

(c) Top view of the 3-D BBox and the heading is arbitrary given the isotropic 2-D Gaussian.

Fig. 10. Orientation degeneration cases. Note that the red and blue boxes/cubes share the same Gaussian distribution representation.



(a) Post-processing pipeline of our Gaussian-based loss.



(b) Condition rule table for our post-processing.

Fig. 11. The proposed processing logic after the end-to-end training regression network to derive the final heading angle. The condition rule table deals with three cases according to the shape of the bounding box: i) square object; ii) the rectangle object with its heading direction confined to the angle forming an acute angle to one of a certain bi-direction, e.g. a vehicle with its long edge as heading direction. iii) otherwise, the final direction is chosen as the one (red triangle) that forms an acute angle with the heading regression result (black arrow).

- Indirect regression: *Reg.** $(\sin\theta,\ \cos\theta)$. The model predicts two vectors ($t^*_{\sin\theta}$ and $t^*_{\cos\theta}$) to match the two targets from the GT ($t_{\sin\theta}$ and $t_{\cos\theta}$):

$$\begin{aligned} t_{\sin\theta} &= \sin\left(\theta\cdot\pi/180\right), \quad t_{\cos\theta} = \cos\left(\theta\cdot\pi/180\right) \\ t^*_{\sin\theta} &= \sin\left(\theta^*\cdot\pi/180\right), \quad t^*_{\cos\theta} = \cos\left(\theta^*\cdot\pi/180\right) \end{aligned} \tag{24}$$

To ensure that $t^{*2}_{\sin\theta} + t^{*2}_{\cos\theta} = 1$ is satisfied, we will perform the following normalization processing:

$$t^*_{\sin\theta} = \frac{t^*_{\sin\theta}}{\sqrt{t^{*2}_{\sin\theta} + t^{*2}_{\cos\theta}}}, \quad t^*_{\cos\theta} = \frac{t^*_{\cos\theta}}{\sqrt{t^{*2}_{\sin\theta} + t^{*2}_{\cos\theta}}} \tag{25}$$

Indirect regression is a simpler way to avoid boundary discontinuity. The multi-task loss is defined as follows:

$$L = \frac{\lambda_1}{N}\sum_{n=1}^{N} obj_n \cdot L_{reg}(b_n, gt_n) + \frac{\lambda_2}{N}\sum_{n=1}^{N} L_{cls}(p_n, t_n) \tag{26}$$

where N is the number of anchors, and obj_n is a binary value ($obj_n = 1$ for foreground and otherwise background – no regression for background). b_n denotes the n-th predicted BBox, gt_n is the n-th target ground-truth. t_n denotes the label of the 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 [15] in our experiments. The regression loss L_{reg} is set by Eq. 9 for GWD loss (or for the BCD and KL loss).

4.5 Extending Gaussian Modeling to 3-D Detection

Now we show how to (non-trivially) extend the Gaussian parametric model from 2-D detection to 3-D. Rather than only using the oriented box in birds-eye-view (BEV), practical 3-D detectors are often required to predict the object's height and altitude in 3-D. More specifically, consider the common simplification in many 3-D applications and existing benchmarks assuming a constant pitch and roll angle of moving objects, we can extend Eq. 5 to 3-D by:

$$\mathbf{\Sigma}^{1/2}_{3d} = \mathbf{R}\mathbf{\Lambda}\mathbf{R}^\top, \ \boldsymbol{\mu}_{3d} = (x, y, z)^\top \tag{27}$$

$$\mathbf{R} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}, \ \mathbf{\Lambda} = \begin{pmatrix} \frac{w}{2} & 0 & 0 \\ 0 & \frac{h}{2} & 0 \\ 0 & 0 & \frac{l}{2} \end{pmatrix} \tag{28}$$

where w, h, l represents the width, height and length of the 3-D BBox, respectively. Heading angle is denoted by θ.

In contrast to the 2-D case whereby the object bounding box IoU is the main performance interest, the object's heading information is often required in 3-D. For instance, in Waymo Open Dataset [65] which is a widely used benchmark for autonomous driving, the metric mAP weighted by heading accuracy (mAPH) is particularly designed.

However, the heading angle is ambiguous given only the parameters of detected BBox. In the previous OpenCV/long edge definitions, the ambiguity refers to the choice of θ and $\theta + 180°$ as shown in Fig. 10(a), which is often resolved by a binary classifier as introduced by existing 3-D detectors e.g. in SECOND [43], PointPillar [44] (see the left of Fig. 10(a)), and a similar scheme has also been devised in a recent 2-D detector [48]. While for our Gaussian-based BBox parameterization, we not only need to address the above ambiguity,

Algorithm 1 Post-processing for square-like degradation cases in Gaussian-based 3-D object detection.

Input: $(x, y, z, w, h, l, \theta)$: predicted cube parameters, (d_x, d_y): predicted heading vector, c: cube's predicted class, r: ratio threshold, C: classes whose head along the long side.
Output: post-processed cube parameters: $(x, y, z, w, h, l, \theta)$.

$\theta_d \leftarrow atan2(d_x, d_y)$;
if $r^{-1} < w/h < r$ **then**
  $\theta \leftarrow \theta_d$
  $w, h \leftarrow max(w, h), max(w, h)$
end if
if $c \in C$ **then** // This cube's heading direction should be parallel to the long border of its BEV projected box
  if $w < h$ **then**
    $\theta \leftarrow \theta + \frac{\pi}{2}$
    $w, h \leftarrow h, w$
  end if
  $n = [\frac{\theta_d - \theta}{\pi}]$
  $\theta \leftarrow limit\_period((\theta + \pi(n \mod 2)), [-\pi, \pi))$ // We select the heading angle closer to the angle decoded from the predict heading vector.
else
  $n = [\frac{\theta_d - \theta}{\pi/2}]$
  $\theta \leftarrow limit\_period((\theta + \frac{\pi}{2}(n \mod 4)), [-\pi, \pi))$ // We select the heading angle closer to the angle decoded from the predict heading vector.
  if n is odd **then** // If we rotate the box by odd multiples of $pi/2$, we have to swap the cubes w and h to guarantee the box's shape does not change.
    $w, h \leftarrow h.w$
  end if
end if

but also to handle a degenerate case when the object in top view is in square form as shown in Fig. 10(b). In this case, as shown in Fig. 10(c), the Gaussian distribution becomes geometrically isotropic resulting in the loss no matter KLD or others is inherently agnostic to the heading of the object[5].

To handle the above degenerating case for square-like detecting BBox in Gaussian modeling, we devise a post-processing pipeline which considers the three exclusive cases as shown in Fig. 11(a). Specifically, we introduce a network layer to regress a coarse heading angle θ_c, in addition to the BBox regression (see the left top in Fig. 11(a)). Then we divide the BBox into three cases for deciding the value of the final heading θ: i) a square object such that the Gaussian parameterization becomes agnostic to the heading angle, and we use the coarse angle as the final heading output. ii) for rectangle object, the model can utilize an important common prior that the heading must be along a certain bi-direction e.g. the long edge direction for a vehicle and in this case, the final heading is chosen by forming an acute angle to that prior direction. iii) otherwise, the heading is taken by forming an acute angle to the coarse heading θ_c. The post-processing logic is shown in Algorithm 1.

5 Experiments

For 2-D rotated object detection, we use Tensorflow [66] for implementation under our previously released rotation detection framework [67] by default unless otherwise specified. We have made our proposed detectors open sourced[6]. For 3-D object detection, we use third-party tools, MMDetection3D [68] and the source code is also publicly available[7]. All the 2-D experiments are performed with GeForce RTX 2080 Ti and 12G memory, while the 3-D experiments are performed with GeForce RTX 3090 Ti and 24G memory.

5.1 2-D Datasets and Implementation Details

DOTA [53] is one of the largest datasets for oriented object detection in aerial images with three 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 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 (two new 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 $\times$ 600 subimages with an overlap of 150 pixels and scale it to 800 $\times$ 800, in line with the cropping protocol in literature.

DIOR-R [69] is an aerial image dataset annotated by rotated BBoxes. There are 23,463 images and 190,288 instances, covering 20 object classes. DIOR-R has a high variation of object size, both in spatial resolutions, and in the aspect of inter-class and intra-class size variability across objects. Different imaging conditions, weathers, seasons, image quality are the major challenges of DIOR-R. Besides, it has high inter-class similarity and intra-class diversity.

UCAS-AOD [70] contains 1,510 aerial images of about 659 $\times$ 1,280 pixels, with 2 categories of 14,596 instances. In line with [16], [53], we sample 1,110 images for training and 400 for testing.

HRSC2016 [54] contains images from two scenarios including ships on sea and ships close inshore. All images are collected from six famous harbors. The training, validation and test set include 436,181 and 444 images, respectively.

ICDAR2015 [71] is commonly used for oriented scene text detection and spotting. This dataset includes 1,000 training images and 500 testing images.

ICDAR2017 MLT [72] is a multi-lingual text dataset, which includes 7,200 training images, 1,800 validation images and 9,000 testing images. The dataset is composed of complete scene images in 9 languages, and text regions in this dataset can be in arbitrary orientations, being more diverse and challenging.

MSRA-TD500 [73] is proposed for detecting long and oriented texts. It contains 300 training and 200 test images annotated in terms of text lines.

FDDB [55] is a dataset designed for unconstrained face detection, in which faces have a wide variability of face scales, poses, and appearance. This dataset contains annotations for 5,171 faces in a set of 2,845 images taken from the

5. Such a degeneration can also happen in 2-D for square objects. Fortunately in many 2-D benchmarks, the heading information is not of interest (mAP cannot reflect the heading accuracy) or the object itself is heading-invariant (e.g. storage-tank and roundabout in aerial images).

6. https://github.com/yangxue0827/RotationDetection
7. https://github.com/zhanggefan/mmdet3d-gaussian

TABLE 1
Ablation test of GWD-based regression loss form and hyperparameter on DOTA. The based detector is RetinaNet.

$1-\frac{1}{\tau+f(\mathbf{D}_w^2)}$	$\tau=1$	$\tau=2$	$\tau=3$	$\tau=5$	$\mathbf{D}_w^2$	Baseline
$f(\mathbf{D}_w^2)=sqrt(\mathbf{D}_w^2)$	68.56	**68.93**	68.37	67.77	49.11	65.73
$f(\mathbf{D}_w^2)=\ln(\mathbf{D}_w^2+1)$	67.87	68.09	67.48	66.49		

TABLE 2
Ablation of KLD regression losses using RetinaNet as based detector.

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

TABLE 3
Ablation study of normalization. The based detector is RetinaNet.

Loss	Norm by Eq. 9	HRSC2016			DOTA-v1.0
		Hmean_{50}	Hmean_{75}	$\text{Hmean}_{50:95}$	AP_{50}
Smooth L1	w/	78.99	43.12	43.47	64.95
	w/o	**84.80**	**48.42**	**47.76**	**65.73**

TABLE 4
Ablation study under different BBox definitions.

Base Detector	Box Def.	Reg. Loss	Dataset	mAP_{50}
RetinaNet [15]	D_{le}	Smooth L1	DOTA-v1.0	64.17
		GWD		66.31 **(+2.14)**
		BCD		68.56 **(+4.39)**
		KLD		**68.88 (+4.71)**
	D_{oc}	Smooth L1		65.73
		GWD		68.93 **(+3.20)**
		BCD		71.23 **(+5.50)**
		KLD		**71.28 (+5.55)**

faces in the Wild dataset [74]. In our paper, we manually use 70% as the training set and the rest as the validation set.

The model uses ResNet50 [75] as the default backbone unless otherwise specified, and is initialized with ImageNet [76] pretrained weights. We perform experiments on six aerial benchmarks, three scene text benchmarks and one face benchmark to verify the generality of our techniques. Weight decay and momentum are set 0.0001 and 0.9, respectively. We use MomentumOptimizer over 4 GPUs with a total of 4 images per mini-batch (1 image per GPU). All the models are trained by 20 epochs in total, and learning rate is reduced tenfold at 12 epochs and 16 epochs, respectively. The initial learning rate for RetinaNet is 1e-3. The number of iterations per epoch for DOTA-v1.0, DOTA-v1.5, DOTA-v2.0, DIOR-R, UCAS-AOD, HRSC2016, ICDAR2015, MLT, MSRA-TD500 and FDDB are 54k, 64k, 80k, 17k, 5k, 10k, 10k, 10k, 5k and 4k respectively, and doubled if data augmentation (random rotation, flipping, and graying) and multi-scale training are used.

5.2 3-D Datasets and Implementation Details

KITTI [77] contains 7,481 training and 7,518 testing samples for 3-D object detection benchmark. The training samples are generally divided into the train split (3,712 samples) and the val split (3,769 samples). The evaluation is classified into Easy, Moderate or Hard according to the object size, occlusion and truncation. All results are evaluated by the mean average precision with a 3-D SkewIoU threshold of 0.7 for cars and 0.5 for pedestrian and cyclists.

Waymo Open Dataset [65] (WOD) is a dataset for autonomous driving. There are totally 1,150 sequences, including 798 sequences in training set with 158,801 LiDAR frames, 202 sequences in validation set with 39,987 LiDAR frames, and 150 sequences in test set with 29,647 LiDAR frames. The official 3-D detection evaluation metrics include the 3-D bounding box mean average precision (mAP) and mAP weighted by heading accuracy (mAPH). The mAP and mAPH are based on an 3-D SkewIoU threshold of 0.7 for vehicles and 0.5 for pedestrians and cyclists.

For experiments on KITTI and WOD, we use PointPillars [44] as the baseline by plugging our new loss. Experiments are all conducted with a single model for 3-class joint detection: vehicle, cyclists, and pedestrian. As a common protocol in 3-D detection, all the detectors are trained from scratch. The training schedule of PointPillars on KITTI follows that of MMDetection3D [68]: AdamW optimizer [78] with 48 samples per mini-batch (12 samples per GPU), a cosine-shaped one-cycle learning rate scheduler that spans 160 epochs. The learning rate starts from 1e-4 and reaches its peak value 1e-3 at the 60 epochs, and then goes down gradually to 1e-7 in the end. We train PointPillars on WOD with FP16 mixed-precision enabled and with similar schedule used by MMDetection3D: AdamW optimizer with 32 samples per mini-batch (8 samples per GPU), and with a linear learning rate scheduler that spans 24 epochs. The learning rate starts from 1e-3 and decays to 1e-4 and 1e-5 at the beginning of the 21^{st} and 24^{th} epochs respectively.

5.3 Ablation Study and Further Comparison

Ablation study of GWD-based regression loss form and hyperparameter: Tab. 1 compares two forms of GWD-based loss. The performance of directly using GWD ($\mathbf{D}_w^2$) as the regression loss is extremely poor: 49.11%, due to its rapid growth trend, as shown in the blue curve on the left of Fig. 7. In other words, the regression loss $\mathbf{D}_w^2$ is too sensitive to large errors. In contrast, Eq. 9 achieves a significant improvement by fitting IoU loss. Eq. 9 introduces two new hyperparameters, the non-linear function $f(\cdot)$ to transform the Wasserstein distance, and the constant τ to modulate the entire loss. From Tab. 1, the overall performance of using $sqrt$ outperforms that using $\ln$, about 0.98±0.3% higher. For $f(\cdot)=sqrt$ with $\tau=2$, the model achieves the best performance: 68.93%. The results are consistent for BCD and KLD. All the subsequent experiments follow this setting for hyperparameters unless otherwise specified.

Ablation study of KLD variants: Keeping the same loss pattern, we compare four KLD-based distance functions in Tab. 2, and conclude that the asymmetry of KLD does not have much impact on performance. In subsequent experiments, we use $\mathbf{D}_{kl}(\mathcal{N}_p||\mathcal{N}_t)$ as the basic setting.

Ablation study of normalization: Note the extra normalization in Eq. 9 questions if the GWD/BCD/KLD actually contributes or simply produces noise in the results. Hence, we also perform a normalization operation on the Smooth L1 loss to eliminate the interference caused by normalization. Tab. 3 shows a significant performance drop after normalization. These results show that the effectiveness of GWD/BCD/KLD does not come from Eq. 9.

Ablation under different rotating box definitions: Tab. 4 studies RetinaNet under different regression losses on DOTA-v1.0, and both rotating box definitions: D_{le} and D_{oc} are

TABLE 5
High-precision detection experiment under different regression loss. 'R', 'F' and 'G' indicate random rotation, flipping, and graying, respectively. The resolution of HRSC2016, MSRA-TD500, ICDAR2015 and FDDB are 500×500, $800 \times 1,000$, $800 \times 1,000$ and 800×800, respectively.

Base Detector	Dataset	Data Aug.	Reg. Loss	$Hmean_{50}/AP_{50}$	$Hmean_{60}/AP_{60}$	$Hmean_{75}/AP_{75}$	$Hmean_{85}/AP_{85}$	$Hmean_{50:95}/AP_{50:95}$
RetinaNet [15]	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)
			BCD	86.38 (+2.10)	85.32 (+10.58)	68.50 (+20.08)	15.67 (+3.11)	55.09 (+7.33)
			KLD	**87.45** (+3.17)	**86.72** (+11.98)	**72.39** (+23.97)	**27.68** (+15.12)	**57.80** (+10.04)
R^3Det [18]			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)
			BCD	**90.06** (+1.54)	**89.75** (+10.74)	**76.24** (+32.82)	**23.42** (+18.84)	**60.26** (+14.08)
			KLD	**89.97** (+1.45)	**89.73** (+10.72)	**77.38** (+33.96)	**25.12** (+20.54)	**61.40** (+15.22)
RetinaNet [15]	MSRA-TD500	R+F	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)
			BCD	75.24 (+4.26)	69.50 (+7.08)	**48.13** (+11.40)	**20.33** (+7.77)	**45.26** (+7.37)
			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)
			BCD	**76.63** (+6.85)	**71.07** (+6.92)	43.10 (+6.13)	10.24 (+1.53)	**42.78** (+5.05)
			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)
			BCD	**78.03** (+3.20)	72.50 (+3.04)	45.44 (+3.42)	10.53 (-1.06)	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 [18]		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)
			BCD	**79.02** (+4.74)	**72.82** (+4.70)	**45.68** (+9.95)	**10.42** (+2.41)	**44.22** (+5.12)
			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)
			BCD	**80.49** (+4.96)	**74.73** (+5.04)	**45.42** (+7.73)	**10.89** (+1.86)	**44.55** (+3.99)
			KLD	79.63 (+4.63)	73.30 (+3.61)	43.51 (+5.82)	10.61 (+1.58)	43.61 (+3.05)
RetinaNet [15]	FDDB	F	Smooth L1	95.92	87.50	55.81	12.67	52.77
			GWD	97.44 (+1.52)	94.68 (+7.18)	80.84 (+25.03)	36.38 (+23.71)	65.77 (+13.00)
			BCD	96.67 (+0.75)	94.60 (+7.10)	83.09 (+27.28)	40.72 (+28.05)	67.03 (+14.26)
			KLD	**97.51** (+1.59)	**95.40** (+7.90)	**85.33** (+29.52)	**42.20** (+29.53)	**68.01** (+15.24)

TABLE 6
Performance comparison on the KITTI *val* split. Numbers are AP scores with 40 recall positions i.e. mAP quoted from [68]. 'Mod.' denotes the moderate level of detection as defind by the dataset and the column 'mAP mod.' refers to the overall mAP for the moderate level of objects.

Method	mAP	Car - 3-D Detection			Ped. - 3-D Detection			Cyc. - 3-D Detection			mAP	Car - BEV Detection			Ped. - BEV Detection			Cyc. - BEV Detection		
	Mod.	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard	Mod.	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
PointPillars [44]	64.28	88.26	78.90	**76.06**	57.10	50.96	46.38	83.77	62.99	59.65	70.10	**93.81**	88.08	**86.80**	61.49	55.51	51.13	87.20	66.69	63.02
+ GWD	65.50	87.38	78.57	75.87	**61.69**	**55.19**	**50.04**	81.61	62.74	59.18	71.48	92.02	88.30	85.72	**64.67**	**58.49**	**53.45**	86.92	67.66	63.37
+ BCD	66.07	87.56	78.69	75.86	58.44	52.91	48.12	**87.08**	**66.62**	**62.68**	**72.02**	92.07	**88.38**	85.68	63.24	57.75	53.23	**90.14**	**69.95**	**65.78**
+KLD	**66.19**	**89.55**	**80.36**	76.02	59.95	52.94	48.22	85.61	65.27	61.45	71.18	93.33	88.11	85.44	64.46	57.26	52.53	87.40	68.19	64.47

TABLE 7
Performance comparison on Waymo Open Dataset *val* set with 202 sequences for 3-D object detection.

Method	Difficulty	mAP				mAPH			
		Veh.	Ped.	Cyc.	Overall	Veh.	Ped.	Cyc.	Overall
PointPillars	Level 1	71.07	72.89	63.26	69.08	70.52	57.95	61.05	63.17
+GWD		72.89	72.86	62.75	69.50	72.35	59.23	60.74	64.11
+BCD		72.86	**73.50**	**63.33**	**69.90**	72.34	**59.74**	**61.30**	**64.46**
+KLD		**72.93**	71.87	62.83	69.21	**72.41**	58.07	60.83	63.77
PointPillars	Level 2	62.84	64.82	60.89	62.85	62.34	51.31	58.76	57.47
+GWD		64.60	64.71	60.43	63.25	64.11	52.41	58.49	58.34
+BCD		64.58	**65.47**	**60.95**	**63.66**	64.10	**53.01**	**59.00**	**58.70**
+KLD		**64.62**	63.73	60.48	62.94	**64.14**	51.32	58.56	58.01

TABLE 8
Ablation of heading post-processing on Waymo Open Dataset *val* set.

Method	Post-Proc.	Difficulty	mAP				mAPH			
			Veh.	Ped.	Cyc.	Overall	Veh.	Ped.	Cyc.	Overall
PointPillars +GWD	w/o	Level 1	**73.54**	**74.09**	**64.24**	**70.62**	**72.99**	55.15	**61.77**	63.30
	w/		72.89	72.86	62.75	69.50	72.35	**59.23**	60.74	**64.11**
	w/o	Level 2	**64.94**	**65.76**	**61.88**	**64.19**	**64.44**	48.76	**59.49**	57.57
	w/		64.60	64.71	60.43	63.25	64.11	**52.41**	58.49	**58.34**

TABLE 9
More ablation experiments on more datasets (MLT and UCAS-AOD).

Base Detector	Reg. Loss	MLT	UCAS-AOD		
		$Hmean_{50}$	car	plane	mAP_{50}
RetinaNet	Smooth L1	48.42	92.62	96.50	94.56
	GWD	54.58 (+6.16)	94.03 (+1.41)	96.86 (+0.36)	95.44 (+0.88)
	BCD	56.79 (+8.37)	**94.99** (+2.37)	**98.10** (+1.60)	**96.54** (+1.98)
	KLD	**57.59** (+9.17)	94.34 (+1.72)	97.94 (+1.44)	96.14 (+1.58)

tested. For the Smooth L1 loss, the accuracy of D_{le}-based method is 1.56% lower than the D_{le}-based ones, at 64.17% and 65.73%, respectively. GWD/BCD/KLD-based methods obtain an increase by 2.14%/4.39%/4.71% and 3.20%/5.50%/5.55% under the above two definitions. Although Gaussian distribution modeling makes our method free from the choice of box definitions, it does not mean that the final performance of the two definition methods will be the same, as shown in Tab. 4 (GWD: 66.31% vs. 68.93%; BCD: 68.56% vs. 71.23%; KLD: 68.88% vs. 71.28%). Different factors, such as order of edges and angle regression range, will still cause differences in model learning, but the methods based on Gaussian distribution need not to bind a certain definition. Therefore, D_{oc} is used in all the subsequent experiments, unless otherwise specified.

High-precision detection: Tab. 5 shows results by two detectors on three datasets where the IoU for AP is at least 50%. For HRSC2016 with a large number of ship of high aspect ratios, GWD achieves 11.89% improvement over Smooth L1 by AP_{75}, BCD and KLD even get 20.08% and 23.97% gain. Even with a stronger R^3Det detector, GWD/BCD/KLD still obtains improvement by 22.46%/32.82%/33.96% by AP_{75}, and 9.89%/14.08%/15.22% by $AP_{50:95}$. Similar results are obtained on MASR-TD500, ICDAR2015, FDDB that BCD/KLD output higher quality BBoxes than GWD and Smooth L1.

Ablation study on 3-D detection tasks: We extended the framework based on Gaussian distribution modeling from

TABLE 10
Comparison between different solutions for inconsistency between metric and loss (IML), boundary discontinuity (BD) and square-like problem (SLP) on DOTA dataset. The ✓ indicates that the method has corresponding problem. † 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.

Base Detector	Solution	Box Def.	IML	BD		SLP	v1.0 tranval/test									v1.0 train/val			v1.5	v2.0
				EoE	PoA		BR†	SV†	LV†	SH†	HA†	ST‡	RA‡	7-mAP$_{50}$	mAP$_{50}$	mAP$_{50}$	mAP$_{75}$	mAP$_{50:95}$	mAP$_{50}$	mAP$_{50}$
RetinaNet [15]	Reg. ($\Delta\theta$)	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
	Reg. ($\Delta\theta$)	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
	Reg.* ($\sin\theta$, $\cos\theta$)	D_{le}	✓	×	×	✓	41.52	63.94	44.95	71.18	53.22	78.11	60.54	59.07	65.78	63.22	30.63	33.19	57.17	43.92
	IoU-Smooth L1 [2]	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	59.17	46.31
	Modulated [35]	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	57.75	45.17
	Modulated [35]	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
	RIL [59]	Quad.	✓	×	×	×	40.81	67.63	55.45	72.42	55.49	78.09	64.75	62.09	66.06	64.07	40.98	39.05	58.91	45.35
	CSL [21]	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) [22]	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 (ours)	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
	BCD (ours)	D_{oc}	×	×	×	×	**45.16**	**74.04**	**72.19**	**84.07**	**65.07**	**80.23**	**64.52**	**69.33**	**71.23**	**67.83**	**42.40**	**41.24**	60.78	**47.48**
	KLD (ours)	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^{3}Det [18]	Reg. ($\Delta\theta$)	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) [22]	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 (ours)	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
	BCD (ours)	D_{oc}	×	×	×	×	**47.80**	**76.23**	**79.20**	**86.90**	**65.24**	83.07	61.20	**71.38**	**72.22**	**69.23**	**43.96**	**41.91**	**63.53**	**49.71**
	KLD (ours)	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**

TABLE 11
Accuracy (%) on DIOR-R. The short names c1-c20 for categories in our experiment are defined as: Airplane, Airport, Baseball field, Basketball court, Bridge, Chimney, Dam, Expressway service area, Expressway toll station, Golf field, Ground track field, Harbor, Overpass, Ship, Stadium, Storage tank, Tennis court, Train station, Vehicle, and Wind mill. ‡ indicates that data augmentation and multi-scale training and testing are used.

Base Detector	Solution	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11	c12	c13	c14	c15	c16	c17	c18	c19	c20	mAP$_{50}$
RetinaNet [15]	*Reg.* ($\Delta\theta$)	57.10	28.85	66.69	80.50	21.00	72.38	17.72	59.90	50.56	73.05	73.18	16.10	37.44	54.20	63.64	43.57	79.56	35.40	25.77	51.59	50.41
	*Reg.** ($\sin\theta$, $\cos\theta$)	57.67	28.48	68.65	80.52	19.29	72.51	22.25	64.16	52.45	73.58	76.26	18.59	39.82	54.21	58.00	43.50	79.74	40.49	25.73	53.24	51.46
	IoU-Smooth L1 [2]	57.17	24.86	67.62	80.51	21.36	72.60	22.48	60.15	51.15	73.01	74.26	19.80	37.02	54.58	63.09	43.55	79.85	37.61	25.96	52.69	50.97
	RIL [59]	50.02	34.91	66.83	80.44	22.73	72.19	25.06	61.39	48.83	68.69	73.26	27.71	41.19	56.50	63.09	42.30	75.40	33.91	24.50	51.47	51.02
	CSL [21]	57.25	26.35	67.60	80.17	19.10	72.55	20.54	65.25	53.10	72.65	73.61	25.18	37.13	56.43	63.56	41.21	78.62	40.82	26.34	52.47	51.50
	DCL [22]	57.21	21.51	67.35	80.68	18.84	72.43	20.24	64.33	51.48	72.96	72.18	24.13	37.21	55.93	61.94	47.50	80.21	39.82	25.53	51.47	51.15
	Modulated [35]	56.70	34.22	68.11	82.44	25.03	72.60	26.91	70.77	53.45	75.75	74.55	29.78	45.79	61.59	62.76	42.85	78.57	38.16	26.72	52.87	53.98
	GWD (ours)	59.22	20.55	69.43	80.85	16.99	72.58	20.18	64.03	53.18	72.24	76.33	17.10	35.27	58.34	68.11	44.15	80.81	37.90	26.13	54.23	51.38
	BCD (ours)	59.75	30.48	69.24	81.08	24.93	72.38	23.85	67.87	53.59	75.03	73.18	31.42	45.26	65.49	64.64	49.17	81.10	35.40	29.01	54.37	54.36
	KLD (ours)	59.12	33.23	68.91	81.25	27.82	75.45	26.76	73.15	53.16	76.56	77.46	33.29	47.05	66.03	65.56	43.83	81.12	37.41	28.52	54.43	55.50
	KLD‡ (ours)	78.45	**48.09**	76.53	89.80	34.38	**77.37**	33.13	84.64	66.56	76.03	82.66	40.69	52.61	79.61	74.68	61.37	88.18	47.90	38.40	63.56	64.73
FPN [14]	*Reg.* ($\Delta\theta$)	62.36	30.31	71.13	80.62	29.65	72.26	22.91	72.50	65.66	73.53	76.55	26.19	45.62	78.89	68.83	71.18	80.95	38.58	47.04	63.50	58.91
	KLD (ours)	62.57	30.10	70.84	81.04	33.39	72.48	22.89	73.40	67.01	76.33	75.59	36.57	50.19	80.55	63.09	70.71	81.07	49.86	48.53	64.29	60.52
	KLD‡ (ours)	**87.77**	42.82	**80.69**	**90.01**	**44.89**	72.70	**34.59**	**85.88**	**76.20**	**77.41**	**84.18**	**45.46**	**56.29**	**88.85**	**78.64**	**79.81**	**89.33**	**57.26**	**56.02**	**72.19**	**70.05**

TABLE 12
Performance comparison by mAP$_{50}$ of using different approximate SkewIoU losses on DOTA-v1.0 dataset. Base model is RetinaNet.

Reg. Loss	Implement	Consistency	Scale Invariance	mAP$_{50}$
Smooth L1 [13]	easy	×	×	64.55
PIoU [11]	medium	✓	✓	65.85
plain SkewIoU [23]	hard	✓	✓	68.27
GWD (ours)	easy	×	×	67.05
KLD (ours)	easy	✓	✓	**69.94**

TABLE 13
Evaluation by mAP of the combination of different strategies for label assignments and regression losses.

Base Detector	Label Assignment	Regression Loss	AP$_{50}$
RetinaNet-D_{le} [15]	Max-IoU	Smooth L1	68.56
	ATSS-IoU	Smooth L1	70.63 (+2.07)
	Max-IoU	KLD	70.35 (+1.79)
	ATSS-IoU	KLD	71.13 (+2.57)
	ATSS-KLD	KLD	**72.13** (+3.57)

TABLE 14
Performance evaluation of our KLD loss on horizontal detection.

Detector	Regression Loss	AP	AP$_{50}$	AP$_{75}$	AP$_s$	AP$_m$	AP$_l$
RetinaNet [15]	Smooth L1	37.2	56.6	39.7	21.4	41.1	48.0
	GIoU	37.4	**56.7**	39.7	22.2	41.7	48.1
	KLD	**38.0**	56.4	**40.6**	**23.3**	**43.2**	**49.3**
Faster RCNN [13]	Smooth L1	37.9	**58.8**	41.0	22.4	41.4	49.1
	GIoU	**38.3**	58.7	41.5	22.5	41.7	**49.7**
	KLD	38.2	58.7	**41.7**	**22.6**	**41.8**	49.3
FCOS [79]	IoU	36.6	56.0	38.8	21.0	40.6	47.0
	KLD	**36.8**	**56.3**	**39.1**	**21.7**	**40.8**	**47.5**

2-D to 3-D object detection. Tab. 6 shows the performance comparison in 3-D detection and BEV detection on KITTI val split, and significant performance improvements are also achieved. On the moderate level of 3-D detection, GWD, BCD and KLD improve the PointPillars by 1.22%, 1.79% and 1.91%. On the moderate level of BEV detection, BCD and GWD achieve gains of 1.38%, 1.92% and 1.08%. Tab. 7 shows similar trend on Waymo Open Dataset.

Ablation study of heading post-processing: We also conducted experiments to validate the effectiveness of the proposed heading post-processing technique. We trained PointPillars-GWD with heading direction binary classifier used in the original work as a reference. As shown in Tab. 8, our post-processing method applied to PointPillars improves the mAPH metric for the square-like category (pedestrian) by 4.08%/3.65% (from a 2.8%/2.55% decrease to a 1.28%/1.10% increase) in terms of Level 1/2, but it also brings a slight performance degradation to mAP due to the need to learn new parameters (heading vectors), which are meaningless for mAP. Therefore, our proposed post-processing method can be useful to the case when the heading information is needed e.g. in autonomous driving, especially for square-like categories.

Ablation study on more datasets: To make the results more credible, we continue to verify on the other two 2-D datasets, as shown in Tab. 9. The improvements of GWD/BCD/KLD on MLT and UCAS-AOD are still considerable, with an increase of 6.16%/8.37%/9.17% and 0.88%/1.98%/1.58% respectively.

Comparison between different solutions: For the three issues of inconsistency between metric and loss, boundary discontinuity and square-like problem, Tab. 10 compares the five different solutions, including IoU-Smooth L1 Loss [2], Modulated loss [35], RIL [59], CSL [21], DCL [22] on DOTA dataset. For fairness, these methods are all implemented on the same baseline, and are trained and tested under the same environment and hyperparameters. In particular,

TABLE 15
Average precision (AP) of 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 [80] and Hourglass-104 [81], respectively. MS indicates that multi-scale training/testing is used. † means that the label assignment strategy is ATSS-KLD. The bold **red** and **blue** fonts indicate the top two performances respectively.

	Method	Backbone	MS	PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SBF	RA	HA	SP	HC	mAP_{50}
Two-stage	ICN [16]	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. [17]	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 [2]	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 [33]	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 [49]	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
	FPN-CSL [21]	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 [35]	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++ [56]	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 [34]	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**
	FPN-BCD (ours)	R-152	✓	**90.33**	85.43	**59.33**	**82.11**	**79.35**	**83.02**	**87.31**	**90.88**	**88.04**	**87.18**	**75.30**	66.73	**76.72**	74.77	**75.61**	**80.14**
Single-stage	PIoU [11]	DLA-34 [82]		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	61.90	64.00	60.50
	O^2-DNet [83]	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 [57]	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
	BBAVectors [84]	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 [12]	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 [85]	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
	DCL [22]	R-152	✓	89.10	84.13	50.15	73.57	71.48	58.13	78.00	**90.89**	86.64	86.78	67.97	67.25	65.63	74.06	67.05	74.06
	RDD [86]	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 (ours)	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
	BCD (ours)	R-152		88.80	84.41	53.73	70.26	77.85	76.31	85.18	90.83	85.91	85.61	64.77	64.15	76.60	**77.19**	71.27	76.86
			✓	89.35	**86.35**	54.79	**80.98**	77.79	74.83	84.19	90.79	86.26	85.74	**72.02**	69.19	**77.20**	76.39	71.91	**78.52**
	KLD (ours)	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
	KLD† (ours)	R-50	✓	89.00	83.70	**56.03**	78.12	**80.77**	**85.35**	**88.22**	**90.90**	84.51	**87.23**	66.71	**73.44**	76.06	**81.94**	70.91	**79.53**
Refine-stage	CFC-Net [62]	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 [18]	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 [57]	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 [22]	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
	RIDet [59]	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 [36]	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-GWD (ours)	R-50	✓	88.89	83.58	55.54	80.46	76.86	83.07	86.85	89.09	83.09	86.17	71.38	64.93	76.21	73.23	64.39	77.58
		R-152	✓	89.28	83.70	**59.26**	79.85	76.42	83.87	86.53	89.06	85.53	86.50	**73.04**	67.56	76.92	77.09	71.58	79.08
	R^3Det-BCD (ours)	R-50		89.53	85.02	54.32	73.08	79.11	84.83	88.05	90.82	85.92	85.75	66.80	63.96	77.11	73.82	66.31	77.63
		R-50	✓	89.77	**86.11**	56.48	**81.94**	79.37	**85.03**	88.15	90.83	**86.77**	87.04	71.87	66.67	78.08	74.31	68.80	79.41
		R-152	✓	89.82	**86.62**	57.36	79.68	79.73	**84.86**	87.92	**90.88**	86.74	86.92	**73.44**	69.95	**78.83**	75.01	76.11	**80.26**
	R^3Det-KLD (ours)	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**

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 contain many corner cases in the dataset. These categories are assumed can better reflect the real-world challenges and advantages of our method. Many methods that solve the boundary discontinuity have achieved significant improvements in the large aspect ratio object category, and the methods that take into account the square-like problem perform well in the square-like object.

We compare the two baselines, *Reg.* ($\Delta\theta$) vs. *Reg.** ($\sin\theta$, $\cos\theta$), provided in Sec. 4.4. Under the same BBox definition D_{le}, the indirect regression method has obvious performance advantages, especially for large aspect ratio objects, due to its immunity to boundary discontinuity.

There is rarely a unified method to solve all problems, and methods are proposed for part of problems e.g. IoU-Smooth L1 Loss. However, the gradient direction of IoU-Smooth L1 Loss is still dominated by Smooth L1 loss, so the metric and loss cannot be regarded as truly consistent. In contrast, due to the three nice properties of Gaussian distribution modeling, it need not to make additional judgments and can elegantly solve all the problems. Without bells and whistles, the combination of RetinaNet and BCD/KLD directly surpasses R^3Det (71.23%/71.28% vs. 70.66% in AP_{50} and 69.33%/69.41% vs. 68.31% in 7-AP_{50}). Even combined with R^3Det, BCD/KLD can still further improve performance of the large aspect ratio object (3.07%/2.82% in 7-AP_{50}) and high-precision detection (5.55%/6.07% in AP_{75} and 3.45%/3.65% in $AP_{50:95}$). BCD-based and KLD-based methods show the best performance in almost all indicators. Fig. 1 and Fig. 4 visualize the comparison between Smooth L1 loss-based and GWD-based detector. Similar conclusions can still be drawn on the more challenging datasets (DOTA-v1.5, DOTA-v2.0, and DIOR-R in Tab. 11), with more tiny objects (less than 10 pixels).

Comparing different approximate SkewIoU losses: Tab. 12 compares the proposed techniques with the plain SkewIoU Loss [23] and the recently proposed approximate SkewIoU loss (PIoU) [11] on DOTA-v1.0 under the same experimental conditions. Since the official PIoU and plain SkewIoU are implemented based on PyTorch [87], we also implemented the PyTorch version[8] [88] of KLD and GWD. Clearly, KLD outperforms all other losses, at 69.94%.

Comparing combined strategies for different label assignments and regression losses: Not limited to regression loss, the proposed KLD can also be used as an effective division basis for label assignment. Tab. 13 shows that when KLD is applied to both label assignment and regression loss, the performance rises to 72.13% with a gain of 3.57% compared with the baseline. Since KLD does not have a clear physical meaning like IoU, dynamic threshold calculation by ATSS [31] is required. Compared to the combination of ATSS-IoU label assignment and KLD loss, using ATSS-KLD can obtain further improvement from 71.13% to 72.13%.

Horizontal detection verification: KLD can be degenerated into the common regression loss in horizontal detection (see Eq. 19). Tab. 14 compares the regression loss Smooth L1 and IoU/GIoU for horizontal detection with our KLD loss on MS COCO [89]. KLD still performs competitively on the Faster RCNN, RetinaNet and FCOS, and even has an improvement

8. https://github.com/open-mmlab/mmrotate

TABLE 16
Performance on HRSC2016 via different backbones. For mAP, the numbers in bracket '07' or '12' means following the 2007 or 2012 evaluation metric. '-' means the raw papers did not provide the results.

Method	Backbone	mAP_{50} (07)	mAP_{50} (12)
RoI-Trans. [17]	R-101	86.20	–
RSDet [35]	R-50	86.50	–
DRN [12]	H-104	–	92.70
SBD [90]	R-50	–	93.70
Gliding Vertex [33]	R-101	88.20	–
OPLD [91]	R-101	88.44	–
BBAVectors [84]	R-101	88.60	–
S^2A-Net [36]	R-101	**90.17**	95.01
R^3Det [18]	R-101	89.26	96.01
R^3Det-DCL [22]	R-101	89.46	96.41
FPN-CSL [21]	R-101	89.62	96.10
DAL [57]	R-101	89.77	–
R^3Det-GWD (ours)	R-101	89.85	97.37
R^3Det-BCD (ours)	R-101	**90.07**	**97.42**
R^3Det-KLD (ours)	R-101	89.87	**97.62**

of 0.6% on RetinaNet. The GT for rotation detection is the minimum circumscribed rectangle, which means that GT can well reflect the true scale and direction information. The "horizontal special case" described in this paper also meets the above requirements, and the horizontal circumscribed rectangle is equal to the minimum circumscribed rectangle at this time. Although the GT of MS COCO is a horizontal box, it is not the minimum circumscribed rectangle. It means that it loses direction and accurate scale information of the object. For example, for a baseball bat placed obliquely in the image, the height and width of its horizontal circumscribed rectangle do not represent the exact height and width of the object. This causes that when KLD is applied to MS COCO, the optimization mechanism of KLD that dynamically adjusts the angle gradient according to the aspect ratio is meaningless, which in our analysis may affect the final performance. Hence, we believe this is a defect in the dataset annotation itself. In fact, it is inappropriate to use MS COCO to discuss $\theta = 0°$, because this dataset discards θ. In addition, $\theta = 0°$ describes the instances in the horizontal position, but not mean all instances of the dataset are in a horizontal position. This paper uses MS COCO to discuss the "horizontal special case" to show that even if the dataset has certain labeling defects, our KLD can still be robust.

5.4 Overall Comparison

Evaluation is performed on DOTA-v1.0 and HRSC2016.

Results on DOTA-v1.0: Tab. 15 compares with state-of-the-art methods, which in fact use different image resolutions, network structures, training strategies and tricks, which make the comparison less direct. For overall performance, BCD-based and KLD-based methods have achieved the best mAP among the two-stage, single-stage and refine-stage methods, in 80.14%, 79.53% and 80.63% respectively.

Results on HRSC2016: It contains large aspect ratio ship instances with arbitrary orientation, posing a challenge to detection accuracy. Tab. 16 shows that R^3Det variants: -GWD, -BCD and -KLD achieve competitive performances: 89.85%/90.07%/89.87% and 97.37%/97.42%/97.62% by the 2007/2012 evaluation metrics, respectively.

6 Conclusion

We have shown how to model the rotated objects as Gaussian distributions and present a novel regression loss and an efficient label assignment strategy, to model the deviation between rotated BBoxes for object detection.

We extend Gaussian modeling to 3-D detection by devising a heading regression and post-processing method that takes advantage of the Gaussian loss while preserving the supervision to object's heading angle in degeneration cases during training. Experimental results on benchmarks (2-D/3-D, aerial/text/face images) show its effectiveness.

Acknowledgment

This work was partly supported by National Key Research and Development Program of China (2020AAA0107600), Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102), and National Science of Foundation China (U20B2068, 61972250, 72061127003).

References

[1] 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*, vol. 10, no. 1, p. 132, 2018.

[2] 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 *IEEE International Conference on Computer Vision*, 2019, pp. 8232–8241.

[3] Q. Ming, L. Miao, Z. Zhou, J. Song, and X. Yang, "Sparse label assignment for oriented object detection in aerial images," *Remote Sensing*, vol. 13, no. 14, p. 2664, 2021.

[4] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang, "East: an efficient and accurate scene text detector," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5551–5560.

[5] X. Liu, D. Liang, S. Yan, D. Chen, Y. Qiao, and J. Yan, "Fots: Fast oriented text spotting with a unified network," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5676–5685.

[6] 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*, vol. 20, no. 11, pp. 3111–3122, 2018.

[7] M. Liao, Z. Zhu, B. Shi, G.-s. Xia, and X. Bai, "Rotation-sensitive regression for oriented scene text detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5909–5918.

[8] X. Shi, S. Shan, M. Kan, S. Wu, and X. Chen, "Real-time rotation-invariant face detection with progressive calibration networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2295–2303.

[9] Y. Zheng, D. Zhang, S. Xie, J. Lu, and J. Zhou, "Rotation-robust intersection over union for 3d object detection," in *European Conference on Computer Vision*, 2020, pp. 464–480.

[10] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3d object detection and tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 11 784–11 793.

[11] 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," in *European Conference on Computer Vision*, 2020, pp. 195–211.

[12] 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 *IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 207–11 216.

[13] 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*, 2015, pp. 91–99.

[14] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2117–2125.

[15] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *IEEE International Conference on Computer Vision*, 2017, pp. 2980–2988.

[16] 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*, 2018, pp. 150–165.

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

[18] X. Yang, J. Yan, Z. Feng, and T. He, "R3det: Refined single-stage detector with feature refinement for rotating object," in *AAAI Conference on Artificial Intelligence*, vol. 35, no. 4, 2021, pp. 3163–3171.

[19] 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 *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 658–666.

[20] 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 *AAAI Conference on Artificial Intelligence*, 2020, pp. 12 993–13 000.

[21] X. Yang and J. Yan, "Arbitrary-oriented object detection with circular smooth label," in *European Conference on Computer Vision*, 2020, pp. 677–694.

[22] X. Yang, L. Hou, Y. Zhou, W. Wang, and J. Yan, "Dense label encoding for boundary discontinuity free rotation detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15 819–15 829.

[23] D. Zhou, J. Fang, X. Song, C. Guan, J. Yin, Y. Dai, and R. Yang, "Iou loss for 2d/3d object detection," in *International Conference on 3D Vision*, 2019, pp. 85–94.

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

[25] A. Bhattacharyya, "On a measure of divergence between two statistical populations defined by their probability distributions," *Bull. Calcutta Math. Soc.*, vol. 35, pp. 99–109, 1943.

[26] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[27] X. Yang, J. Yan, Q. Ming, W. Wang, X. Zhang, and Q. Tian, "Rethinking rotated object detection with gaussian wasserstein distance loss," in *International Conference on Machine Learning*. PMLR, 2021, pp. 11 830–11 841.

[28] X. Yang, X. Yang, J. Yang, Q. Ming, W. Wang, Q. Tian, and J. Yan, "Learning high-precision bounding box for rotated object detection via kullback-leibler divergence," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[29] Y. Li, "Detecting lesion bounding ellipses with gaussian proposal networks," in *International Workshop on Machine Learning in Medical Imaging*, 2019, p. 337–344.

[30] S. Pan, S. Fan, S. W. Wong, J. V. Zidek, and H. Rhodin, "Ellipse detection and localization with applications to knots in sawn lumber images," in *IEEE Winter Conference on Applications of Computer Vision*, 2021, p. 3892–3901.

[31] S. Zhang, C. Chi, Y. Yao, Z. Lei, and S. Z. Li, "Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9759–9768.

[32] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikainen, "Toward arbitrary-oriented ship detection with rotated region proposal and discrimination networks," *International journal of computer vision*, vol. 128, no. 2, pp. 261–318, 2020.

[33] 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*, vol. 43, no. 4, pp. 1452–1459, 2020.

[34] J. Han, J. Ding, N. Xue, and G.-S. Xia, "Redet: A rotation-equivariant detector for aerial object detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2786–2795.

[35] W. Qian, X. Yang, S. Peng, J. Yan, and Y. Guo, "Learning modulated loss for rotated object detection," in *AAAI Conference on Artificial Intelligence*, vol. 35, no. 3, 2021, pp. 2458–2466.

[36] 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.

[37] M. Liao, B. Shi, and X. Bai, "Textboxes++: A single-shot oriented scene text detector," *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 3676–3690, 2018.

[38] 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*, 2016, pp. 21–37.

[39] S. Shi, X. Wang, and H. Li, "Pointrcnn: 3d object proposal generation and detection from point cloud," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 770–779.

[40] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.

[41] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4490–4499.

[42] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 652–660.

[43] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, 2018.

[44] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 697–12 705.

[45] E. Xie, P. Sun, X. Song, W. Wang, X. Liu, D. Liang, C. Shen, and P. Luo, "Polarmask: Single shot instance segmentation with polar representation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 193–12 202.

[46] Y. Yang, J. Chen, X. Zhong, and Y. Deng, "Polygon-to-polygon distance loss for rotated object detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

[47] X. Yang, Y. Zhou, G. Zhang, J. Yang, W. Wang, J. Yan, X. Zhang, and Q. Tian, "The kfiou loss for rotated object detection," *arXiv preprint arXiv:2201.12558*, 2022.

[48] X. Yang and J. Yan, "On the arbitrary-oriented object detection: Classification based approaches revisited," *International Journal of Computer Vision*, vol. 130, no. 5, pp. 1340–1365, 2022.

[49] 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*, vol. 11, no. 24, p. 2930, 2019.

[50] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *IEEE International Conference on Computer Vision*, 2017, pp. 2961–2969.

[51] X. Wang, T. Kong, C. Shen, Y. Jiang, and L. Li, "Solo: Segmenting objects by locations," in *European Conference on Computer Vision*, 2020, pp. 649–665.

[52] Z. Tian, C. Shen, and H. Chen, "Conditional convolutions for instance segmentation," in *European Conference on Computer Vision*. Springer, 2020, pp. 282–298.

[53] 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 *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3974–3983.

[54] Z. Liu, L. Yuan, L. Weng, and Y. Yang, "A high resolution optical satellite image dataset for ship recognition and some new baselines," in *International Conference on Pattern Recognition Applications and Methods*, vol. 2, 2017, pp. 324–331.

[55] V. Jain and E. Learned-Miller, "Fddb: A benchmark for face detection in unconstrained settings," UMass Amherst technical report, Tech. Rep., 2010.

[56] X. Yang, J. Yan, W. Liao, X. Yang, J. Tang, and T. He, "Scrdet++: Detecting small, cluttered and rotated objects via instance-level feature denoising and rotation loss smoothing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[57] Q. Ming, Z. Zhou, L. Miao, H. Zhang, and L. Li, "Dynamic anchor learning for arbitrary-oriented object detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 3, 2021, pp. 2355–2363.

[58] L. Ding and A. Fridman, "Object as distribution," *arXiv preprint arXiv:1907.12929*, 2019.

[59] Q. Ming, L. Miao, Z. Zhou, X. Yang, and Y. Dong, "Optimization for arbitrary-oriented object detection via representation invariance loss," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2021.

[60] Z. Guo, C. Liu, X. Zhang, J. Jiao, X. Ji, and Q. Ye, "Beyond bounding-box: Convex-hull feature adaptation for oriented and densely packed object detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8792–8801.

[61] L. Hou, K. Lu, X. Yang, Y. Li, and J. Xue, "G-rep: Gaussian representation for arbitrary-oriented object detection," *arXiv preprint arXiv:2205.11796*, 2022.

[62] 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," *IEEE Transactions on Geoscience and Remote Sensing*, 2021.

[63] H. Jeffreys, "An invariant form for the prior probability in estimation problems," *Royal Society of London. Series A. Mathematical and Physical Sciences*, vol. 186, no. 1007, pp. 453–461.

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

[65] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2020, pp. 2446–2454.

[66] 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)*, 2016, pp. 265–283.

[67] X. Yang, Y. Zhou, and J. Yan, "Alpharotate: A rotation detection benchmark using tensorflow," *arXiv preprint arXiv:2111.06677*, 2021.

[68] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, Z. Zhang, D. Cheng, C. Zhu, T. Cheng, Q. Zhao, B. Li, X. Lu, R. Zhu, Y. Wu, J. Dai, J. Wang, J. Shi, W. Ouyang, C. C. Loy, and D. Lin, "MMDetection: Open mmlab detection toolbox and benchmark," *arXiv preprint arXiv:1906.07155*, 2019.

[69] G. Cheng, J. Wang, K. Li, X. Xie, C. Lang, Y. Yao, and J. Han, "Anchor-free oriented proposal generator for object detection," *IEEE Transactions on Geoscience and Remote Sensing*, 2022.

[70] 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 *IEEE International Conference on Image Processing*, 2015, pp. 3735–3739.

[71] 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*. IEEE, 2015, pp. 1156–1160.

[72] 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 *International Conference on Document Analysis and Recognition*, vol. 1, 2017, pp. 1454–1459.

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

[74] T. L. Berg, A. C. Berg, J. Edwards, and D. A. Forsyth, "Who's in the picture," in *Advances in Neural Information Processing Systems*, 2005, pp. 137–144.

[75] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[76] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition*. Ieee, 2009, pp. 248–255.

[77] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3354–3361.

[78] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2018.

[79] Z. Tian, C. Shen, H. Chen, and T. He, "Fcos: Fully convolutional one-stage object detection," in *IEEE International Conference on Computer Vision*, 2019, pp. 9627–9636.

[80] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1492–1500.

[81] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *European Conference on Computer Vision*, 2016, pp. 483–499.

[82] F. Yu, D. Wang, E. Shelhamer, and T. Darrell, "Deep layer aggregation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2403–2412.

[83] 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*, vol. 169, pp. 268–279, 2020.

[84] J. Yi, P. Wu, B. Liu, Q. Huang, H. Qu, and D. Metaxas, "Oriented object detection in aerial images with box boundary-aware vectors," in *IEEE Winter Conference on Applications of Computer Vision*, 2021, pp. 2150–2159.

[85] P. Zhao, Z. Qu, Y. Bu, W. Tan, and Q. Guan, "Polardet: A fast, more precise detector for rotated target in aerial images," *International Journal of Remote Sensing*, vol. 42, no. 15, pp. 5821–5851, 2021.

[86] B. Zhong and K. Ao, "Single-stage rotation-decoupled detector for oriented object," *Remote Sensing*, vol. 12, no. 19, p. 3262, 2020.

[87] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *Advances in Neural Information Processing Systems Workshop*, 2017.

[88] Y. Zhou, X. Yang, G. Zhang, J. Wang, Y. Liu, L. Hou, X. Jiang, X. Liu, J. Yan, C. Lyu, W. Zhang, and K. Chen, "Mmrotate: A rotated object detection benchmark using pytorch," *arXiv preprint arXiv:2204.13317*, 2022.

[89] 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*, 2014, pp. 740–755.

[90] Y. Liu, S. Zhang, L. Jin, L. Xie, Y. Wu, and Z. Wang, "Omnidirectional scene text detection with sequential-free box discretization," in *27th International Joint Conference on Artificial Intelligence*, 2019.

[91] Q. Song, F. Yang, L. Yang, C. Liu, M. Hu, and L. Xia, "Learning point-guided localization for detection in remote sensing images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2020.

Xue Yang is a Ph.D. candidate in Computer Science, Shanghai Jiao Tong University, Shanghai, China. He received the B. E. in Automation, Central South University, Hunan, China, in 2016. He also received the M. S. from Chinese Academy of Sciences University, Beijing, China, in 2019. His research interest is computer vision. He published first-authored papers in TPAMI, IJCV, CVPR, ECCV, ICCV, ICML, NeurIPS, AAAI and ACM MM. He is also the leading contributor to the MMRotate and AlphaRotate open-source projects for oriented object detection, and with 5000+ stars in Github.

Gefan Zhang received his B.E. degree in automotive engineering at Tongji University in 2016. In 2018 he became a part-time student in pursuit of M.S. degree at the Department of Computer Science and Engineering, Shanghai Jiao Tong University. His research interest is 3-D object detection, especially lidar point cloud object detection. He is currently also working as a lidar perception engineer at COWA Robot and with focus on 3-D detection for autonomous driving.

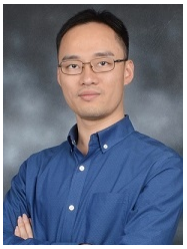
Xiaojiang Yang received the B.E. degree in Physics from Nankai University in 2018 (with minor in Math). He is currently working toward the Ph.D. degree in the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China. His research interests include machine learning and vision, especially for generative models and representation learning. He has first-authored papers in top venues including ICLR and IEEE TNNLS.

Yue Zhou received the B.S. degree in Electronic and Information Engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 2017. He is currently pursuing the Ph.D. degree in Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China. His research interest is computer vision, especially for object detection. He has published papers in CVPR and the leading contributor to the MMRotate open-source project for rotation detection.

Wentao Wang received the B.E. degree from the Beijing University of Chemistry Technology, Beijing, China, in 2016 and received the M.S. degree from Institute of Electrics, Chinese Academy of Sciences, Beijing, China, in 2019. He is pursuing the Ph.D. degree in Computer Science, Shanghai Jiao Tong University, Shanghai, China. His research interests include deep learning and computer vision. He has published first-authored papers in CVPR, ICCV, ACM MM.

Jin Tang received the B.Eng. degree in automation and the Ph.D. degree in computer science from Anhui University, Hefei, China, in 1999 and 2007, respectively. He is currently a Professor with the School of Computer Science and Technology, Anhui University, Hefei, China. His current research interests include computer vision, pattern recognition, and deep learning, especially for remote sensing and object detection.

Tao He received the B. E. and M. E. degrees from Shanghai Jiao Tong University in Electrical Engineering in 2005 and 2008, respectively. He received his PhD in Mechanical and Aerospace Engineering, from Tokyo Institute of Technology, Tokyo, Japan in 2012. He was also once a postdoc with CMU. He has been working on autonomous driving for over one decade and currently he is the founder and the Chief Executive Officer (CEO) of COWAROBOT Co., Ltd. He is the winner of Forbes 40 under 40 China.



Junchi Yan (S'10-M'11-SM'21) is an Associate Professor with Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China. He is also affiliated with Shanghai AI Laboratory, Shanghai, China. Before that, he was a Senior Research Staff Member with IBM Research where he started his career since April 2011, and obtained his PhD in Electrical Engineering, Shanghai Jiao Tong University in 2015. His research interest is machine learning. He served Area Chair for CVPR/AAAI/ICML/NeurIPS, Associate Editor for Pattern Recognition.