remote sensing

MDPI

Article

Class-Conditioned Evidence Recovery in Frozen Vision Transformers for Remote Sensing Object Detection

Haoxuan Xiang, Xiaodong Zhao, Sheng Yang and Peng Lin *

School of Automation, Hangzhou Dianzi University, Hangzhou 310018, China;
24030119@hdu.edu.cn (H.X.); xdzhao@hdu.edu.cn (X.Z.); 24150639@hdu.edu.cn (S.Y.)
* Correspondence: penglin@hdu.edu.cn

Highlights

What are the main findings?

- An image-level class prior guides the residual updates in a frozen vision transformer. The adapted features condition the denoiser that generates box prompts.
- On VisDrone, RS-Adapter achieved 39.2% AP and 28.4% AP_s, with a 3.1-point AP_s gain over DEIM-D-FINE-L. Its AP50 was 78.3% on DIOR, 84.5% on NWPU VHR-10, and 87.7% on RSOD.

What are the implications of the main findings?

- Freezing the backbone reduced the number of trainable parameters to 16.8 million, 83.7% fewer than full tuning.
- Generating box prompts from the image allows the prompt-based decoder to run automatically at inference.

Abstract

Aerial object detectors must resolve small, closely spaced objects while retaining enough context to recognize large structures. RS-Adapter uses an image-level class prior to update intermediate features in a frozen vision transformer. The prior combines a learned classifier, fixed class prototypes, and class-conditional descriptor statistics. Cross-attention links the prior to local features, which are fused across scales and gated before being added to the transformer. The adapted features guide a denoiser that generates box prompts for the detection decoder. On VisDrone, RS-Adapter achieved 39.2% average precision (AP) and 28.4% small-object AP, 3.1 percentage points above DEIM-D-FINE-L on the latter metric. AP50 was 78.3% on DIOR, 84.5% on NWPU VHR-10, and 87.7% on RSOD. RS-Adapter trained 16.8 million of its 103.2 million parameters and used less peak training memory than full tuning in the VisDrone comparison.

Keywords: remote sensing object detection; frozen vision transformer; class-conditioned adaptation; small objects; cross-attention; denoised box prompts; parameter-efficient learning

Check for updates

Academic Editors: Guangliang Cheng, Shuchang Lyu and Hossein M. Rizeei

Received: 11 October 2026
Revised: 7 November 2026
Accepted: 13 November 2026
Published: 16 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Vehicles in an aerial image may occupy only a few pixels each, while a harbor or stadium can span much of the frame. Detecting adjacent vehicles requires enough detail to distinguish their boundaries. Recognizing a large facility also requires context. VisDrone, DIOR, NWPU VHR-10, and RSOD cover settings ranging from dense road traffic to overhead views of large facilities, with different amounts of training data [1–4].

Pretrained vision transformers offer transferable image representations [5,6], but features that identify a scene may provide too little spatial detail to separate individual objects. Feature pyramids combine resolutions [7], and PKINet uses kernels of different sizes to cover a broader range of object extents [8]. Spectral cues, selective attention, and position uncertainty have also been studied for small targets [9–11]. With the backbone frozen, these spatial cues must enter through the trainable parts of the detector.

Promptable segmentation pairs pretrained encoders with prompts [12–14]. SkySense, RemoteCLIP, DOFA, and RobSense extend pretrained representations to Earth observation [15–18], and parameter-efficient tuning limits the number of weights updated during training [19–21]. Spatial adapters and image-driven prompts provide ways to connect pretrained representations with dense prediction [22,23]. For remote sensing detection, this adaptation must retain the spatial cues that distinguish individual objects. Image-level class information can guide feature updates, but its influence should depend on the local evidence in the image. We use local features to supply spatial detail and an image-level class prior to guide how the adapter incorporates those features.

RS-Adapter estimates a class prior from the current image and injects class-conditioned residuals into a frozen transformer (Figure 1). The Subspace Cross-Attention Block (SCAB) associates the prior with local features. The Hierarchical Attention Fusion Block (HAFB) propagates the resulting features across scales, and the Gaussian Correlation Interaction Block (GCIB) gates their interaction with the backbone. The adapted features also condition a denoiser whose refined boxes serve as prompts for a separate detection decoder.

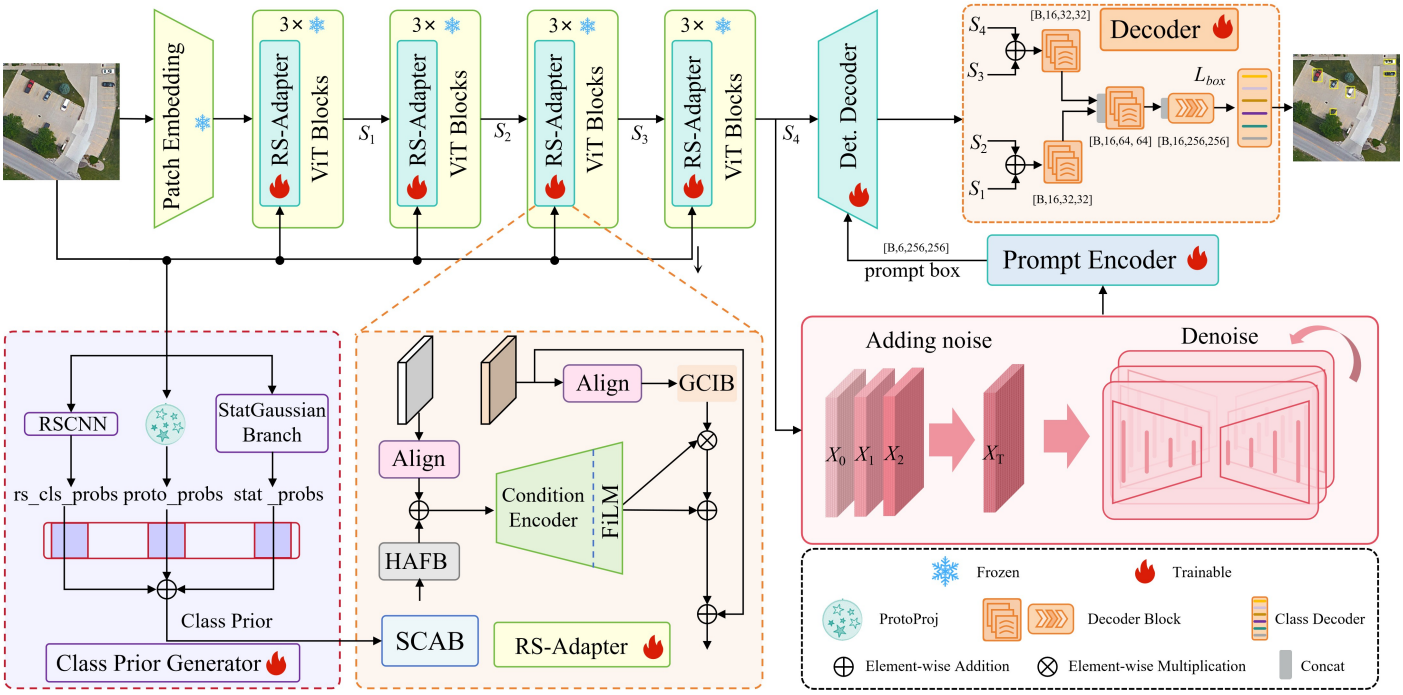


Figure 1. Overall RS-Adapter architecture. A frozen patch embedding and four groups of ViT blocks are augmented with trainable adapters. The three-branch class prior conditions feature recovery, and automatic box refinement supplies prompts to the detection decoder. Black arrows trace the flow of images, features, and prompts. Thick pink arrows show the forward-noise and reverse-denoising path, and orange dashed outlines identify the enlarged adapter and decoder schematics. Pale yellow groups contain the frozen ViT blocks, cyan blocks denote the adapters and prompt or detection modules, and orange blocks denote decoder operations. Snowflakes and flames indicate frozen and trainable components, respectively.

The design combines established operations in a single adaptation pathway. The image-level prior conditions the residuals inside the frozen encoder, and the resulting

features guide box-prompt generation. Image-level labels supervise the prior, and bounded area weights give smaller boxes more influence on the localization loss. The consistency loss encourages geometric and class predictions for the same hypotheses to agree across nearby noise levels.

The main contributions are as follows.

1. We introduce a class-conditioned residual adapter for frozen vision transformers. A three-branch prior derived from the image guides local feature recovery, multiscale fusion, and interaction with the frozen backbone.
2. We connect the adapted features to a denoiser that generates box prompts for automatic detection. The training objective combines scale-normalized localization, class-prior supervision, and consistency across noise levels.
3. We evaluate RS-Adapter on VisDrone, DIOR, NWPU VHR-10, and RSOD against 26 baseline configurations, with component, loss, capacity, and prompt controls, to examine the design. On VisDrone, the model used 16.8 million trainable parameters and achieved 39.2% AP and 28.4% small-object AP (AP_s). These scores exceeded DEIM-D-FINE-L by 2.2 and 3.1 percentage points, respectively.

2. Related Work

2.1. Detection Under Spatial and Semantic Scale Variation

Faster R-CNN and Cascade R-CNN refine proposals before producing detections [24,25]. Dense detectors address sample assignment and localization through focal loss, anchor-free regression, adaptive sample selection, and task alignment [26–29]. These choices matter in aerial scenes, where many small targets can compete for predictions against repetitive backgrounds.

The real-time baselines include YOLOX, YOLOv9, and YOLOv10, which use anchor-free prediction, programmable gradient information, and dual assignments, respectively [30–32]. We also include YOLO11 and YOLO26 from Ultralytics (Ultralytics Inc., Frederick, MD, USA) [33,34], as well as YOLO12 [35]. The comparisons identify each model scale and backbone and report both trainable and total parameters.

PKINet uses multiple kernel sizes to cover objects of different extents [8]. Other work uses spectral information, position Gaussian estimation, or selective attention for small targets [9–11]. Liu et al. [36] detect rip currents by dividing large satellite scenes into patches and merging detections in the original image coordinates. RS-Adapter processes full images and uses class scores to condition intermediate features. Zhang et al. [37] use adaptive dual cross-attention to fuse visible and thermal features across scales. Here, cross-attention links local features to a class prior derived from the same image.

2.2. Set Prediction and Generative Box Refinement

DETR casts detection as set prediction with bipartite matching [38]. Deformable DETR changes how queries sample spatial features, DAB-DETR represents queries as anchor boxes, and DINO uses denoising during training [39–41]. RT-DETR, RT-DETRv2, D-FINE, and DEIM further change inference, optimization, box regression, or matching [42–45].

DiffusionDet applies denoising to a set of noisy boxes to obtain detections [46–48]. RS-Adapter uses a related reverse update conditioned on features from the adapted encoder. The refined boxes are rasterized as prompts, and the detection decoder combines those prompts with features from several encoder depths to predict classes and boxes.

2.3. Foundation Models and Parameter-Efficient Adaptation

ViT-Adapter adds spatial inductive biases for dense prediction, while LoRA updates the model through low-rank parameter changes [19,22]. Other studies vary where adapters

are inserted, how encoder blocks interact, and which tokens are retained [21,49–52]. RS-Adapter leaves the encoder weights fixed and conditions the residual added before each block group on the current image.

SkySense and RobSense learn multimodal representations, DOFA handles different observation modalities, and RemoteCLIP and GeoChat connect geospatial images with language [15–18,53]. SkySense-O, SegEarth-OV, YOLO-World, Grounding DINO, and YOLOE broaden the categories available at inference [54–58]. We use a fixed category vocabulary and horizontal boxes for each dataset.

RSPrompter learns image-driven prompts for instance segmentation with SAM [23]. We refer to its box-detection configuration as RSPrompter-det. In RS-Adapter, image-level class scores guide the encoder updates that feed the horizontal-box decoder and the prompt denoiser. AeroGen uses diffusion to generate training images [59]. RS-Adapter uses denoising within the detector to refine box hypotheses.

3. Materials and Methods

3.1. Task Definition and Overall Architecture

Let $I \in \mathbb{R}^{3\times H_0\times W_0}$ be an image with annotations $\mathcal{Y} = \{(b_i, c_i)\}_{i=1}^{M}$. A normalized horizontal box is $b_i = (x_i, y_i, w_i, h_i) \in [0,1]^4$, and its class is $c_i \in \{1,\dots,K\}$. The detector returns $\widehat{\mathcal{Y}} = \{(\hat{b}_j, \hat{p}_j)\}_{j=1}^{Q}$, with a no-object category for unused predictions. The number of annotations M varies by image, while Q sets the prediction capacity. The prior is estimated from I. Each dataset has its own class prototypes and classification head.

RS-Adapter freezes the patch embedding, four transformer groups, and the prototype projection. The residual adapters, the other components of the prior generator, the prompt encoder, and the detection decoder are trainable. The decoder receives the four group outputs $S_1,\dots,S_4$. Learned channel projections and spatial resampling align the side features with the encoder and decoder.

3.2. Frozen Encoder with Residual Evidence Injection

The pretrained patch embedding maps the input to tokens $Z_0 = E_{\theta_0}(I)$. We divide the 12 ViT blocks into four groups of three and freeze them along with the patch embedding. Before group ℓ, the adapter computes the residual $\mathcal{A}_{\phi_\ell}$ from the preceding state, the side feature R_ℓ, and the class prior π.

$$S_\ell = \mathcal{B}_{\theta_\ell}\big(S_{\ell-1} + a_\ell \mathcal{A}_{\phi_\ell}(S_{\ell-1}, R_\ell, \pi)\big), \qquad S_0 = Z_0. \tag{1}$$

We initialize a_ℓ near zero and learn it with the adapter. The block weights θ_ℓ stay fixed, and gradients pass through the blocks to the residual adapters.

The convolutional side encoder produces four maps $R_1,\dots,R_4$ at successively coarser resolutions. Align_ℓ projects their channels and resizes them bilinearly to the ViT token grid. The ViT groups share one token resolution, so separate operators $\widetilde{S}_\ell = \mathrm{Align}_\ell^{\mathrm{dec}}(S_\ell)$ resample their outputs for decoding.

3.3. Three-Branch Class Prior Generation

The class prior combines three scores. Let $u = \mathrm{GAP}(R_4)$ be a pooled side-encoder descriptor and $v = \mathrm{sg}(\mathrm{GAP}(S_4^{\mathrm{base}}))$ the pooled descriptor from the unadapted frozen encoder. The CNN branch predicts class presence as $q^{\mathrm{cnn}} = \sigma(W_u u + d_u)$. The prototype branch compares v with class prototypes m_c, giving

$$q_c^{\mathrm{proto}} = \sigma\left(\frac{\langle v, m_c\rangle}{\tau_p(\|v\|_2\|m_c\|_2 + \epsilon)} + d_c\right). \tag{2}$$

We construct m_c from frozen-encoder descriptors of training crops in class c and keep the prototypes fixed during detector training. The temperature $\tau_p > 0$ and offset d_c calibrate the cosine score before the sigmoid. The branch uses the unadapted descriptor v, which is available before the adapter computes its residual.

The statistical branch models within-class variation in descriptor space. We use a regularized diagonal Gaussian whose parameters are initialized from training crops and refined through the prior objective. Its score is

$$g_c(u) = -\frac{1}{2}\sum_{r=1}^{D}\left[\frac{(u_r-\mu_{cr})^2}{\sigma_{cr}^2+\epsilon}+\log(\sigma_{cr}^2+\epsilon)\right], \qquad q_c^{\text{stat}} = \sigma(g_c(u)/\tau_s + d_c^{\text{s}}). \tag{3}$$

A softplus parameterization with a positive floor keeps the variances positive. The Gaussian score includes the log-variance term to account for descriptor dispersion within a class.

An image-dependent softmax supplies the weights for the three branch outputs.

$$\alpha = \text{softmax}(h_\phi([u;v])), \qquad \bar{q}_c = \sum_{r\in\{\text{cnn,proto,stat}\}} \alpha_r q_c^r, \qquad \pi_c = \frac{\bar{q}_c+\epsilon}{\sum_{k=1}^{K}(\bar{q}_k+\epsilon)}. \tag{4}$$

The mixture scores $\bar{q}_c$ represent class presence. Normalization gives the relative weights π_c, while $\rho = \max_c \bar{q}_c$ controls the strength of class conditioning. This lets the residual use a normalized prior even when the image provides little evidence of any foreground class.

We form $e_\pi = \sum_c \pi_c e_c$ from the learned class embeddings e_c. At side-encoder level ℓ, the conditioned feature is $P_\ell = \text{Conv}_\ell(R_\ell) + \rho\ \text{Broadcast}(W_\ell e_\pi)$. The projected side feature retains spatial variation. The broadcast term adds the same image-level class information at each location. The resulting map supplies the prior-conditioned input to SCAB.

3.4. Subspace Cross-Attention for Local Semantic Association

SCAB takes two aligned inputs $F_1, F_2 \in \mathbb{R}^{N\times C}$. The first contains local features and the second contains prior-conditioned features. Their concatenation supplies a shared query, while each input supplies its own keys and values (Figure 2a). We partition the channels into n subspaces of width $d = C/n$. In subspace m, the query Q_m and the keys and values K_m^r, V_m^r for branches $r \in \{1,2\}$ all lie in $\mathbb{R}^{N\times d}$. The attention acts across channels.

$$\begin{aligned} \overline{Q}_m &= \text{softmax}_N(Q_m/\sqrt{d}), & \overline{K}_m^r &= \text{softmax}_d(K_m^r), \\ A_m^r &= (\overline{K}_m^r)^\top \overline{Q}_m \in \mathbb{R}^{d\times d}, & U_m^r &= V_m^r A_m^r \in \mathbb{R}^{N\times d}. \end{aligned} \tag{5}$$

The softmax subscripts specify the normalized axes. Each column of A_m^r sums to one. The matrix products cost $O(NC^2/n)$ in addition to the projections, and the value matrix retains all N spatial locations.

We concatenate the subspaces within each branch and then concatenate the two branches. A learned scalar η scales the result before element-wise multiplication with the input. The output projection restores the C channels (Figure 2a).

$$F_{\text{SCAB}} = W_o\Big(\eta[U^1;U^2]\odot[F_1;F_2]\Big) \in \mathbb{R}^{N\times C}, \qquad U^r = \text{Concat}_{m=1}^{n} U_m^r. \tag{6}$$

3.5. Hierarchical Attention Fusion Across Spatial Scales

Let $F_{g\ell} \in \mathbb{R}^{C_\ell\times H_\ell\times W_\ell}$ denote the four conditioned levels. With the base spatial size $H\times W$ in Figure 2c, level ℓ has size $(H/2^\ell)\times(W/2^\ell)$ and $2^\ell C$ channels, for $\ell = 1,\ldots,4$. The top-down pass starts at $G_4 = \text{Conv}_4(F_{g4})$. For $\ell = 3,2,1$, we use a channel projection and bilinear upsampling to match $G_{\ell+1}$ to the shape of $F_{g\ell}$, giving U_ℓ.

$$\chi_\ell = \sigma(\text{MLP}_\ell(\text{GAP}(U_\ell)) + \text{MLP}_\ell(\text{GMP}(U_\ell))), \tag{7}$$

We combine average and maximum pooling to compute the channel gate χ_ℓ. The gate weights the upsampled features before fusion with the local map.

$$G_\ell = \mathrm{Conv}^{\mathrm{o}}_\ell\Big(\mathrm{Conv}^{\mathrm{l}}_\ell(F_{g\ell}) \odot \Big[F_{g\ell} + \chi_\ell \odot U_\ell\Big]\Big), \qquad Y_g = \mathrm{Conv}^{\mathrm{g}}(G_1). \tag{8}$$

The projections and resizing match the channel and spatial dimensions before fusion. The local map $F_{g\ell}$ supplies spatial detail, and the attention on U_ℓ selects which coarse features are passed down.

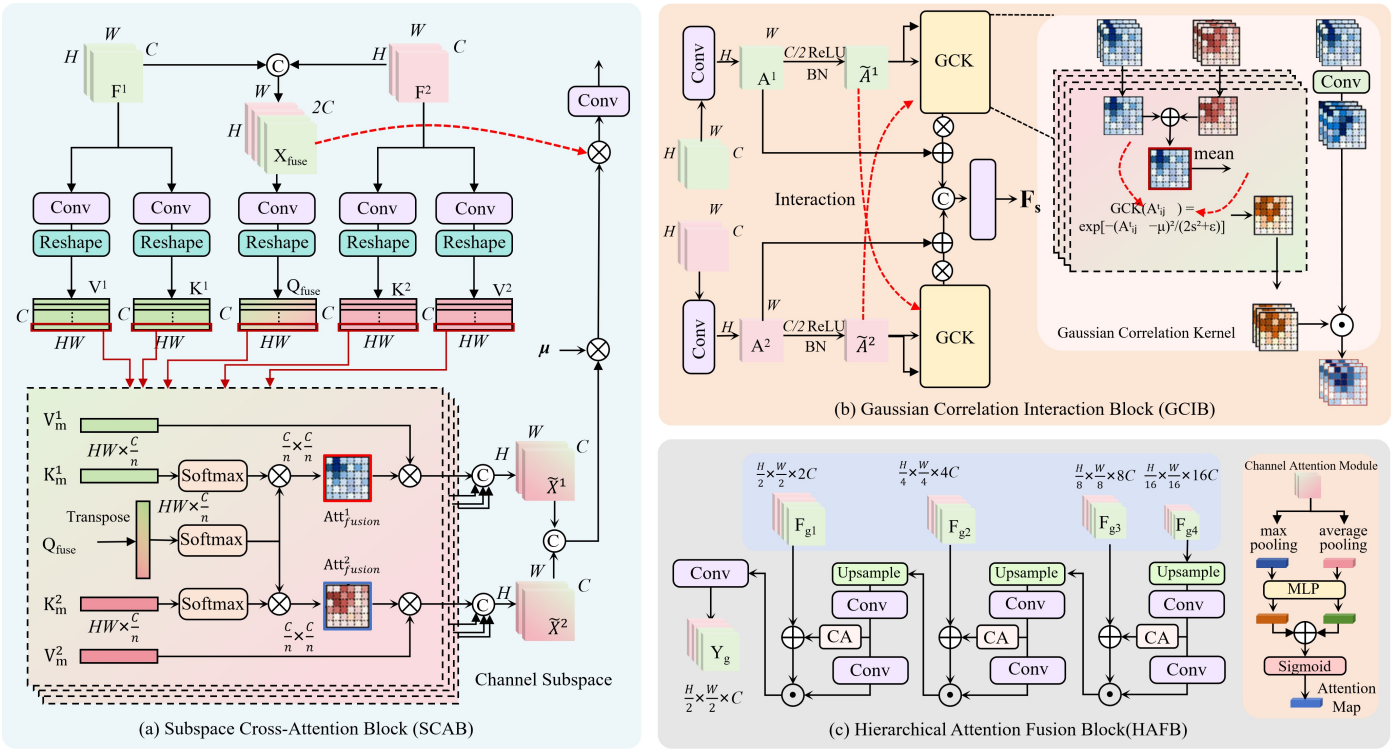


Figure 2. Adapter blocks. (**a**) SCAB combines local and prior-conditioned features within channel subspaces. (**b**) GCIB uses Gaussian responses to gate the two feature branches. (**c**) HAFB fuses features across spatial scales. Green and pink input stacks distinguish the two feature branches in (**a**,**b**). Black arrows indicate feature flow, red dashed arrows show the shared input in (**a**) and cross-branch interactions in (**b**), and stacked dashed outlines show repeated subspaces or channels. In (**c**), CA denotes the channel attention module illustrated on the right.

3.6. Gaussian Correlation Interaction and Feature-Wise Conditioning

GCIB projects the aligned backbone map Z and the recovered context map C_g to half-width, yielding A^1 and A^2. Batch normalization and ReLU give $\widetilde{A}^1$ and $\widetilde{A}^2$ (Figure 2b). For each channel c, we compute a shared spatial mean and variance from the average of the two maps.

$$\begin{aligned} M_c(p) &= \tfrac{1}{2}\big(\widetilde{A}^1_c(p) + \widetilde{A}^2_c(p)\big), \\ \mu_c &= N^{-1}\sum_p M_c(p), \qquad s_c^2 = N^{-1}\sum_p (M_c(p) - \mu_c)^2, \\ \kappa^r_c(p) &= \exp\left[-\frac{(\widetilde{A}^r_c(p) - \mu_c)^2}{2s_c^2 + \epsilon}\right]. \end{aligned} \tag{9}$$

The Gaussian response scores the compatibility of each activation with the joint reference. The variance s_c^2 measures spatial dispersion. A convolution and sigmoid produce a gate $g^r = \sigma(\mathrm{Conv}^r(\kappa^r))$ that modulates the opposite branch.

$$F_s = \mathrm{Conv}^{\mathrm{s}}\Big([A^1 + g^2 \odot \widetilde{A}^1 \,;\, A^2 + g^1 \odot \widetilde{A}^2]\Big). \tag{10}$$

The gated responses are added to the original projections before concatenation.

The condition encoder takes the sum of the aligned backbone feature and HAFB output, $C_\ell = \mathrm{Enc}_\ell(\mathrm{Align}_\ell(S_{\ell-1}) + \mathrm{Align}_\ell(Y_g))$. Feature-wise linear modulation [60] uses C_ℓ to predict a scale centered at one and an additive shift.

$$\gamma_\ell = 1 + \delta \tanh(W_\gamma C_\ell), \qquad \beta_\ell = W_\beta C_\ell, \qquad \mathcal{A}_{\phi_\ell} = W_\ell^{\mathrm{a}}(\gamma_\ell \odot F_{s\ell} + \beta_\ell + \mathrm{Align}_\ell(S_{\ell-1})). \tag{11}$$

For $\delta > 0$, the scale lies between $1 - \delta$ and $1 + \delta$. The adapter adds the projected residual before the next frozen block group, as specified in Equation (1).

3.7. Automatic Denoised Box Prompts and Detection Decoding

During training, the prompt pathway maps annotated boxes $b = (x, y, w, h)$ to unconstrained coordinates $z_0 = \mathrm{logit}(\mathrm{clip}(b, \epsilon, 1 - \epsilon))$. The Q initial hypotheses include annotated boxes sampled with replacement and background slots. A binary mask m_j records whether slot j came from an annotation. The forward process corrupts the base coordinates according to

$$z_t = \sqrt{\bar{\alpha}_t}\, z_0 + \sqrt{1 - \bar{\alpha}_t}\, \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, I_4), \quad \bar{\alpha}_t = \prod_{s=1}^{t}(1 - \beta_s). \tag{12}$$

A foreground hypothesis keeps the clean logit box of the annotation from which it was sampled. It contributes to direct denoising and the geometric part of cross-noise consistency. Background hypotheses have no clean object box target. Classification remains active, and the final decoder assigns the no-object label to unmatched outputs. Repeated sampling in the prompt pathway does not duplicate targets in the final one-to-one matching.

The denoiser $D_\psi(z_t, t, S_4)$ estimates the clean logit coordinates $\widehat{z}_0^{(t)}$. We use the following deterministic reverse update.

$$\widehat{\varepsilon}_t = \frac{z_t - \sqrt{\bar{\alpha}_t}\widehat{z}_0^{(t)}}{\sqrt{1 - \bar{\alpha}_t} + \epsilon}, \qquad z_{t'} = \sqrt{\bar{\alpha}_{t'}}\widehat{z}_0^{(t)} + \sqrt{1 - \bar{\alpha}_{t'}}\widehat{\varepsilon}_t, \tag{13}$$

The inference schedule chooses $t' < t$ in each deterministic reverse update [47]. We draw the initial logit coordinates of each of the Q box hypotheses from $\mathcal{N}(0, I_4)$ and use three updates in the main configuration. The reverse process runs in logit space. Before rasterization, $\hat{b} = \sigma(\hat{z}_0)$ bounds the center and size components. Decoded corners can extend past the image even when these components are bounded. The final prediction filter rejects invalid boxes before ranking.

We rasterize the boxes on the 256×256 prompt grid shown in Figure 1. Its six planes encode the two center coordinates, width, height, objectness, and diffusion time. A weighted average combines overlapping hypotheses. We bound the occupancy denominator away from zero and set empty positions to zero. This six-plane grid is the box-prompt representation in Figure 1.

The decoder fuses the four aligned transformer states in two pairs and adds the encoded prompt feature P_D after spatial and channel alignment.

$$D_{12} = \mathcal{D}_{12}(\widetilde{S}_1 + \widetilde{S}_2), \qquad D_{34} = \mathcal{D}_{34}(\widetilde{S}_3 + \widetilde{S}_4), \qquad F_D = \mathcal{D}_{\mathrm{o}}([D_{12}; D_{34}] + P_D). \tag{14}$$

Figure 1 shows the two depth-pair outputs on 32×32 grids with 16 channels and the subsequent 16-channel decoder maps on 64×64 and 256×256 grids. We resize the two depth-pair outputs to a common resolution before concatenation, and we then align P_D to the spatial size and channel width of the concatenated features before addition. The classification and regression heads use query features sampled at the refined box centers. The classification head returns $K + 1$ scores. The decoder feature width is independent of K, and the prompt-grid resolution is independent of the input canvas.

3.8. Scale-Normalized Objective and Matching

For $M \leq Q$, we match the M annotations to distinct final predictions. The injective assignment σ minimizes a cost based on class probability, normalized box distance, and generalized intersection over union.

$$\sigma^* = \arg\min_{\sigma} \sum_{i=1}^{M} \Big[-\lambda_c \log \hat{p}_{\sigma(i),c_i} + \lambda_1 \|b_i - \hat{b}_{\sigma(i)}\|_1 + \lambda_g (1 - \mathrm{GIoU}(b_i, \hat{b}_{\sigma(i)})) \Big]. \tag{15}$$

Unmatched predictions are assigned the no-object label. GIoU uses decoded boxes with positive widths and heights. Loss ablations retain the same matching cost to isolate the effect of the objective. Supervised auxiliary decoder outputs use independent matching.

Small boxes are unusually sensitive to absolute coordinate error. We therefore introduce a bounded target-area weight

$$\omega_i = \mathrm{clip}\left(\sqrt{a_{\mathrm{ref}} / (w_i h_i + \epsilon)}, 1, \omega_{\max} \right), \qquad \tilde{\omega}_i = \omega_i \Big/ \left(M^{-1} \sum_{k=1}^{M} \omega_k \right). \tag{16}$$

The reference area a_{ref} is expressed in normalized image coordinates, and $\omega_{\max}$ caps the weight assigned to small boxes. For $M > 0$, normalization gives the positive weights unit mean within an image. For $M = 0$, classification remains active and positive localization contributes zero.

We also represent each box by a Gaussian with the same first and second moments as a uniform axis-aligned rectangle. Its mean is (x, y) and its covariance is $\mathrm{diag}(w^2/12, h^2/12)$. The squared Wasserstein distance between two such diagonal distributions simplifies to

$$d_{\mathrm{W}}^2(b, \hat{b}) = (x - \hat{x})^2 + (y - \hat{y})^2 + \frac{(w - \hat{w})^2 + (h - \hat{h})^2}{12}, \qquad \ell_{\mathrm{W}} = 1 - \exp\left(-\frac{d_{\mathrm{W}}^2}{2 s_0^2} \right). \tag{17}$$

The scale s_0 sets how rapidly the Wasserstein term saturates. It uses the same normalized coordinates as the box center and dimensions.

$$\mathcal{L}_{\mathrm{loc}} = \frac{1}{\max(M, 1)} \sum_{i=1}^{M} \tilde{\omega}_i \Big[\lambda_1 \|b_i - \hat{b}_{\sigma^*(i)}\|_1 + \lambda_g (1 - \mathrm{GIoU}(b_i, \hat{b}_{\sigma^*(i)})) + \lambda_{\mathrm{W}} \ell_{\mathrm{W}}(b_i, \hat{b}_{\sigma^*(i)}) \Big]. \tag{18}$$

We apply focal classification to all queries [26] and weight the no-object class separately. The loss ablation holds the assignment cost fixed.

Let $y_c^{\mathrm{pres}} = \mathbf{1}_{n_c > 0}$ be the presence label for class c, and let $r_c = (n_c + \epsilon)/(M + K\epsilon)$ be its smoothed share of the M annotated instances. The first two terms below supervise sigmoid scores, and the last term supervises their normalized mixture:

$$\mathcal{L}_{\mathrm{prior}} = \sum_{h \in \mathcal{H}} \mathrm{BCE}(q^h, y^{\mathrm{pres}}) + \mathrm{BCE}(\bar{q}, y^{\mathrm{pres}}) + \mathbf{1}_{M>0} \, \mathrm{KL}(r \,\|\, \pi). \tag{19}$$

The set $\mathcal{H}$ includes the branches with trainable scores. With fixed prototype calibration, the prototype score contributes to the mixture $\bar{q}$ only. Presence BCE and frequency KL share $\bar{q}$

and can favor different score ratios. For example, with nine objects of one class and one of another, BCE favors high scores for both classes, whereas KL favors a ratio close to nine to one after normalization. Each term in Equation (19) has unit weight before the common multiplier $\lambda_p = 0.2$ is applied in the full objective. For empty images, the presence labels are all zero and the frequency term is disabled.

To connect training at different noise levels, the same base hypotheses are independently corrupted at two nearby steps t and t'. The denoising branch predicts clean geometry and class distributions $p^{(t)}$ and $p^{(t')}$. A stopped-gradient target provides a consistency term,

$$\mathcal{L}_{\text{con}} = \frac{1}{Q}\sum_{j=1}^{Q}\Big[\mathrm{KL}\left(\mathrm{sg}(p_j^{(t')}) \,\|\, p_j^{(t)}\right) + m_j\,\ell_{\text{W}}\big(\hat{b}_j^{(t)}, \mathrm{sg}(\hat{b}_j^{(t')})\big)\Big], \tag{20}$$

Sharing the base hypotheses aligns slots across the two noise levels. The classification consistency term covers all Q slots, while m_j restricts the box term to foreground slots. Direct denoising also uses smooth-L_1 between $\hat{z}_{0,j}^{(t)}$ and $z_{0,j}$ at foreground slots.

The final objective combines these terms,

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{loc}} + \lambda_p\mathcal{L}_{\text{prior}} + \lambda_d\mathcal{L}_{\text{den}} + \lambda_{\text{con}}\mathcal{L}_{\text{con}}. \tag{21}$$

We set $\lambda_1 = 5$, $\lambda_g = 2$, $\lambda_{\text{W}} = 1$, $\lambda_p = 0.2$, $\lambda_d = 1$, and $\lambda_{\text{con}} = 0.1$. The consistency weight is warmed up after initial prior calibration. Algorithm 1 summarizes optimization and image-only inference.

Algorithm 1 Training and inference with RS-Adapter

Require: Images I; training annotations $\mathcal{Y}$; frozen encoder E_θ; query capacity Q
Require: Trainable adapter and prior parameters ϕ; prompt and decoder parameters ψ
Build class prototypes from training crops only; freeze ProtoProj and θ
for each training minibatch $(I, \mathcal{Y})$ **do**
  Compute unadapted descriptor v and side-encoder features $R_{1:4}$
  Obtain $q^{\text{cnn}}, q^{\text{proto}}, q^{\text{stat}}$, then (π, ρ) using (4)
  Recover local evidence with SCAB and transport it with HAFB
  Inject GCIB/FiLM residuals into the frozen encoder; retain gradients to adapters
  Build Q base hypotheses from sampled foreground boxes and background slots
  Sample nearby times (t, t') and independent Gaussian noise; form $(z_t, z_{t'})$
  Predict clean hypotheses, encode automatic box prompts, and decode both sets
  Compute σ^* by (15); label unmatched queries as no-object
  Compute focal, scale-normalized geometry, prior, denoising, and consistency losses
  Update (ϕ, ψ) with (21); keep encoder and prototype weights fixed
end for
Image-only inference:
Compute $(\pi, \rho, S_{1:4})$ from I; sample $z_{T,j} \sim \mathcal{N}(0, I_4)$ for $j = 1, \ldots, Q$
for t in the fixed decreasing reverse schedule **do**
  Predict $\hat{z}_0^{(t)}$; update hypotheses with (13)
end for
Map refined logits to normalized boxes, rasterize prompts, and decode
Reject no-object and invalid boxes; rank the remaining predictions
return The top predictions under the stated evaluation cap

4. Evaluation Protocol

4.1. Datasets

VisDrone-DET covers ten foreground classes, including small, densely spaced road users [1]. We used 6471 training, 548 validation, and 1610 test-development images. The maintainers provide box annotations for test-development at https://github.com/

VisDrone/VisDrone-Dataset (accessed on 12 November 2026). The separate challenge split with withheld annotations falls outside our evaluation.

DIOR contains 23,463 optical remote sensing images in twenty categories [2]. We used horizontal annotations and the official split of 5862 training, 5863 validation, and 11,738 test images. The images include compact objects and large facilities. The dataset is available at https://gcheng-nwpu.github.io/ (accessed on 12 November 2026).

NWPU VHR-10 includes 650 positive images across ten classes and 150 background images [3]. We split the positive images into 455 training, 65 validation, and 130 test images. The test set also includes all 150 background images, giving 280 test images for this study.

RSOD contains 976 images and four categories (aircraft, oil tank, playground, and overpass) [4]. We use 586 training, 195 validation, and 195 test images. Table 1 gives the partition sizes. The source data are available at https://github.com/RSIA-LIESMARS-WHU/RSOD-Dataset- (accessed on 12 November 2026). The NWPU VHR-10 and RSOD scores characterize performance on the image partitions in Table 1. A source-scene holdout is a separate evaluation.

Table 1. Dataset composition and training, validation, and test partitions.

Dataset	Images	Classes	Train	Val.	Test	Primary Metric
VisDrone-DET	8629	10	6471	548	1610	AP/dense-scene protocol
DIOR	23,463	20	5862	5863	11,738	AP and AP50
NWPU VHR-10	800	10	455	65	280	AP50
RSOD	976	4	586	195	195	AP50

VisDrone excludes the separate 1580-image challenge split. The NWPU VHR-10 test set contains 130 positive and 150 background images; its partition is specific to this study.

4.2. Evaluation Metrics and Implementation

For VisDrone and DIOR, AP averages per-class average precision across classes and IoU thresholds from 0.50 to 0.95 in increments of 0.05 [61]. AP50 and AP75 use a single IoU threshold of 0.50 and 0.75, respectively. AP_s applies the COCO small-object area definition in original image coordinates. NWPU VHR-10 and RSOD use AP50 as their primary metric. Scores are given as percentages and differences in percentage points.

We evaluate VisDrone test-development with COCO-style AP and up to 1000 predictions per image. Set predictors use the same query capacity. Evaluation follows the VisDrone protocol for ignored regions and other-category annotations, and the main comparisons use full images without tiling. The public VisDrone challenge toolkit (AISKYEYE, Tianjin University, Tianjin, China) computes AP with a cap of 500 detections per image (https://github.com/VisDrone/VisDrone2018-DET-toolkit/blob/master/evalDET.m, accessed on 12 November 2026).

Images are resized with their aspect ratio preserved and padded to a 1024×1024 canvas. RS-Adapter training uses horizontal flips and photometric augmentation. The detector comparison uses common input resolution and partitions, while the methods retain their native assignment mechanisms. The tuning comparison holds the encoder architecture and prediction setting fixed.

We used four NVIDIA RTX A6000 GPUs with CUDA 12.1 (NVIDIA Corporation, Santa Clara, CA, USA) and PyTorch 2.2 (PyTorch Foundation, The Linux Foundation, San Francisco, CA, USA). We trained RS-Adapter with AdamW at an initial learning rate of 1×10^{-4} and a weight decay of 0.05 (Table 2). The 150-epoch schedule used cosine decay after five warm-up epochs, with an effective batch size of 16 images. For latency, we report the median of 1000 synchronized batch-one runs after 100 warm-up runs. Timing covers

the full path from image encoding to box decoding, including the reverse updates. We report trainable parameters and peak memory separately.

Table 2. Training and evaluation settings for RS-Adapter.

Item	Setting	Configuration Detail
Encoder	ViT-B-sized; four depth groups	Frozen patch embedding and transformer blocks
Input canvas	1024×1024 pixels	Preserve aspect ratio; pad
Trainable groups	Side encoder, adapters, prior, prompts, heads	Frozen encoder and ProtoProj
Optimizer	AdamW; learning rate 1×10^{-4}	Cosine learning-rate schedule
Weight decay	0.05	Apply consistently within adaptation track
Training schedule	150 epochs	Cosine decay; 5 warm-up epochs
Effective batch	16 images	Images per optimizer update
Subspaces	8	Channel width divisible by subspace count
Prediction capacity	1000	Same cap across compared methods
Reverse updates	3	Deterministic reverse updates at fixed timesteps
Prototype source	Training object crops only	No validation or test annotations
Numerical precision	FP32	Training and timed inference
Timing	Batch 1; 100 warm-up runs	Median over 1000 timed runs

Effective batch size accounts for gradient accumulation. The primary configuration uses three reverse updates.

4.3. Comparison Methods and Experimental Controls

Tables 3 and 4 compare RS-Adapter with 26 baseline configurations under the stated partitions and evaluation settings. These include dense and proposal-based detectors, set predictors, diffusion-based detection, and adaptations of remote sensing and foundation models.

Table 3. VisDrone test-development comparison at 1024-pixel input with a maximum of 1000 detections per image. Scores are percentages. Latency covers the complete prediction procedure.

Model	Backbone	VisDrone Accuracy (%)				Parameters (M)		Latency (ms)
		AP	AP_{50}	AP_{75}	AP_s	Trainable	Total	
Region-proposal detectors								
Faster R-CNN [24]	R50	26.8	48.7	26.1	16.9	41.5	41.5	49.5
Cascade R-CNN [25]	R50	29.4	51.6	29.9	18.5	69.1	69.1	64.2
Dense-prediction detectors								
RetinaNet [26]	R50	24.9	44.8	24.3	14.5	37.7	37.7	38.6
FCOS [27]	R50	28.1	49.6	27.8	17.7	32.3	32.3	34.8
ATSS [28]	R50	29.0	50.8	28.6	18.1	32.1	32.1	36.1
TOOD [29]	R50	30.6	52.9	30.4	19.8	32.0	32.0	37.2
YOLOX-L [30]	CSP	31.2	53.6	31.0	20.0	54.2	54.2	22.3
YOLOv9-C [31]	GELAN	33.5	56.8	33.4	21.9	25.3	25.3	19.8
YOLOv10-L [32]	CSP	33.8	57.4	33.9	22.3	24.4	24.4	18.4
YOLO11-L [33]	C3k2	34.5	58.0	34.6	23.0	25.3	25.3	18.0
YOLO12-L [35]	A2C2f	35.1	58.7	35.3	23.5	26.4	26.4	20.1
YOLO26-L [34]	CSP	36.0	59.8	36.1	24.4	24.8	24.8	17.2

Table 3. *Cont.*

Model	Backbone	VisDrone Accuracy (%)				Parameters (M)		Latency (ms)
		AP	AP_{50}	AP_{75}	AP_s	Trainable	Total	
Transformer set predictors								
DETR [38]	R50	24.3	43.9	24.0	13.8	41.3	41.3	39.6
Deformable DETR [39]	R50	31.6	54.2	31.8	20.4	40.2	40.2	45.0
DAB-DETR [40]	R50	30.2	52.8	30.1	19.0	44.0	44.0	41.2
DINO [41]	R50	35.2	59.2	35.4	23.1	47.7	47.7	51.0
RT-DETR-R50 [42]	R50	33.7	56.9	33.8	21.8	42.0	42.0	26.5
RT-DETRv2-R50 [43]	R50	34.8	58.4	35.0	23.0	42.0	42.0	26.7
D-FINE-L [44]	HGNetv2	36.3	60.1	36.5	24.7	31.4	31.4	23.8
DEIM-D-FINE-L [45]	HGNetv2	37.0	60.9	37.4	25.3	31.4	31.4	23.8
Diffusion-based detection								
DiffusionDet [48]	R50	34.4	58.3	34.8	22.7	41.2	41.2	66.4
Task-specific and adapted configurations								
PKINet-S + FPN [8]	PKINet-S	35.6	59.6	35.8	23.9	35.5	35.5	38.4
ViT-Adapter-B [22]	ViT-B	35.8	59.9	36.1	24.1	123.0	123.0	70.3
RSPrompter-det [23]	ViT-B	36.5	60.4	36.8	24.6	19.2	105.6	76.1
Vision–language detectors								
YOLO-World-L [56]	CSP	34.6	58.1	34.8	22.9	48.1	48.1	27.4
YOLOE-L [58]	CSP	35.4	59.1	35.7	23.8	27.5	27.5	22.0
RS-Adapter (ours)	**ViT-B**	**39.2**	**63.1**	**40.1**	**28.4**	**16.8**	**103.2**	**72.8**

RSPrompter-det denotes the detection configuration. Latency includes all prediction stages; trainable and total parameter counts are reported separately. Italics identify detector families. Bold in the table body identifies RS-Adapter.

RSPrompter-det is the box-detection configuration of RSPrompter [23]. We evaluate PKINet-S with an FPN detection head and give YOLO-World and YOLOE fixed prompts from each dataset's category vocabulary.

We selected configurations on the validation set and used the held-out split for evaluation. The detectors shared the dataset-specific category mapping and, where applicable, the prediction cap. The component ablation held the frozen encoder, decoder, and prompt schedule fixed. The loss ablation held the architecture and matching cost fixed.

Table 4. Detection performance by benchmark and metric. Scores are percentages. Models are trained separately on each dataset.

Model	VisDrone AP	DIOR AP	DIOR AP50	NWPU VHR-10 AP50	RSOD AP50
Region-proposal detectors					
Faster R-CNN	26.8	50.5	65.2	75.8	81.2
Cascade R-CNN	29.4	52.0	67.7	77.4	82.5
Dense-prediction detectors					
RetinaNet	24.9	48.6	62.9	74.6	80.3
FCOS	28.1	51.7	66.9	76.6	81.9
ATSS	29.0	52.5	68.4	77.2	82.3
TOOD	30.6	54.2	68.9	78.2	83.1
YOLOX-L	31.2	54.6	70.6	78.5	83.4
YOLOv9-C	33.5	56.9	71.6	80.0	84.5
YOLOv10-L	33.8	56.1	72.5	80.2	84.7
YOLO11-L	34.5	58.1	73.6	80.6	85.0
YOLO12-L	35.1	57.8	74.0	81.0	85.3
YOLO26-L	36.0	59.0	74.1	81.6	85.7

Table 4. *Cont.*

Model	VisDrone AP	DIOR AP	DIOR AP50	NWPU VHR-10 AP50	RSOD AP50
Transformer set predictors					
DETR	24.3	47.3	63.4	74.2	80.0
Deformable DETR	31.6	54.8	69.8	78.8	83.6
DAB-DETR	30.2	53.7	68.4	77.9	82.9
DINO	35.2	58.9	73.4	81.1	85.3
RT-DETR-R50	33.7	56.4	71.8	80.1	84.6
RT-DETRv2-R50	34.8	58.2	73.3	80.8	85.1
D-FINE-L	36.3	59.6	74.7	81.8	85.9
DEIM-D-FINE-L	37.0	60.1	75.9	82.2	86.2
Diffusion-based detection					
DiffusionDet	34.4	57.9	72.7	80.6	84.9
Task-specific and adapted configurations					
PKINet-S + FPN	35.6	59.2	74.6	81.3	85.5
ViT-Adapter-B	35.8	58.6	73.8	81.4	85.6
RSPrompter-det	36.5	59.7	75.0	81.9	86.0
Vision–language detectors					
YOLO-World-L	34.6	57.3	72.9	80.7	85.0
YOLOE-L	35.4	58.8	74.2	81.2	85.4
RS-Adapter (ours)	**39.2**	**63.8**	**78.3**	**84.5**	**87.7**

NWPU VHR-10 and RSOD use the study partitions in Table 1. Italics identify detector families. Bold in the table body identifies RS-Adapter.

5. Results and Analysis

5.1. Overall Detection and Cross-Benchmark Comparisons

At the stated resolution and prediction cap, RS-Adapter achieved 39.2% AP on VisDrone (Table 3). It exceeded DEIM-D-FINE-L by 2.2 percentage points and YOLO26-L by 3.2 points.

On small objects, RS-Adapter reached 28.4% AP_s, compared with 25.3% for DEIM-D-FINE-L. At the stricter overlap threshold, the two models scored 40.1% and 37.4% AP75, respectively.

On DIOR, RS-Adapter achieved 63.8% AP and 78.3% AP50, compared with 60.1% and 75.9% for DEIM-D-FINE-L (Table 4). RS-Adapter also reached 84.5% AP50 on NWPU VHR-10 and 87.7% on RSOD under the partitions in Table 1.

5.2. Category-Resolved Analysis

RS-Adapter had the highest AP50 among the four models in each of the ten VisDrone categories (Table 5). The class macro-average was 63.1%.

On DIOR, RS-Adapter led the three models in 19 of 20 categories (Table 6). DEIM-D-FINE-L scored highest on golf fields. Averaging the 20 displayed category scores gives 73.4% AP50 for DINO, 75.9% for DEIM-D-FINE-L, and 78.3% for RS-Adapter after rounding to one decimal place. These means agree with Table 4 to the reported precision.

Table 5. Category AP50 on VisDrone (%). The final row is the ten-class average.

Category	DINO	DEIM-D-FINE-L	YOLO26-L	RS-Adapter
Pedestrian	58.3	59.0	58.2	61.5
People	46.6	47.7	46.7	50.6
Bicycle	36.1	37.6	36.3	40.0
Car	93.5	94.5	94.0	95.0
Van	65.6	66.6	65.4	68.4
Truck	66.5	68.9	67.8	71.2
Tricycle	46.5	48.2	47.1	51.0
Awning-tricycle	33.3	35.1	34.0	38.1
Bus	86.3	87.7	87.0	89.3
Motor	59.3	63.7	61.5	65.9
Macro-average	**59.2**	**60.9**	**59.8**	**63.1**

Bold in the table body identifies the macro-average.

Table 6. Category AP50 on DIOR (%). The final row is the 20-class average.

Category	DINO	DEIM-D-FINE-L	RS-Adapter
Airplane	81.2	83.6	86.1
Airport	72.7	75.5	77.4
Baseball field	82.3	83.2	84.7
Basketball court	74.4	77.8	79.9
Bridge	61.1	61.8	65.4
Chimney	75.9	78.6	80.4
Dam	66.8	70.1	72.3
Expressway service area	65.5	68.8	70.7
Expressway toll station	65.2	67.5	69.8
Golf field	77.3	79.2	78.8
Ground track field	79.8	81.7	85.2
Harbor	74.1	76.6	78.4
Overpass	60.4	63.5	67.7
Ship	79.6	79.1	83.2
Stadium	80.4	84.1	86.2
Storage tank	83.3	85.8	88.9
Tennis court	82.8	85.3	87.2
Train station	68.7	72.4	74.5
Vehicle	58.1	62.0	66.4
Windmill	78.2	80.7	82.8
Macro-average	**73.4**	**75.9**	**78.3**

Bold in the table body identifies the macro-average.

5.3. Qualitative Evidence in Dense Aerial Scenes

Figure 3 compares the boxes displayed by RT-DETR, YOLO26, and RS-Adapter in crowded daytime and low-light scenes.

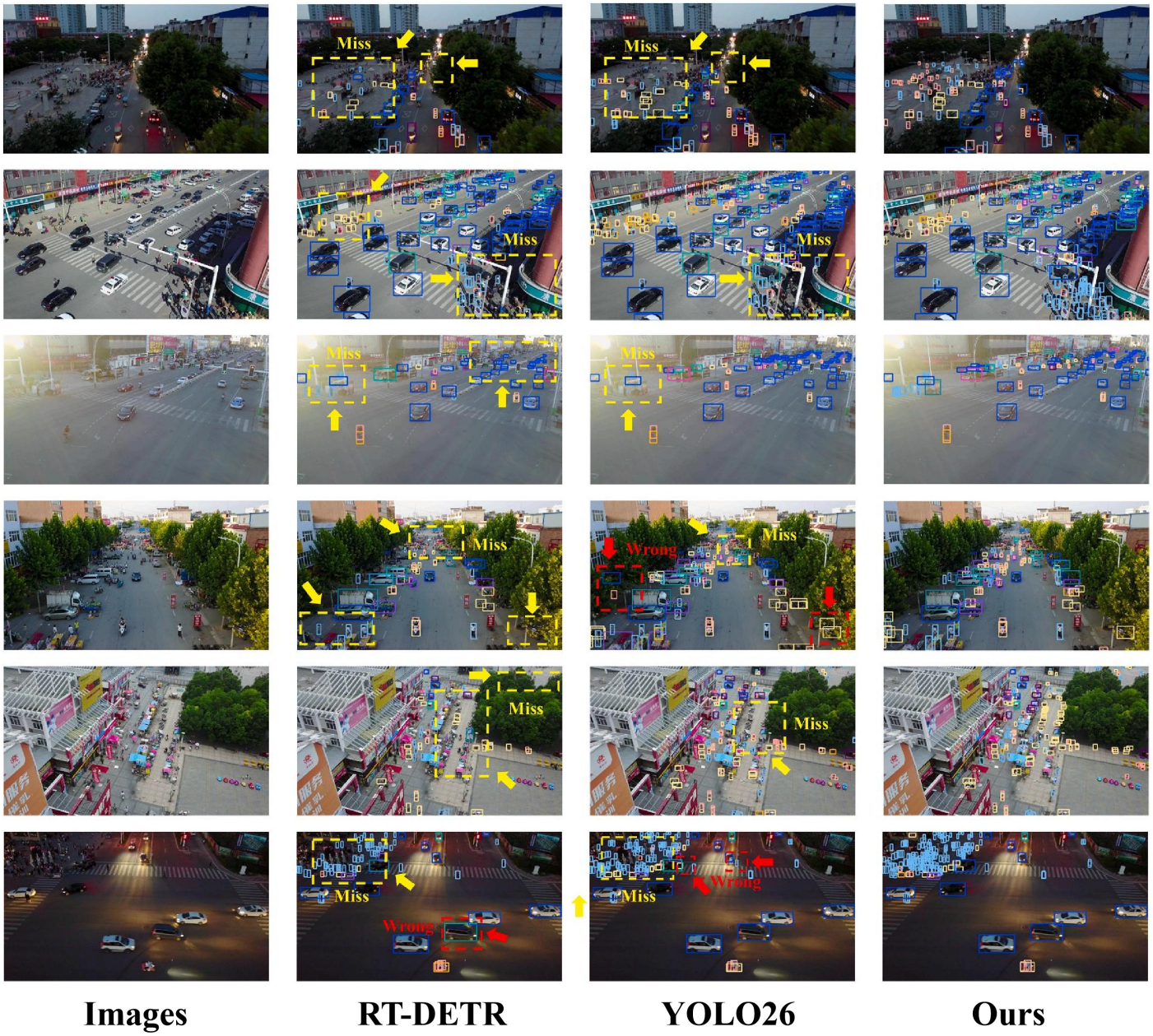


Figure 3. VisDrone examples. Columns show the input image, RT-DETR, YOLO26, and RS-Adapter. A yellow "Miss" highlights a region with a visible target that has no corresponding displayed box. A red "Wrong" flags a displayed box whose class, placement, or target association appears inconsistent with the image. Solid rectangular outlines show predicted boxes, with color distinguishing object classes. Arrows and dashed outlines point to the flagged regions, with yellow for missed targets and red for apparent errors. Both callouts are based on visual inspection of the displayed boxes.

The insets in Figure 4 show how DINO, DiffusionDet, and RS-Adapter place boxes around small objects at intersections and on arterial roads.

The response overlays in Figure 5 show where each model responds around road users and along the roadway.

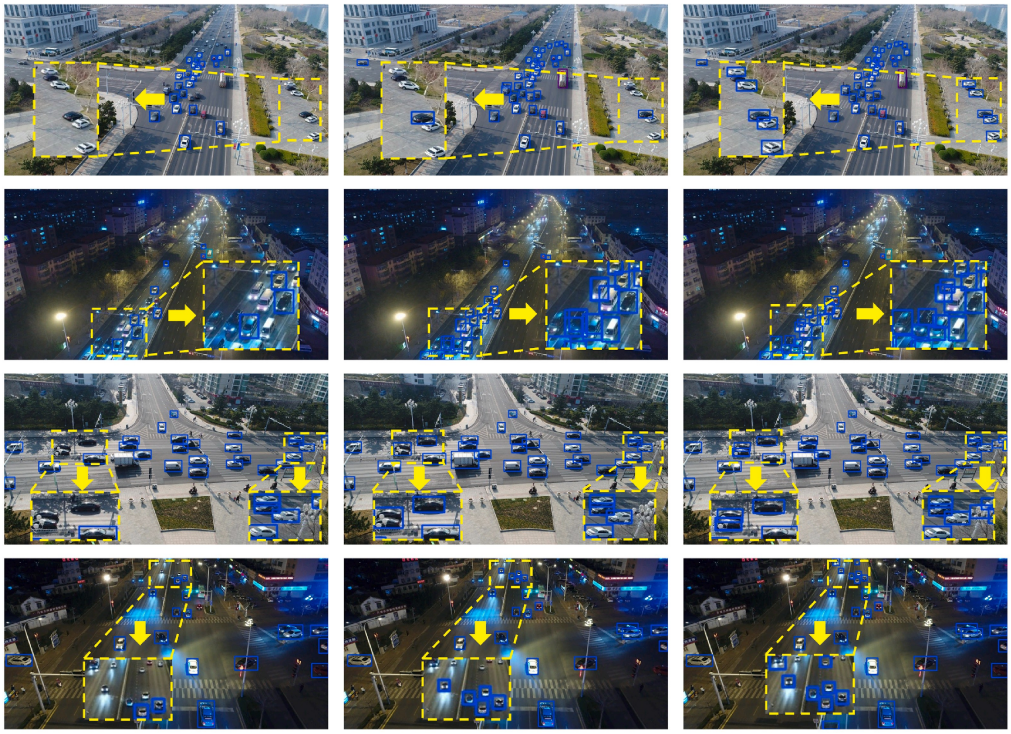


Figure 4. VisDrone examples from DINO, DiffusionDet, and RS-Adapter, from left to right. Yellow dashed rectangles and arrows link each highlighted region to its enlarged view. Solid colored outlines show predicted boxes.

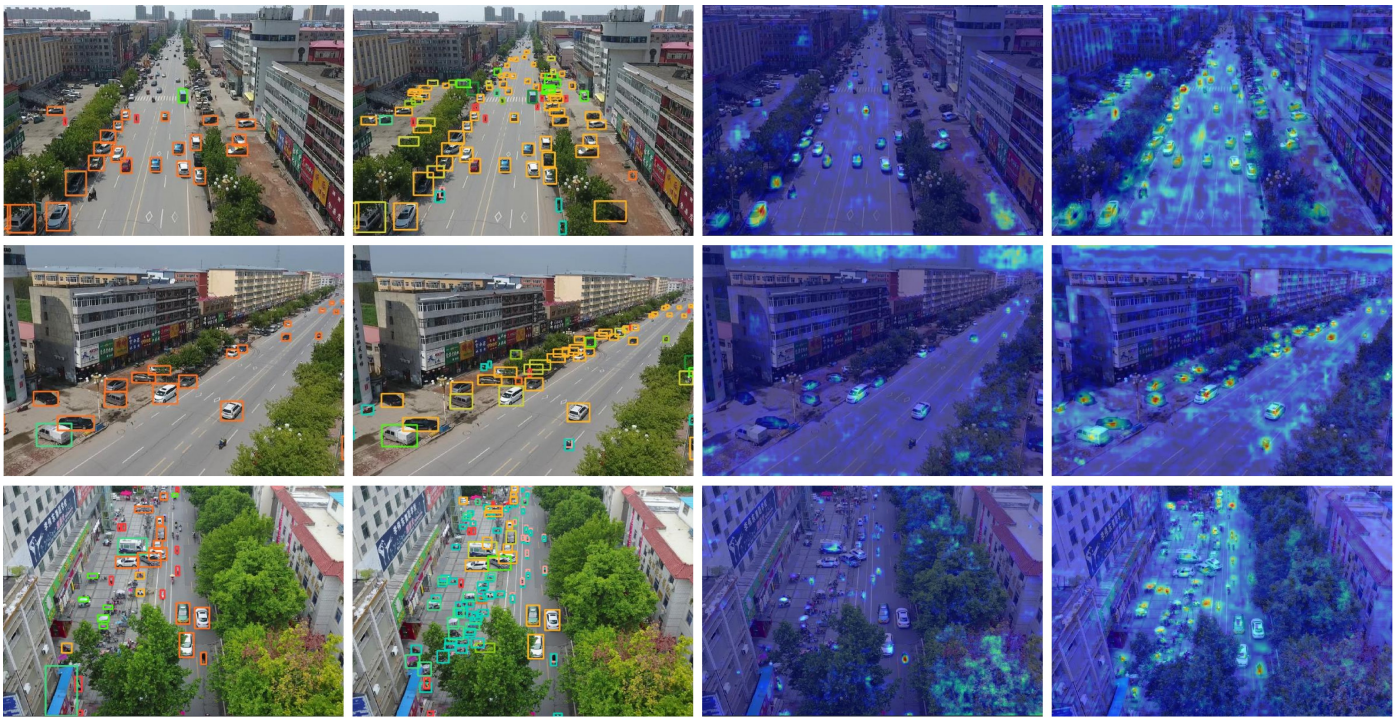


Figure 5. VisDrone detections from (**a**) ViT-Adapter and (**b**) RS-Adapter, with the response overlays for ViT-Adapter in (**c**) and RS-Adapter in (**d**). The colored outlines in (**a**,**b**) show predicted boxes, with different colors distinguishing object categories. The qualitative color key runs from ▬ lower to ▬ higher displayed response within each panel.

5.4. Overhead Categories and Extended Structures

The DIOR examples in Figure 6 include courts with repeated markings, ships in water, and small storage tanks near larger structures.



Figure 6. RS-Adapter detections on DIOR across compact objects and extended facilities, including aircraft, ships, courts, chimneys, storage tanks, and harbors. Colored rectangles show predicted boxes. The adjacent text gives the predicted class and confidence score; colors distinguish classes within each panel.

Figure 7 compares RSPrompter and RS-Adapter on scenes that contain multiple categories, including courts beside a track and vehicles near a station or a highway service area.

Figure 8 shows detections of compact objects and larger structures on NWPU VHR-10.

The ground-truth column in Figure 9 provides a reference for box placement. In row (b), the two long boxes from RS-Adapter follow the annotated bridge spans. In row (d), RS-Adapter displays boxes along the berths on the right, where the two baselines show fewer boxes.

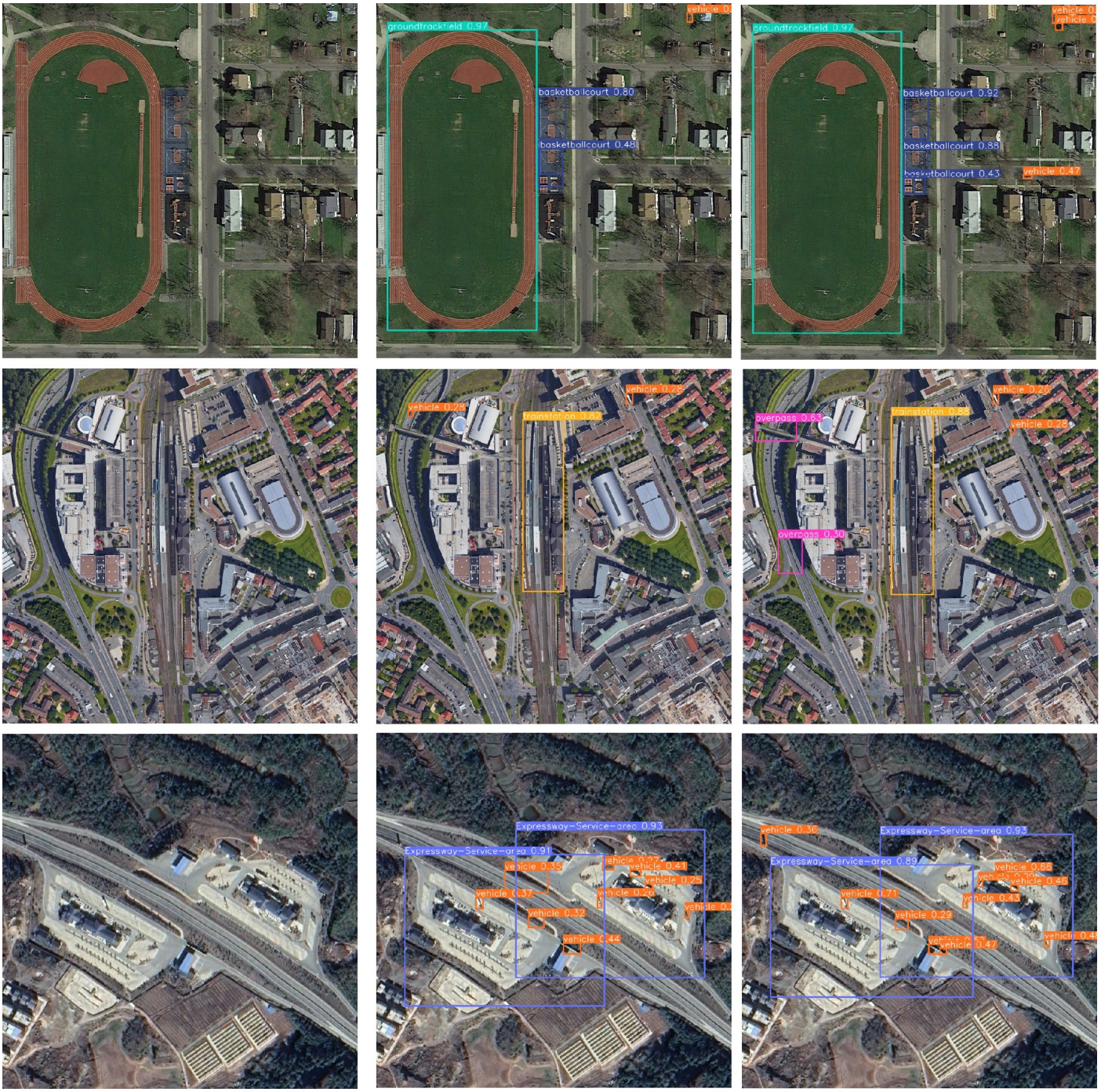


Figure 7. DIOR examples. Columns show the input image, RSPrompter detection output, and RS-Adapter detection output. Colored rectangles show predicted locations. Cyan marks ground track fields, dark blue marks basketball courts, orange marks vehicles, yellow marks train stations, magenta marks overpasses, and violet-blue marks expressway service areas. Text labels show the predicted class and confidence score.

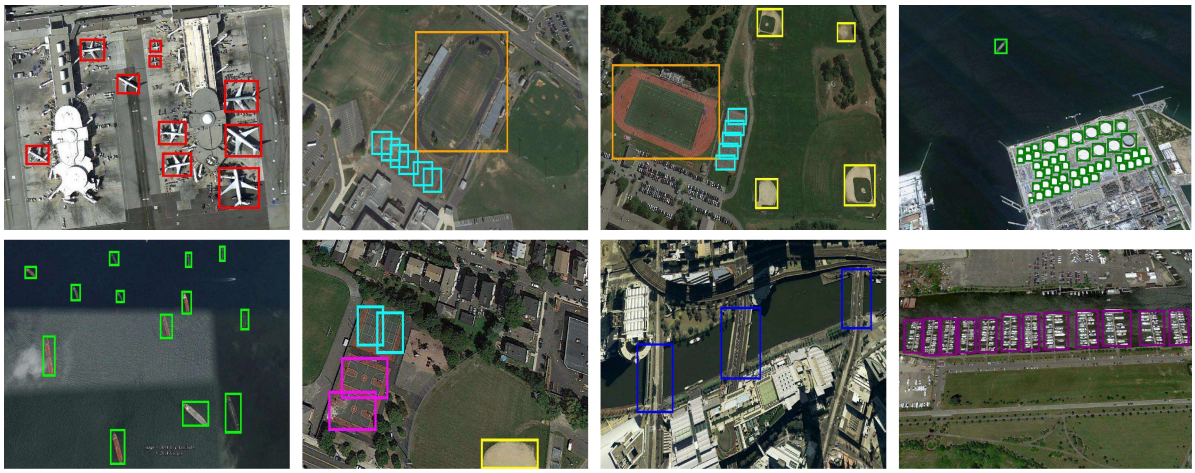


Figure 8. RS-Adapter detections on NWPU VHR-10, including repeated compact targets, elongated bridges, and sports facilities. Colored outlines show predicted boxes. Examples include red boxes around aircraft in the upper left, blue boxes around bridges in the lower row, and magenta boxes around harbors in the lower right.

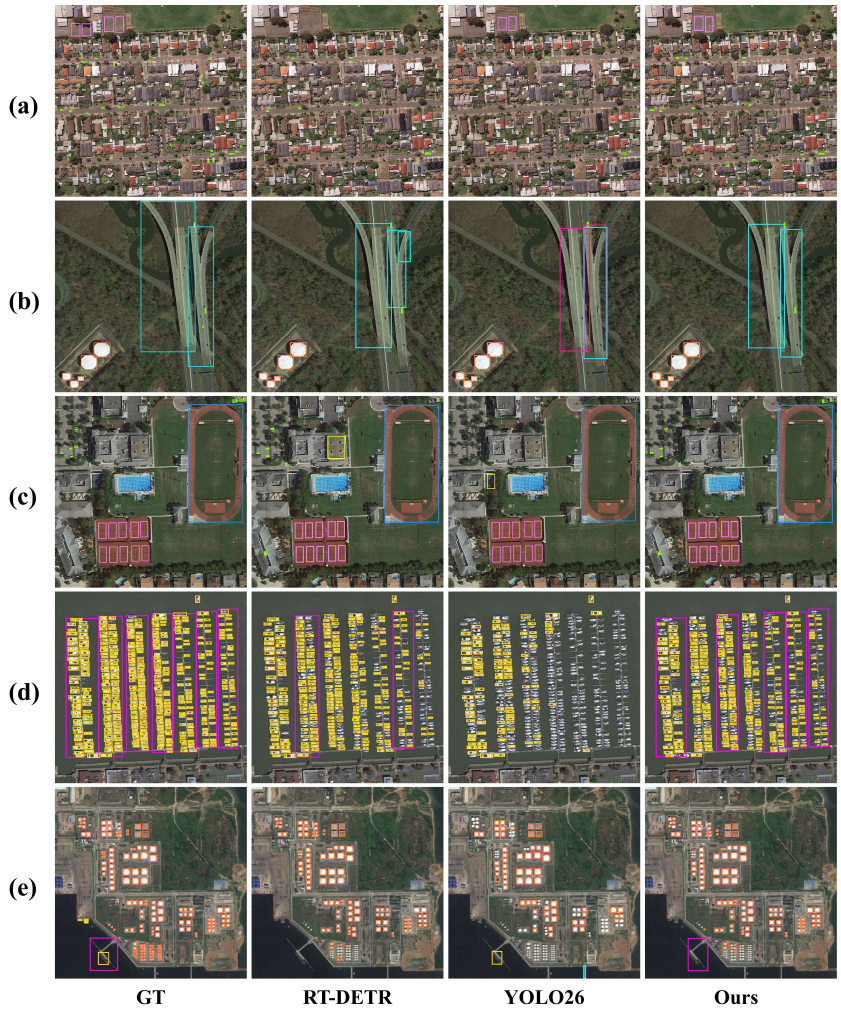


Figure 9. NWPU VHR-10 comparisons. The GT column shows dataset annotations. The next three columns show predictions by RT-DETR, YOLO26, and RS-Adapter. Box colors denote object categories, such as cyan for bridges, yellow for ships, and magenta for harbors in the GT column. (**a**) Residential area with small vehicles and sports courts. (**b**) Bridge spans beside storage tanks. (**c**) Sports complex with courts and a running track. (**d**) Rows of densely moored ships in a marina. (**e**) Storage tanks and harbor structures at the waterfront.

5.5. RSOD Detections and Response Overlays

The RSOD examples in Figure 10 show boxes for aircraft, oil tanks, overpasses, and playgrounds.

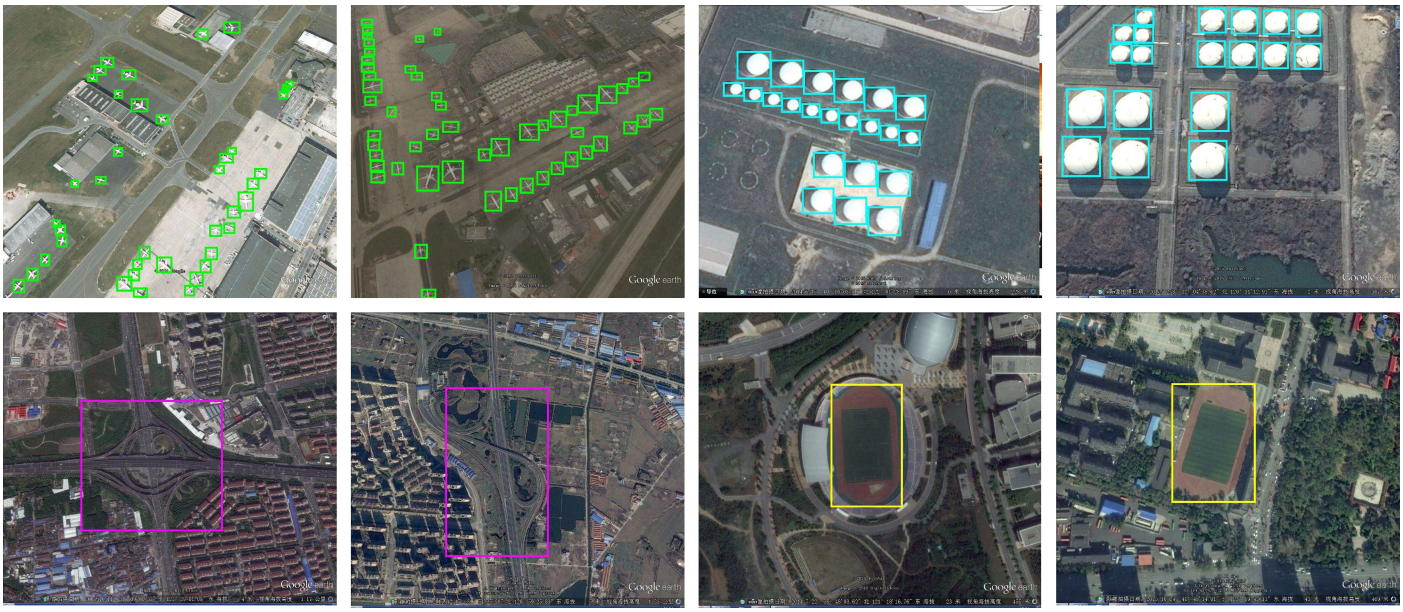


Figure 10. RS-Adapter detection gallery on RSOD. The predicted boxes are green for aircraft, cyan for oil tanks, magenta for overpasses, and yellow for playgrounds. The Chinese Google Earth status text identifies the imagery date, geographic coordinates (N and E), terrain elevation, and viewing altitude. Distances are shown in metres (m) or kilometres (km).

In the DEIM and RS-Adapter overlays (Figure 11), warm regions tend to cluster around individual aircraft. They cover broader areas for the overpass and playground.

Figures 3–11 illustrate box placement and spatial response patterns in selected scenes. The quantitative comparisons use AP computed on the full test partitions.

Figure 12 shows the VisDrone metrics from Table 3. RS-Adapter achieved 63.1% AP50, 6.2 points above RT-DETR-R50, and 39.2% AP across IoU thresholds.

5.6. Adapter Component and Class-Prior Ablations

With the encoder, decoder, and prompt schedule fixed, VisDrone AP rose from 34.1% in the base configuration to 38.3% with the complete adapter (Table 7). Adding the proposed loss raised AP to 39.2%. The component rows change trainable capacity as well as the design, from 10.6 million parameters in the base to 16.8 million in the complete adapter.

The prior alone reached 35.0% AP. Adding SCAB, HAFB, or GCIB separately gave 36.1%, 35.8%, and 35.5%, respectively. SCAB and HAFB together reached 37.4%, and adding GCIB brought AP to 38.3%. The standard unconditioned adapter uses 14.4 million trainable parameters (Section 5.9). Section 5.10 compares adapters with similar trainable parameter counts.

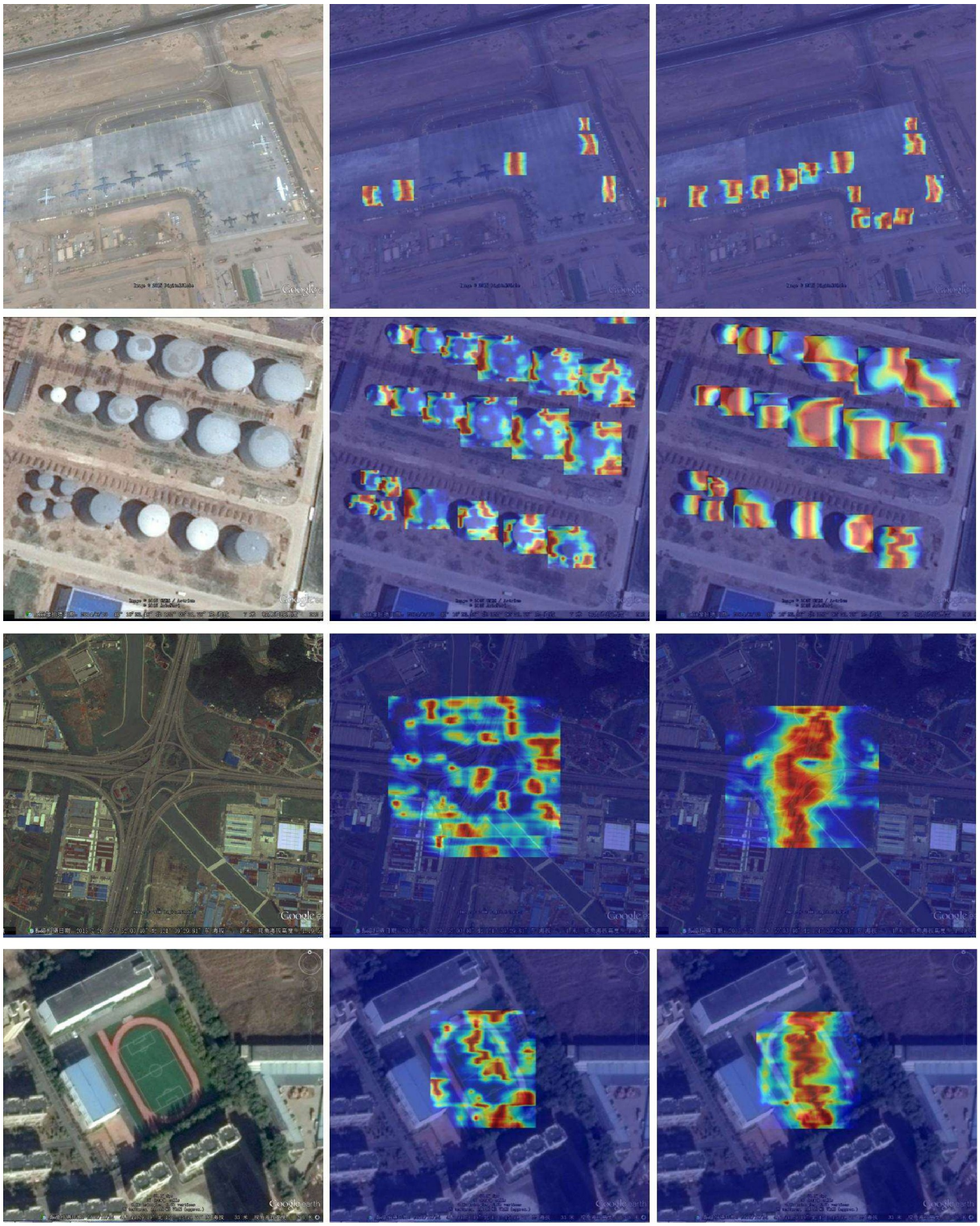


Figure 11. RSOD input images and response overlays from DEIM and RS-Adapter, from left to right. The qualitative color key runs from lower to higher displayed response within each panel. The Chinese Google Earth status text identifies the imagery date, geographic coordinates (N and E), terrain elevation, and viewing altitude. Distances are shown in metres (m) or kilometres (km).

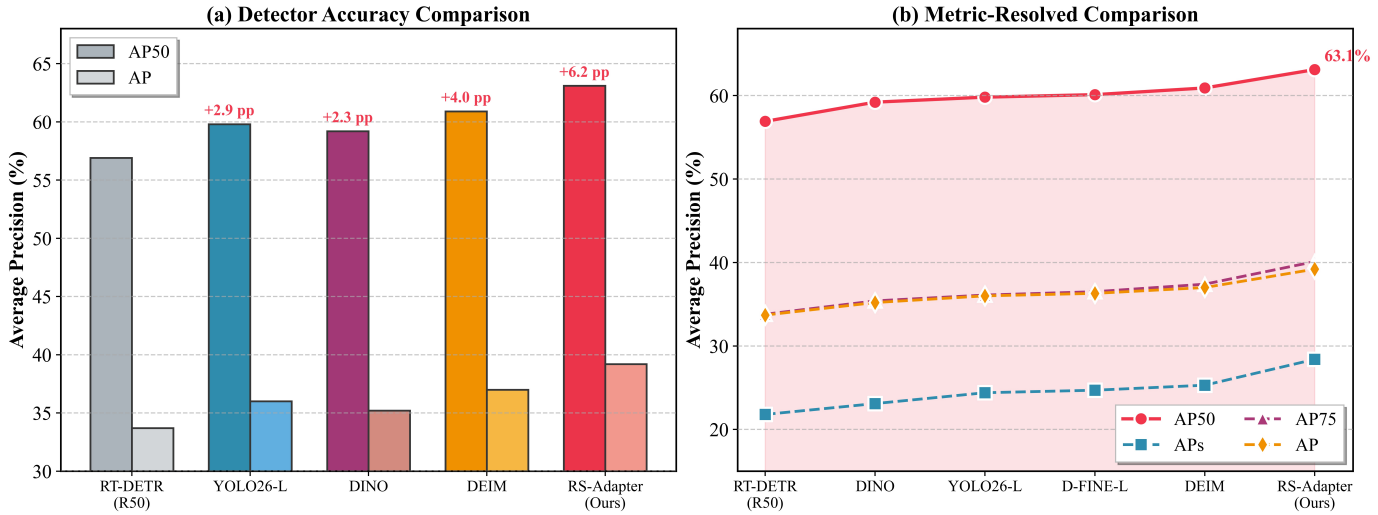


Figure 12. VisDrone detection accuracy for representative configurations from Table 3: (**a**) paired AP50 and AP bars, with annotations giving AP50 gains in percentage points over RT-DETR-R50; and (**b**) AP50, AP_S, AP75, and AP across six detector configurations. In (**a**), each color identifies a detector; darker bars show AP50 and lighter bars show AP. In (**b**), red circles, blue squares, purple triangles, and orange diamonds represent AP50, AP_S, AP75, and AP, respectively. The pink shading is a visual guide under the AP50 curve. DEIM denotes DEIM-D-FINE-L.

Table 7. Component ablation of class-conditioned evidence recovery with the frozen encoder, prediction capacity, and denoising schedule fixed.

Configuration	Adapter Components				VisDrone (%)		DIOR (%)	Trainable (M)
	Prior	SCAB	HAFB	GCIB	AP	AP_s	AP	
Base	–	–	–	–	34.1	22.0	58.6	10.6
Prior	Yes	–	–	–	35.0	22.9	59.4	11.4
Prior + SCAB	Yes	Yes	–	–	36.1	24.2	60.7	12.9
Prior + HAFB	Yes	–	Yes	–	35.8	24.0	60.3	12.5
Prior + GCIB	Yes	–	–	Yes	35.5	23.4	60.4	12.1
Prior + SCAB + HAFB	Yes	Yes	Yes	–	37.4	26.0	61.8	14.0
Complete adapter	Yes	Yes	Yes	Yes	38.3	27.3	63.0	16.8
Adapter + proposed loss	**Yes**	**Yes**	**Yes**	**Yes**	**39.2**	**28.4**	**63.8**	**16.8**

The complete adapter uses the reference detection objective; the final row adds the proposed objective. Loss terms introduce no additional inference parameters. Bold in the table body identifies the adapter with the proposed objective.

With the adapter and objective fixed, the three-branch prior yielded 39.2% AP on VisDrone (Table 8). A uniform prior gave 36.8% AP. The best individual branch reached 37.1%, and the CNN-plus-prototype prior reached 38.1%.

Shuffling the mixture's class entries at inference reduced VisDrone AP from 39.2% to 34.7%.

Table 8. Class-prior controls with the complete adapter and objective fixed.

Prior Condition	Prior Branches			VisDrone (%)		DIOR (%)
	CNN	ProtoProj	Statistical	AP	AP_s	AP
Uniform	–	–	–	36.8	25.2	61.6
Shuffled mixture	Yes	Yes	Yes	34.7	22.8	59.1
CNN only	Yes	–	–	37.1	25.6	62.0
Prototype only	–	Yes	–	36.7	25.0	61.3
Statistical only	–	–	Yes	36.5	24.8	61.4
CNN + prototype	Yes	Yes	–	38.1	26.9	62.8
CNN + statistical	Yes	–	Yes	37.8	26.4	62.5
Three-branch mixture	**Yes**	**Yes**	**Yes**	**39.2**	**28.4**	**63.8**

The uniform prior assigns equal class weights. The learned priors use the current image, and shuffling permutes the mixture's class entries at inference. The prototype branch remains frozen. Bold in the table body identifies the three-branch mixture.

5.7. Loss Contributions and Geometric Sensitivity

With the architecture and matching fixed, the reference loss gave 38.3% AP and 27.3% AP_s on VisDrone (Table 9). The full loss gave 39.2% and 28.4%, respectively. Adding one term at a time gave gains of 0.3–0.5 AP points in these runs.

Table 9. Objective ablation with the architecture and matching cost fixed.

Objective	Loss Terms				VisDrone (%)		DIOR (%)
	Area	Wasserstein	Prior	Consistency	AP	AP_s	AP
Reference	–	–	–	–	38.3	27.3	63.0
+Area weighting	Yes	–	–	–	38.7	27.8	63.3
+Wasserstein	–	Yes	–	–	38.8	27.9	63.5
+Prior supervision	–	–	Yes	–	38.6	27.6	63.2
+Noise consistency	–	–	–	Yes	38.6	27.5	63.1
Full objective	**Yes**	**Yes**	**Yes**	**Yes**	**39.2**	**28.4**	**63.8**

All rows keep direct denoiser supervision. Area denotes Equation (16); Wasserstein denotes Equation (17). Bold in the table body identifies the full objective.

The prior-supervision row adds presence BCE and frequency KL together. Section 5.10 reports their separate effects and the variability across independent trainings.

5.8. Prompt Perturbation and Denoising Budget

We apply perturbations to DINO's reference boxes, DiffusionDet's sampled starting boxes, and the normalized boxes corresponding to RS-Adapter's initial hypotheses. The scale σ is defined in normalized coordinates. The effective perturbation can differ after each model's coordinate transforms and refinement. Table 10 gives the RS-Adapter drop relative to its own AP at $\sigma = 0$.

In Table 11, going from one to three reverse updates raised VisDrone AP from 37.9% to 39.2% and latency from 43.6 to 72.8 ms. Five updates reached 39.4% AP at 102.0 ms. We use three updates for the main results. Section 5.10 compares three-update denoising with two other prompt generators at similar inference costs.

Table 10. VisDrone AP under coordinate perturbation of initial query hypotheses.

Noise Level σ	VisDrone AP (%)			RS-Adapter AP Drop (Points)
	DINO	DiffusionDet	RS-Adapter	
0	35.2	34.4	39.2	0.0
0.05	34.8	34.2	39.0	0.2
0.1	33.9	33.7	38.5	0.7
0.15	32.4	32.8	37.7	1.5
0.2	30.2	31.5	36.6	2.6
0.3	26.4	28.9	33.9	5.3

The VisDrone prediction cap is unchanged. At $\sigma = 0$ each method uses its unperturbed initialization. We perturb boxes at each model's initialization interface, before its own refinement procedure.

Table 11. Effect of the number of reverse updates on detection accuracy and latency.

Reverse Updates	VisDrone (%)		DIOR (%)	Latency (ms)
	AP	AP_s	AP	
1	37.9	27.1	62.7	43.6
2	38.7	27.8	63.4	58.2
3	39.2	28.4	63.8	72.8
5	39.4	28.6	64.0	102.0
8	39.5	28.7	64.1	145.8

Latency includes the encoder, prior generator, prompt refinement, and decoder. Three updates define the primary configuration.

Tables 7–11 report point estimates for the listed configurations.

5.9. Computational Cost and Adaptation Scope

The full detector has 103.2 million parameters, of which 16.8 million are trainable. Table 12 reports the training and inference costs separately. Compared with full tuning of the same ViT-B-sized encoder, RS-Adapter trained 83.7% fewer parameters. Peak memory fell from 19.8 to 14.1 GB. The reported AP values were 38.5% for full tuning and 39.2% for RS-Adapter.

Table 12. Accuracy and resource use of tuning strategies with a common ViT-B-sized encoder and three reverse updates.

Strategy	Tuning Scope	Parameters (M)		VisDrone AP (%)	Peak Memory (GB)	Compute (GFLOPs)	Latency (ms)
		Total	Trainable				
Full tuning	All	103.2	103.2	38.5	19.8	284	68.5
Frozen + decoder	Decoder	97.0	10.6	34.1	8.7	248	63.4
LoRA + decoder	Low-rank	99.9	13.5	36.4	11.9	259	66.1
Spatial adapter	Unconditioned	100.8	14.4	37.0	13.2	278	70.2
RS-Adapter	**Class-conditioned**	**103.2**	**16.8**	**39.2**	**14.1**	**284**	**72.8**

Training memory uses the common effective batch and precision. Compute includes encoding and refinement. The full-tuning reference uses the controlled adaptation configuration. Bold in the table body identifies RS-Adapter.

RS-Adapter and full tuning both use 284 GFLOPs (Figure 13). Batch-one throughput was about 13.7 and 14.6 frames per second, respectively, including prompt refinement. Section 5.10 reports the AP variation across three independent trainings for both strategies.

5.10. Loss Combinations, Matched Controls, and Repeatability

With the reference detection and denoising losses fixed, presence BCE alone reached 38.5% AP on VisDrone and frequency KL alone reached 38.4%, compared with 38.3% with

both terms disabled (Table 13). Using both terms gave 38.6%. Combining area weighting and Wasserstein localization gave 38.9%, and adding consistency to the two prior terms gave 38.7%.

The capacity comparison in Table 14 uses 16.7 million trainable parameters for the widened unconditioned adapter and 16.8 million for the class-conditioned adapter. Their VisDrone AP values were 38.3% and 38.6%, respectively. With the class-conditioned pathway fixed, direct box regression scored 38.4% AP and deterministic refinement scored 38.8%. Three-update denoising reached 39.2% AP at 72.8 ms, compared with 71.9 ms for deterministic refinement.

Across three independent training runs, RS-Adapter averaged 39.17% AP with a sample standard deviation of 0.21 points. Full tuning averaged 38.57% with a standard deviation of 0.25 points. Table 15 reports training variability across the configurations. The difference between the means was 0.60 points, compared with the 0.7-point difference in the tuning comparison in Table 12. For the single-term loss controls, mean AP was 0.33–0.50 points above the reference objective and the sample standard deviations ranged from 0.15 to 0.26 points.

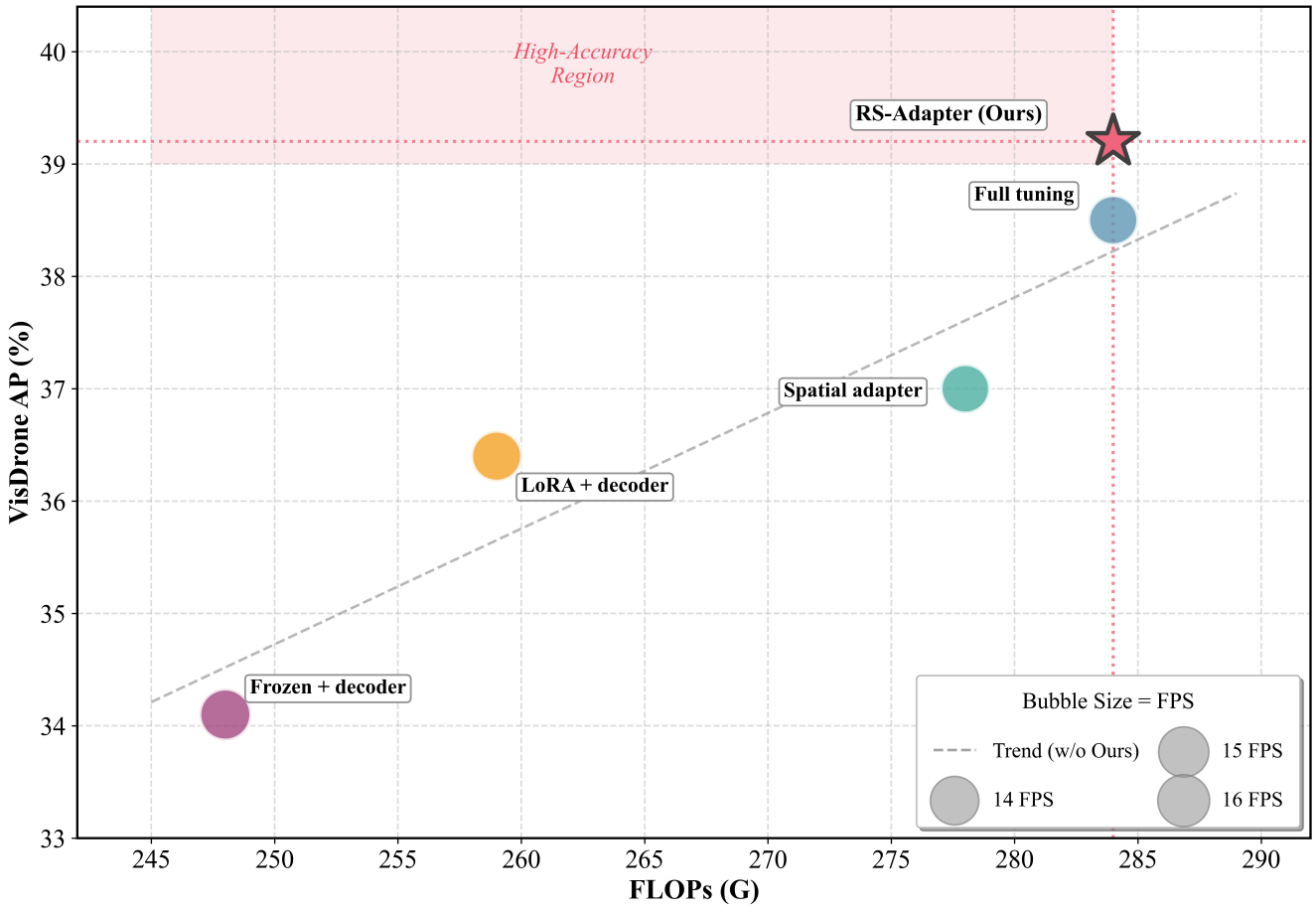


Figure 13. Accuracy and inference cost of the controlled tuning strategies in Table 12. Bubble area is proportional to batch-one throughput, computed as 1000 divided by latency in milliseconds; the star marks RS-Adapter. The dashed line is a descriptive least-squares fit to the four reference strategies. The shaded region highlights AP of at least 39% at compute no greater than RS-Adapter, and the dotted lines mark its operating point.

Table 13. Separate and joint controls for prior supervision and other loss terms.

Configuration	Presence BCE	Frequency KL	Additional Losses	VisDrone AP (%)	VisDrone AP_s (%)	DIOR AP (%)
No prior supervision	Off	Off	None	38.3	27.3	63.0
Presence only	On	Off	None	38.5	27.6	63.2
Frequency only	Off	On	None	38.4	27.4	63.1
Both prior terms	On	On	None	38.6	27.6	63.2
Geometric pair	Off	Off	Area + Wasserstein	38.9	28.0	63.6
Prior and consistency	On	On	Consistency	38.7	27.8	63.4

All configurations retain the reference detection and direct denoising losses. The rows with both prior terms disabled or enabled repeat the reference and prior-supervision results in Table 9, respectively.

Table 14. Adapter capacity and box-prompt controls on VisDrone.

Configuration	Trainable (M)	GFLOPs	Latency (ms)	AP (%)	AP_s (%)
Capacity comparison with the shared objective					
Widened unconditioned adapter	16.7	282	71.5	38.3	27.1
Class-conditioned adapter	16.8	284	72.8	38.6	27.5
Prompt comparison with the class-conditioned pathway					
Direct box regression	16.7	279	68.4	38.4	27.6
Deterministic iterative refinement	16.9	283	71.9	38.8	28.0
Three-update denoising	16.8	284	72.8	39.2	28.4

The adapter capacity comparison uses the same objective in both rows. The prompt comparison uses the same class-conditioned feature pathway and detection decoder at similar inference costs. The three-update row reports the main RS-Adapter result. Italics distinguish the capacity and prompt controls.

Table 15. Results from three independent training runs on VisDrone.

Configuration	Run A AP (%)	Run B AP (%)	Run C AP (%)	AP Mean (%) $\pm$ Sample SD (Points)
RS-Adapter	39.0	39.4	39.1	39.17 ± 0.21
Full tuning	38.6	38.8	38.3	38.57 ± 0.25
Reference objective	38.2	38.5	38.1	38.27 ± 0.21
Area weighting	38.5	38.8	38.7	38.67 ± 0.15
Wasserstein	38.8	38.6	38.9	38.77 ± 0.15
Joint prior supervision	38.6	38.4	38.8	38.60 ± 0.20
Noise consistency	38.4	38.9	38.5	38.60 ± 0.26
Widened unconditioned adapter	38.3	38.7	38.4	38.47 ± 0.21
Direct box regression	38.4	38.6	38.7	38.57 ± 0.15
Deterministic iterative refinement	38.8	39.0	38.7	38.83 ± 0.15

The final column reports the mean AP and sample standard deviation computed from the three runs.

6. Discussion

Compared with DEIM-D-FINE-L on VisDrone, RS-Adapter gained 3.1 points in small-object AP and 2.7 points in AP75. These differences compare the complete detectors. The class-prior controls vary the conditioning signal, and the capacity control compares conditioned and unconditioned adapters with similar trainable parameter counts.

ViT-Adapter-B, RSPrompter-det, and DiffusionDet scored 35.8%, 36.5%, and 34.4% AP on VisDrone, with 123.0, 19.2, and 41.2 million trainable parameters, respectively (Table 3). ViT-Adapter-B adapts a transformer for dense prediction. RSPrompter learns prompts from remote sensing images, and DiffusionDet refines boxes to produce detections. In

RS-Adapter, the image-level class prior conditions the encoder residuals, and the adapted features guide the refinement of boxes used as decoder prompts.

With the adapter fixed, permuting the class scores reduced VisDrone AP from 39.2% to 34.7%, suggesting that the correspondence between scores and classes matters. At similar trainable parameter counts, class conditioning gave a 0.3-point AP gain over the widened unconditioned adapter (Table 14). Three-update denoising scored 0.4 points above deterministic refinement at a similar inference cost. These are point estimates from the two controls. The sample standard deviations in Table 15 give context for differences of a few tenths of an AP point.

The two Gaussian components operate on different statistics. The class prior models how descriptors vary across instances of a class. GCIB computes statistics across spatial locations to gate the interaction between feature branches.

In the tuning comparison, RS-Adapter has fewer trainable parameters and uses less peak training memory than full tuning. At inference, the full encoder and prompt denoiser still run. Table 11 shows how additional reverse updates affect latency.

The evaluation is restricted to horizontal boxes and a fixed vocabulary for each dataset. Oriented detection would require representing angles in the box state and prompt encoding. An open-vocabulary version would require class prototypes that cover categories beyond the training labels.

7. Conclusions

RS-Adapter conditions the residual updates of a frozen vision transformer on an image-level class prior. The adapted features also guide the denoiser that generates box prompts. On VisDrone, the detector reached 39.2% AP and 28.4% AP_s with 16.8 million trainable parameters. AP50 was 78.3% on DIOR, 84.5% on NWPU VHR-10, and 87.7% on RSOD. In the VisDrone tuning comparison, RS-Adapter used 14.1 GB of peak training memory compared with 19.8 GB for full tuning.

Author Contributions: Conceptualization, H.X. and P.L.; methodology, H.X. and X.Z.; software, H.X.; validation, H.X., X.Z., and S.Y.; formal analysis, H.X.; investigation, H.X.; data curation, H.X.; writing—original draft preparation, H.X.; writing—review and editing, X.Z., S.Y., and P.L.; visualization, H.X.; supervision, P.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets are publicly available from VisDrone https://github.com/VisDrone/VisDrone-Dataset (accessed on 12 November 2026), the DIOR and NWPU VHR-10 author page https://gcheng-nwpu.github.io/ (accessed on 12 November 2026), and RSOD https://github.com/RSIA-LIESMARS-WHU/RSOD-Dataset- (accessed on 12 November 2026).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zhu, P.; Wen, L.; Du, D.; Bian, X.; Fan, H.; Hu, Q.; Ling, H. Detection and Tracking Meet Drones Challenge. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 7380–7399. https://doi.org/10.1109/TPAMI.2021.3119563.
2. Li, K.; Wan, G.; Cheng, G.; Meng, L.; Han, J. Object detection in optical remote sensing images: A survey and a new benchmark. *ISPRS J. Photogramm. Remote Sens.* **2020**, *159*, 296–307. https://doi.org/10.1016/j.isprsjprs.2019.11.023.
3. Cheng, G.; Han, J. A survey on object detection in optical remote sensing images. *ISPRS J. Photogramm. Remote Sens.* **2016**, *117*, 11–28. https://doi.org/10.1016/j.isprsjprs.2016.03.014.
4. Long, Y.; Gong, Y.; Xiao, Z.; Liu, Q. Accurate Object Localization in Remote Sensing Images Based on Convolutional Neural Networks. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 2486–2498. https://doi.org/10.1109/TGRS.2016.2645610.

5. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv* **2021**, https://doi.org/10.48550/arXiv.2010.11929.
6. He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; Girshick, R. Masked Autoencoders Are Scalable Vision Learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2022; pp. 16000–16009.
7. Lin, T.Y.; Dollar, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2017.
8. Cai, X.; Lai, Q.; Wang, Y.; Wang, W.; Sun, Z.; Yao, Y. Poly Kernel Inception Network for Remote Sensing Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2024; pp. 27706–27716.
9. Sun, H.; Wang, R.; Li, Y.; Yang, L.; Lin, S.; Cao, X.; Zhang, B. SET: Spectral Enhancement for Tiny Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2025; pp. 4713–4723.
10. Zhao, T.; Kiblawi, S.; Usuyama, N.; Lee, H.H.; Preston, S.; Poon, H.; Wei, M. Boltzmann Attention Sampling for Image Analysis with Small Objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2025; pp. 25950–25959.
11. Bian, J.; Feng, M.; Dong, W.; Wu, F.; Luo, J.; Wang, Y.; Shi, G. Feature Information Driven Position Gaussian Distribution Estimation for Tiny Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2025; pp. 30376–30386.
12. Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A.C.; Lo, W.Y.; et al. Segment Anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*; IEEE: New York, NY, USA, 2023; pp. 4015–4026.
13. Ravi, N.; Gabeur, V.; Hu, Y.T.; Hu, R.; Ryali, C.; Ma, T.; Khedr, H.; Radle, R.; Rolland, C.; Gustafson, L.; et al. SAM 2: Segment Anything in Images and Videos. *arXiv* **2024**, https://doi.org/10.48550/arXiv.2408.00714.
14. Xiong, Y.; Varadarajan, B.; Wu, L.; Xiang, X.; Xiao, F.; Zhu, C.; Dai, X.; Wang, D.; Sun, F.; Iandola, F.; et al. EfficientSAM: Leveraged Masked Image Pretraining for Efficient Segment Anything. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2024; pp. 16111–16121.
15. Guo, X.; Lao, J.; Dang, B.; Zhang, Y.; Yu, L.; Ru, L.; Zhong, L.; Huang, Z.; Wu, K.; Hu, D.; et al. SkySense: A Multi-Modal Remote Sensing Foundation Model Towards Universal Interpretation for Earth Observation Imagery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2024; pp. 27672–27683.
16. Liu, F.; Chen, D.; Guan, Z.; Zhou, X.; Zhu, J.; Ye, Q.; Fu, L.; Zhou, J. RemoteCLIP: A Vision Language Foundation Model for Remote Sensing. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 1–16. https://doi.org/10.1109/TGRS.2024.3390838.
17. Xiong, Z.; Wang, Y.; Zhang, F.; Stewart, A.J.; Hanna, J.; Borth, D.; Papoutsis, I.; Saux, B.L.; Camps-Valls, G.; Zhu, X.X. Neural Plasticity-Inspired Multimodal Foundation Model for Earth Observation. *arXiv* **2024**, https://doi.org/10.48550/arXiv.2403.15356.
18. Do, M.K.; Han, K.; Lai, P.; Phan, K.T.; Xiang, W. RobSense: A Robust Multi-modal Foundation Model for Remote Sensing with Static, Temporal, and Incomplete Data Adaptability. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2025; pp. 7427–7436.
19. Hu, E.J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W. LoRA: Low-Rank Adaptation of Large Language Models. *arXiv* **2022**, https://doi.org/10.48550/arXiv.2106.09685.
20. Scheibenreif, L.; Mommert, M.; Borth, D. Parameter Efficient Self-Supervised Geospatial Domain Adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2024; pp. 27841–27851.
21. Mai, Z.; Zhang, P.; Tu, C.H.; Chen, H.Y.; Nguyen, Q.H.; Zhang, L.; Chao, W.L. Lessons and Insights from a Unifying Study of Parameter-Efficient Fine-Tuning (PEFT) in Visual Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2025; pp. 14845–14857.
22. Chen, Z.; Duan, Y.; Wang, W.; He, J.; Lu, T.; Dai, J.; Qiao, Y. Vision Transformer Adapter for Dense Predictions. *arXiv* **2023**. https://doi.org/10.48550/arXiv.2205.08534
23. Chen, K.; Liu, C.; Chen, H.; Zhang, H.; Li, W.; Zou, Z.; Shi, Z. RSPrompter: Learning to Prompt for Remote Sensing Instance Segmentation Based on Visual Foundation Model. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 1–17. https://doi.org/10.1109/TGRS.2024.3356074.
24. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031.
25. Cai, Z.; Vasconcelos, N. Cascade R-CNN: Delving Into High Quality Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2018.
26. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollar, P. Focal Loss for Dense Object Detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*; IEEE: New York, NY, USA, 2017.
27. Tian, Z.; Shen, C.; Chen, H.; He, T. FCOS: Fully Convolutional One-Stage Object Detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*; IEEE: New York, NY, USA, 2019.

28. Zhang, S.; Chi, C.; Yao, Y.; Lei, Z.; Li, S.Z. Bridging the Gap Between Anchor-Based and Anchor-Free Detection via Adaptive Training Sample Selection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2020.
29. Feng, C.; Zhong, Y.; Gao, Y.; Scott, M.R.; Huang, W. TOOD: Task-Aligned One-Stage Object Detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, IEEE: New York, NY, USA, 2021; pp. 3510–3519.
30. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. YOLOX: Exceeding YOLO Series in 2021. *arXiv* **2021**, https://doi.org/10.48550/arXiv.2107.08430.
31. Wang, C.Y.; Yeh, I.H.; Liao, H.Y.M. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. *arXiv* **2024**, https://doi.org/10.48550/arXiv.2402.13616.
32. Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J.; Ding, G. YOLOv10: Real-Time End-to-End Object Detection. *arXiv* **2024**, https://doi.org/10.48550/arXiv.2405.14458.
33. Ultralytics. Ultralytics YOLO11, 2024. Official model documentation. Available online: https://docs.ultralytics.com/models/yolo11/ (accessed on 7 October 2026).
34. Jocher, G.; Qiu, J.; Liu, M.; Lyu, S.; Akyon, F.C.; Kalfaoglu, M.E. Ultralytics YOLO26: Unified Real-Time End-to-End Vision Models. *arXiv* **2026**, https://doi.org/10.48550/arXiv.2606.03748.
35. Tian, Y.; Ye, Q.; Doermann, D. YOLOv12: Attention-Centric Real-Time Object Detectors. *arXiv* **2025**, https://doi.org/10.48550/arXiv.2502.12524.
36. Liu, Y.; Yang, Y.; Li, X.; Yang, F.; Xie, H.; Wang, W.; Dong, C. A Deep Learning-Based Pipeline for Detecting Rip Currents from Satellite Imagery. *Remote Sens.* **2026**, *18*, 368. https://doi.org/10.3390/rs18020368
37. Zhang, J.; Song, X.; Li, Y.; Liang, D.; Zhang, Z.; Cai, J. Adaptive Dual Cross-Attention Network for Multispectral Object Detection in Autonomous Driving. *Expert Syst. Appl.* **2026**, *318*, 132012. https://doi.org/10.1016/j.eswa.2026.132012.
38. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-End Object Detection with Transformers. *arXiv* **2020**, https://doi.org/10.48550/arXiv.2005.12872.
39. Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; Dai, J. Deformable DETR: Deformable Transformers for End-to-End Object Detection. *arXiv* **2021**, https://doi.org/10.48550/arXiv.2010.04159.
40. Liu, S.; Li, F.; Zhang, H.; Yang, X.; Qi, X.; Su, H.; Zhu, J.; Zhang, L. DAB-DETR: Dynamic Anchor Boxes are Better Queries for DETR. *arXiv* **2022**, https://doi.org/10.48550/arXiv.2201.12329.
41. Zhang, H.; Li, F.; Liu, S.; Zhang, L.; Su, H.; Zhu, J.; Ni, L.M.; Shum, H.Y. DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection. *arXiv* **2023**, https://doi.org/10.48550/arXiv.2203.03605.
42. Zhao, Y.; Lv, W.; Xu, S.; Wei, J.; Wang, G.; Dang, Q.; Liu, Y.; Chen, J. DETRs Beat YOLOs on Real-time Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2024; pp. 16965–16974.
43. Lv, W.; Zhao, Y.; Chang, Q.; Huang, K.; Wang, G.; Liu, Y. RT-DETRv2: Improved Baseline with Bag-of-Freebies for Real-Time Detection Transformer. *arXiv* **2024**, https://doi.org/10.48550/arXiv.2407.17140.
44. Peng, Y.; Li, H.; Wu, P.; Zhang, Y.; Sun, X.; Wu, F. D-FINE: Redefine Regression Task in DETRs as Fine-grained Distribution Refinement. *arXiv* **2024**, https://doi.org/10.48550/arXiv.2410.13842.
45. Huang, S.; Lu, Z.; Cun, X.; Yu, Y.; Zhou, X.; Shen, X. DEIM: DETR with Improved Matching for Fast Convergence. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); IEEE: New York, NY, USA, 2025; pp. 15162–15171.
46. Ho, J.; Jain, A.; Abbeel, P. Denoising Diffusion Probabilistic Models. *arXiv* **2020**, https://doi.org/10.48550/arXiv.2006.11239.
47. Song, J.; Meng, C.; Ermon, S. Denoising Diffusion Implicit Models. *arXiv* **2021**, https://doi.org/10.48550/arXiv.2010.02502.
48. Chen, S.; Sun, P.; Song, Y.; Luo, P. DiffusionDet: Diffusion Model for Object Detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*; IEEE: New York, NY, USA, 2023; pp. 19830–19843.
49. Peng, Z.; Xu, Z.; Zeng, Z.; Xie, L.; Tian, Q.; Shen, W. Parameter Efficient Fine-tuning via Cross Block Orchestration for Segment Anything Model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2024; pp. 3743–3752.
50. Steitz, J.M.O.; Roth, S. Adapters Strike Back. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2024; pp. 23449–23459.
51. Mercea, O.B.; Gritsenko, A.; Schmid, C.; Arnab, A. Time- Memory- and Parameter-Efficient Visual Adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2024; pp. 5536–5545.
52. Lei, C.; Li, A.; Yao, H.; Zhu, C.; Zhang, L. Rethinking Token Reduction with Parameter-Efficient Fine-Tuning in ViT for Pixel-Level Tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2025; pp. 14954–14964.

53. Kuckreja, K.; Danish, M.S.; Naseer, M.; Das, A.; Khan, S.; Khan, F.S. GeoChat: Grounded Large Vision-Language Model for Remote Sensing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2024; pp. 27831–27840.
54. Zhu, Q.; Lao, J.; Ji, D.; Luo, J.; Wu, K.; Zhang, Y.; Ru, L.; Wang, J.; Chen, J.; Yang, M.; et al. SkySense-O: Towards Open-World Remote Sensing Interpretation with Vision-Centric Visual-Language Modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2025; pp. 14733–14744.
55. Li, K.; Liu, R.; Cao, X.; Bai, X.; Zhou, F.; Meng, D.; Wang, Z. SegEarth-OV: Towards Training-Free Open-Vocabulary Segmentation for Remote Sensing Images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2025; pp. 10545–10556.
56. Cheng, T.; Song, L.; Ge, Y.; Liu, W.; Wang, X.; Shan, Y. YOLO-World: Real-Time Open-Vocabulary Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2024; pp. 16901–16911.
57. Liu, S.; Zeng, Z.; Ren, T.; Li, F.; Zhang, H.; Yang, J.; Jiang, Q.; Li, C.; Yang, J.; Su, H.; et al. Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection. *arXiv* **2024**, https://doi.org/10.48550/arXiv.2303.05499.
58. Wang, A.; Liu, L.; Chen, H.; Lin, Z.; Han, J.; Ding, G. YOLOE: Real-Time Seeing Anything. *arXiv* **2025**, https://doi.org/10.48550/arXiv.2503.07465.
59. Tang, D.; Cao, X.; Wu, X.; Li, J.; Yao, J.; Bai, X.; Jiang, D.; Li, Y.; Meng, D. AeroGen: Enhancing Remote Sensing Object Detection with Diffusion-Driven Data Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2025; pp. 3614–3624.
60. Perez, E.; Strub, F.; de Vries, H.; Dumoulin, V.; Courville, A. FiLM: Visual Reasoning with a General Conditioning Layer. *arXiv* **2018**, https://doi.org/10.48550/arXiv.1709.07871.
61. Lin, T.Y.; Maire, M.; Belongie, S.; Bourdev, L.; Girshick, R.; Hays, J.; Perona, P.; Ramanan, D.; Zitnick, C.L.; Dollar, P. Microsoft COCO: Common Objects in Context. *arXiv* **2014**, https://doi.org/10.48550/arXiv.1405.0312.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.