LEARNING SYMMETRY-AWARE REGION REPRESENTATIONS FOR ORIENTED OBJECT DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Oriented object detection uses rotated bounding boxes to localize objects in applications such as remote sensing, scene text detection, and industrial inspection. Two-stage oriented detectors typically extract features from rotated proposals, then use these features for classification and box refinement in each proposal's local coordinate frame. However, their detection heads typically do not explicitly account for the symmetry of the proposal's local coordinate frame. Reversing either local axis gives an equivalent description of the same proposal, but affects the prediction targets differently. Class and size remain unchanged, while center offsets and relative angles undergo predictable sign changes. We argue that region features should follow the same transformation laws as the targets they predict. To this end, we propose Symmetry-Aware Region Representations (SRR). We extract features in four equivalent local frames and combine them through the proposed Geometry-Consistent Aggregation (GCA), which weights the frames according to how each target changes sign. We further propose Symmetry-Aware Feature Learning (SFL), which divides features into types by their transformation laws and learns a separate linear map for each type, keeping the head equivariant to local axis reversals. Extensive experiments on DOTA-v1.0, HRSC2016, DIOR-R, RSAR, and FAIR1M show that SRR improves oriented object detection performance, with consistent localization gains across multiple detectors and backbones on DOTA-v1.0.

1 INTRODUCTION

Oriented object detection aims to recognize and localize objects at arbitrary orientations, with applications in remote sensing image analysis (Xia et al., 2018; Yang et al., 2021b), scene text detection (Liao et al., 2018), and industrial inspection (Liu et al., 2019; Wu et al., 2022). Compared with horizontal bounding boxes, oriented boxes fit tilted objects more closely and help distinguish neighboring instances in crowded scenes. Two-stage detectors remain among the strongest oriented detectors (Yu et al., 2024; Yuan et al., 2026). They generate rotated proposals and extract features aligned with each proposal for classification and box refinement (Ding et al., 2019; Xie et al., 2021). Classification and box refinement have different feature requirements. Recognition should not depend on how the box is parameterized, whereas box refinement must track the proposal geometry.

Two-stage oriented detectors commonly use a shared RoI representation for classification and box regression (Ding et al., 2019; Xie et al., 2021; Han et al., 2021b). Other methods tailor region features to different prediction targets. STD (Yu et al., 2024) assigns position, angle, size, and class predictions to successive Transformer stages. Strip R-CNN (Yuan et al., 2026) uses strip convolutions for position and size prediction, while angle prediction shares features with classification. These designs account for the different feature requirements of the prediction targets, but do not explicitly specify how region features should transform under reversals of the proposal's local axes.

Consider reversing the local axes of a rotated proposal. As shown in Figure 1, independently reversing the width and height axes gives four equivalent coordinate frames for the same box. Class and size remain unchanged across these frames. Each center offset changes sign when its corresponding axis is reversed, while the relative angle changes sign when exactly one axis is reversed. These relations arise from the coordinate choice and hold regardless of object appearance. This motivates us to learn region features with the same transformation laws as their targets. A standard detection

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

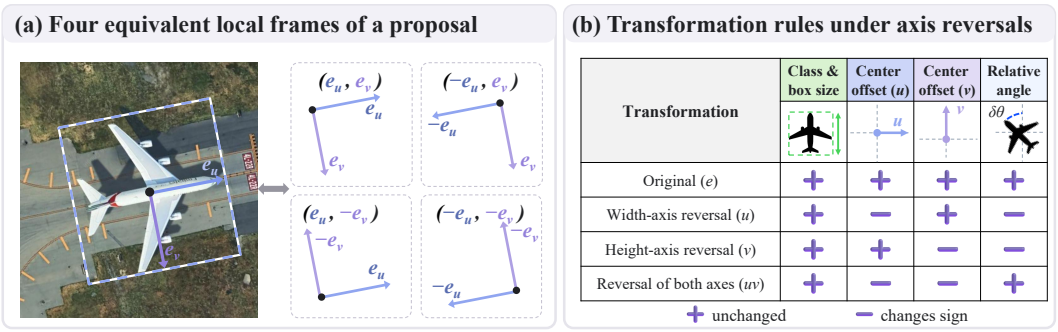


Figure 1: **Prediction targets in equivalent local coordinate frames.** (a) Independently reversing the width and height axes gives four local frames for the same rotated proposal. The proposal and its position in the image remain fixed. (b) Class and size are invariant across frames. Each center offset changes sign when its corresponding axis is reversed, while the relative angle $\delta\theta$, measured in the proposal's local frame, changes sign when exactly one axis is reversed.

head does not enforce this, since it predicts all targets from one shared vector computed in a single frame. To this end, we propose **Symmetry-Aware Region Representations (SRR)** to build these relations into the detection head. We first extract features aligned with each proposal in the four equivalent local frames. The proposed **Geometry-Consistent Aggregation (GCA)** combines these features using the signs associated with each target. This produces four feature types: invariant features for class and size, and features for the two center offsets and the angle that follow their respective sign changes. The resulting head is equivariant to local axis reversals, but ties the linear maps across feature types. We further propose **Symmetry-Aware Feature Learning (SFL)**, which learns a separate linear map for each type while preserving this equivariance.

We evaluate SRR on five benchmark datasets, including DOTA-v1.0, HRSC2016, DIOR-R, RSAR, and FAIR1M. We integrate SRR into Oriented R-CNN, RoI Transformer, and ReDet and evaluate its performance with multiple backbones. As shown in Figure 2, SRR consistently improves localization across different detectors and backbones on DOTA-v1.0, with AP_{75} gains of up to 2.69 points. On HRSC2016, SRR improves AP_{75} and $AP_{50:95}$ by 1.83 and 1.81 points over the Oriented R-CNN baseline. These results show that learning region features according to the transformation laws of their targets provides an effective approach to improving oriented object detection. Our main contributions are summarized as follows.

- We use the local coordinate symmetry of oriented boxes as a guide for region representation learning. The key insight is to make features follow their targets' transformation laws.
- We propose GCA to aggregate frame features according to target transformation laws. SFL learns separate linear maps per type while preserving local equivariance.
- Extensive experiments on five datasets demonstrate the effectiveness of SRR. SRR consistently improves localization across multiple detectors and backbones, demonstrating its broad applicability to two-stage oriented detection.

2 Related Work

Oriented object detection. Conventional oriented detectors follow one-stage or two-stage detection paradigms. One-stage detectors predict class scores and oriented boxes from dense feature maps without a separate proposal stage (Liao et al., 2018; Yang et al., 2021a; Han et al., 2021a). Two-stage detectors first obtain region proposals, then extract aligned region features for classification and box refinement. RoI Transformer (Ding et al., 2019) transforms horizontal proposals into rotated ones, while Oriented R-CNN (Xie et al., 2021) generates rotated proposals directly. Aligning features with these proposals allows the detection head to refine each box in its local coordinate frame. More recently, DETR-based methods formulate oriented detection as set prediction (Dai et al., 2022; Zeng et al., 2024; Lee et al., 2025). Alongside these architectural developments, prior work improves box regression through angle encoding (Yang & Yan, 2020; Yu & Da, 2023), target match-

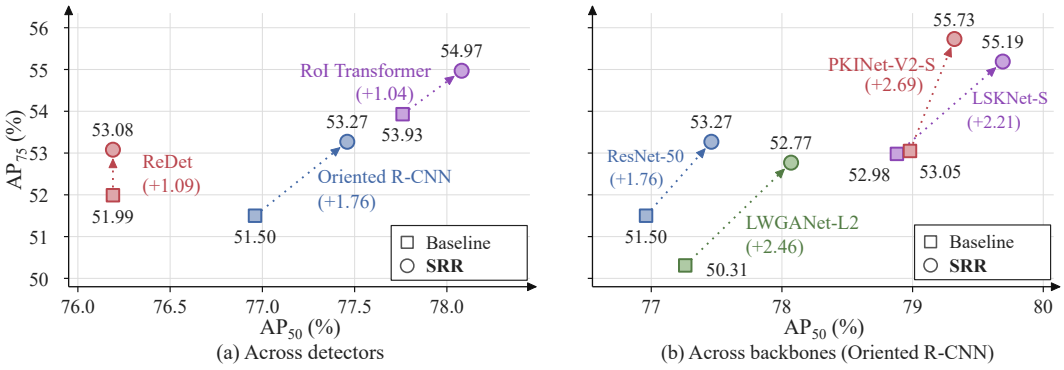


Figure 2: **SRR across detectors and backbones on DOTA-v1.0.** Squares denote the baseline and circles denote SRR. Each arrow connects a baseline to SRR under the same detector and backbone, and labels show AP_{75} gains. ReDet in panel (a) uses ReResNet-50.

ing (Ming et al., 2021), and geometric loss design (Yang et al., 2021b;c). SRR retains the detector's box encoding and losses and instead learns region features that follow the same transformation laws as their targets under local axis reversals.

Feature learning for oriented detection. Feature alignment adapts spatial sampling to the geometry of predicted boxes (Yang et al., 2021a; Han et al., 2021a; Zeng et al., 2024). FAA (Gu et al., 2026) aligns region features to a common reference orientation and combines the aligned and original features for prediction. Task-specific feature learning addresses the different needs of classification and localization. RRD (Liao et al., 2018) and S^2A-Net (Han et al., 2021a) use rotation-sensitive features for regression and invariant features for classification. STD (Yu et al., 2024) assigns position, angle, size, and class predictions to successive Transformer stages. Strip R-CNN (Yuan et al., 2026) uses strip convolutions for position and size prediction, while angle prediction shares features with classification. SRR groups features by the transformation laws of their targets under local axis reversals and learns region features with the corresponding transformation properties.

Equivariant networks. Equivariant networks build a known symmetry into the architecture, so that a transformed input produces features transformed in a prescribed way. Group equivariant CNNs (Cohen & Welling, 2016) and steerable CNNs (Cohen & Welling, 2017) do this for rotations and reflections of the image plane, and Weiler & Cesa (2019) extend steerable CNNs to general subgroups of E(2). Kondor & Trivedi (2018) show that equivariant linear layers take the form of group convolutions. Frame averaging (Puny et al., 2022) makes an arbitrary network equivariant by averaging its outputs over a group. In oriented detection, ReDet (Han et al., 2021b) and FRED (Lee et al., 2024) build rotation-equivariant backbones and derive invariant region features through spatial or orientation alignment, and Wu et al. (2025) study how rotation equivariance affects aerial detectors. These methods handle rotations of the image. SRR handles the reversal of a proposal's local axes, a symmetry that comes from the box parameterization. It applies frame averaging to this four-element group, assigns each prediction target the character that matches its sign changes, and learns equivariant maps between the frames.

3 Method

SRR replaces the detection head of a two-stage oriented detector. We first define the target transformation laws, then describe GCA and SFL in Sections 3.2 and 3.3. Figure 3 shows the forward computation: shared feature extraction is followed by SFL and then GCA for prediction.

3.1 Local Coordinate Symmetry

For a fixed proposal and feature field, independently reversing the two local axes gives four coordinate frames for the same rectangle. A single reversal produces a left-handed frame, while reversing both rotates the frame by π. These frames serve only to compute features. Regression outputs are decoded in the original frame as in the baseline detector.

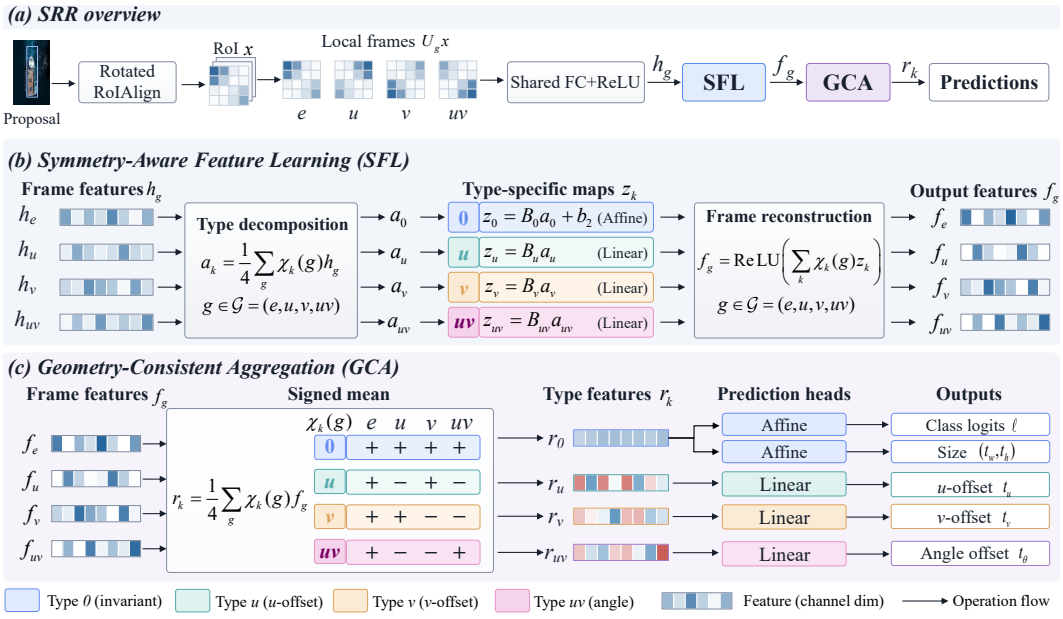


Figure 3: **Overall architecture of SRR.** (a) SRR extracts RoI features using Rotated RoIAlign and represents each RoI in four equivalent local frames obtained by reversing its axes. A shared FC layer processes the four frames, followed by SFL, GCA, and prediction heads. (b) SFL decomposes the frame features into four types with distinct sign transformation laws and applies a separate mapping to each type with a bias only for the invariant type. It reconstructs the frame features before applying ReLU, preserving equivariance to local axis reversals. (c) GCA aggregates the frame features by signed averaging. Each prediction head uses the feature type that follows the transformation law of its target.

Let $B_\theta = [e_u, e_v]$ contain the proposal's orthonormal width and height axes. An axis reversal is $g = \mathrm{diag}(s_u, s_v)$, where $+1$ keeps an axis and -1 reverses it. The four choices form a group:

$$\mathcal{G} = \{g = \mathrm{diag}(s_u, s_v) : s_u, s_v \in \{-1, +1\}\} \simeq C_2 \times C_2. \tag{1}$$

The elements e, u, v, uv correspond to $(s_u, s_v) = (1,1), (-1,1), (1,-1), (-1,-1)$, respectively. Let $x \in \mathbb{R}^{C\times H\times W}$ denote the pooled RoI feature map in the proposal's original local frame. The transformed frame is $B_\theta g$, and $U_g x$ denotes the same RoI feature map in that frame. On a symmetric sampling grid, U_g reverses the corresponding spatial dimensions of the fixed feature map.

For a proposal with center c and side lengths w, h, the center target is $\mathrm{diag}(w,h)^{-1} B_\theta^\top (c^* - c)$, where c^* is the target center. Its components change by s_u and s_v, while class and logarithmic size targets are unchanged. A single axis reversal changes the frame's handedness, giving $\delta\theta \mapsto s_u s_v \delta\theta \pmod{\pi}$ for the relative angle of an undirected box.

We retain the baseline box coder, whose angle residual follows $\delta\theta \mapsto s_u s_v \delta\theta$ under local axis reversals. The head predicts standardized targets $t = S^{-1}(\bar{t} - \mu)$, where $\bar{t}$ denotes unstandardized targets, S is positive diagonal, and $\mu_u = \mu_v = \mu_\theta = 0$. These conditions preserve the target transformation laws. The detector inverts standardization before decoding, as detailed in Appendix A.1.

Writing the class logits as ℓ and the regression outputs as $t = (t_u, t_v, t_w, t_h, t_\theta)$, we require

$$\ell(U_g x) = \ell(x), \qquad t(U_g x) = \rho(g) t(x), \qquad \rho(g) = \mathrm{diag}(s_u, s_v, 1, 1, s_u s_v). \tag{2}$$

Thus each output either keeps or reverses its sign. We use the same four transformation laws for the hidden features. Type 0 is invariant and serves class and size prediction. Types u and v serve the two center offsets, and type uv serves the angle. Their signs are

$$\chi_0(g) = 1, \qquad \chi_u(g) = s_u, \qquad \chi_v(g) = s_v, \qquad \chi_{uv}(g) = s_u s_v. \tag{3}$$

These functions are the characters of $C_2 \times C_2$. Note that a type is defined by a sign pattern, not by an object class, and one type can hold many learned channels.

3.2 GEOMETRY-CONSISTENT AGGREGATION (GCA)

GCA combines features from the four local frames according to the characters of the prediction targets. We first describe the GCA head, which uses two shared fully connected layers before aggregation. SFL replaces the second layer of this head while retaining the final GCA step. The GCA head computes the frame features as

$$\begin{aligned} h_g &= \mathrm{ReLU}(W_1 \,\mathrm{vec}(U_g x) + b_1)\,, \\ f_g &= \mathrm{ReLU}(W_2 h_g + b_2), \end{aligned} \tag{4}$$

where $h_g, f_g \in \mathbb{R}^d$. Since the weights are shared, a local axis reversal only permutes the frame features f_g. For each type $k \in \{0, u, v, uv\}$, GCA computes

$$r_k = \frac{1}{4} \sum_{g \in \mathcal{G}} \chi_k(g) f_g. \tag{5}$$

For class and size, this is an ordinary average. For center and angle, the signs make the aggregated feature follow the same transformation law as its target. For example, reversing the width axis changes the sign of the feature r_u for the center offset while leaving the invariant feature r_0 unchanged. We use the aggregated features to predict the corresponding outputs:

$$\ell = W_{\mathrm{cls}} r_0 + b_{\mathrm{cls}}, \qquad (t_w, t_h) = W_{\mathrm{size}} r_0 + b_{\mathrm{size}}, \tag{6}$$

$$t_u = w_u^\top r_u, \qquad t_v = w_v^\top r_v, \qquad t_\theta = w_\theta^\top r_{uv}. \tag{7}$$

Class and size are predicted from the same invariant feature through separate affine maps. The outputs that change sign carry no bias, because a constant term cannot change sign with the feature.

With these predictors, aggregating features before prediction is equivalent to predicting in each frame, expressing the predictions in the original frame, and averaging them. GCA is thus frame averaging over the full group $\mathcal{G}$ (Puny et al., 2022), which guarantees Equation (2) by construction.

3.3 SYMMETRY-AWARE FEATURE LEARNING (SFL)

The GCA head satisfies the target transformation laws but shares W_2 across all four types in the linear part of its second layer. Local equivariance does not require this sharing: each type can use its own linear map while retaining its sign transformation law. SFL therefore learns a separate map for each type.

Type decomposition. We decompose the intermediate frame features h_g into four types by signed averaging and learn a separate linear map for each type:

$$a_k = \frac{1}{4} \sum_{g} \chi_k(g) h_g, \qquad z_k = B_k a_k + \mathbf{1}[k = 0] b_2. \tag{8}$$

Decomposing the GCA head's second-layer responses before ReLU gives $W_2 a_k + \mathbf{1}[k = 0] b_2$, with W_2 shared across types. SFL replaces W_2 with a separate B_k for each type. For every reversal $q \in \mathcal{G}$, the component $a_k \in \mathbb{R}^d$ follows $a_k(U_q x) = \chi_k(q) a_k(x)$. Each B_k preserves this sign transformation law, with a bias only for the invariant type.

Frame reconstruction. Applying ReLU directly to a component that changes sign would break its transformation law. We instead reconstruct the four frame features before ReLU:

$$f_g = \mathrm{ReLU}\left(\sum_k \chi_k(g) z_k \right). \tag{9}$$

Since each z_k follows its type's transformation law, an axis reversal only permutes the reconstructed frames. As in the GCA head, a shared ReLU preserves equivariance while allowing interactions between types. The components a_k serve intermediate feature learning, whereas GCA aggregates the resulting f_g into r_k for prediction using Equations (5)–(7). Intuitively, the GCA head transforms each frame on its own before ReLU, whereas SFL lets each frame draw on all four.

Equivariant maps and initialization. The separate type maps follow from the equivariance constraint. An equivariant block M_{kj} that sends type j to type k must satisfy

$$\chi_k(g) M_{kj} = M_{kj} \chi_j(g) \qquad \text{for all } g \in \mathcal{G}. \tag{10}$$

For distinct types, some axis reversal gives opposite signs on the two sides, so $M_{kj} = -M_{kj}$ and hence $M_{kj} = 0$. Within a type, equivariance permits any matrix $B_k \in \mathbb{R}^{d\times d}$. We learn the four type matrices using a shared matrix W_2 and three increments,

$$B_0 = W_2, \qquad B_k = W_2 + \Delta_k, \quad k \in \{u, v, uv\}. \tag{11}$$

We initialize W_2 as an ordinary second fully connected layer and the increments to zero, so all four maps initially equal W_2 and SFL starts from the GCA head. We train W_2 and the increments jointly with the detector. The four maps B_k parameterize a group convolution over the frames (Kondor & Trivedi, 2018), as derived in Appendix A.3. By construction, both heads satisfy Equation (2) for any weights. Appendix A.2 gives the proof.

3.4 Efficient Computation in SRR

GCA and SRR share the same first-layer design. Computing four frame responses separately would cost $4md$ matrix FLOPs per RoI, where $m = CHW$. We avoid repeating the full multiplication by decomposing x into four components

$$x_k = \frac{1}{4}\sum_{g\in\mathcal{G}} \chi_k(g) U_g x, \qquad U_g x = \sum_k \chi_k(g) x_k. \tag{12}$$

We recover all four first-layer preactivations exactly, since $W_1 \operatorname{vec}(U_g x) = \sum_k \chi_k(g) W_1 \operatorname{vec}(x_k)$. The saving comes from the fixed relations between mirrored entries. For a pair of entries exchanged by an axis reversal, their sum stays unchanged and their difference changes sign. These two quantities replace the original pair, and combining the corresponding weights preserves its linear response. Applying this decomposition along both spatial axes gives four components with m_k independent coordinates each, where $\sum_k m_k = m$. Other entries follow from fixed signs and need no separate computation. The four reduced matrix products therefore cost md FLOPs in total, matching one ordinary fully connected layer. This count excludes elementwise decomposition and reconstruction. Appendix A.4 gives the construction.

Both heads use four matrix products in the second layer: GCA shares W_2 across frames, while SFL uses one B_k per type. SFL adds $3d^2$ weights but retains the same $4d^2$ matrix FLOPs per RoI.

4 Experiments

4.1 Experimental Settings

Datasets and Evaluation Metrics. We evaluate SRR on DOTA-v1.0 (Xia et al., 2018), HRSC2016 (Liu et al., 2017), DIOR-R (Cheng et al., 2022), RSAR (Zhang et al., 2025), and FAIR1M-v1.0 and v2.0 (Sun et al., 2022). We train on the training and validation sets of DOTA-v1.0, HRSC2016, DIOR-R, and FAIR1M-v2.0, and on the training sets of RSAR and FAIR1M-v1.0, evaluating on the respective official test sets. We report AP_{50}, AP_{75}, and $AP_{50:95}$, averaging over classes and, for $AP_{50:95}$, ten IoU thresholds from 0.50 to 0.95. DOTA-v1.0 results come from the official server. DIOR-R and RSAR use VOC2007 interpolation. Appendices B.1 and B.2 give more details.

Implementation Details. Our experiments use a common MMRotate 1.0 implementation. Unless otherwise specified, we train with AdamW and ImageNet-pretrained backbones, and all ablation studies use Oriented R-CNN with ResNet-50 (He et al., 2016). DOTA-v1.0 and FAIR1M use 1024×1024 patches, with patch detections merged before evaluation. Other images are resized to fit within 800×800 pixels, preserving aspect ratios. Training lasts 36 epochs on HRSC2016 and 12 elsewhere. For PKINet-V2-S, we use 12 and 36 epochs on DOTA-v1.0 and 36 on DIOR-R. We apply random flipping and random rotation, except on RSAR where rotation is disabled. We report means over three training runs with seeds 0–2 and use $\pm$ for sample standard deviations. Gains are calculated before rounding. Appendix B.3 gives the remaining settings.

4.2 Main Results

Results on DOTA-v1.0. SRR achieves competitive performance against published methods in Table 1(a), reaching 80.99 AP_{50} with PKINet-V2-S trained for 36 epochs. Table 2 isolates the effect of

SRR by keeping the detector, backbone, and training schedule fixed within each pair. With Oriented R-CNN and ResNet-50, SRR raises AP_{50} by 0.50 points, AP_{75} by 1.76, and $AP_{50:95}$ by 1.35. As shown in Figure 2, AP_{75} gains exceed AP_{50} gains across all six default configurations, consistent with more accurate box refinement. ReDet gains in AP_{75} despite unchanged AP_{50}, and localization also improves with the longer PKINet-V2-S schedule.

Table 1: **Comparison with published methods on DOTA-v1.0 and DIOR-R.** The lower blocks contain our baseline and SRR results.

(a) DOTA-v1.0

Method	Backbone	AP_{50}
O-RCNN (Xie et al., 2021)	ResNet-50	75.87
ReDet (Han et al., 2021b)	ReResNet	76.25
FAA (Gu et al., 2026)	ResNet-50	76.55
LSKNet (Li et al., 2023)	LSKNet-S	77.49
Strip R-CNN (Yuan et al., 2026)	StripNet-S	78.09
PKINet (Cai et al., 2024)	PKINet-S	78.39
O-RCNN	ResNet-50	76.96
O-RCNN+**SRR**	ResNet-50	77.46
O-RCNN†	PKINet-V2-S	80.49
O-RCNN+**SRR**†	PKINet-V2-S	**80.99**

(b) DIOR-R

Method	Backbone	AP_{50}
O-RCNN (Xie et al., 2021)	ResNet-50	64.30
DCFL (Xu et al., 2023)	ResNet-50	66.80
PKINet (Cai et al., 2024)	PKINet-S	67.03
LWGANet (Lu et al., 2026)	LWGANet-L2	68.53
Strip R-CNN (Yuan et al., 2026)	StripNet-S	68.70
PKINet-v2 (Cai et al., 2026)	PKINet-V2-S	69.40
O-RCNN	ResNet-50	65.07
O-RCNN+**SRR**	ResNet-50	65.97
O-RCNN†	PKINet-V2-S	71.15
O-RCNN+**SRR**†	PKINet-V2-S	**71.65**

DOTA-v1.0 entries use single-scale training and testing. Published results retain their original training and evaluation settings. We train our models for 12 epochs with ResNet-50 and 36 epochs with PKINet-V2-S, marking the latter with †. O-RCNN denotes Oriented R-CNN. Appendix C.1 gives the full comparisons.

Table 2: **SRR across detectors, backbones, and training schedules on DOTA-v1.0.** Each pair compares the original detection head with SRR under the same training and evaluation settings.

Detector	Backbone	Epochs	SRR	AP_{50}	AP_{75}	$AP_{50:95}$
Oriented R-CNN	ResNet-50	12		76.96 ± 0.25	51.50 ± 0.32	47.83 ± 0.28
			✓	$\mathbf{77.46} \pm 0.22$	$\mathbf{53.27} \pm 0.20$	$\mathbf{49.18} \pm 0.26$
RoI Transformer	ResNet-50	12		77.76 ± 0.25	53.93 ± 0.08	49.85 ± 0.17
			✓	$\mathbf{78.08} \pm 0.47$	$\mathbf{54.97} \pm 0.19$	$\mathbf{50.70} \pm 0.12$
ReDet	ReResNet-50	12		76.19 ± 0.06	51.99 ± 0.52	48.13 ± 0.04
			✓	$\mathbf{76.19} \pm 0.27$	$\mathbf{53.08} \pm 0.27$	$\mathbf{48.98} \pm 0.09$
Oriented R-CNN	LSKNet-S	12		78.88 ± 0.42	52.98 ± 0.78	48.98 ± 0.32
			✓	$\mathbf{79.69} \pm 0.47$	$\mathbf{55.19} \pm 0.79$	$\mathbf{50.40} \pm 0.29$
Oriented R-CNN	LWGANet-L2	12		77.26 ± 0.18	50.31 ± 0.29	47.02 ± 0.05
			✓	$\mathbf{78.07} \pm 0.33$	$\mathbf{52.77} \pm 0.45$	$\mathbf{48.81} \pm 0.29$
Oriented R-CNN	PKINet-V2-S	12		78.98 ± 0.38	53.05 ± 0.35	49.16 ± 0.32
			✓	$\mathbf{79.32} \pm 0.50$	$\mathbf{55.73} \pm 0.11$	$\mathbf{50.28} \pm 0.14$
Oriented R-CNN	PKINet-V2-S	36		80.49 ± 0.08	57.65 ± 0.34	52.03 ± 0.30
			✓	$\mathbf{80.99} \pm 0.11$	$\mathbf{59.61} \pm 0.42$	$\mathbf{52.85} \pm 0.39$

Results on DIOR-R. With Oriented R-CNN and ResNet-50, SRR improves all three metrics in Table 4, increasing AP_{50} by 0.90 points and $AP_{50:95}$ by 0.92. With PKINet-V2-S trained for 36 epochs, it raises AP_{50} from 71.15 to 71.65 in Table 1(b). SRR thus improves both the default and stronger baselines. Appendix C.2 gives the full PKINet-V2-S results.

Results on HRSC2016. Using VOC2012 interpolation, SRR improves AP_{75} by 1.83 points and $AP_{50:95}$ by 1.81 with ResNet-50 in Table 3, while AP_{50} changes from 97.41 to 97.35. Both localization metrics also improve with the stronger LSKNet-S and PKINet-V2-S baselines. Appendix C.3 provides results under both VOC interpolation rules.

Results on RSAR and FAIR1M. We evaluate SRR on SAR imagery with RSAR and on fine-grained oriented object detection with FAIR1M. Since the target transformation laws do not depend on sensing modality or category granularity, both use the same GCA and SFL construction. As shown in Table 3, SRR raises $AP_{50:95}$ on RSAR from 36.17 to 37.01 and also improves AP_{75}. AP_{50} improves on both versions of FAIR1M. These results support the use of the same region representation design for SAR imagery and fine-grained detection.

Table 3: **Results on HRSC2016, RSAR, and FAIR1M.** All entries use Oriented R-CNN. HRSC2016 uses 36 training epochs and VOC2012 interpolation. RSAR and FAIR1M use 12 epochs.

Dataset	Backbone	Head	AP_{50}	AP_{75}	$AP_{50:95}$
HRSC2016	ResNet-50	Baseline	$\mathbf{97.41} \pm 0.29$	88.48 ± 1.10	71.35 ± 0.56
		SRR	97.35 ± 0.17	$\mathbf{90.31} \pm 0.05$	$\mathbf{73.16} \pm 0.30$
	LSKNet-S	Baseline	98.49 ± 0.20	91.03 ± 0.55	74.02 ± 0.74
		SRR	$\mathbf{98.55} \pm 0.19$	$\mathbf{91.80} \pm 0.76$	$\mathbf{75.79} \pm 0.18$
	PKINet-V2-S	Baseline	98.40 ± 0.02	91.34 ± 0.60	74.32 ± 0.29
		SRR	$\mathbf{98.48} \pm 0.13$	$\mathbf{92.11} \pm 0.41$	$\mathbf{75.84} \pm 0.32$
RSAR	ResNet-50	Baseline	69.17 ± 0.82	34.26 ± 0.50	36.17 ± 0.19
		SRR	$\mathbf{70.09} \pm 0.42$	$\mathbf{35.18} \pm 0.13$	$\mathbf{37.01} \pm 0.40$
FAIR1M-v1.0	ResNet-50	Baseline	42.04 ± 0.29	—	—
		SRR	$\mathbf{43.02} \pm 0.27$	—	—
FAIR1M-v2.0	ResNet-50	Baseline	47.15 ± 0.31	—	—
		SRR	$\mathbf{47.83} \pm 0.19$	—	—

4.3 Ablation Studies and Head Analysis

Component Ablation. Table 4 compares the baseline, GCA, and full SRR. GCA and SRR satisfy the same local equivariance constraint, but SFL lets each feature type learn a separate map. $AP_{50:95}$ improves at each step on all three datasets. On DOTA-v1.0, GCA gains 0.77 points and SFL adds 0.58 points. On HRSC2016, AP_{50} is near saturation and barely changes, but SFL raises AP_{75} from 89.27 to 90.31 over GCA. The improvement thus comes from the two proposed designs in SRR.

Table 4: **Ablation of GCA and SFL on DOTA-v1.0, HRSC2016, and DIOR-R.** SRR includes both components. We use VOC2012 interpolation for HRSC2016.

Dataset	GCA	SFL	AP_{50}	AP_{75}	$AP_{50:95}$
DOTA-v1.0			76.96 ± 0.25	51.50 ± 0.32	47.83 ± 0.28
	✓		77.03 ± 0.16	52.56 ± 0.66	48.60 ± 0.59
	✓	✓	$\mathbf{77.46} \pm 0.22$	$\mathbf{53.27} \pm 0.20$	$\mathbf{49.18} \pm 0.26$
HRSC2016 VOC2012			$\mathbf{97.41} \pm 0.29$	88.48 ± 1.10	71.35 ± 0.56
	✓		97.29 ± 0.31	89.27 ± 0.50	72.44 ± 0.36
	✓	✓	97.35 ± 0.17	$\mathbf{90.31} \pm 0.05$	$\mathbf{73.16} \pm 0.30$
DIOR-R			65.07 ± 0.44	42.95 ± 0.04	41.17 ± 0.17
	✓		65.96 ± 0.56	43.53 ± 0.38	41.93 ± 0.20
	✓	✓	$\mathbf{65.97} \pm 0.40$	$\mathbf{43.90} \pm 0.15$	$\mathbf{42.09} \pm 0.16$

Localization on Shared Proposals. We compare box refinement using the same baseline RPN proposals with initial IoU of at least 0.5. Each model uses its own learned features, and refined boxes are evaluated before NMS. As shown in Appendix C.5, mean output IoU on HRSC2016 increases from 0.8504 for the baseline to 0.8578 with GCA and 0.8597 with SRR. On DIOR-R, it increases from 0.7966 to 0.8004 and 0.8026, respectively. Center, size, and orientation errors also decrease at each step. Since the proposals are held fixed, these results indicate that improved localization contributes to the detection gains.

Head Comparisons and Model Capacity. We compare SRR with alternative heads under the same detector and training settings in Table 5. Compared with the re-implemented FAA (Gu et al., 2026) and Strip Head (Yuan et al., 2026), SRR achieves higher AP_{75} and $AP_{50:95}$ with fewer parameters and FLOPs. To examine whether increased capacity can explain these gains, we also widen both hidden layers of the baseline and GCA. The widened baseline matches SRR in parameters and FLOPs but achieves only 47.86 $AP_{50:95}$, compared with 49.18 for SRR. Widening GCA retains a shared linear map across feature types and improves $AP_{50:95}$ only from 48.60 to 48.69. These results show that learning separate maps for different feature types is more effective than simply increasing hidden width.

Table 5: **Comparison of detection accuracy and model cost across head architectures and hidden widths on DOTA-v1.0.** Both hidden layers have the width in parentheses. Parameters and FLOPs refer to the full detector. FLOPs are computed for a 1024×1024 input and 2,000 RoIs, excluding elementwise operations and FAA's Fourier transform and bilinear sampling.

Head	Params	FLOPs	AP_{50}	AP_{75}	$AP_{50:95}$
Baseline (1,024)	41.37M	224.59G	76.96 ± 0.25	51.50 ± 0.32	47.83 ± 0.28
Baseline (1,236)	44.51M	230.88G	76.87 ± 0.10	51.75 ± 0.38	47.86 ± 0.17
GCA (1,024)	41.37M	230.88G	77.03 ± 0.16	52.56 ± 0.66	48.60 ± 0.59
GCA (1,236)	44.51M	240.04G	77.09 ± 0.18	52.57 ± 0.36	48.69 ± 0.32
FAA (Gu et al., 2026)	44.84M	231.64G	77.06 ± 0.36	51.66 ± 0.54	47.85 ± 0.26
Strip Head (Yuan et al., 2026)	55.98M	318.28G	77.33 ± 0.07	51.45 ± 0.36	47.99 ± 0.34
SRR	44.51M	230.88G	$\mathbf{77.46} \pm 0.22$	$\mathbf{53.27} \pm 0.20$	$\mathbf{49.18} \pm 0.26$

Comparison with Task-Specific Branches. We compare SFL with task-specific branches that follow the same target transformation laws. Each branch processes all four local frames, whereas SFL learns a separate mapping for each feature type. As shown in Table 6, SRR improves $AP_{50:95}$ by 0.34 points over the branch head with the same parameter count, while using 42% fewer FLOPs and reducing head latency from 1.39 to 1.09 ms. Reducing the branch width to approximately match FLOPs gives a slightly lower latency of 1.00 ms, but SRR achieves 0.90 points higher $AP_{50:95}$. Learning by feature type thus offers a better balance between accuracy and computation than separate task branches. Appendix B.3 provides the branch design and measurement settings.

Table 6: **Task-specific branches and SRR on DOTA-v1.0.** Hidden dim. denotes the second-layer output width in each branch or in SRR. We count head parameters and compute FLOPs for one RoI. Head latency is measured for 512 RoIs on an RTX 6000D GPU using FP32 inference.

Head	Hidden dim.	Params	FLOPs	Latency	AP_{50}	AP_{75}	$AP_{50:95}$
Task-specific	1,024	17.06M	29.64M	1.39 ms	77.30 ± 0.09	53.11 ± 0.26	48.84 ± 0.12
Task-specific	256	13.90M	17.05M	1.00 ms	77.20 ± 0.03	52.33 ± 0.44	48.28 ± 0.22
SRR	1,024	17.06M	17.06M	1.09 ms	$\mathbf{77.46} \pm 0.22$	$\mathbf{53.27} \pm 0.20$	$\mathbf{49.18} \pm 0.26$

5 Conclusion

We propose Symmetry-Aware Region Representations (SRR) for oriented object detection. Equivalent local frames describe the same proposal, but its targets follow different transformation laws. Through GCA and SFL, SRR learns region features that follow these laws, preserving equivariance to local axis reversals. Experiments on five benchmarks demonstrate its effectiveness, with consistent localization gains across multiple detectors and backbones. These results show that the local coordinate symmetry of oriented boxes provides a useful basis for designing region representations. The current design uses explicit proposal frames in two-stage detectors. Future work will explore this principle for feature learning in one-stage and DETR-based oriented detectors.

AI USE STATEMENT

In this work, we used generative AI tools to provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, and interpret results. We have not used generative AI tools to help develop theoretical models or conceptual frameworks, formulate mathematical claims, assist with translation, or support qualitative and thematic data analysis, and the rest of the required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools to create or edit software code, draft parts of this research paper, summarize or analyse existing literature, and edit this research paper to improve readability. We have reviewed all AI-assisted work. We checked LLM-generated research ideas for potential plagiarism through a manual literature survey. LLM-generated code was verified and tested for correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We describe SRR in Section 3. Appendix A fixes the box encoding conventions, proves that both heads satisfy the required transformation laws, and derives the efficient computation of the first layer, together with the changes needed to integrate SRR into an existing detector. Appendix B specifies the datasets, evaluation metrics, and training hyperparameters, and Appendix C reports the full comparisons and the additional analyses. The code, trained checkpoints, and training and evaluation logs will be released to reproduce the findings reported above.

REFERENCES

Xinhao Cai, Qiuxia Lai, Yuwei Wang, Wenguan Wang, Zeren Sun, and Yazhou Yao. Poly kernel inception network for remote sensing detection. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27706–27716. IEEE, 2024.

Xinhao Cai, Liulei Li, Gensheng Pei, Zeren Sun, Yazhou Yao, and Wenguan Wang. Pkinet-v2: Towards powerful and efficient poly-kernel remote sensing object detection. *arXiv preprint arXiv:2603.16341*, 2026.

Gong Cheng, Jiabao Wang, Ke Li, Xingxing Xie, Chunbo Lang, Yanqing Yao, and Junwei Han. Anchor-free oriented proposal generator for object detection. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–11, 2022.

Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pp. 2990–2999. PMLR, 2016.

Taco S. Cohen and Max Welling. Steerable CNNs. In *International Conference on Learning Representations*, 2017.

Linhui Dai, Hong Liu, Hao Tang, Zhiwei Wu, and Pinhao Song. Ao2-detr: Arbitrary-oriented object detection transformer. *IEEE transactions on circuits and systems for video technology*, 33 (5):2342–2356, 2022.

Jian Ding, Nan Xue, Yang Long, Gui-Song Xia, and Qikai Lu. Learning roi transformer for oriented object detection in aerial images. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2844–2853. IEEE, 2019.

Chenlin Fu and Yingying Zhu. Bghr: bridging the gap between hbox-supervised and rbox-supervised oriented object detection via adaptive fine-grained sample mining. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 3022–3030, 2025.

Changyu Gu, Linwei Chen, Lin Gu, and Ying Fu. Fourier angle alignment for oriented object detection in remote sensing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 42225–42235, June 2026.

Jiaming Han, Jian Ding, Jie Li, and Gui-Song Xia. Align deep features for oriented object detection. *IEEE transactions on geoscience and remote sensing*, 60:1–11, 2021a.

Jiaming Han, Jian Ding, Nan Xue, and Gui-Song Xia. Redet: A rotation-equivariant detector for aerial object detection. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2785–2794. IEEE, 2021b.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Zhanchao Huang, Wei Li, Xiang-Gen Xia, and Ran Tao. A general Gaussian heatmap label assignment for arbitrary-oriented object detection. *IEEE Transactions on Image Processing*, 31: 1895–1910, 2022.

Ziyue Huang, Yongchao Feng, Qingjie Liu, and Yunhong Wang. Mutdet: Mutually optimizing pretraining for remote sensing object detection. In *European Conference on Computer Vision*, pp. 1–17. Springer, 2024.

Risi Kondor and Shubhendu Trivedi. On the generalization of equivariance and convolution in neural networks to the action of compact groups. In *International conference on machine learning*, pp. 2747–2755. PMLR, 2018.

Chanho Lee, Jinsu Son, Hyounguk Shon, Yunho Jeon, and Junmo Kim. Fred: Towards a full rotation-equivariance in aerial image object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 2883–2891, 2024.

Hakjin Lee, Minki Song, Jamyoung Koo, and Junghoon Seo. Hausdorff distance matching with adaptive query denoising for rotated detection transformer. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1872–1882. IEEE, 2025.

Ke Li, Gang Wan, Gong Cheng, Liqiu Meng, and Junwei Han. Object detection in optical remote sensing images: A survey and a new benchmark. *ISPRS journal of photogrammetry and remote sensing*, 159:296–307, 2020.

Wentong Li, Yijie Chen, Kaixuan Hu, and Jianke Zhu. Oriented RepPoints for aerial object detection. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1819–1828, 2022.

Yuxuan Li, Qibin Hou, Zhaohui Zheng, Ming-Ming Cheng, Jian Yang, and Xiang Li. Large selective kernel network for remote sensing object detection. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16748–16759. IEEE, 2023.

Minghui Liao, Zhen Zhu, Baoguang Shi, Gui-song Xia, and Xiang Bai. Rotation-sensitive regression for oriented scene text detection. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5909–5918. IEEE, 2018.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2999–3007, 2017.

Yuekai Liu, Hongli Gao, Liang Guo, Aoping Qin, Canyu Cai, and Zhichao You. A data-flow oriented deep ensemble learning method for real-time surface defect inspection. *IEEE Transactions on Instrumentation and Measurement*, 69(7):4681–4691, 2019.

Zikun Liu, Liu Yuan, Lubin Weng, and Yiping Yang. A high resolution optical satellite image dataset for ship recognition and some new baselines. In *International conference on pattern recognition applications and methods*, volume 2, pp. 324–331. SciTePress, 2017.

Wei Lu, Xue Yang, and Si-Bao Chen. Lwganet: Addressing spatial and channel redundancy in remote sensing visual tasks with light-weight grouped attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 7574–7582, 2026.

Qi Ming, Lingjuan Miao, Zhiqiang Zhou, Xue Yang, and Yunpeng Dong. Optimization for arbitrary-oriented object detection via representation invariance loss. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5, 2021.

Qi Ming, Lingjuan Miao, Zhiqiang Zhou, Junjie Song, Yunpeng Dong, and Xue Yang. Task interleaving and orientation estimation for high-precision oriented object detection in aerial images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 196:241–255, 2023.

Yifan Pu, Yiru Wang, Zhuofan Xia, Yizeng Han, Yulin Wang, Weihao Gan, Zidong Wang, Shiji Song, and Gao Huang. Adaptive rotated convolution for rotated object detection. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6566–6577. IEEE, 2023.

Omri Puny, Matan Atzmon, Heli Ben-Hamu, Ishan Misra, Aditya Grover, Edward J. Smith, and Yaron Lipman. Frame averaging for invariant and equivariant network design. In *International Conference on Learning Representations*, 2022.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, volume 28, 2015.

Xian Sun, Peijin Wang, Zhiyuan Yan, Feng Xu, Ruiping Wang, Wenhui Diao, Jin Chen, Jihao Li, Yingchao Feng, Tao Xu, et al. Fair1m: A benchmark dataset for fine-grained object recognition in high-resolution remote sensing imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 184:116–130, 2022.

Jiangshan Wang, Yifan Pu, Yizeng Han, Jiayi Guo, Yiru Wang, Xiu Li, and Gao Huang. Gra: Detecting oriented objects through group-wise rotating and attention. In *European Conference on Computer Vision*, pp. 298–315. Springer, 2024.

Maurice Weiler and Gabriele Cesa. General e (2)-equivariant steerable cnns. In *Advances in neural information processing systems*, volume 32, 2019.

Hongjin Wu, Ruoshan Lei, and Yibing Peng. Pcbnet: A lightweight convolutional neural network for defect inspection in surface mount technology. *IEEE Transactions on Instrumentation and Measurement*, 71:1–14, 2022.

Xiuyu Wu, Xinhao Wang, Xiubin Zhu, Lan Yang, Jiyuan Liu, and Xingchen Hu. Measuring the impact of rotation equivariance on aerial object detection. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7329–7339. IEEE, 2025.

Gui-Song Xia, Xiang Bai, Jian Ding, Zhen Zhu, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. Dota: A large-scale dataset for object detection in aerial images. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3974–3983. IEEE, 2018.

Xingxing Xie, Gong Cheng, Jiabao Wang, Xiwen Yao, and Junwei Han. Oriented r-cnn for object detection. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3500–3509. IEEE, 2021.

Chang Xu, Jian Ding, Jinwang Wang, Wen Yang, Huai Yu, Lei Yu, and Gui-Song Xia. Dynamic coarse-to-fine learning for oriented tiny object detection. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7318–7328, 2023.

Xue Yang and Junchi Yan. Arbitrary-oriented object detection with circular smooth label. In *European conference on computer vision*, pp. 677–694. Springer, 2020.

Xue Yang, Junchi Yan, Ziming Feng, and Tao He. R3det: Refined single-stage detector with feature refinement for rotating object. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 3163–3171, 2021a.

Xue Yang, Junchi Yan, Qi Ming, Wentao Wang, Xiaopeng Zhang, and Qi Tian. Rethinking rotated object detection with gaussian wasserstein distance loss. In *International conference on machine learning*, pp. 11830–11841. PMLR, 2021b.

Xue Yang, Xiaojiang Yang, Jirui Yang, Qi Ming, Wentao Wang, Qi Tian, and Junchi Yan. Learning high-precision bounding box for rotated object detection via kullback-leibler divergence. In *Advances in Neural Information Processing Systems*, volume 34, pp. 18381–18394, 2021c.

Hongtian Yu, Yunjie Tian, Qixiang Ye, and Yunfan Liu. Spatial transform decoupling for oriented object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 6782–6790, 2024.

Yi Yu and Feipeng Da. Phase-shifting coder: Predicting accurate orientation in oriented object detection. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13354–13363. IEEE, 2023.

Xinbin Yuan, Zhaohui Zheng, Yuxuan Li, Xialei Liu, Li Liu, Xiang Li, Qibin Hou, and Ming-Ming Cheng. Strip r-cnn: Large strip convolution for remote sensing object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 12259–12267, 2026.

Ying Zeng, Yushi Chen, Xue Yang, Qingyun Li, and Junchi Yan. Ars-detr: Aspect ratio-sensitive detection transformer for aerial oriented object detection. *IEEE transactions on geoscience and remote sensing*, 62:1–15, 2024.

Xin Zhang, Xue Yang, Yuxuan Li, Jian Yang, Ming-Ming Cheng, and Xiang Li. Rsar: Restricted state angle resolver and rotated sar benchmark. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7416–7426. IEEE, 2025.

Jiaqi Zhao, Zeyu Ding, Yong Zhou, Hancheng Zhu, Wen-Liang Du, and Rui Yao. Rediffdet: Rotation-equivariant diffusion model for oriented object detection. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24429–24439. IEEE, 2025.

SUPPLEMENTARY MATERIAL

Appendix A gives the coordinate and box encoding conventions, the equivariance proof, the efficient computation of the first layer, and the integration of SRR into existing detectors. Appendix B provides additional details on the datasets, evaluation metrics, and implementation. Appendix C gives full comparisons with published detectors, results with PKINet-V2-S, and additional HRSC2016 evaluations. It also includes baseline reproduction, an analysis of training settings and localization, and qualitative detection results on DOTA-v1.0 and RSAR.

A MORE METHOD DETAILS

A.1 LOCAL COORDINATES AND DECODING CONSISTENCY

Changing the local axes re-expresses the same sampled region. We describe the coordinate and box encoding conventions, then show that reversing both axes leaves the decoded box unchanged. We retain the baseline's projected-center encoding. Canonical boxes use the long-edge convention (le90), with $w \geq h$ and angles in $[-\pi/2, \pi/2)$. Image coordinates increase rightward and downward. Angles are measured clockwise from the positive horizontal axis. The proposal axes are $e_u = (\cos\theta, \sin\theta)^\top$ and $e_v = (-\sin\theta, \cos\theta)^\top$.

Let $F : \mathbb{R}^2 \to \mathbb{R}^C$ be the image feature field and $B = [e_u, e_v]$ the orthonormal frame of a proposal centered at c. The local feature field is $f(\xi) = F(c + B\xi)$, where ξ measures distance along the local axes. Expressing the region in $B' = Bg$ gives $f'(\xi) = f(g\xi)$. On a symmetric RoI sampling grid, this change reverses the corresponding spatial dimensions, as specified by U_g.

Reversing one axis also reverses the handedness of the frame. For a planar rotation R_α, the identity $gR_\alpha = R_{\det(g)\alpha}g$ gives the resulting transformation law for relative angles. The head uses these left-handed frames internally. The box coder receives the original right-handed proposal, and width and height retain their labels under all axis reversals.

For the unstandardized targets $\bar{t}$ in Section 3.1, the network predicts $t = S^{-1}(\bar{t} - \mu)$. Here μ contains the target means, and S is diagonal with positive target standard deviations. In other words, each target is centered and divided by its standard deviation before the head predicts it. The center and angle means are zero ($\mu_u = \mu_v = \mu_\theta = 0$), preserving their transformation laws.

Edge swapping selects between two equivalent target boxes whose side lengths are exchanged and whose angles differ by $\pi/2$. After wrapping angles to $[-\pi/2, \pi/2)$, we keep the original target only if its absolute angle residual is strictly smaller than that of the exchanged target, so $|\delta\theta| \leq \pi/4$. When the two residuals have equal magnitude, we select the exchanged target. Exact squares keep their side order during long-edge canonicalization. With this tie rule, reversing one axis maps the encoded residual $\delta\theta$ to $-\delta\theta$ in every case, including $|\delta\theta| = \pi/4$.

Under these conventions, the right-handed frames B_θ and $B_{\theta+\pi} = -B_\theta$ describe the same rectangle. With the feature field and proposal support fixed, their RoI feature maps differ by a reversal of both spatial axes. The box outputs then transform by $\mathrm{diag}(-1, -1, 1, 1, 1)$. Let $\bar{t}_u, \bar{t}_v$ denote the unstandardized center offsets. The decoded center is unchanged because

$$B_{\theta+\pi}\,\mathrm{diag}(w, h)(-\bar{t}_u, -\bar{t}_v)^\top = B_\theta\,\mathrm{diag}(w, h)(\bar{t}_u, \bar{t}_v)^\top. \tag{A.1}$$

The decoded side lengths also agree. The angles differ by π before canonicalization, so both frames decode to the same oriented box.

A.2 EQUIVARIANCE TO LOCAL AXIS REVERSALS

An axis reversal permutes the four local frames. Signed averaging converts this permutation into the transformation law required by each output. We use this relation to prove equivariance under the RoI feature map action U_q from Section 3.1.

For feature vectors y_g indexed by local frame, the character transform and its inverse are

$$\widehat{y}_k = \frac{1}{4}\sum_{g\in\mathcal{G}} \chi_k(g) y_g, \qquad y_g = \sum_k \chi_k(g)\widehat{y}_k. \tag{A.2}$$

The inverse follows from character orthogonality, $\frac{1}{4}\sum_g \chi_k(g)\chi_j(g) = \mathbf{1}[k=j]$. If an input action q gives $y'_g = y_{gq}$, changing variables to $j = gq$ yields

$$\widehat{y}'_k = \frac{1}{4}\sum_j \chi_k(jq)y_j = \chi_k(q)\widehat{y}_k. \tag{A.3}$$

Proposition 1 (Equivariance to local axis reversals)**.** *With the proposal region and pooled feature field held fixed, GCA and SRR satisfy Equation* (2) *for arbitrary learned weights under the specified action on the RoI feature map.*

Proof. We now show that the two heads satisfy Equation (2) for arbitrary learned weights. For an input transformation $q \in \mathcal{G}$, the feature vectors from the first layer satisfy

$$h_g(U_q x) = \mathrm{ReLU}(W_1 \,\mathrm{vec}(U_{gq}x) + b_1) = h_{gq}(x). \tag{A.4}$$

Equation (A.3) gives $a_k(U_q x) = \chi_k(q)a_k(x)$. Each linear map B_k acts within one type and commutes with its scalar sign. The bias occurs only in type 0, where $\chi_0(q) = 1$. Thus $z_k(U_q x) = \chi_k(q)z_k(x)$. Reconstructing the features in each frame yields

$$\sum_k \chi_k(g)z_k(U_q x) = \sum_k \chi_k(gq)z_k(x). \tag{A.5}$$

The reconstructed features are therefore permuted across frames. Elementwise ReLU commutes with this permutation, so $f_g(U_q x) = f_{gq}(x)$. Thus the final character transform gives $r_k(U_q x) = \chi_k(q)r_k(x)$. Equations (6)–(7) assign each output its matching type, which establishes Equation (2). For GCA, the shared second layer directly preserves the frame permutation, and the same aggregation argument applies. This completes the proof. □

A.3 Relation to Group Convolution

Recall that SFL decomposes the hidden features into four types and applies one map per type. The decomposition expresses these features in a basis where each axis reversal acts by a sign, and an equivariant linear map acts separately on each type in this basis. Reconstructing the frame features gives the group convolution form of the same map (Cohen & Welling, 2017; Kondor & Trivedi, 2018).

Let L denote the linear part of SFL before the bias and ReLU. Substituting the type decomposition into the reconstruction yields

$$(Lh)_g = \sum_k \chi_k(g)B_k \left(\frac{1}{4}\sum_{j\in\mathcal{G}} \chi_k(j)h_j\right) \tag{A.6}$$

$$= \sum_{q\in\mathcal{G}} K(q)h_{gq}, \qquad K(q) = \frac{1}{4}\sum_k \chi_k(q)B_k. \tag{A.7}$$

Here we set $j = gq$ and use $\chi_k(g)^2 = 1$. The inverse relation is $B_k = \sum_q \chi_k(q)K(q)$, so the four type matrices and the four kernel matrices describe the same family of linear maps. The invariant bias b_2 becomes the same bias in every reconstructed frame.

For the GCA head, all four matrices equal W_2. Character orthogonality then gives

$$K(e) = W_2, \qquad K(q) = 0 \quad (q \neq e). \tag{A.8}$$

Thus the GCA head applies its second-layer linear map independently in each frame. SFL learns the four type matrices separately, allowing interactions between frames in the linear stage. Both heads apply ReLU after summing the type components in each frame. Each final type is therefore obtained from nonlinear responses to these sums and can depend on multiple input types.

A.4 Efficient Computation of the First Layer

A spatial reversal exchanges mirrored entries of the RoI feature map. Their sum stays unchanged, while their difference changes sign. A reversal therefore only flips the sign of the odd part, so the

even and odd components can be computed once and reused in every frame. We use these sums and differences to recover the first-layer responses in all four frames without repeating the full matrix multiplication.

For one spatial axis, pair input entries (x_a, x_b) and their weights (w_a, w_b) at reflected locations. Define the even (e) and odd (o) components as

$$x_e = \frac{x_a + x_b}{2}, \quad x_o = \frac{x_a - x_b}{2}, \quad w_e = w_a + w_b, \quad w_o = w_a - w_b. \tag{A.9}$$

Then $w_a x_a + w_b x_b = w_e x_e + w_o x_o$. Reversing the input leaves the even contribution unchanged and changes the sign of the odd contribution.

Applying this decomposition along both RoI axes splits the flattened input into four 2D parity blocks (e, e), (o, e), (e, o), and (o, o). They follow the transformation laws of feature types 0, u, v, and uv, respectively. Signed sums of their linear responses recover all four first-layer preactivations exactly. An unpaired center entry on an odd-sized axis contributes only to the even component.

Let $m = CHW$ be the input dimension, d the hidden width, and m_k the number of input entries assigned to type k. For a 7×7 grid,

$$(m_0, m_u, m_v, m_{uv}) = C(16, 12, 12, 9), \qquad \sum_k m_k = m. \tag{A.10}$$

The first-layer maps have dimensions $d \times m_k$. Together they require dm FLOPs, the same as one ordinary fully connected layer. Evaluating the four frames separately requires $4dm$ FLOPs, so the decomposition reduces this count by 75%. The bias enters the invariant contribution before reconstruction.

Matrix multiplication in the second hidden layer costs d^2 FLOPs per RoI for the baseline and $4d^2$ for both GCA and SRR. SFL therefore adds $3d^2$ weights to GCA without increasing the cost of these matrix multiplications. These counts exclude prediction layers and biases. Appendix B.3 gives the counting conventions.

A.5 Detector Integration

SRR replaces the detection head of Oriented R-CNN and RoI Transformer (Ding et al., 2019). It takes ordinary Rotated RoIAlign features and predicts class logits and five class-agnostic box regression outputs. The head uses the local coordinates and box encoding from Section 3.1. The detector's standard box coder converts these outputs into one box per proposal by inverting the regression targets relative to the proposal coordinates.

For ReDet (Han et al., 2021b), SRR replaces the detection head. It retains the rotation-equivariant backbone (ReResNet), rotation-equivariant feature pyramid (ReFPN), first-stage head, and second-stage rotation-invariant RoI alignment module (RiRoIAlign).

ReDet features also have orientation channels, so U_g must transform these channels together with the spatial grid. Index the eight orientation channels by $j \in \{0, 1, \ldots, 7\}$. Width reversal flips the spatial width axis and permutes the channels by $j \mapsto (4 - j) \bmod 8$. Height reversal flips the height axis and uses $j \mapsto -j \bmod 8$. Composing them flips both spatial axes and shifts the orientation channels by four ($j \mapsto (j + 4) \bmod 8$). These operations satisfy the same four-element group law, so the proof in Appendix A.2 applies to this combined action.

B Experimental Details

B.1 Datasets

DOTA-v1.0. DOTA-v1.0 (Xia et al., 2018) contains 2,806 aerial images and 188,282 annotated instances across 15 categories. Image sizes range from approximately 800×800 to $4{,}000 \times 4{,}000$ pixels. Each instance is annotated by an arbitrary quadrilateral whose four vertices are ordered clockwise. We use the oriented detection task. The official split allocates approximately half of the images to training and one sixth to validation. The remaining third forms the test set. The 15 categories are plane (PL), baseball diamond (BD), bridge (BR), ground track field (GTF), small

vehicle (SV), large vehicle (LV), ship (SH), tennis court (TC), basketball court (BC), storage tank (ST), soccer ball field (SBF), roundabout (RA), harbor (HA), swimming pool (SP), and helicopter (HC).

HRSC2016. The High Resolution Ship Collection 2016 (HRSC2016) dataset (Liu et al., 2017) contains satellite images from six harbors. Image sizes range from 300×300 to $1{,}500 \times 900$ pixels. The dataset provides horizontal and rotated bounding boxes for 2,976 ship instances. Its annotations distinguish ship types through a three-level category hierarchy. We merge these labels into a single ship category for oriented detection. The test split used in our experiments contains 453 images and 1,228 non-ignored ship instances.

DIOR-R. DIOR-R (Cheng et al., 2022) adds oriented bounding boxes to DIOR (Li et al., 2020), whose original annotations use horizontal boxes. It shares the same 23,463 remote sensing images collected from Google Earth. Each image has 800×800 pixels, and the dataset covers 20 object categories.

RSAR. RSAR (Zhang et al., 2025) is a synthetic aperture radar dataset for oriented object detection, constructed from SARDet-100K. It contains 95,842 images and 183,534 instances across six categories, with 78,837 training images, 8,467 validation images, and 8,538 test images. We train on the training set and evaluate on the test set.

FAIR1M. FAIR1M-v1.0 and FAIR1M-v2.0 (Sun et al., 2022) each contain 37 fine-grained object categories. Version 2.0 expands the validation set of version 1.0. We use the training annotations for version 1.0 and the combined training and validation annotations for version 2.0. The two versions have separate test sets, and we submit predictions to their respective official evaluations.

B.2 Evaluation Metrics

We report AP_{50}, AP_{75}, and $AP_{50:95}$, with AP averaged over object classes. AP_{50} and AP_{75} use IoU thresholds of 0.50 and 0.75, respectively. $AP_{50:95}$ further averages AP over ten IoU thresholds from 0.50 to 0.95 in steps of 0.05.

For HRSC2016, we report both VOC2007 eleven-point interpolation and VOC2012 all-point interpolation at each IoU threshold, and the tables identify the interpolation rule. Both interpolation rules are applied to the same predictions. DIOR-R and RSAR use VOC2007 eleven-point interpolation. For FAIR1M, we map patch detections to the original image coordinates and merge them before submitting predictions to the official evaluation servers, which return AP_{50} but not AP_{75} or $AP_{50:95}$. For DOTA-v1.0, we map patch detections to the original image coordinates and merge them before submitting predictions to the Task1 online evaluation server. The server returns the standard DOTA mAP with per-class AP, together with AP at IoU 0.50 and 0.75 and the mean AP over IoU 0.50:0.95. The AP at IoU 0.50 in the two reports is identical, and we report it as AP_{50}. All DOTA-v1.0 metrics in this paper come from the server.

B.3 Implementation Details

We implement SRR and the reproduced baselines in a modified version of MMRotate 1.0. This implementation fixes numerical issues in mixed precision training, matches the angle normalization behavior of MMRotate 0.x before rotated RoIAlign, and replaces the rotated IoU implementation to improve computational efficiency and numerical stability.

Table 7 summarizes the default training hyperparameters. The PKINet-V2-S comparisons in Table 10 use 36 epochs on DOTA-v1.0, DIOR-R, and HRSC2016, with learning-rate decay at epochs 24 and 33. Resizing preserves aspect ratios, and random flips include horizontal, vertical, and diagonal directions. DOTA-v1.0 trainval and test images are cropped at their original scale into 1024×1024 patches with a nominal overlap of 200 pixels; boundary windows shift inward, and smaller images are padded. RSAR images are resized to fit within 800×800 pixels and then padded to 800×800. FAIR1M patches are resized to fit within 1024×1024 pixels. Random rotation is disabled on RSAR.

Computational cost. Parameters include frozen weights. FLOPs count each multiply–add once and cover convolutions, fully connected layers, and matrix products. Detector FLOPs on DOTA-v1.0 use a 1024×1024 input and 2,000 RoIs. The counts exclude RoI sampling, box decoding,

Table 7: Training settings for each dataset. Learning rates are specified for a total batch size of 8 and scaled linearly with the total batch size.

Setting	DOTA-v1.0	HRSC2016	DIOR-R	RSAR	FAIR1M
Optimizer	AdamW	AdamW	AdamW	AdamW	AdamW
AdamW (β_1, β_2)	$(0.9, 0.999)$	$(0.9, 0.999)$	$(0.9, 0.999)$	$(0.9, 0.999)$	$(0.9, 0.999)$
Learning rate	10^{-4}	4×10^{-4}	10^{-4}	10^{-4}	10^{-4}
Batch size per GPU	2	2	2	2	2
Weight decay	0.05	0.05	0.05	0.05	0.05
Training epochs	12	36	12	12	12
Decay epochs	8, 11	24, 33	8, 11	8, 11	8, 11
Decay factor	0.1	0.1	0.1	0.1	0.1
Warm-up schedule	Linear	Linear	Linear	Linear	Linear
Warm-up iterations	500	500	500	500	500
Warm-up start factor	$1/3$	$1/3$	$1/3$	$1/3$	$1/3$
Gradient clipping (ℓ_2 norm)	35	35	35	35	35
Image resize bound	1024×1024	800×800	800×800	800×800	1024×1024
Flip probability (total)	0.75	0.75	0.75	0.75	0.75
Rotation probability	0.5	0.5	0.5	0	0.5

non-maximum suppression (NMS), elementwise operations, and the Fourier transform and bilinear sampling of the rotation-aligned head. We measure the head latency in Table 6 on one NVIDIA RTX 6000D GPU with FP32 inference. All heads receive the same features extracted by the baseline from one DOTA-v1.0 test image and replicated to 512 RoIs. Each run averages latency over 100 forward passes after 30 warm-up passes. We report the median latency over five runs.

Task-specific branches. The task-specific head uses a shared first fully connected layer followed by four branches: one for class and size, one for each center offset, and one for angle. Each branch processes all four frames and aggregates them with the signs associated with its target. As in SRR, the four matrices are parameterized by a shared matrix and three learned increments. We compare branch widths of 1,024 and 256, keeping the first hidden layer at 1,024 units in both cases. The 1,024-dimensional branches use the same initialization as SFL.

Training runs and result reporting. Detection and localization metrics are averaged equally over three training runs with seeds 0–2, with models paired by seed in the shared-proposal analysis. Differences between methods are calculated before rounding. Sample standard deviations use denominator $n - 1$. In Table 15, N gives the minimum and maximum numbers of instances matched by the retained proposals across the three training runs.

C Additional Experimental Results

C.1 Comparison with Published Methods

Table 8 compares SRR with published detectors on DOTA-v1.0 and reports AP_{50} for each category. All entries use single-scale training and testing. With PKINet-V2-S trained for 36 epochs, SRR achieves 80.99 AP_{50}, compared with 80.49 for the baseline.

Table 9 compares SRR with published detectors on DIOR-R. With PKINet-V2-S trained for 36 epochs, SRR raises AP_{50} from 71.15 to 71.65.

C.2 Results with PKINet-V2-S

Table 10 compares the baseline and SRR with PKINet-V2-S trained for 36 epochs on three datasets. SRR improves all three metrics on each dataset. On DOTA-v1.0 and DIOR-R, AP_{75} rises by 1.97 and 1.05 points, respectively. On HRSC2016 under VOC2012 evaluation, AP_{50} is already above 98, while $AP_{50:95}$ improves from 74.32 to 75.84. The localization gains therefore also hold with this stronger backbone and training schedule.

Table 8: **Full AP_{50} comparison on DOTA-v1.0.** Results for previous methods are taken from FAA (Gu et al., 2026). The bottom block reports our baseline and SRR results. Our models use 12 epochs, or 36 for entries marked †. R. F-RCNN and RoI Trans. denote Rotated Faster R-CNN and RoI Transformer.

Method	Backbone	AP_{50}	PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SBF	RA	HA	SP	HC
BGHR (Fu & Zhu, 2025)	—	71.44	88.21	75.65	42.83	60.17	79.39	74.99	85.76	90.86	83.94	85.34	56.77	65.90	60.18	72.45	49.11
R. F-RCNN (Ren et al., 2015)	ResNet-50	73.17	89.40	81.81	47.28	67.44	73.96	73.12	85.03	**90.90**	85.15	84.90	56.60	64.77	64.70	70.28	62.22
R^3Det (Yang et al., 2021a)	ResNet-50	73.7	89.5	81.2	50.5	66.1	70.9	78.7	78.2	90.8	85.3	84.2	61.8	63.8	68.2	69.8	67.2
RoI Trans. (Ding et al., 2019)	ResNet-50	74.05	89.01	77.48	51.64	72.07	74.43	77.55	87.76	90.81	79.71	85.27	58.36	64.11	76.50	71.99	54.06
S^2A-Net (Han et al., 2021a)	ResNet-50	74.12	89.11	82.84	48.37	71.11	78.11	78.39	87.25	90.83	84.90	85.64	60.36	62.60	65.26	69.13	57.94
MutDet (Huang et al., 2024)	ResNet-50	74.5	87.3	78.7	51.3	68.5	78.9	81.6	88.1	90.7	79.9	83.7	58.0	61.8	76.5	72.1	60.8
O-RCNN (Xie et al., 2021)	ResNet-50	75.87	89.46	82.12	54.78	70.86	78.93	83.00	88.20	**90.90**	87.50	84.68	63.97	67.69	74.94	68.84	52.28
ReDiffDet (Zhao et al., 2025)	ResNet-50	76.18	85.45	77.84	50.39	71.63	80.90	84.85	88.77	90.88	87.00	86.08	63.30	63.43	76.01	76.03	60.16
ReDet (Han et al., 2021b)	ReResNet	76.25	88.79	82.64	53.97	74.00	78.13	84.06	88.04	90.89	87.78	85.75	61.76	60.39	75.96	68.07	63.59
R^3Det-GWD (Yang et al., 2021b)	ResNet-50	76.34	88.82	82.94	55.63	72.75	78.52	83.10	87.46	90.21	86.36	85.44	64.70	61.41	73.46	76.94	57.38
FAA (Gu et al., 2026)	ResNet-50	76.55	89.40	84.15	54.66	72.23	78.42	82.89	87.86	90.89	86.26	85.24	63.95	65.58	75.30	70.44	61.06
ReDiffDet (Zhao et al., 2025)	Swin-T	76.62	82.89	82.90	52.47	70.80	80.48	85.41	**88.97**	**90.90**	86.97	87.35	60.24	65.01	76.61	74.28	64.07
ARC (Pu et al., 2023)	R50-ARC	77.35	89.40	82.48	55.33	73.88	79.37	84.05	88.06	**90.90**	86.44	84.83	63.63	**70.32**	74.29	71.91	65.43
R^3Det-KLD (Yang et al., 2021c)	ResNet-50	77.36	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
LSKNet (Li et al., 2023)	LSKNet-S	77.49	89.66	85.52	57.72	75.70	74.95	78.69	88.24	90.88	86.79	86.38	66.92	63.77	77.77	74.47	64.82
GRA (Wang et al., 2024)	R50-GRA	77.63	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Strip R-CNN (Yuan et al., 2026)	StripNet-S	78.09	89.56	83.86	53.65	76.27	79.31	84.93	88.24	90.89	87.69	85.88	66.15	68.11	75.73	73.32	67.79
PKINet (Cai et al., 2024)	PKINet-S	78.39	**89.72**	84.20	55.81	77.63	80.25	84.45	88.12	90.88	87.57	86.07	66.86	70.23	77.47	73.62	62.94
O-RCNN	ResNet-50	76.96	89.43	83.65	54.98	72.28	78.58	84.00	88.36	90.88	86.73	85.78	66.50	62.96	75.70	70.25	64.32
O-RCNN+**SRR**	ResNet-50	77.46	89.41	84.11	54.42	76.42	79.08	84.45	88.38	90.88	86.49	85.77	65.31	63.91	76.45	71.15	65.73
O-RCNN†	PKINet-V2-S	80.49	89.69	85.76	58.91	78.43	80.81	**86.38**	88.54	90.87	87.92	87.50	68.94	68.89	78.60	81.02	75.10
O-RCNN+**SRR**†	PKINet-V2-S	**80.99**	89.64	**86.50**	**60.53**	**79.15**	**81.36**	86.26	88.59	90.86	**88.53**	**87.62**	**70.88**	68.80	**78.85**	**81.25**	**76.10**

Table 9: **Comparisons on DIOR-R.** Results for previous methods are taken from PKINet-v2 (Cai et al., 2026). The bottom block reports our baseline and SRR results. Our models use 12 epochs, or 36 for entries marked †. R. F-RCNN and RoI Trans. denote Rotated Faster R-CNN and RoI Transformer.

Method	Backbone	AP_{50}
RetinaNet-O (Lin et al., 2017)	ResNet-50	57.55
R. F-RCNN (Ren et al., 2015)	ResNet-50	59.54
TIOE-Det (Ming et al., 2023)	ResNet-50	61.98
RoI Trans. (Ding et al., 2019)	ResNet-50	63.87
O-RCNN (Xie et al., 2021)	ResNet-50	64.30
GGHL (Huang et al., 2022)	DarkNet-53	66.48
ARS-DETR (Zeng et al., 2024)	ResNet-50	66.12
DCFL (Xu et al., 2023)	ResNet-50	66.80
Oriented RepPoints (Li et al., 2022)	ResNet-50	66.71
LSKNet-S (Li et al., 2023)	LSKNet-S	65.90
Strip R-CNN-S (Yuan et al., 2026)	StripNet-S	68.70
LWGANet-L2 (Lu et al., 2026)	LWGANet-L2	68.53
PKINet-S (Cai et al., 2024)	PKINet-S	67.03
PKINet-V2-S (Cai et al., 2026)	PKINet-V2-S	69.40
O-RCNN	ResNet-50	65.07
O-RCNN+**SRR**	ResNet-50	65.97
O-RCNN†	PKINet-V2-S	71.15
O-RCNN+**SRR**†	PKINet-V2-S	**71.65**

C.3 Additional Results on HRSC2016

Table 11 gives the full HRSC2016 results across three backbones under both VOC interpolation rules. SRR improves AP_{75} and $AP_{50:95}$ for each backbone under both rules. The ResNet-50 results also include the component ablation under VOC2007 interpolation, complementing the VOC2012 results in Table 4 with the same predictions.

Table 10: **Baseline and SRR with PKINet-V2-S trained for 36 epochs.** HRSC2016 results use VOC2007 and VOC2012 interpolation of the same predictions.

Dataset	Head	AP_{50}	AP_{75}	$AP_{50:95}$
DOTA-v1.0	Baseline	80.49 ± 0.08	57.65 ± 0.34	52.03 ± 0.30
	SRR	$\mathbf{80.99} \pm 0.11$	$\mathbf{59.61} \pm 0.42$	$\mathbf{52.85} \pm 0.39$
DIOR-R	Baseline	71.15 ± 0.29	50.41 ± 0.24	47.56 ± 0.07
	SRR	$\mathbf{71.65} \pm 0.46$	$\mathbf{51.45} \pm 0.19$	$\mathbf{48.56} \pm 0.03$
HRSC2016 VOC2007	Baseline	90.54 ± 0.02	89.48 ± 0.12	71.02 ± 0.24
	SRR	$\mathbf{90.59} \pm 0.03$	$\mathbf{89.74} \pm 0.09$	$\mathbf{71.88} \pm 0.04$
HRSC2016 VOC2012	Baseline	98.40 ± 0.02	91.34 ± 0.60	74.32 ± 0.29
	SRR	$\mathbf{98.48} \pm 0.13$	$\mathbf{92.11} \pm 0.41$	$\mathbf{75.84} \pm 0.32$

Table 11: **HRSC2016 results with three backbones and component ablation.** Baseline and SRR are evaluated under VOC2007 and VOC2012 interpolation using the same predictions. The VOC2007 results also include GCA with ResNet-50.

Evaluation	Backbone	Head	AP_{50}	AP_{75}	$AP_{50:95}$
VOC2007	ResNet-50	Baseline	90.19 ± 0.04	85.65 ± 4.51	68.95 ± 0.20
	ResNet-50	GCA	90.22 ± 0.11	88.44 ± 0.34	70.00 ± 0.61
	ResNet-50	**SRR**	$\mathbf{90.34} \pm 0.01$	$\mathbf{89.05} \pm 0.16$	$\mathbf{70.32} \pm 0.64$
	LSKNet-S	Baseline	90.59 ± 0.03	89.37 ± 0.22	71.05 ± 0.16
	LSKNet-S	**SRR**	$\mathbf{90.60} \pm 0.02$	$\mathbf{89.59} \pm 0.32$	$\mathbf{72.33} \pm 0.46$
	PKINet-V2-S	Baseline	90.54 ± 0.02	89.48 ± 0.12	71.02 ± 0.24
	PKINet-V2-S	**SRR**	$\mathbf{90.59} \pm 0.03$	$\mathbf{89.74} \pm 0.09$	$\mathbf{71.88} \pm 0.04$
VOC2012	ResNet-50	Baseline	$\mathbf{97.41} \pm 0.29$	88.48 ± 1.10	71.35 ± 0.56
	ResNet-50	**SRR**	97.35 ± 0.17	$\mathbf{90.31} \pm 0.05$	$\mathbf{73.16} \pm 0.30$
	LSKNet-S	Baseline	98.49 ± 0.20	91.03 ± 0.55	74.02 ± 0.74
	LSKNet-S	**SRR**	$\mathbf{98.55} \pm 0.19$	$\mathbf{91.80} \pm 0.76$	$\mathbf{75.79} \pm 0.18$
	PKINet-V2-S	Baseline	98.40 ± 0.02	91.34 ± 0.60	74.32 ± 0.29
	PKINet-V2-S	**SRR**	$\mathbf{98.48} \pm 0.13$	$\mathbf{92.11} \pm 0.41$	$\mathbf{75.84} \pm 0.32$

C.4 Baseline Reproduction and Training Settings

Table 12 compares the training settings behind our Oriented R-CNN baseline and the result reported by Xie et al. (2021) on DOTA-v1.0. The reported model uses SGD on one GPU with a batch size of 2 and no random rotation, achieving 75.87 AP_{50}. Our reproduction uses four GPUs with a total batch size of 8, a learning rate of 0.02 scaled linearly with the batch size, momentum 0.9, and weight decay 10^{-4}. With SGD and no random rotation, it reaches 75.64 AP_{50}, close to the reported result. The AdamW configurations use the default DOTA-v1.0 hyperparameters in Table 7.

Table 12: **Baseline reproduction and training settings on DOTA-v1.0.** The first row is the published result. The remaining rows use our implementation. Batch size is the total across GPUs, with two images per GPU. RR denotes random rotation.

Model	Batch size	Optimizer	RR	AP_{50}	AP_{75}	$AP_{50:95}$
Baseline (reported)	2	SGD	×	75.87	—	—
Baseline (ours)	8	SGD	×	75.64 ± 0.07	49.14 ± 0.79	46.14 ± 0.33
Baseline (ours)	8	AdamW	×	76.13 ± 0.41	50.51 ± 0.71	46.87 ± 0.48
Baseline (ours)	8	AdamW	✓	76.96 ± 0.25	51.50 ± 0.32	47.83 ± 0.28
SRR	8	AdamW	✓	$\mathbf{77.46} \pm 0.22$	$\mathbf{53.27} \pm 0.20$	$\mathbf{49.18} \pm 0.26$

Switching to the AdamW configuration raises the baseline to 76.13 AP_{50}. Adding random rotation with probability 0.5 further improves it to 76.96, giving the default baseline used in this paper. Under the same training settings, SRR reaches 77.46 AP_{50} and improves AP_{75} and $AP_{50:95}$ by 1.76 and 1.35 points, respectively. SRR thus provides additional gains over a baseline already strengthened by the optimizer and augmentation settings.

C.5 Localization Analysis

C.5.1 AP at Each IoU Threshold

Table 13 reports AP at each IoU threshold. On HRSC2016, the largest gain under VOC2012 evaluation occurs at IoU 0.85, where AP rises from 54.73 to 60.69. On DIOR-R, SRR improves on the baseline at all ten thresholds, with the largest gains at IoU 0.80 and 0.85. SFL further improves AP_{75} and $AP_{50:95}$ over GCA on both datasets. At IoU 0.95, AP drops to about 5 points or less for every head, and GCA and SRR stay within one point of each other.

Table 13: **Ablation study of GCA and SFL at different IoU thresholds on HRSC2016 and DIOR-R.** Entries report AP. HRSC2016 results use both VOC2007 and VOC2012 interpolation.

	HRSC2016 VOC2007			HRSC2016 VOC2012			DIOR-R		
IoU	Baseline	GCA	**SRR**	Baseline	GCA	**SRR**	Baseline	GCA	**SRR**
0.50	90.19	90.22	**90.34**	**97.41**	97.29	97.35	65.07	65.96	**65.97**
0.55	90.13	90.18	**90.27**	96.67	96.85	**96.91**	62.03	62.53	**62.84**
0.60	90.08	90.11	**90.20**	95.90	96.09	**96.11**	58.45	58.87	**59.26**
0.65	89.83	89.92	**90.06**	94.51	94.77	**94.78**	54.52	54.97	**55.09**
0.70	89.54	89.59	**89.81**	92.59	92.92	**93.21**	49.11	49.90	**50.10**
0.75	85.65	88.44	**89.05**	88.48	89.27	**90.31**	42.95	43.53	**43.90**
0.80	77.55	78.29	**79.16**	78.24	79.87	**81.90**	35.43	36.53	**36.65**
0.85	54.14	56.15	**57.13**	54.73	58.46	**60.69**	26.15	27.10	**27.44**
0.90	18.77	22.07	**22.85**	14.79	18.63	**20.05**	14.44	**15.47**	15.27
0.95	3.58	**5.05**	4.33	0.18	**0.27**	0.25	3.55	**4.48**	4.41
$AP_{50:95}$	68.95	70.00	**70.32**	71.35	72.44	**73.16**	41.17	41.93	**42.09**

C.5.2 Localization on Shared Proposals

We compare how the baseline, GCA, and SRR refine the same set of proposals on HRSC2016 and DIOR-R. We use the final checkpoints at epoch 36 on HRSC2016 and epoch 12 on DIOR-R. All three models refine proposals from the corresponding baseline RPN. Each proposal is matched to the ground-truth box with which it has the highest rotated IoU. We discard proposals with at least half of their area inside ignored regions and keep those with initial IoU of at least 0.5. The kept proposals cover an average of 99.76% of the test instances outside ignored regions on HRSC2016 and 78.77% on DIOR-R. Each model extracts RoI features with its own trained backbone, and we evaluate the refined boxes before NMS.

We measure box overlap using output IoU and the fraction of proposals reaching IoU 0.90, denoted by P_{90}. Center, size, and orientation errors describe the remaining geometric differences. Before computing these errors, we order the sides of predicted and ground-truth boxes by length and adjust their angles accordingly. Center error is the L2 norm of the center displacement projected onto the ground-truth axes and normalized by the corresponding side lengths. Size error is the mean absolute log ratio of predicted to ground-truth side lengths. Orientation error is the smallest difference between long-axis angles modulo π, and it is evaluated only for ground-truth aspect ratios of at least 1.2, where the long axis is well defined. All metrics are averaged over proposals within each instance, then across instances.

As shown in Table 14, GCA reduces center, size, and orientation errors on both datasets, and SFL reduces them further while increasing output IoU. SRR also increases the fraction of proposals reaching IoU 0.90. On HRSC2016, P_{90} rises from 27.08% for the baseline to 31.79% for GCA and 32.59% for SRR. The corresponding values on DIOR-R are 17.65%, 18.79%, and 19.34%.

Table 15 reports mean IoU after box refinement for different ground-truth aspect ratios. Only instances matched by the retained proposals are included. SRR improves over the baseline in every nonempty group. GCA gives the highest IoU in the HRSC2016 group with aspect ratios from 1.2 to 2, which contains only two instances. Most HRSC2016 instances included in this analysis have aspect ratios of at least 4. Their mean IoU rises from 0.8558 to 0.8650 with SRR. DIOR-R spans a broader range of shapes. For nearly square objects with aspect ratios below 1.2, SRR raises mean IoU from 0.7921 to 0.7976.

Table 14: **Box refinement on shared proposals.** Center is the center error normalized by ground-truth side lengths. Size is the mean absolute log ratio of predicted to ground-truth side lengths. Orientation is the long-axis angle error for aspect ratios of at least 1.2.

Dataset	Head	IoU ↑	Center ↓	Size ↓	Orientation (°) ↓
HRSC2016	Baseline	0.8504 ± 0.0006	0.0439 ± 0.0001	0.0587 ± 0.0002	1.102 ± 0.017
	GCA	0.8578 ± 0.0002	0.0416 ± 0.0001	0.0564 ± 0.0003	1.016 ± 0.008
	SRR	$\mathbf{0.8597} \pm 0.0013$	$\mathbf{0.0410} \pm 0.0002$	$\mathbf{0.0557} \pm 0.0008$	$\mathbf{1.002} \pm 0.024$
DIOR-R	Baseline	0.7966 ± 0.0006	0.0620 ± 0.0001	0.0869 ± 0.0002	3.981 ± 0.055
	GCA	0.8004 ± 0.0005	0.0597 ± 0.0002	0.0854 ± 0.0000	3.939 ± 0.044
	SRR	$\mathbf{0.8026} \pm 0.0007$	$\mathbf{0.0589} \pm 0.0002$	$\mathbf{0.0845} \pm 0.0004$	$\mathbf{3.865} \pm 0.038$

Table 15: Mean IoU after box refinement across ground-truth aspect ratios on HRSC2016 and DIOR-R, using shared proposals with initial IoU ≥ 0.5. Only instances matched by the retained proposals are included. Dashes indicate groups with no instances.

	HRSC2016				DIOR-R			
Aspect ratio	N	Baseline	GCA	SRR	N	Baseline	GCA	SRR
$[1, 1.2)$	0	–	–	–	21,179–21,309	0.7921	0.7957	0.7976
$[1.2, 2)$	2	0.7362	0.7501	0.7456	17,845–17,942	0.7662	0.7710	0.7738
$[2, 4)$	143–144	0.8114	0.8199	0.8215	50,976–50,992	0.8194	0.8230	0.8250
$[4, \infty)$	1,076–1,082	0.8558	0.8631	0.8650	7,895–7,925	0.7300	0.7345	0.7368

C.6 Qualitative Results

Figure 4 compares the baseline and SRR heads with PKINet-V2-S on DOTA-v1.0. At the display threshold of 0.3, SRR detects objects missing from the baseline results, including ships, vehicles, swimming pools, and storage tanks. Figures 5 and 6 show additional detection results of SRR on DOTA-v1.0 and RSAR, respectively.

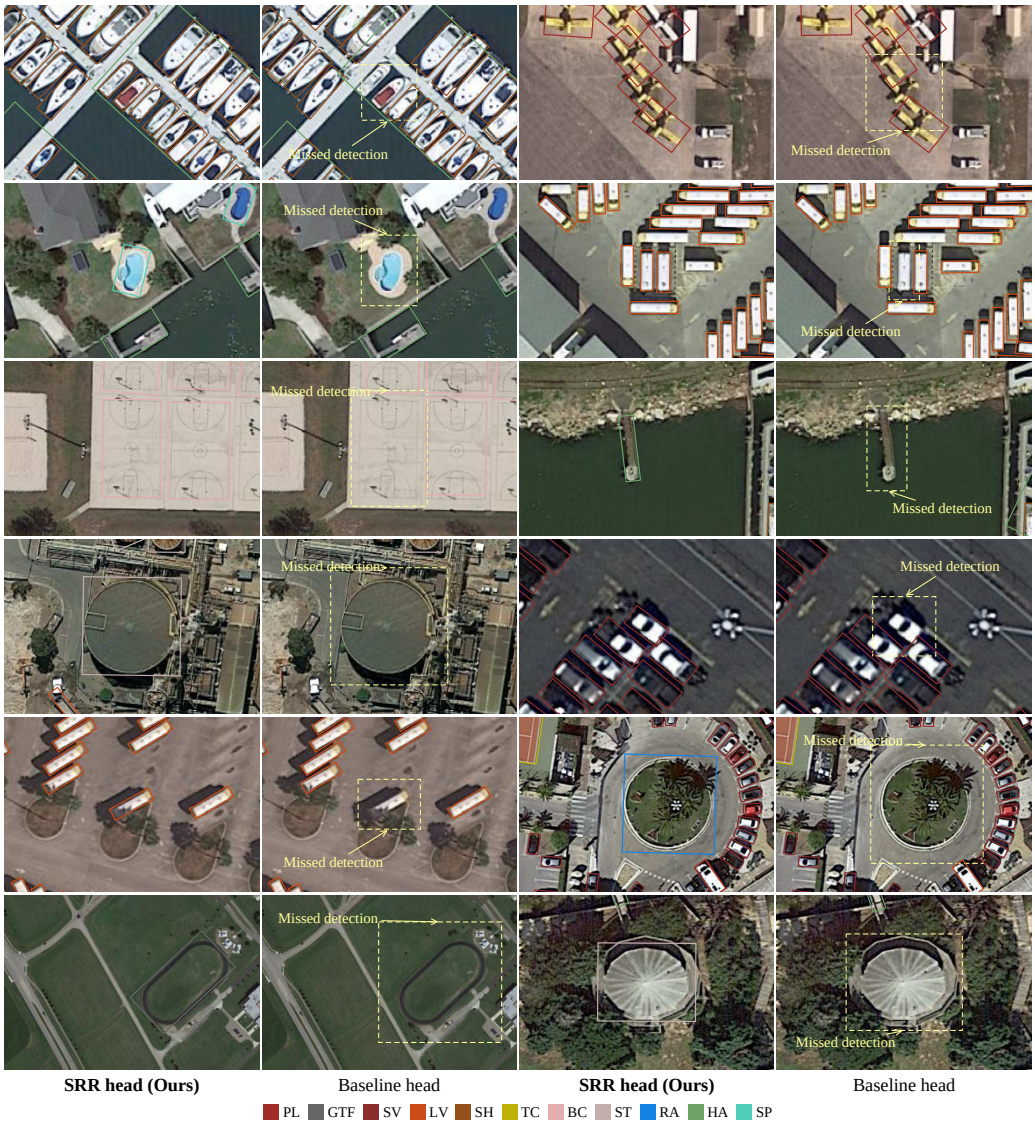


Figure 4: Qualitative comparison of the baseline and SRR heads on DOTA-v1.0 with a PKINet-V2-S backbone. Each row shows two scenes, with SRR predictions on the left and baseline predictions on the right for each scene. Detections are shown at a score threshold of 0.3, with colors indicating predicted categories. Yellow dashed boxes highlight objects detected by SRR but missing from the displayed baseline results.

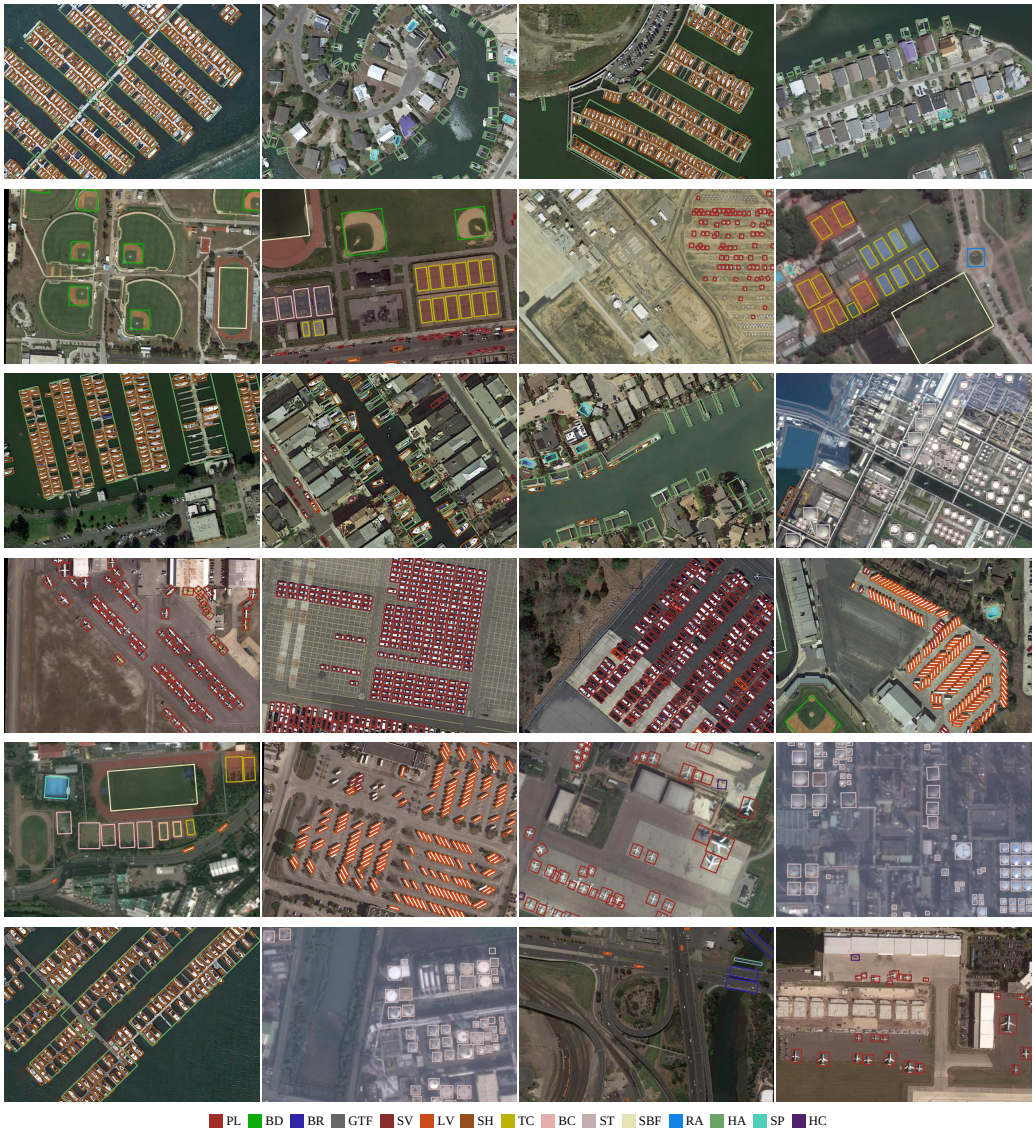


Figure 5: **Detection results of SRR on the DOTA-v1.0 test set.** Detections are shown at a score threshold of 0.3, with colors indicating predicted categories. Examples include harbors, airports, sports fields, storage tank sites, and parking lots.

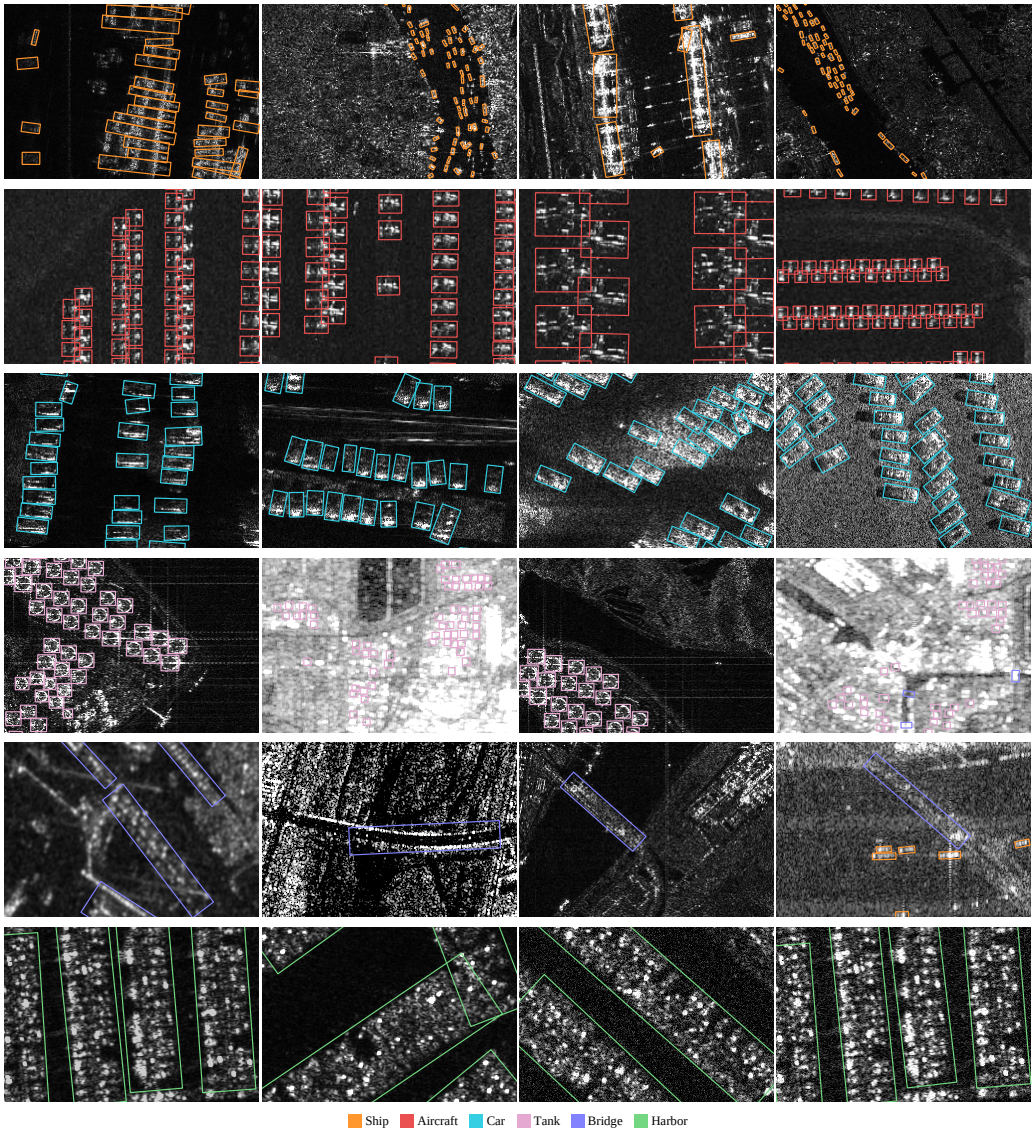


Figure 6: **Qualitative results of SRR on RSAR.** From top to bottom, the six rows present examples of ships, aircraft, cars, tanks, bridges, and harbors, with four examples per row. Oriented boxes show SRR predictions, with colors indicating the categories listed in the legend.