J. Vis. Commun. Image R. 119 (2026) 104869

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Full Length Article

LARANet: A lightweight defect detection network with reinforcement-guided perception and energy-aware statistical attention☆

Yuhan Shao [a], Yu Zhou [a,*], Taolue Chen [b], Zixuan Li [c]

[a] College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China
[b] School of Computing and Mathematical Sciences, Birkbeck, University of London, London, United Kingdom
[c] College of Chemistry and Chemical Engineering, Xiamen University, Xiamen, China

ARTICLE INFO

Keywords:
Defect detection
Lightweight object detection
Multi-agent reinforcement learning
Attention mechanism
Cross-scene generalization

ABSTRACT

Defect detection under complex visual conditions remains challenging due to irregular geometries, weak textures, and spatial sparsity while satisfying efficiency constraints. To address these challenges, we propose LARANet (Lightweight Attention and Reinforcement-enhanced Network), a lightweight defect detection framework that unifies compact modeling, adaptive perception, and robust localization. Specifically, LARANet integrates: (i) a RepGhost lightweight backbone for efficient semantic encoding at reduced computational cost; (ii) a Collaborative Reinforcement Perception Module (CRPM), where multi-agent Actor–Critic policies guided by dual GRUs dynamically expand the receptive field to capture defect-relevant cues; (iii) an Energy-Aware Statistical Attention Module (ESAM), which fuses multi-scale statistical features with edge-aware responses to enhance subtle texture and boundary perception; and (iv) an Adaptive Weighted IoU (AWIoU) loss for stable localization across varied object shapes and scales. Experiments on AircraftDefect-5 and two public benchmarks (NEU-DET and IDID) show that LARANet surpasses state-of-the-art lightweight detectors in mAP, small-defect recognition, and generalization across scenes.

1. Introduction

Modern automated inspection increasingly demands defect detectors that deliver reliable performance across diverse environments and imaging conditions, ranging from industrial components to outdoor systems subject to dynamic illumination and sensor noise [1–3]. Beyond accuracy, these models must achieve computational efficiency under resource constraints, remain robust to small and irregular defects, and generalize effectively across domain shifts in camera setups or production processes with minimal annotation overhead [4,5].

Early defect detection relied on manual visual inspection and non-destructive testing or process-monitoring methods, including X-ray computed tomography, laser-based monitoring, and terahertz spectroscopy/imaging [6–8]. While effective for targeted evaluation, these approaches depend on specialized instrumentation and expert interpretation, limiting scalability and consistency. The advent of machine vision and handcrafted feature learning improved automation [9], yet shallow representations and limited tolerance to background variation restricted adaptability in complex scenarios.

Deep learning has substantially advanced defect detection by enabling end-to-end hierarchical representation learning. YOLO-based and other deep detectors have achieved strong accuracy–efficiency trade-offs in industrial inspection [10,11]. Recent developments incorporate attention mechanisms, multi-scale fusion, and limited-label learning to enhance robustness under challenging conditions; contrastive and self-supervised industrial inspection studies further seek to reduce annotation dependence for pipeline and magnetic-flux-leakage defect scenarios [12,13]. For instance, Lu et al. proposed WSS-YOLO for steel surface defect detection, improving robustness under complex surface conditions [14]. Liu et al. developed a real-time anchor-free detector with global-local feature enhancement to better handle cluttered backgrounds, small defects, and irregular shapes [15]. Meanwhile, Sun et al. introduced SDD-DETR, a transformer-based framework tailored for aero-engine blade surface inspection [16], further underscoring the strong potential of DETR-style architectures in industrial defect detection. Comparative studies of lightweight detectors for aircraft inspection [17] and benchmarking efforts across industrial datasets [18] highlight the need for stronger robustness against complex geometries and domain shifts.

Despite these advances, most existing approaches address such challenges in a decoupled or static manner. Lightweight detectors commonly pursue efficiency through aggressive architectural simplification

☆ This paper has been recommended for acceptance by Junsong Yuan.

* Corresponding author.
E-mail address: zhouyu@nuaa.edu.cn (Y. Zhou).

https://doi.org/10.1016/j.jvcir.2026.104869
Received 3 April 2026; Received in revised form 28 May 2026; Accepted 8 June 2026
Available online 11 June 2026

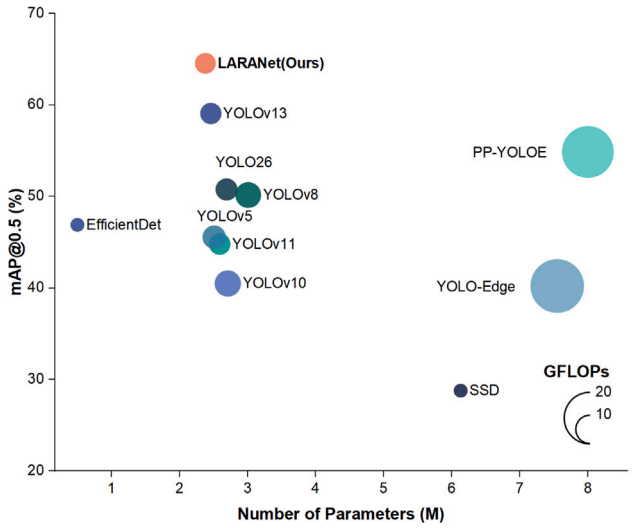

Fig. 1. Performance–complexity comparison among detection models on AircraftDefect-5. The x-axis denotes parameter count (M), the y-axis represents mAP@0.5 (%), and bubble diameter corresponds to GFLOPs.

and static compression strategies. However, this typically compromises contextual modeling capacity, resulting in diminished robustness under complex textures and domain shifts. Attention-based and transformer-driven models improve feature selectivity, yet their receptive fields and focus patterns become fixed after training, limiting adaptability to spatially sparse, irregular, or scene-dependent defect distributions. Moreover, conventional bounding-box regression losses are primarily designed for regular object shapes and scale consistency, making them less effective when handling defects with ambiguous boundaries and irregular geometries. Consequently, several key challenges remain in intelligent defect detection:

1. Severe scale variation and cluttered backgrounds hinder reliable detection of small or low-contrast defects;
2. Ambiguous boundaries, irregular shapes, and high inter-class similarity reduce feature discriminability;
3. Imbalanced or scarce defect samples degrade generalization and increase overfitting risk;
4. The trade-off between accuracy and efficiency constrains deployment in resource-limited and domain-variant scenarios.

Collectively, these limitations indicate that existing lightweight or attention-based detectors alone are insufficient to simultaneously address spatial sparsity, geometric irregularity, and cross-scene variability under strict efficiency constraints. To overcome these issues, we propose LARANet, a lightweight detection framework designed for efficient and robust defect recognition across diverse scenarios, as illustrated in Fig. 1. Specifically:

- A RepGhost lightweight backbone for efficient semantic representation with reduced parameters and FLOPs;
- A CRPM that introduces multi-agent reinforcement learning to actively explore feature space and adaptively focus on defect-relevant regions, enabling dynamic spatial perception beyond static attention;
- An ESAM, derived from a SimAM variant, to enhance texture discrimination and boundary sensitivity under complex surface patterns;
- An AWIoU loss, inspired by adaptive box-scaling ideas, to enable robust localization for defects with irregular shapes and ambiguous boundaries.

The main contributions of this paper are summarized as follows:

1. We propose LARANet, a lightweight defect detection network built on a YOLOv13 baseline, achieving robust performance across diverse industrial inspection scenarios while balancing accuracy and efficiency.
2. We integrate four complementary modules to improve compactness, representation, and optimization stability, especially introducing a reinforcement learning-driven CRPM for adaptive spatial perception, together with a RepGhost lightweight backbone, ESAM, and an AWIoU loss.
3. Extensive experiments on AircraftDefect-5, NEU-DET, and IDID demonstrate that LARANet consistently improves mAP and maintains strong precision–recall performance, while achieving favorable trade-offs in complexity, speed, and robustness across diverse domains.

2. Related work

2.1. Modern defect detection

Modern defect detection has rapidly advanced through deep learning-based visual frameworks, enabling end-to-end automation and broad adaptability across industrial domains [19]. Recent supervised and transformer-based detectors improve proposal refinement, query-based defect modeling, and tiny-object sensitivity through feature pyramids, attention, and spectral enhancement [20–22]. Lightweight YOLO-family designs support real-time deployment, while Swin-based and other hybrid models strengthen contextual reasoning under illumination and texture variations [11,23].

To reduce labeling demands, contrastive and self-supervised paradigms have been explored for intelligent industrial inspection, complementing supervised pipelines when annotations are scarce [12,13]. Combined with architectural advances and lightweight optimization, these strategies yield accurate, efficient, and robust systems for large-scale industrial inspection [4,5].

2.2. Evolution of YOLO detectors

The YOLO series reformulates detection as a unified one-pass regression task, balancing speed and accuracy for industrial applications [11]. Recent YOLO-based studies emphasize lightweight fusion and architectural refinement: YOLOv8 variants have been adapted for PCB and industrial defect detection [24], while YOLOv9/10 variants further explore advanced backbones and context-aware adaptive modules to enhance small-object detection and robustness [25,26].

Subsequent studies extend the YOLO line to YOLOv11, YOLOv12, and YOLOv13 variants, emphasizing efficient attention, re-parameterized modules, and benchmarking across application domains [27–30]. Building on this progression, LARANet integrates a RepGhost lightweight backbone, a multi-agent CRPM for active spatial exploration, and an ESAM for defect-focused enhancement to strengthen precision and robustness across diverse inspection scenarios.

2.3. Multi-agent reinforcement learning

Reinforcement learning (RL) learns optimal policies through interaction, while multi-agent RL (MARL) extends this paradigm to cooperative agents operating under partial observability [31]. A common formalism is the Dec-POMDP with state S, joint action A, transition T, reward R, and discount γ [32]. Typical paradigms include CTCE, CTDE, and DTDE, with DTDE enabling flexible local coordination through message passing [33]. Surveys further highlight MARL's role in resource allocation optimization [34], while recent advances explore theory-of-mind inspired reasoning models for adaptive partner modeling [35].

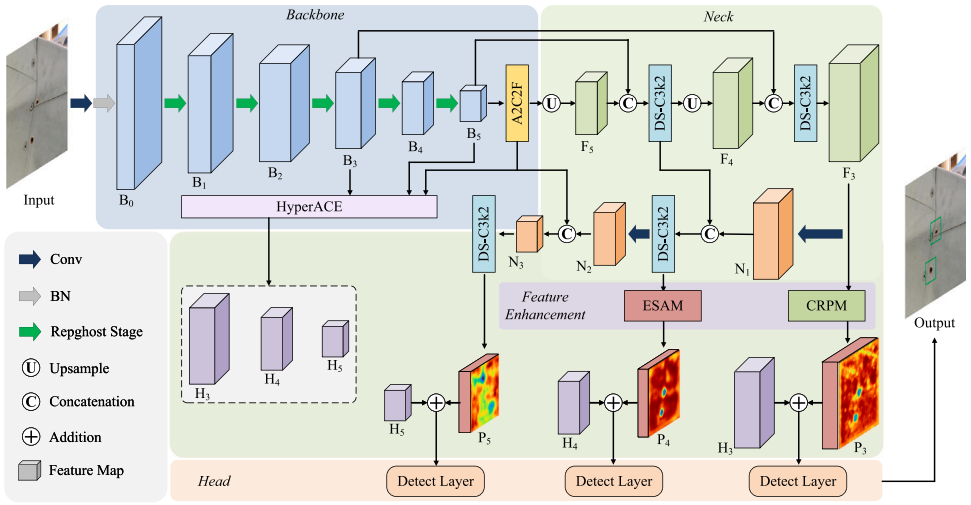

Fig. 2. Overall architecture of the proposed LARANet.

In visual perception, MARL supports active attention and collaborative exploration [31]. Our CRPM adopts DTDE with a dual-GRU design for temporal modeling and inter-agent communication: agents iteratively act, adjust positions, and build a visitation map that generates an adaptive attention mask. CRPM is trained with a composite reward that balances classification accuracy and spatial focus [31,33].

3. Methodology

In this section, we first provide an overview of the proposed defect detector. We then discuss the details of the RepGhost lightweight backbone, CRPM, and ESAM. Finally, we present the AWIoU loss.

3.1. Overall architecture

The overall architecture of LARANet is illustrated in Fig. 2. The framework follows the standard backbone-neck-head paradigm, where the backbone extracts multi-scale features, the neck fuses representations, and the head produces detection outputs. LARANet inherits the multi-scale aggregation strategy of YOLOv13 while integrating two novel modules – CRPM and ESAM – at the P3 and P4 layers to enable reinforcement-driven spatial exploration and texture-aware enhancement within the feature pyramid.

For feature extraction, the RepGhost lightweight backbone delivers efficient semantic representation via a lightweight, structurally re-parameterized design. At the middle- and high-level stages, CRPM introduces multi-agent reinforcement learning to actively focus on defect-related regions and improve contextual awareness. ESAM is then applied in the fusion stage, where an improved SimAM attention mechanism enhances sensitivity to defect boundaries and fine-grained textures while suppressing background noise. Finally, we employ the AWIoU loss in the detection head to stabilize bounding-box regression and improve localization accuracy.

Overall, LARANet achieves a compact design with high inference efficiency while delivering collaborative gains in accuracy, robustness, and generalization. The integration of re-parameterized representation learning, reinforcement-driven perception, and adaptive attention contributes to resilient performance across varied materials, textures, and imaging conditions.

3.2. RepGhost lightweight backbone

In industrial surface-defect detection, YOLOv13 exhibits strong feature-extraction capability. However, its deep hierarchical backbone architecture entails considerable parameter overhead, imposing a heavy computational burden on resource-constrained real-time systems. To balance accuracy and lightweight performance, LARANet integrates the RepGhost lightweight backbone, which leverages structural re-parameterization for implicit feature reuse, thereby reducing computation and memory access costs without degrading representational capacity. The structure is shown in Fig. 3. RepGhost transfers feature reuse from the feature space to the weight space. During training, multiple branch transformations are introduced; during inference, they are linearly fused into a single convolution, enabling implicit feature reuse:

$$Y_{\text{rep}} = \text{Add}\big([X, \Phi_1(X), \ldots, \Phi_{s-1}(X)]\big) \;=\; \Phi^*(X), \tag{1}$$

where s is the number of training branches, $\Phi_i(\cdot)$ denotes the ith branch transformation, $X \in \mathbb{R}^{C\times H\times W}$ denotes the input feature map, and $\Phi^*(\cdot)$ denotes the equivalent fused convolution operator. As shown in Eq. (1), the multi-branch responses are added and represented by an equivalent fused operator. The Add operation replaces concatenation to achieve low-latency feature aggregation. A key advantage is that the fusion occurs at the weight level without incurring additional inference delay, while maintaining the expressive power of multi-branch extraction.

Each RepGhost block consists of a linear projection, a convolution, and a depthwise separable branch. The input feature is first convolved to generate a primary feature map X_1, then passed through a depthwise convolution to obtain an auxiliary feature X_2. The final block output is obtained by implicit addition:

$$Y_{\text{blk}} = \text{ReLU}(X_1 + X_2), \qquad X_1 = \text{Conv}(X), \quad X_2 = \text{DWConv}(X_1), \tag{2}$$

where DWConv denotes depthwise convolution and ReLU is the activation function. To satisfy re-parameterization constraints, a Batch Normalization layer is introduced in the identity branch during training, providing nonlinear adaptability during backpropagation. In inference, this layer is fused, resulting in:

$$Y_{\text{inf}} = \text{ReLU}\big(W^* * X + b^*\big), \tag{3}$$

where $*$ denotes convolution, and W^* and b^* represent the equivalent fused kernel and bias, respectively. This fusion merges all branches in the weight space into a single equivalent convolution, yielding a compact inference topology.

Within the YOLOv13 backbone, the original feature extraction module is replaced by a RepGhost Bottleneck, which contains two sequential RepGhost blocks: the first compresses channels while abstracting spatial features, and the second expands channels through dilated convolution to enhance expressiveness. An SE (Squeeze-and-Excitation) module is integrated in the downsampling path to suppress redundant

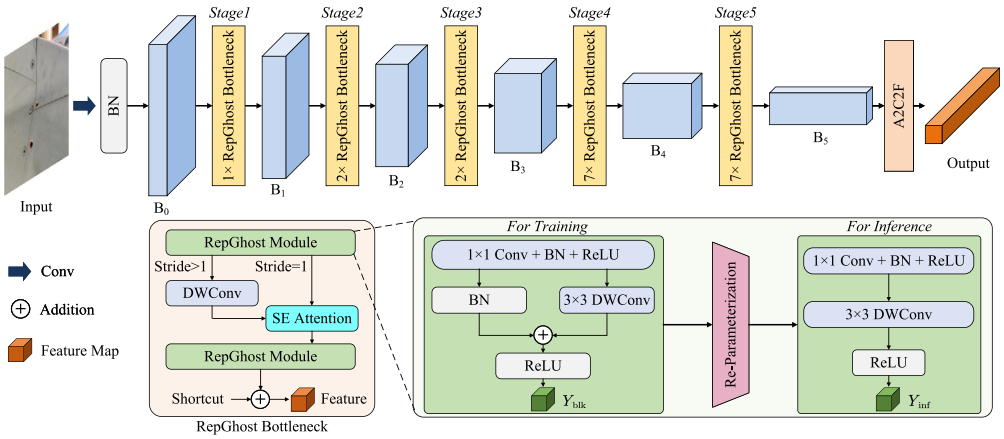

Fig. 3. Structure of the RepGhost lightweight backbone.

responses, and skip connections are employed to stabilize gradients. The intermediate channel number is set as:

$$C_{\text{mid}} = \frac{1}{2} C_{\text{out}}, \tag{4}$$

and the residual connection is used when $C_{\text{in}} = C_{\text{out}}$, thereby preserving feature integrity while compressing parameters.

To ensure compatibility with the YOLOv13 architecture, LARANet retains the original multi-scale output layers (P3–P5) and controls the overall channel width with a scaling coefficient κ:

$$C_i' = \kappa \times C_i, \qquad \kappa \in \{0.5,\ 1.0,\ 1.3\}. \tag{5}$$

Here, i indexes the multi-scale stages and C_i denotes the original channel width of stage i. This scaling design allows flexible trade-offs between inference speed and detection accuracy according to task complexity.

3.3. CRPM

After extracting mid-level semantic features, we introduce a reinforcement learning-driven CRPM, as illustrated in Fig. 4, with two representative agents shown to highlight cooperative perception and interaction. At its core, CRPM employs a multi-agent cooperative decision mechanism to carry out dynamic perception and action sampling within the feature space, thereby constructing a learnable spatial-exploration strategy. Through iterative interaction during training, agents progressively transition from coarse exploration to fine-grained defect-focused behavior.

Let the mid-level feature map output from the backbone be $I_f \in \mathbb{R}^{B\times C_f\times H\times W}$, where C_f denotes the channel dimension, and let N agents be indexed by $i \in \{1, \ldots, N\}$. The interaction process lasts for T time steps, and the trajectory sampling count C specifies the number of sampled policy trajectories used to estimate rewards and returns during training. The position of the ith agent at time step t is denoted as $p_i(t) \in [0,1]^2$. The window size w is a configurable parameter, and the single-step displacement is proportional to w and normalized by the feature-map dimensions. A discrete action set $\mathcal{A}$ is defined, consisting of nine actions: eight directional shifts and one stationary action. At each time step, the ith agent samples a movement action $a_i(t) \in \mathcal{A}$ from its policy distribution to determine its movement direction. A local observation operator $\mathcal{O}(\cdot)$ extracts a region $o_i(t)$ from I_f around $p_i(t)$, which is then encoded by the observation encoder $\Phi_{\text{obs}}(\cdot)$ into a feature vector $f_i(t)$:

$$o_i(t) = \mathcal{O}(I_f, p_i(t), w), \quad f_i(t) = \Phi_{\text{obs}}(o_i(t)), \tag{6}$$

and the normalized coordinates are mapped into a d_p-dimensional embedding through the position encoder $\Phi_{\text{pos}}(\cdot)$:

$$q_i(t) = \Phi_{\text{pos}}(p_i(t)) = W_p p_i(t) + b_p, \tag{7}$$

where W_p and b_p denote the corresponding projection weights and biases.

To visualize the perceptual process of multiple agents, Fig. 5 illustrates local observation and motion updates. The ith agent moves one step rightward from time t to $t+1$, while the $(i+1)$th agent moves diagonally left and down; the step length is proportional to the observation window.

Building on this interaction process, we address the common concern that reinforcement learning can incur high training costs and inference overhead. CRPM is designed to mitigate these issues. Specifically, interaction is confined to mid-level feature maps with few agents and discrete actions, avoiding external environment rollouts or pixel-level sampling. Feature transformations are optimized end-to-end with the detection network, while discrete actions are trained through the policy objective without extra inference loops. At inference, CRPM reduces to deterministic attention modulation with learned parameters and incurs negligible overhead compared to standard convolutional attention. To further stabilize and accelerate training, we adopt an Actor–Critic framework with Generalized Advantage Estimation (GAE), reducing variance and enabling efficient credit assignment over short trajectories.

The module further adopts a dual-GRU architecture that explicitly separates prediction and decision processes. The Prediction GRU aggregates local observations and inter-agent messages to form a stable environmental representation, while the Decision GRU generates the policy distribution and updates positions. This decoupling ensures clear functional separation between evidence accumulation and action selection, improving training stability. Inter-agent communication is implemented by a function $\Phi_{\text{comm}}(\cdot)$, through which each agent broadcasts a message based on its predictive state $h_i(t-1)$ and decision state $\hat{h}_i(t-1)$. A self-decoupled averaging scheme is then applied to suppress self-reinforcement and enhance cooperative diversity:

$$m_i(t) = \Phi_{\text{comm}}(h_i(t-1), \hat{h}_i(t-1)), \tag{8}$$

$$m_i^{\text{avg}}(t) = \begin{cases} \dfrac{1}{N-1}\displaystyle\sum_{j\neq i} m_j(t), & N > 1, \\ \mathbf{0}, & N = 1. \end{cases} \tag{9}$$

The Prediction GRU updates the hidden environmental state by integrating temporal observations and communication messages:

$$h_i(t) = G_{\text{pred}}(h_i(t-1), x_i(t)), \tag{10}$$

$$x_i(t) = [f_i(t), m_i^{\text{avg}}(t), q_i(t)]. \tag{11}$$

The Decision GRU fuses the projected prediction output $g_i(t) = W_g h_i(t) + b_g$ and inter-agent messages to generate an actionable policy representation:

$$\hat{h}_i(t) = G_{\text{dec}}(\hat{h}_i(t-1), [g_i(t), m_i^{\text{avg}}(t)]), \tag{12}$$

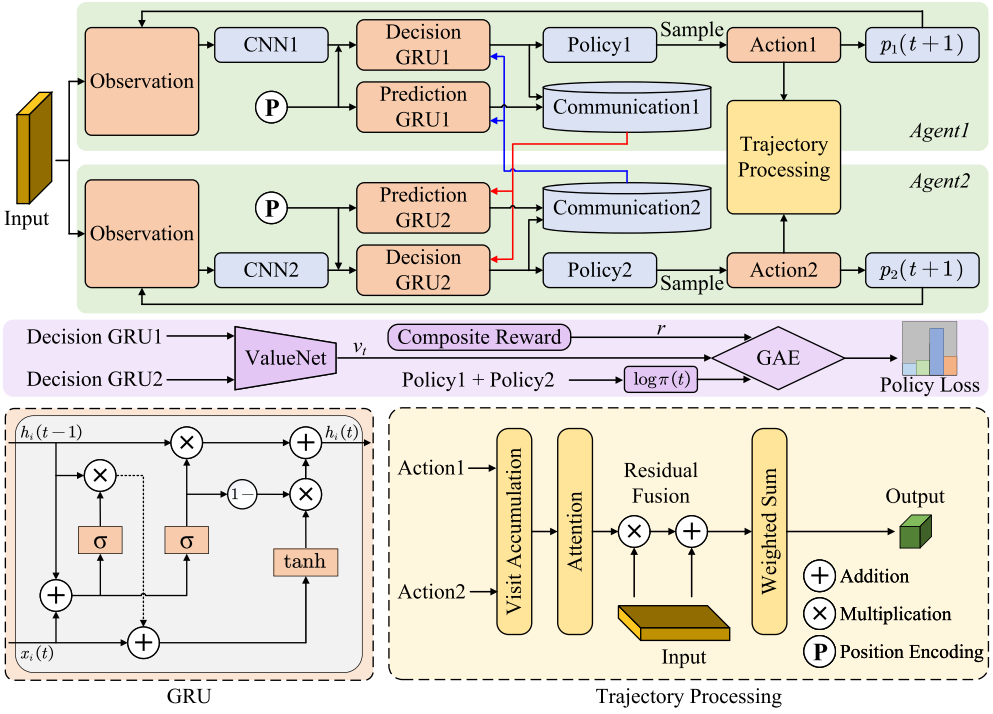

Fig. 4. Structure of the CRPM. Two representative agents are illustrated to show cooperative reinforcement perception and interaction.

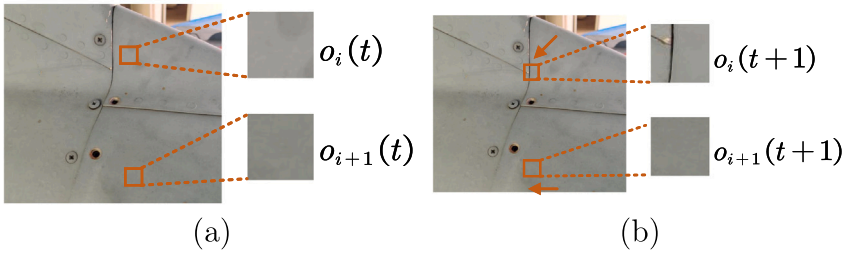

Fig. 5. Illustration of local observation in the CRPM: (a) Observation windows of agents i and $(i+1)$ at time step t; (b) Updated observations at time $t+1$.

$$\mathbf{z}_i(t) = W_\pi \hat{h}_i(t) + b_\pi, \tag{13}$$

where W_g, b_g, W_π, and b_π are learnable parameters. Given a temperature coefficient τ_t, the policy network produces a discrete probability distribution over actions:

$$\boldsymbol{\pi}_i(t) = \mathrm{Softmax}\left(\frac{\mathbf{z}_i(t)}{\tau_t}\right), \quad \pi_i^{(k)}(t) = \frac{\exp(z_i^{(k)}(t)/\tau_t)}{\sum_{\ell=1}^{|\mathcal{A}|}\exp(z_i^{(\ell)}(t)/\tau_t)}. \tag{14}$$

Let $\{\Delta_a\}_{a=1}^{|\mathcal{A}|}$ represent the displacement dictionary with $\Delta_a \in \{-1,0,1\}^2$, β_s the step-scaling coefficient, and the normalization factor $[1/W, 1/H]$. The step size and updated agent position are then:

$$s = \beta_s \cdot w \cdot \left[\frac{1}{W}, \frac{1}{H}\right], \tag{15}$$

$$p_i(t+1) = \mathrm{clip}(p_i(t) + s \odot \Delta_{a_i(t)}, [0,1]^2). \tag{16}$$

As shown in Eqs. (15) and (16), the motion magnitude is tied to the observation window and normalized by the feature-map size, keeping agent updates scale-aware across pyramid resolutions. Throughout the agent's trajectory, each visited location contributes to a visit map $M \in \mathbb{R}^{B\times 1\times H\times W}$, which is normalized and sharpened to produce the visit-based attention map A_{visit}:

$$\hat{M} = \frac{M}{\max(M)+\varepsilon}, \quad A_{\text{visit}} = \mathrm{Sigmoid}\left(\gamma_v(\hat{M} - 0.5)\right), \tag{17}$$

where $\gamma_v = 3.0$ is the temperature parameter. Finally, the input feature is modulated through a residual mechanism during fusion:

$$Y_{\text{CRPM}} = I_f \odot (1 + \alpha_t A_{\text{visit}}), \tag{18}$$

where α_t is a linearly scheduled warm-up coefficient that increases from 0 to 1 over the first T_{warm} training epochs (set to 3), enabling a gradual activation of the RL branch. All network parameters remain trainable during training.

For reinforcement optimization, CRPM employs an Actor–Critic framework with GAE. The terminal detection reward is defined as:

$$r_{\text{det}} = \alpha \cdot (-L_{\text{det}}) + (1-\alpha) \cdot \text{acc}. \tag{19}$$

In Eq. (19), $\alpha \in [0,1]$ balances detection loss and accuracy in the CRPM reward; L_{det} denotes the normalized detection loss, and acc denotes the mini-batch detection accuracy proxy. Both L_{det} and acc are detached from the computation graph when constructing the reward, ensuring that no gradients are backpropagated through this path. An intrinsic penalty encourages exploration:

$$r_{\text{int}}(t) = -\frac{\eta}{N}\sum_{i=1}^{N}\text{Visited}(p_i(t)), \tag{20}$$

where η denotes the intrinsic penalty against revisits, and Visited$(\cdot)$ indicates whether a location has been previously visited.

With value prediction $V_\theta(t)$, discount factor ρ, GAE coefficient λ_{GAE}, and terminal return $R_T = r_{\text{int}}(T-1) + r_{\text{det}}$ for a trajectory of length T,

the temporal difference and advantage terms are given by:

$$\delta(t) = \begin{cases} r_{\text{int}}(t) + \rho\, V_\theta(t+1) - V_\theta(t), & t < T-1, \\ R_T - V_\theta(t), & t = T-1, \end{cases} \tag{21}$$

$$A(t) = \delta(t) + \rho\lambda_{\text{GAE}} A(t+1), \quad A(T) = 0. \tag{22}$$

The return target is $\bar{R}(t) = \text{sg}(A(t) + V_\theta(t))$, where $\text{sg}(\cdot)$ denotes stop-gradient. The advantage $A(t)$ is standardized and clipped to yield $\hat{A}(t)$. The policy loss, aggregated over time steps, is defined as:

$$\mathcal{L}_\pi = -\mathbb{E}\left[\sum_t \left(\sum_i \log \pi_i^{(a_i(t))}(t) \cdot \hat{A}(t)\right)\right] - \lambda_{\text{ent}}(t)\mathbb{E}[\mathcal{H}(\boldsymbol{\pi}(t))], \tag{23}$$

where $\mathcal{H}(\boldsymbol{\pi}(t)) = -\sum_i \sum_a \pi_i^{(a)}(t) \log \pi_i^{(a)}(t)$ denotes the policy entropy, and $\lambda_{\text{ent}}(t)$ is the cosine-decayed entropy-regularization weight. The value-function loss is formulated as a mean squared error:

$$\mathcal{L}_{\text{value}} = \mathbb{E}\left[\sum_t \|V_\theta(t) - \bar{R}(t)\|_2^2\right]. \tag{24}$$

The expectations in the policy and value losses are estimated over the C sampled trajectories and the mini-batch. The final optimization objective is expressed as:

$$\mathcal{L}_{\text{CRPM}} = \mathcal{L}_\pi + \mathcal{L}_{\text{value}}. \tag{25}$$

The policy branch follows the RL objective, whereas the detection path follows the supervised loss; reward metrics are detached to avoid gradient leakage.

Collectively, CRPM integrates an Actor–Critic framework with GAE, temperature and entropy scheduling, reinforcement warm-up, and composite rewards under a multi-agent paradigm. Coupled with dual-GRU decoupling and residual attention, it links spatial exploration with feature focusing through a compact optimization loop, thereby improving detection performance in complex textures and low-contrast defect regions.

3.4. ESAM

In industrial surface defect detection, targets frequently exhibit significant scale variations, indistinct boundaries, and bimodal size distributions. To address these challenges, we propose ESAM. ESAM extends the SimAM module [36] by introducing energy-aware variance modeling and edge-aware modulation, enabling more discriminative attention responses to defects with complex textures and ambiguous boundaries. The overall architecture is illustrated in Fig. 6.

The SimAM module computes neuron-wise attention weights based on a linear separability criterion:

$$e_t^* = \frac{4(\hat{\sigma}^2 + \lambda)}{(x_t - \hat{\mu})^2 + 2\hat{\sigma}^2 + 2\lambda}, \tag{26}$$

$$X_t' = X_t \odot \text{Sigmoid}\left(\frac{1}{E}\right), \tag{27}$$

where x_t denotes the value of a single neuron, X_t the spatial feature map, $\hat{\mu}$ and $\hat{\sigma}^2$ the mean and variance within the same channel, λ a regularization term, and Sigmoid constrains the output range. Here, E denotes the energy map assembled from e_t^*. However, due to complex surface textures and frequent high-frequency noise in industrial defect images, the uniform-distribution assumption of SimAM across spatial regions limits its responsiveness to fine-grained local patterns. ESAM enhances the original SimAM from three perspectives to improve sensitivity to multi-scale and edge-defect characteristics.

First, ESAM fuses global and local variances with adaptive weighting. Given an input feature map $X \in \mathbb{R}^{C\times H\times W}$, the global variance is computed channel-wise:

$$\mu_g(c) = \frac{1}{HW}\sum_{i,j} X_{cij}, \quad V_g(c,i,j) = (X_{cij} - \mu_g(c))^2. \tag{28}$$

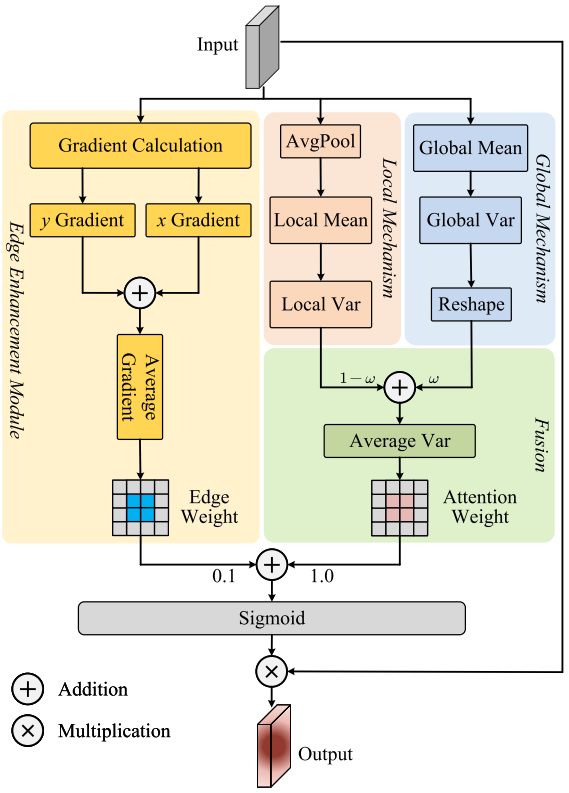

Fig. 6. Structure of the ESAM.

The local variance is computed via a 3×3 average pooling operation:

$$\mu_l = \text{AvgPool}_{3\times3}(X), \quad V_l = (X - \mu_l)^2. \tag{29}$$

The fused representation is then expressed as:

$$V_f = \omega V_g + (1-\omega)V_l. \tag{30}$$

As shown in Eq. (30), the ESAM variance-fusion weight $\omega = 0.7$ balances global (large-object) and local (fine-texture) sensitivities. This design enables responsiveness to both small-scale defects and large structural patterns in datasets with bimodal object-size distributions.

In addition, ESAM incorporates a defect-sensitive adaptive normalization strategy to selectively enhance responses of small-scale defects, which account for most instances in the dataset. The spatial mean of the fused variance V_f is computed as:

$$\bar{V}_f(c) = \frac{1}{HW}\sum_{i=1}^{H}\sum_{j=1}^{W} V_f(c,i,j), \tag{31}$$

and the normalized attention weight is formulated as:

$$A_{\text{var}} = \frac{V_f}{3.5(\bar{V}_f + \varepsilon)} + \gamma_0, \tag{32}$$

where $\bar{V}_f$ is broadcast spatially, ε ensures numerical stability, and γ_0 is initialized to 0.6 as a learnable bias term to maintain a stable baseline response during early training. Because γ_0 is optimized end-to-end rather than kept fixed, this value only provides an initial mild positive prior that prevents weak defect responses from being suppressed before stable variance statistics are learned. This normalization amplifies pixels with higher variance, effectively highlighting regions corresponding to small defects.

Finally, to further enhance boundary sensitivity, ESAM integrates a gradient-based edge enhancement module. Horizontal and vertical

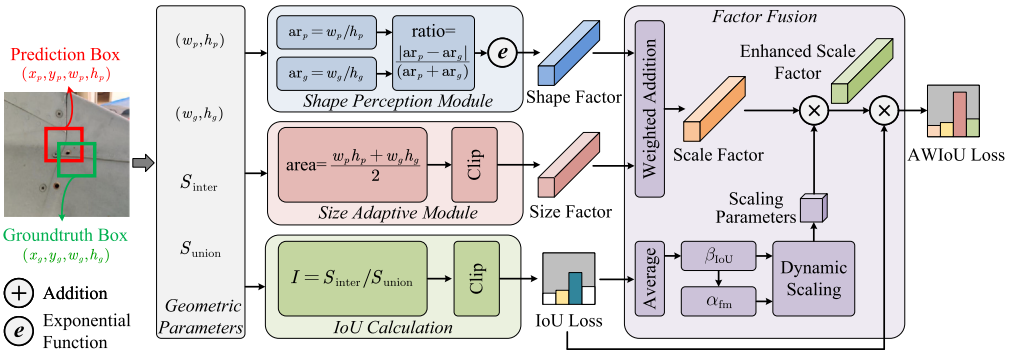

Fig. 7. Flow diagram of the AWIoU loss.

gradients of the feature map are computed as:

$$G_x(i,j) = |X_{:,i,\min(j+1,W)} - X_{:,i,j}|, \quad G_y(i,j) = |X_{:,\min(i+1,H),j} - X_{:,i,j}|, \tag{33}$$

and the edge response is normalized to form:

$$W_{\text{edge}} = \frac{(G_x + G_y)/2}{\mathbb{E}[(G_x + G_y)/2] + \varepsilon}, \tag{34}$$

where $\mathbb{E}[\cdot]$ averages over spatial positions. The final attention map combines the variance-based and edge-based components, producing the module output:

$$Y_{\text{ESAM}} = X \odot \text{Sigmoid}\left(A_{\text{var}} + 0.1 W_{\text{edge}}\right), \tag{35}$$

thereby reinforcing feature responses along high-gradient regions such as crack trajectories and corrosion boundaries. This design preserves a lightweight structure while providing robust enhancement for fine-grained defect structures, further improving the clarity of boundary-related representations.

Taken as a whole, ESAM builds on energy-based attention modeling through global–local variance modeling, adaptive normalization, and edge-aware modulation, strengthening defect localization while preserving computational efficiency.

3.5. AWIoU loss

Industrial surface defects often exhibit diverse shapes and highly imbalanced size distributions, where small targets are prone to vanishing gradients during training. To tackle this challenge, we propose the AWIoU loss function, which extends adaptive box-scaling ideas [37] through shape-aware constraints and scale-adaptive weighting to improve localization stability and geometric robustness. The overall process is illustrated in Fig. 7.

Let (x_p, y_p, w_p, h_p) and (x_g, y_g, w_g, h_g) denote the center coordinates, width, and height of the predicted and ground-truth boxes, respectively. The center and shape distances are defined as:

$$D_{\text{center}} = (x_p - x_g)^2 + (y_p - y_g)^2, \tag{36}$$

$$D_{\text{shape}} = \frac{(w_p - w_g)^2 + (h_p - h_g)^2}{4}. \tag{37}$$

Defining intersection-over-union as $I = S_{\text{inter}}/S_{\text{union}}$ with base loss $\mathcal{L}_{\text{IoU}} = 1 - I$, a dynamic modulation term is introduced to emphasize hard samples:

$$\beta_{\text{IoU}} = \frac{\mathcal{L}_{\text{IoU}}}{m + \varepsilon}, \quad \alpha_{\text{fm}} = \delta_{\text{w}} \gamma_{\text{w}}^{(\beta_{\text{IoU}} - \delta_{\text{w}})}, \quad S_{\text{fm}} = \frac{\beta_{\text{IoU}}}{\alpha_{\text{fm}}}, \tag{38}$$

where $\gamma_{\text{w}} = 1.9$, $\delta_{\text{w}} = 3$, m is the running mean of IoU loss, and ε is a small constant for numerical stability. This term adaptively amplifies gradients for low-IoU samples, facilitating stable hard-sample learning.

To further adapt to varying shapes, the predicted and ground-truth aspect ratios $\text{ar}_p = w_p/h_p$ and $\text{ar}_g = w_g/h_g$ are used to derive shape and scale modulation factors:

$$S_{\text{shape}} = 1 + 0.1\left(1 - \exp\left[-\frac{|\text{ar}_p - \text{ar}_g|}{\text{ar}_p + \text{ar}_g + \varepsilon}\right]\right), \tag{39}$$

$$S_{\text{size}} = \text{clip}\left(\frac{w_p h_p + w_g h_g}{2 \cdot 64^2}, 0.5, 2.0\right). \tag{40}$$

The overall adaptive scaling factor is:

$$S_{\text{adp}} = S_{\text{fm}} \cdot S_{\text{shape}} \cdot S_{\text{size}}. \tag{41}$$

The adaptive weighted IoU-regression term is then:

$$\mathcal{L}_{\text{WIoU}} = S_{\text{adp}} \mathcal{L}_{\text{IoU}}. \tag{42}$$

To enhance geometric consistency, Wasserstein-based size and shape similarity terms are introduced. The Wasserstein size factor S'_{size}, geometric similarity S_{geo}, and enhanced distance D_{enh} are computed as:

$$S'_{\text{size}} = \text{clip}\left(\sqrt{\frac{w_p h_p + w_g h_g}{2 \cdot 32^2}}, 0.1, 3.0\right), \tag{43}$$

$$S_{\text{geo}} = \exp\left(-\frac{|w_p h_p - w_g h_g|}{w_p h_p + w_g h_g + \varepsilon}\right), \tag{44}$$

$$D_{\text{enh}} = D_{\text{center}} + D_{\text{shape}} + 0.5 \cdot \frac{|\text{ar}_p - \text{ar}_g|}{\text{ar}_p + \text{ar}_g + \varepsilon} \cdot S'_{\text{size}} + 0.1(1 - S_{\text{geo}}). \tag{45}$$

This formulation acts as a transport-cost surrogate by jointly modeling spatial displacement and scale-aware shape variation. Based on this, the Wasserstein-based similarity score is defined as:

$$S_{\text{WD}} = \exp\left(-\frac{\sqrt{D_{\text{enh}} + \varepsilon}}{12.8 \cdot \text{clip}(S'_{\text{size}}, 0.5, 2.0)}\right). \tag{46}$$

Here, S_{WD} is not a strict Wasserstein distance, but a heuristic approximation that mimics its behavior by combining center displacement and shape discrepancy. It can be interpreted as an exponential form of transport cost under simplified assumptions, enabling efficient optimization while preserving geometric consistency. Finally, the overall loss is formulated as:

$$\mathcal{L}_{\text{AWIoU}} = \lambda_{\text{AW}} \mathcal{L}_{\text{WIoU}} + (1 - \lambda_{\text{AW}})(1 - S_{\text{WD}}), \quad \lambda_{\text{AW}} = 0.5. \tag{47}$$

In summary, AWIoU integrates adaptive hard-sample reweighting with shape- and scale-aware geometric similarity, following an adaptive box-scaling design to balance gradient contributions across objects of different sizes. This design improves localization accuracy, particularly for small or elongated defects on complex industrial surfaces.

After defining the core modules and loss function, we next evaluate LARANet and its key components through comprehensive experiments.

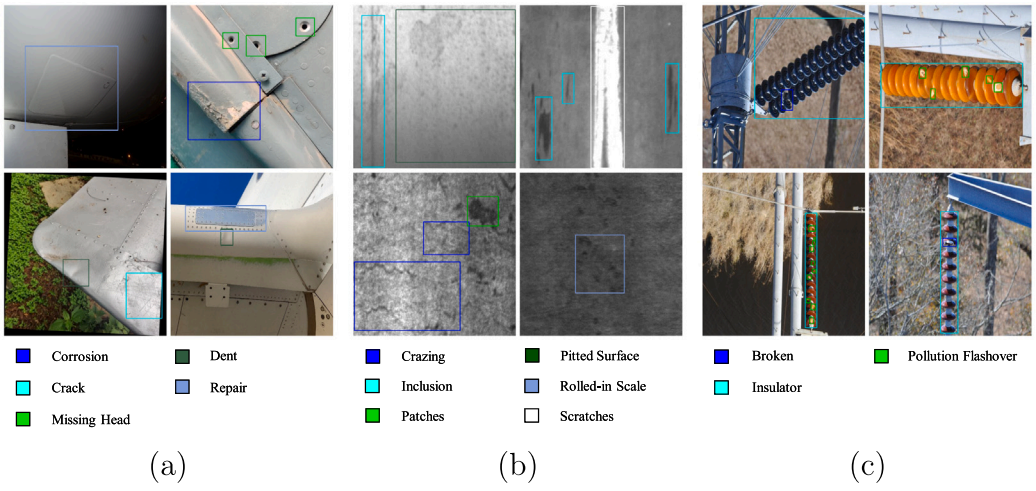

Fig. 8. Sample visualization of datasets: (a) AircraftDefect-5, (b) NEU-DET, (c) IDID.

4. Experiments and analysis

In this section, we conduct extensive experiments on one industrial aircraft-defect dataset (AircraftDefect-5) and two public benchmarks (NEU-DET and IDID) to systematically evaluate the effectiveness and cross-scene generalization capability of the proposed LARANet. We first introduce the experimental setup, followed by comprehensive comparative evaluations, ablation studies, key design analyses, and qualitative visual analyses.

4.1. Experimental setup

4.1.1. Dataset description

To comprehensively evaluate the effectiveness and cross-domain generalization of LARANet, three representative datasets are employed: AircraftDefect-5, NEU-DET, and IDID. These datasets cover diverse materials, textures, and acquisition setups. Since each image may contain multiple defect categories, class statistics are reported by the number of annotated targets. Typical samples are illustrated in Fig. 8. As shown, the datasets exhibit substantial variations in background texture, defect morphology, and scale distribution, providing a diverse evaluation environment for robustness testing. To enhance sample diversity and mitigate overfitting, standard augmentations – including random scaling, brightness adjustment, Gaussian blurring, contrast and saturation perturbation, and random rotation – are applied to AircraftDefect-5 and NEU-DET.

(1) AircraftDefect-5

AircraftDefect-5 is collected from key aircraft structural regions, including wings, fuselage, doors, and tail sections, with an image resolution of 640 × 480. It contains five defect classes: Corrosion, Crack, Missing Head, Repair, and Dent. After augmentation, the training/validation/test splits contain 3515/439/439 images, respectively, with 2872 Corrosion, 1140 Crack, 4761 Missing Head, 1102 Repair, and 1839 Dent instances.

(2) NEU-DET

NEU-DET is adopted as a public benchmark for cross-domain validation on metallic surface textures. It contains six defect classes: Crazing, Inclusion, Patches, Pitted Surface, Rolled-in Scale, and Scratches. After augmentation, the training/validation/test splits contain 2234/279/279 images, respectively, with 965 Crazing, 981 Inclusion, 977 Patches, 969 Pitted Surface, 968 Rolled-in Scale, and 998 Scratches instances.

(3) IDID

IDID is adopted for outdoor power-equipment inspection and is sourced from the publicly available Roboflow platform. It contains three classes: Broken, Insulator, and Pollution Flashover. To reflect natural imaging conditions, weather-style augmentation is applied with rain, snow, fog, haze, and sandstorm effects. The training/validation/test splits contain 1280/160/160 images, respectively, with 1093 Broken, 1832 Insulator, and 2448 Pollution Flashover instances.

To characterize object-size diversity, Fig. 9 plots the width-height distribution of annotated boxes. Targets span small scratches, corrosion patches, and large-scale components, exhibiting balanced coverage without extreme outliers. AircraftDefect-5 contains the most complex small-scale defect instances; NEU-DET emphasizes morphological variability on steel textures; IDID evaluates robustness under outdoor illumination and environmental noise. Together, these datasets form a comprehensive benchmark for assessing industrial defect detection across both indoor and outdoor domains.

Table 1
Main experimental parameters.

Epochs	Learning rate	Momentum	Batch size	Input size	Optimizer
100	0.01	0.937	64	640 × 640	SGD

4.1.2. Experimental environment and configuration

All experiments were conducted on an Ubuntu 22.04 LTS system equipped with an Intel Xeon Gold 5418Y CPU (10 cores) and an NVIDIA RTX 4090 GPU (24 GB VRAM). The implementation was based on PyTorch 2.2.2 with Python 3.10 and CUDA 11.8. The main hyperparameters are summarized in Table 1.

4.1.3. Evaluation metrics

Model performance was assessed from three perspectives: detection accuracy, localization precision, and computational efficiency, reflecting the practical needs of reliable lightweight defect detection in industrial settings. The following metrics were adopted:

$$P = \frac{TP}{TP + FP}, \tag{48}$$

$$R = \frac{TP}{TP + FN}, \tag{49}$$

$$AP = \int_0^1 P(R)\,\mathrm{d}R, \tag{50}$$

$$mAP = \frac{1}{K}\sum_{i=1}^{K} AP_i, \tag{51}$$

where TP, FP, and FN denote true positives, false positives, and false negatives, respectively, and K denotes the number of categories. Precision (P) quantifies the proportion of correct detections, recall (R)

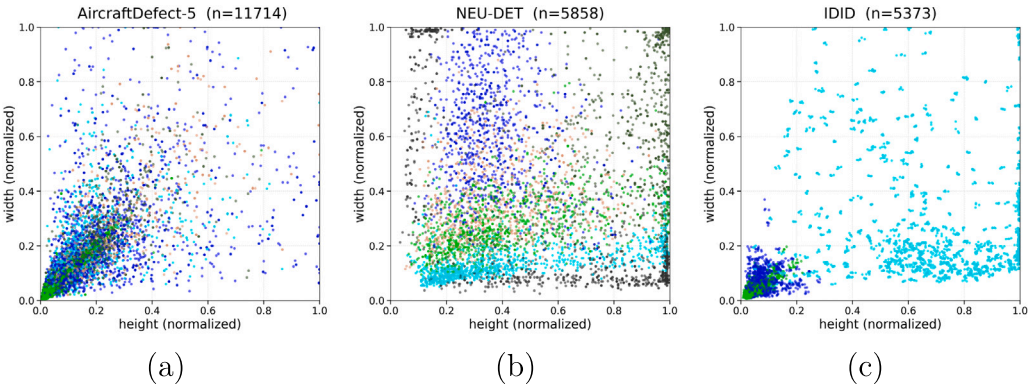

Fig. 9. Scatter plots depicting the distributions of bounding box widths and heights across the three datasets: (a) AircraftDefect-5, (b) NEU-DET, and (c) IDID. Colors denote different defect types.

measures the ability to capture all relevant defects, and mAP reflects overall detection accuracy across categories. Model complexity was assessed using parameter count (Params, M), frames per second (FPS), and computational cost (GFLOPs). FPS in the main comparison tables was measured on a single NVIDIA RTX 4090 GPU with batch size 1 under FP32 precision using the same detection-evaluation pipeline. All model-comparison, ablation, and design-analysis experiments were conducted under identical settings, and metrics reported with "±" denote the mean and standard deviation over three independent runs. Unless otherwise noted, GFLOPs refer to inference computation, whereas training GFLOPs are additionally reported for reinforcement-related analyses to more accurately characterize lightweight efficiency.

4.2. Model comparison

To evaluate LARANet under practical inspection settings, we compare it on AircraftDefect-5, NEU-DET, and IDID with representative one-stage, two-stage, transformer-based, lightweight, and defect-oriented detectors. The baselines include YOLOv13, recent YOLO variants, NanoDet, PicoDet, EfficientDet, RT-DETR, LW-DETR, DEIM, Faster R-CNN, and SSD. In addition, WSS-YOLO [14] and AFD-RTNet [15] are reproduced from public implementations and evaluated under the same protocol.

As shown in Tables 2 and 3, LARANet provides the strongest accuracy–efficiency balance among the compared methods. On AircraftDefect-5, it obtains 64.56%/43.38% mAP@0.5/mAP@0.5:0.95, improving over YOLOv13 by 5.49/5.19 points and over AFD-RTNet by 9.46/10.45 points, while maintaining 108.54 FPS at only 6.08 GFLOPs. The larger gain on mAP@0.5:0.95 indicates more accurate high-IoU localization for small and irregular defects. Compared with ultra-lightweight detectors such as NanoDet and PicoDet, transformer-based baselines such as RT-DETR and LW-DETR, and high-capacity methods such as PP-YOLOE and DEIM, LARANet achieves a more favorable trade-off between detection accuracy and computational cost.

The two public benchmarks show a consistent trend under different material and acquisition conditions. On NEU-DET, LARANet records 83.87%/53.47% mAP@0.5/mAP@0.5:0.95, improving over YOLOv13 by 7.49/7.54 points and exceeding AFD-RTNet by 3.73/4.87 points. It also surpasses EfficientDet, PP-YOLOE, and DETR-based baselines in high-IoU accuracy. On IDID, LARANet obtains 84.86%/53.38%, which is 8.80/4.58 points higher than YOLOv13 and clearly ahead of YOLO26, EfficientDet, RT-DETR, and LW-DETR, while achieving the highest precision and recall on this dataset. These results suggest stronger robustness to repetitive metallic textures, weak boundaries, outdoor illumination variation, and background clutter.

The qualitative comparisons in Fig. 10 further support the numerical results. On AircraftDefect-5, LARANet produces more complete detections with compact, well-aligned boxes, whereas YOLOv13 and YOLO26 occasionally miss defects under reflective surfaces. On NEU-DET and IDID, EfficientDet is more prone to missing small or low-contrast defects, and RT-DETR shows reduced recall in cluttered or illumination-variant scenes. In contrast, LARANet preserves higher confidence and tighter scale alignment across different materials and lighting conditions.

Overall, the model comparison demonstrates that LARANet consistently improves detection accuracy and high-IoU localization while retaining a compact computational profile. Its performance on small targets, complex textures, and cross-domain scenes supports the effectiveness and generalizability of the proposed architectural design.

4.3. Ablation study

To quantify the contribution of each component, we conducted a comprehensive ablation study on the primary dataset. Starting from the YOLOv13 baseline, the RepGhost lightweight backbone, CRPM, ESAM, and the AWIoU loss were evaluated individually and in selected combinations.

As summarized in Table 4, each proposed module provides distinct and complementary improvements. Integrating the RepGhost lightweight backbone raises mAP@0.5 from 59.07% to 61.83% while reducing parameters by approximately 12.6%, validating its efficiency in compact feature extraction. The CRPM module notably increases mAP@0.5:0.95 by 3.90 points, reflecting enhanced high-IoU localization through multi-agent reinforcement perception. ESAM strengthens edge and texture sensitivity via energy-aware statistical edge fusion, while AWIoU improves geometric consistency and regression stability in the full configuration. Among the two-module variants, RepGhost+CRPM achieves the highest mAP@0.5, whereas RepGhost+ESAM achieves the highest mAP@0.5:0.95, indicating different complementary roles. The full LARANet delivers the best overall accuracy with a +5.49/+5.19-point gain in mAP@0.5/mAP@0.5:0.95 over the baseline while preserving a compact computational profile.

Feature-response visualizations corroborate these findings (Fig. 11). Across the ablation variants, activation maps evolve from diffuse, background-heavy patterns to concentrated highlights on true defect regions. In particular, the joint effect of CRPM and ESAM enhances responses to small-scale defects and weak boundaries, while maintaining contextual sensitivity to surrounding textures. This cooperative attention behavior demonstrates that the proposed modules jointly enable fine-grained, context-consistent, and robust defect localization across complex industrial surfaces.

4.4. Analysis of key design choices

This subsection analyzes the main design choices of LARANet, including the CRPM optimization strategy, P3-stage module replacement, ESAM components, key hyperparameters, and the IoU-based regression loss.

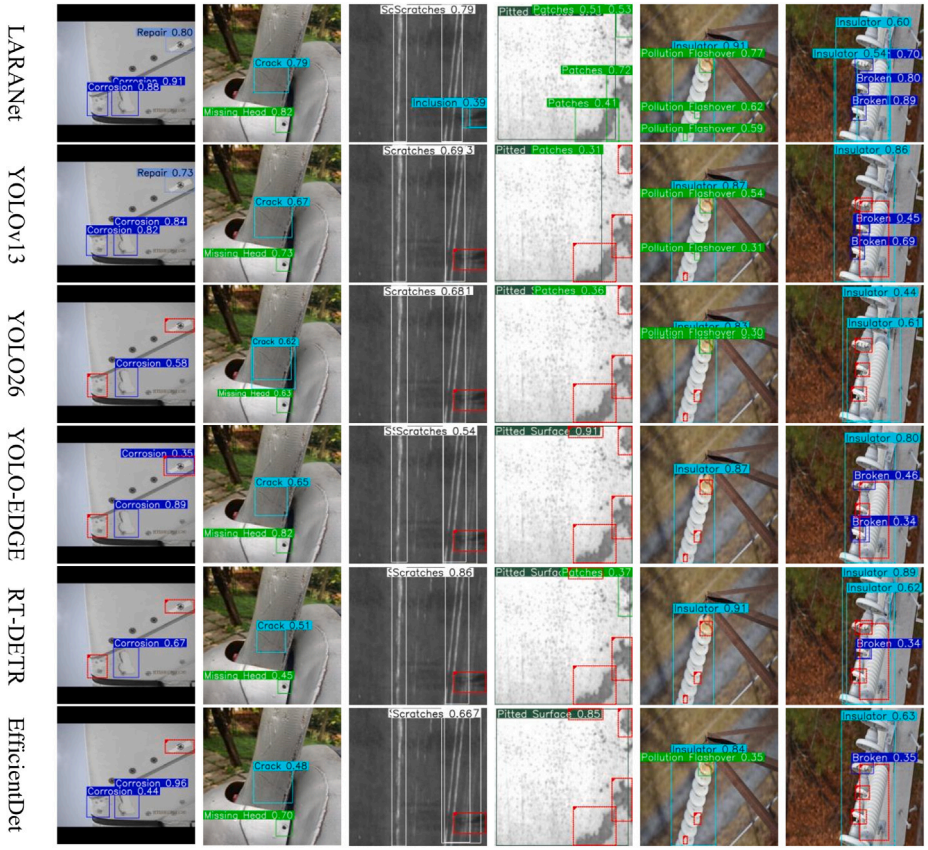

Fig. 10. Model-comparison visualization of detection results across three datasets.

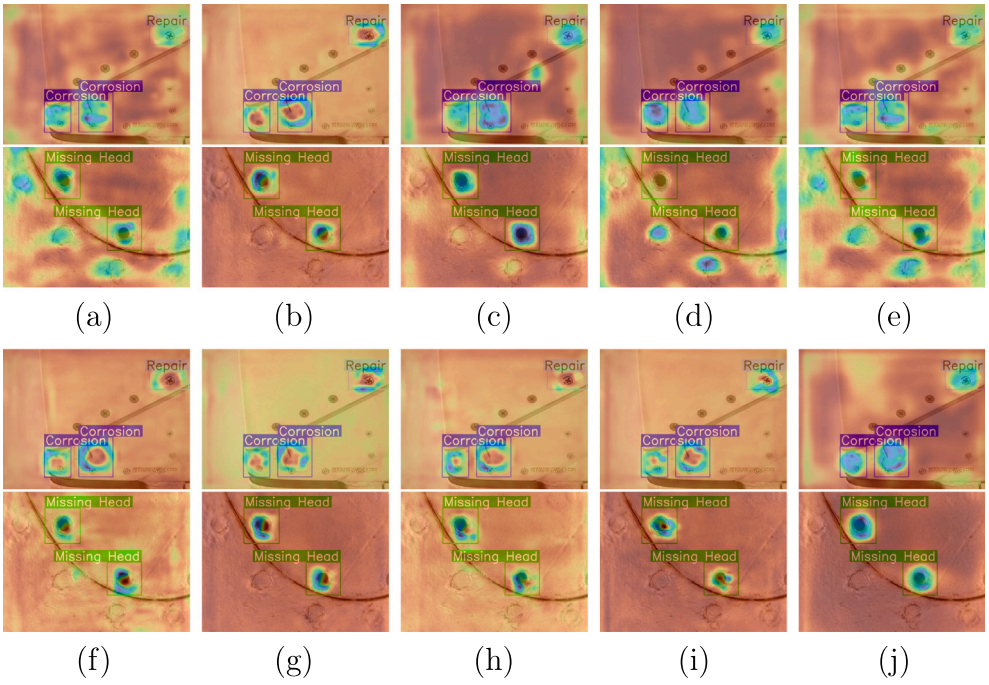

Fig. 11. Visualization of mid-layer feature responses under different ablation settings: (a) Baseline only, (b) +RepGhost, (c) +CRPM, (d) +ESAM, (e) +AWIoU, (f) RepGhost + CRPM, (g) RepGhost + ESAM, (h) RepGhost + CRPM + ESAM, (i) RepGhost + ESAM + AWIoU, (j) Full LARANet.

Table 2
Detection performance comparison on AircraftDefect-5.

Model	mAP@0.5 (%)	mAP@0.5:0.95 (%)	P	R	Params (M)	GFLOPs	FPS
YOLOv13	59.07 ± 0.15	38.19 ± 0.14	82.80	52.07	2.46	6.29	100.22
YOLO26	50.77 ± 0.23	30.11 ± 0.16	72.93	46.45	2.69	6.60	66.17
YOLOv11	44.81 ± 0.31	24.91 ± 0.29	60.44	43.94	2.59	6.44	104.80
YOLOv10	40.50 ± 0.27	23.44 ± 0.19	55.70	40.41	2.71	8.40	72.23
YOLOv8	50.17 ± 0.17	29.67 ± 0.13	71.50	46.78	3.01	8.20	72.54
YOLOv5	45.56 ± 0.29	26.22 ± 0.12	62.93	42.77	2.51	7.18	89.24
YOLO-Edge	40.23 ± 0.37	17.46 ± 0.15	40.23	20.13	7.55	20.22	19.37
PP-YOLOE	54.87 ± 0.32	34.51 ± 0.18	75.98	50.55	8.00	19.45	35.46
EfficientDet	46.91 ± 0.35	28.20 ± 0.21	66.36	43.34	0.50	3.03	87.20
NanoDet	9.74 ± 0.26	8.71 ± 0.18	10.25	11.27	0.94	2.77	155.72
PicoDet	11.25 ± 0.37	10.07 ± 0.10	11.85	13.03	1.06	3.27	153.03
RT-DETR	52.45 ± 0.26	29.80 ± 0.22	71.74	48.16	20.23	59.97	8.15
LW-DETR	48.93 ± 0.27	24.56 ± 0.19	64.79	44.14	12.18	11.26	12.69
DEIM	47.88 ± 0.31	29.66 ± 0.18	77.63	43.46	17.44	18.42	21.54
Faster R-CNN	4.32 ± 0.12	1.21 ± 0.08	3.01	24.65	18.96	41.84	7.34
SSD	28.79 ± 0.16	22.68 ± 0.16	2.04	37.55	6.13	3.04	147.36
WSS-YOLO [14]	51.86 ± 0.31	25.06 ± 0.18	72.14	42.82	3.20	7.74	106.44
AFD-RTNet [15]	55.10 ± 0.29	32.93 ± 0.28	80.58	47.62	11.65	10.02	119.39
Ours	**64.56 ± 0.19**	**43.38 ± 0.16**	**85.36**	**57.85**	2.38	6.08	108.54

Table 3
Detection performance comparison on NEU-DET and IDID.

Model	mAP@0.5 (%)		mAP@0.5:0.95 (%)		P		R	
	NEU-DET	IDID	NEU-DET	IDID	NEU-DET	IDID	NEU-DET	IDID
YOLOv13	76.38	76.06	45.93	48.80	69.16	81.96	72.41	71.30
YOLO26	76.63	66.93	41.54	42.77	70.77	70.45	72.30	66.60
YOLOv11	75.54	61.62	41.85	35.26	65.79	67.80	68.13	60.20
YOLOv10	74.60	66.89	42.72	42.54	67.01	69.91	70.78	64.54
YOLOv8	74.75	66.60	42.07	42.20	62.70	72.62	75.25	64.09
YOLOv5	71.62	66.22	38.80	40.36	58.00	69.94	69.80	64.75
YOLO-Edge	73.60	65.07	38.67	38.72	73.60	65.07	61.75	61.26
PP-YOLOE	79.31	60.22	44.64	38.95	68.53	67.63	**77.33**	58.85
EfficientDet	78.94	64.65	40.24	37.37	72.63	70.27	74.09	62.76
NanoDet	48.07	45.85	38.71	29.55	36.02	49.74	39.93	53.51
PicoDet	46.83	33.85	34.96	16.06	48.76	23.90	51.64	25.79
RT-DETR	68.92	64.93	41.65	40.68	63.08	59.73	64.79	64.19
LW-DETR	64.71	60.71	36.82	35.38	59.42	55.95	60.19	61.67
DEIM	77.13	57.47	40.00	39.12	72.06	72.54	70.83	57.60
Faster R-CNN	35.33	23.29	13.13	9.74	14.15	3.28	13.15	18.89
SSD	27.80	4.36	9.52	1.63	2.83	2.03	19.84	15.22
WSS-YOLO [14]	75.26	68.41	43.23	44.05	66.85	73.96	69.52	66.52
AFD-RTNet [15]	80.14	72.94	48.60	45.78	68.31	76.16	72.09	70.02
Ours	**83.87**	**84.86**	**53.47**	**53.38**	**81.21**	**88.64**	76.76	**80.76**

Table 4
Effect of different modules on detection performance.

RepGhost	CRPM	ESAM	AWIoU	mAP@0.5 (%)	mAP@0.5:0.95 (%)	Params (M)	GFLOPs	FPS
				59.07	38.19	2.46	6.29	100.22
✓				61.83 (+2.76)	39.83 (+1.64)	2.15	6.07	109.32
	✓			60.89 (+1.82)	42.09 (+3.90)	2.76	6.31	97.22
		✓		60.87 (+1.80)	40.49 (+2.30)	2.46	6.29	102.62
			✓	60.92 (+1.85)	41.39 (+3.20)	2.46	6.29	104.35
✓	✓			62.60 (+3.53)	42.22 (+4.03)	2.38	6.08	107.17
✓		✓		62.22 (+3.15)	42.57 (+4.38)	2.15	6.07	109.06
✓	✓	✓		64.02 (+4.95)	42.58 (+4.39)	2.38	6.08	108.33
✓		✓	✓	63.92 (+4.85)	39.98 (+1.79)	2.15	6.07	109.02
✓	✓	✓	✓	64.56 (+5.49)	43.38 (+5.19)	2.38	6.08	108.54

4.4.1. Policy optimization algorithms

We systematically assessed alternative reinforcement learning strategies for the CRPM module on AircraftDefect-5, including Proximal Policy Optimization (PPO), Graph-Regularized Policy Optimization (GRPO), and the vanilla REINFORCE algorithm, under multiple hyperparameter configurations.

As summarized in Table 5, PPO(A) achieves the highest mAP@0.5 among the tested variants, but requires substantially higher training GFLOPs. REINFORCE has the lowest training cost but provides slightly lower mAP@0.5 than AC+GAE, while GRPO introduces additional communication and graph-regularization costs without achieving the best accuracy. In contrast, AC+GAE achieves competitive mAP@0.5 with much lower training cost than PPO(A). Therefore, AC+GAE is adopted as a balanced policy optimization scheme rather than the absolute best-performing choice in terms of mAP alone.

Fig. 12(a) and (b) further support this stability-oriented choice by comparing the epoch-wise mAP@0.5 and total-loss trajectories of AC+GAE and REINFORCE. Although the two methods reach comparable final mAP, AC+GAE exhibits fewer oscillations during training. This suggests that the critic-guided advantage estimate helps reduce update variance and stabilize policy learning.

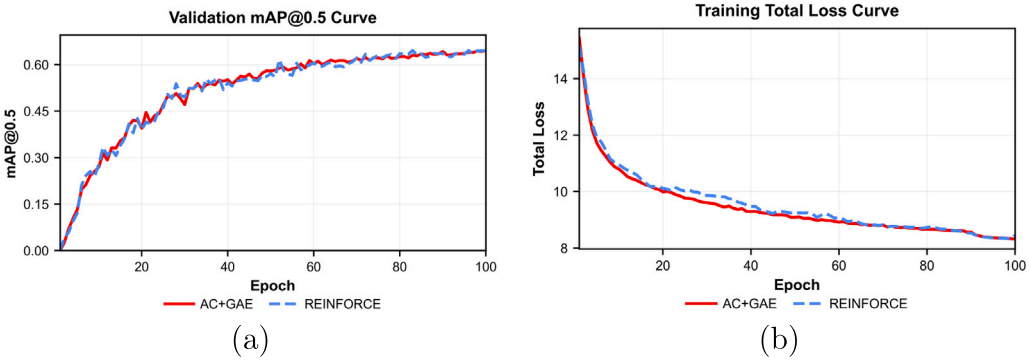

Fig. 12. Training stability comparison between AC+GAE and REINFORCE in CRPM: (a) validation mAP@0.5; (b) training total loss.

Table 5
Comparison of policy optimization algorithms for CRPM. Representative fast (F) and accuracy (A) settings for PPO/GRPO, the best REINFORCE setting, and our GAE-regularized Actor–Critic are reported.

Algorithm	Policy hyperparameters	mAP@0.5	Training GFLOPs
GRPO (F)	lr = 2e−4/upd = 2	63.55 ± 0.28	44.22
GRPO (A)	lr = 2e−4/upd = 4	63.72 ± 0.26	54.82
PPO (F)	ϵ = 0.1/upd = 2	64.22 ± 0.22	40.10
PPO (A)	ϵ = 0.2/upd = 4	**64.82 ± 0.21**	56.26
REINFORCE	lr = 5e−5/b = 128	64.49 ± 0.17	**21.44**
Ours	AC+GAE ($N = 4$)	64.56 ± 0.19	23.48

The reinforcement learning objective is used only during training. During inference, CRPM performs deterministic feature modulation with learned parameters, without reward computation, policy updating, or environment interaction. Therefore, LARANet improves adaptive perception while maintaining deployment efficiency.

4.4.2. P3 module replacement study

To evaluate whether CRPM offers advantages over conventional P3 module replacements, we conducted a controlled study with three representative alternatives: an attention-based receptive-field module, a dynamic convolution module, and a learnable sampling module. The attention-based module enhances contextual modeling via adaptive receptive fields, the dynamic convolution module generates input-dependent kernels for local adaptation, and the learnable sampling module emphasizes informative regions. As shown in Table 6, CRPM achieves the highest detection performance with only a slight increase in parameter count, while maintaining comparable GFLOPs and the highest FPS. Notably, CRPM improves mAP@0.5 by +2.71 percentage points over the attention-based receptive-field module, demonstrating a more favorable accuracy–efficiency trade-off than conventional attention-based designs.

Unlike conventional attention-based replacements that perform deterministic one-shot reweighting, CRPM formulates feature perception at P3 as a sequential decision process. Agents explore the mid-level feature space, exchange contextual messages, and accumulate visit maps to adapt spatial focus to sparse, irregular, or low-contrast defects. Dynamic convolution and learnable sampling provide input-dependent local adaptation, but they do not explicitly model cooperative exploration or training feedback from detection objectives. These characteristics help explain why CRPM achieves a stronger accuracy–complexity trade-off.

4.4.3. ESAM component analysis

This decomposition study analyzes the two key components of ESAM within the full LARANet configuration. As shown in Table 7, the variance-focused design (SimAM + Variance Branch, 63.36%) provides the stronger standalone benefit, whereas the edge-only branch is less effective when used alone. However, combining variance-aware modulation with edge-aware enhancement achieves the best overall result, indicating that boundary cues provide complementary gains when coupled with statistical variance modeling.

To justify the choice of the ESAM variance-fusion weight ω, we evaluate different global–local variance fusion weights on AircraftDefect-5. As shown in Table 8, $\omega = 0.7$ achieves the best overall performance, indicating that a slightly stronger global variance contribution preserves structural consistency while local variance retains fine-grained defect sensitivity.

4.4.4. CRPM hyperparameter analysis

To investigate the influence of CRPM hyperparameters on AircraftDefect-5, five key variables were analyzed within the full LARANet framework: the number of agents N, time steps T, attention-window size w, trajectory sampling count C, and reward weight α. A one-factor-at-a-time protocol was used, varying only the target hyperparameter while keeping the others fixed.

As shown in Figs. 13 and 14, performance exhibits a clear nonlinear dependence on hyperparameter choices. Moderate agent and trajectory settings improve exploration, whereas overly large N, T, or C introduce redundant sampling and higher training cost. The window size $w = 7$ provides sufficient local context without absorbing excessive background, and $\alpha = 0.7$ gives a stable balance between detection-loss reduction and accuracy-oriented reward. Therefore, $N = 4$, $T = 3$, $w = 7$, $C = 3$, and $\alpha = 0.7$ are adopted as the final CRPM configuration.

4.4.5. IoU-based regression loss comparison

We compare four IoU-based losses: AWIoU (ours), WIoU [37], MPDIoU [38], and SIoU [39]. These baselines cover adaptive box scaling, point-distance modeling, and revised angular penalties, respectively. Both theoretical characteristics and practical training behavior are analyzed on AircraftDefect-5, with a focus on small objects (area ratio $\leq$0.5%) and partial-overlap scenarios.

As illustrated in Fig. 15(a)–(c), AWIoU imposes stronger penalties in small-object and partial-overlap regions while maintaining gradient stability comparable to WIoU, leading to improved optimization behavior. Fig. 15(a) presents theoretical analyses, including loss–IoU curves, gradient trends, and size sensitivity. Fig. 15(b) reports training dynamics, showing mAP@0.5 convergence and final performance comparisons. Fig. 15(c) further examines partial-overlap cases, highlighting loss behavior and convergence incentives.

As summarized in Table 9, AWIoU achieves the best final mAP@0.5 of 64.56%, outperforming WIoU (63.34%), MPDIoU (63.00%), and SIoU (62.76%), while also producing lower box loss. Although AWIoU converges slightly slower at early stages, it provides stronger convergence incentives and higher loss reduction rates in partial-overlap scenarios, indicating more effective optimization signals for challenging small-defect localization.

Table 6
Comparison with alternative receptive-field adaptation modules.

Model	mAP@0.5 (%)	P (%)	R (%)	Params (M)	GFLOPs	FPS
Attention-based receptive-field module	61.85 ± 0.19	80.72	57.35	2.16	6.16	102.62
Dynamic-convolution module	62.44 ± 0.20	82.86	56.95	**2.15**	**6.07**	106.84
Learnable-sampling module	61.93 ± 0.14	80.84	57.11	2.18	6.47	101.80
CRPM	**64.56 ± 0.19**	**85.36**	**57.85**	2.38	6.08	**108.54**

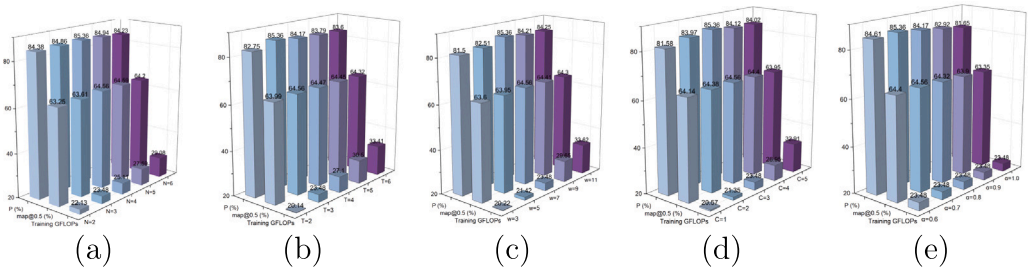

Fig. 13. CRPM hyperparameter sensitivity analysis. Subplots (a)–(e) correspond to N, T, w, C, and α, respectively.

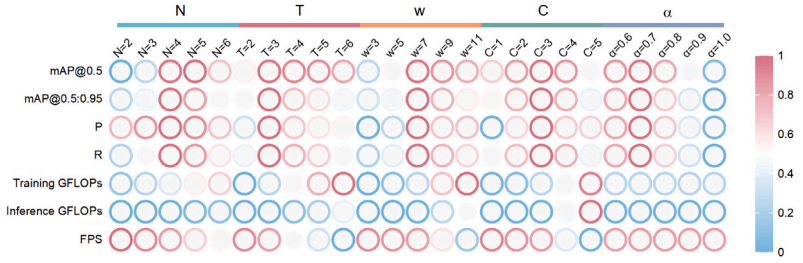

Fig. 14. Heatmap summarizing performance across seven metrics: mAP@0.5, mAP@0.5:0.95, precision, recall, training GFLOPs, inference GFLOPs, and FPS. Scores are normalized; higher is better for accuracy metrics and FPS, lower is better for FLOPs.

Table 7
ESAM decomposition ablation.

Model	mAP@0.5 (%)	P (%)	R (%)
Baseline SimAM	62.71 ± 0.15	81.58	57.07
SimAM + Edge branch	61.58 ± 0.21	83.64	55.38
SimAM + Variance branch	63.36 ± 0.19	83.71	56.98
ESAM	**64.56 ± 0.19**	**85.36**	**57.85**

Table 8
Sensitivity analysis of the variance-fusion weight ω in ESAM.

ω	mAP@0.5 (%)	P (%)	R (%)
0.5	63.92 ± 0.16	84.63	56.91
0.6	64.21 ± 0.21	84.87	57.28
0.7	**64.56 ± 0.19**	**85.36**	**57.85**
0.8	64.33 ± 0.15	85.04	57.42
0.9	63.97 ± 0.21	84.58	57.06

Table 9
Comparison of IoU losses on AircraftDefect-5.

Loss	mAP@0.5 (%)	P (%)	R (%)
WIoU [37]	63.34 ± 0.19	81.23	57.44
MPDIoU [38]	63.00 ± 0.22	81.68	57.36
SIoU [39]	62.76 ± 0.14	80.07	57.56
AWIoU (Ours)	**64.56 ± 0.19**	**85.36**	**57.85**

4.5. Model visualization and analysis

4.5.1. Performance visualization

Fig. 16 presents the confusion matrices of LARANet on AircraftDefect-5, NEU-DET, and IDID. The clear diagonal distributions indicate reliable class discrimination across the three datasets. Residual errors mainly arise from background-related confusion, especially on AircraftDefect-5 and IDID, where small or weakly contrasted defects are visually close to background regions. NEU-DET shows more concentrated diagonal responses, suggesting stable recognition under relatively regular surface textures.

The convergence curves in Fig. 17 further show that LARANet maintains stable optimization behavior across all three datasets. The trajectories of mAP@0.5, mAP@0.5:0.95, precision, and recall generally increase and then stabilize with only minor fluctuations, indicating that the reinforcement-guided branch does not noticeably destabilize the training process. Reported numerical results are obtained from the original evaluation logs.

Taken together, these visualization results suggest that many remaining errors are related to background interference rather than severe inter-class confusion, supporting the effectiveness of the proposed modules in enhancing boundary precision and reducing texture ambiguity.

4.5.2. Detection visualization

To demonstrate the model's effectiveness qualitatively, representative detection results on the three datasets are visualized in Fig. 18.

On AircraftDefect-5, LARANet captures fine-grained corrosion, cracks, missing-head defects, repairs, and dents with tight boxes and high confidence, even on small or blurred-edge instances. On NEU-DET, it maintains clear responses to low-contrast crazing and pitted surfaces amid repetitive steel textures and specular noise. On IDID, it localizes insulator defects under outdoor lighting, reflections, and occlusions, with clear scale and boundary alignment. These visual results are consistent with the quantitative cross-scene evaluations and further support the robustness of LARANet across metallic, composite, and outdoor infrastructure domains.

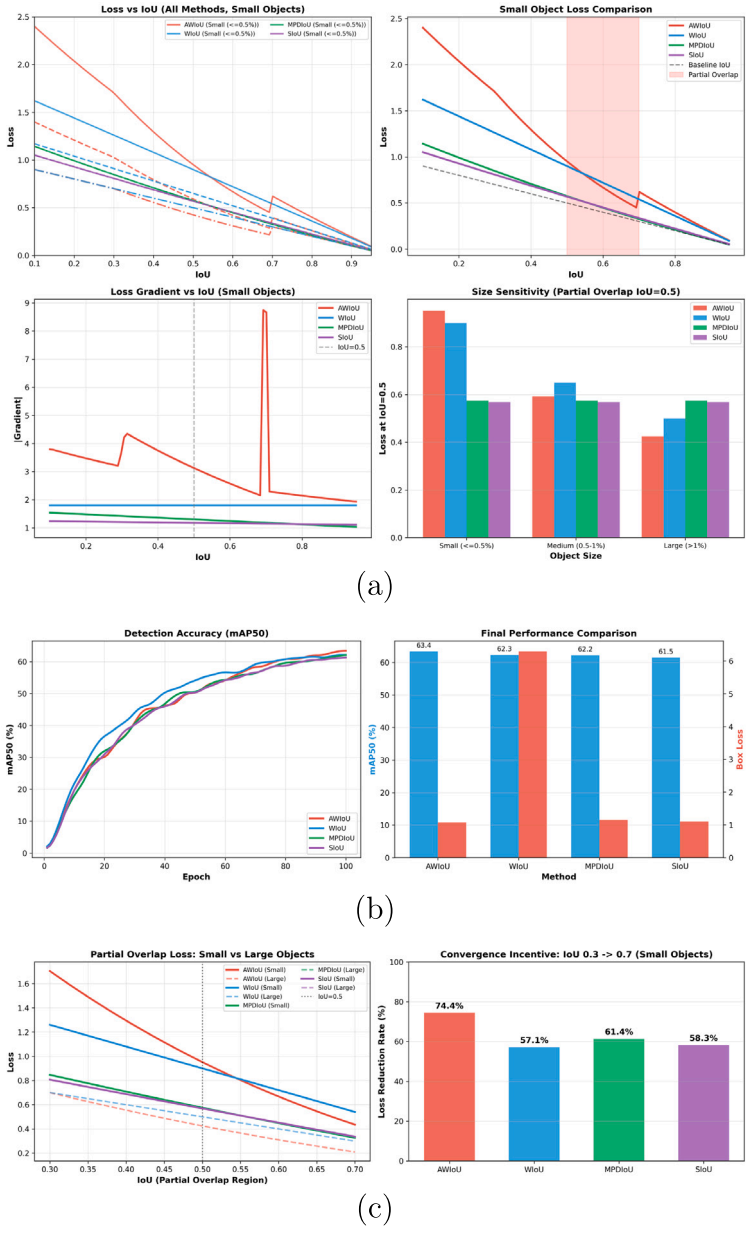

Fig. 15. IoU-loss analysis and training results: (a) analytical comparison, (b) convergence from training logs, and (c) partial-overlap analysis.

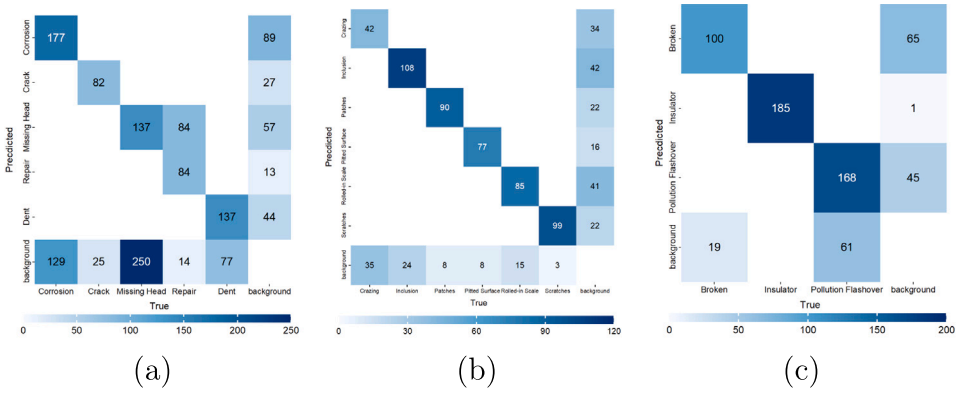

Fig. 16. Confusion matrices of LARANet across three datasets: (a) AircraftDefect-5, (b) NEU-DET, (c) IDID.

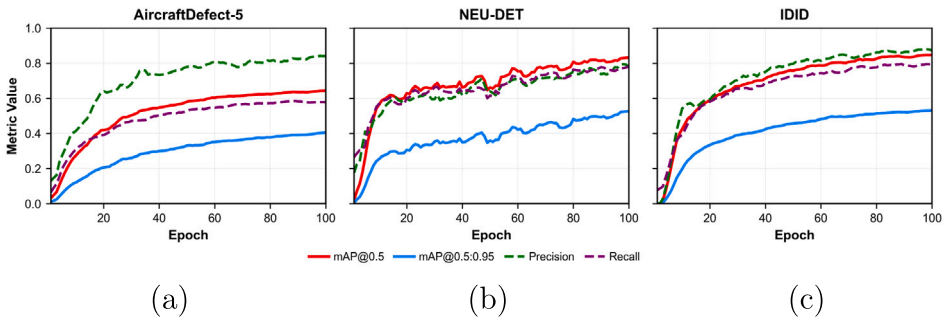

Fig. 17. Convergence curves of LARANet on (a) AircraftDefect-5, (b) NEU-DET, and (c) IDID. Curves are mildly smoothed for readability.

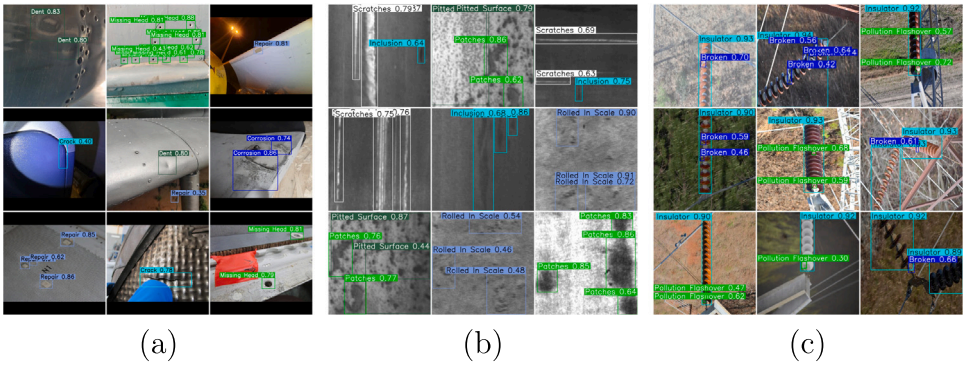

Fig. 18. Detection results of LARANet on three datasets: (a) AircraftDefect-5, (b) NEU-DET, (c) IDID.

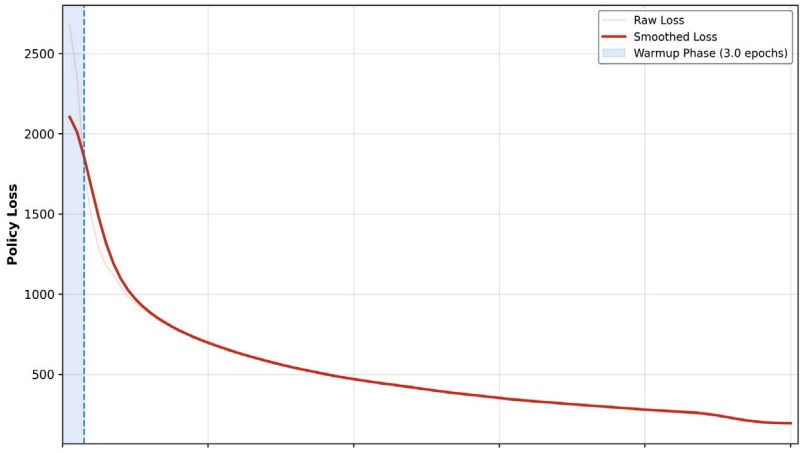

Fig. 19. CRPM policy-loss curve during training. Light-red: raw loss; dark-red: smoothed loss; blue-shaded region: warm-up stage (epochs 1–3).

4.5.3. CRPM convergence and behavior analysis

We report the policy-loss evolution of CRPM in Fig. 19. The light-red curve denotes the raw loss and exhibits noticeable oscillations, while the thick dark-red curve denotes the smoothed trend. The blue-shaded area marks the warm-up stage (first three epochs). Overall, the policy loss decreases from above 2600 to around 200, indicating stable convergence after warm-up.

To further examine CRPM behavior, we visualize agent trajectories at epochs 40, 60, 80, and 100 (Fig. 20). Each subplot shows three-step movements in feature space: green dots indicate initial positions, red dots indicate final positions, and arrows (dark → light) represent temporal motion. Trajectories are extracted on the P3 feature map and overlaid on full-resolution images with proportional upscaling for readability. At epoch 40, movements are relatively scattered; by epochs 60–80, trajectories gradually concentrate on defect-relevant regions; by epoch 100, agents consistently move toward target areas, demonstrating stable and defect-oriented spatial attention.

5. Conclusion

This paper proposes LARANet, a lightweight defect detection framework for practical industrial inspection. By combining RepGhost representation learning, reinforcement-guided CRPM, energy-aware ESAM, and the AWIoU loss, LARANet improves adaptive feature focusing and geometric localization while maintaining low computational complexity.

Experiments on AircraftDefect-5, NEU-DET, and IDID show that LARANet achieves the highest mAP among the compared methods and strong precision–recall performance while retaining low FLOPs, few parameters, and high FPS. The results indicate that reinforcement-guided

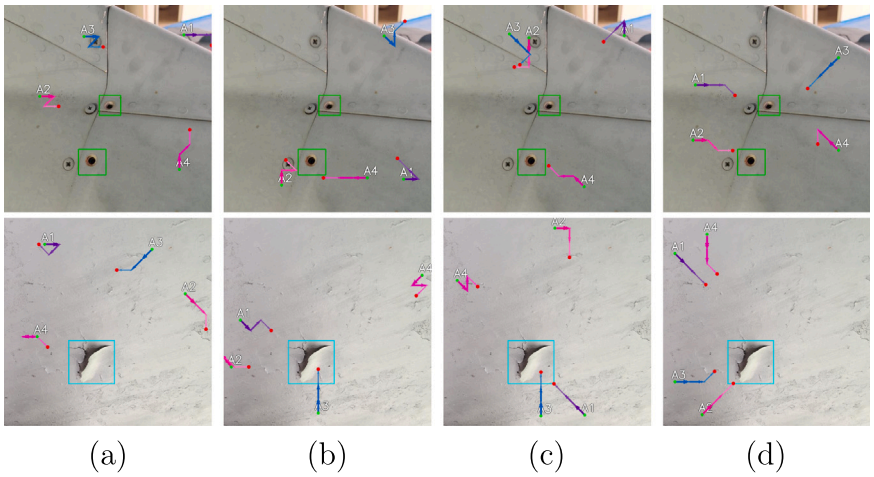

Fig. 20. Agent trajectory visualization across training epochs: (a) epoch 40, (b) epoch 60, (c) epoch 80, (d) epoch 100.

perception and statistical edge-aware modulation improve robustness to small defects, weak boundaries, and cross-scene appearance changes.

Several limitations remain. Broader evaluation across more industrial domains is still needed to further substantiate cross-scene generalization. In addition, although CRPM does not increase inference-time deployment complexity, its reinforcement-guided optimization introduces extra training cost. Future work will explore more efficient policy learning, self-supervised training, and multimodal sensing for robust real-world inspection.

CRediT authorship contribution statement

Yuhan Shao: Writing – original draft, Methodology, Formal analysis, Conceptualization. **Yu Zhou:** Writing – review & editing, Visualization, Validation, Supervision, Project administration. **Taolue Chen:** Writing – review & editing, Validation, Resources, Investigation. **Zixuan Li:** Writing – review & editing, Methodology, Data curation.

Declaration of generative AI and AI-assisted technologies in the manuscript preparation process

During the preparation of this work, the authors used ChatGPT for language polishing and editing assistance. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors sincerely thank the anonymous reviewers and the editors for their valuable comments and insightful suggestions, which have greatly helped improve the quality and clarity of this manuscript. The authors are also grateful to the open-source community for providing valuable tools and resources that facilitated this research.

Data availability

The data that has been used is confidential.

References

[1] A. Baitieva, D. Hurych, V. Besnier, et al., Supervised anomaly detection for complex industrial images, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2024, pp. 17754–17762.
[2] Y. Ma, J. Yin, F. Huang, et al., Surface defect inspection of industrial products with object detection deep networks: A systematic review, Artif. Intell. Rev. 57 (12) (2024) 333.
[3] B. Wan, X. Zhou, B. Zhu, et al., CANet: Context-aware aggregation network for salient object detection of surface defects, J. Vis. Commun. Image Represent. 93 (2023) 103820.
[4] F. Yan, X. Jiang, Y. Lu, et al., Global context guided refinement and aggregation network for lightweight surface defect detection, Pattern Recognit. (2025) 112893.
[5] J. Liu, Z. Wang, L. Ma, et al., Benchmarking object detection robustness against real-world corruptions, Int. J. Comput. Vis. 132 (10) (2024) 4398–4416.
[6] M.V. Bimrose, T. Hu, D.J. McGregor, et al., Detecting and classifying hidden defects in additively manufactured parts using deep learning and X-ray computed tomography, J. Intell. Manuf. 36 (5) (2025) 3465–3479.
[7] T. Herzog, M. Brandt, A. Trinchi, et al., Process monitoring and machine learning for defect detection in laser-based metal additive manufacturing, J. Intell. Manuf. 35 (4) (2024) 1407–1437.
[8] W. Nsengiyumva, S. Zhong, L. Zheng, W. Liang, B. Wang, Y. Huang, X. Chen, Y. Shen, Sensing and nondestructive testing applications of terahertz spectroscopy and imaging systems: State-of-the-art and state-of-the-practice, IEEE Trans. Instrum. Meas. 72 (2023) 1–83.
[9] X. Tong, Z. Yu, X. Tian, et al., Improving accuracy of automatic optical inspection with machine learning, Front. Comput. Sci. 16 (1) (2022) 161310.
[10] Y. Zhou, Z. Zhao, MPA-YOLO: Steel surface defect detection based on improved YOLOv8 framework, Pattern Recognit. (2025) 111897.
[11] A. Vijayakumar, S. Vairavasundaram, YOLO-based object detection models: A review and its applications, Multimedia Tools Appl. 83 (35) (2024) 83535–83574.
[12] H. Zhao, J. Liu, H. Zhang, Q. Wang, L. Wang, K. Zhai, A cross-view masked contrastive learning method for intelligent pipeline platform with unlabeled samples, IEEE/ASME Trans. Mechatronics (2026) http://dx.doi.org/10.1109/TMECH.2025.3602163.
[13] H. Zhao, J. Liu, J. Tang, X. Shen, S. Lu, Q. Wang, A MFL mechanism-based self-supervised method for defect detection with limited labeled samples, IEEE Trans. Instrum. Meas. 72 (2023) 1–10, http://dx.doi.org/10.1109/TIM.2022.3212041.
[14] M. Lu, W. Sheng, Y. Zou, et al., WSS-YOLO: An improved industrial defect detection network for steel surface defects, Measurement 236 (2024) 115060.
[15] Q. Liu, et al., A real-time anchor-free defect detector with global and local feature enhancement for surface defect detection, Expert Syst. Appl. 246 (2024) 123199.
[16] X. Sun, K. Song, X. Wen, Y. Wang, Y. Yan, SDD-DETR: surface defect detection for no-service aero-engine blades with detection transformer, IEEE Trans. Autom. Sci. Eng. (2024).
[17] N. Suvittawat, C. Kurniawan, J. Datephanyawat, et al., Advances in aircraft skin defect detection using computer vision: A survey and comparison of YOLOv9 and RT-DETR performance, Aerospace 12 (4) (2025) 356.
[18] D.G. Lema, L. Sánchez-González, R. Usamentiaga, et al., Benchmarking deep learning models for surface defect detection: a reproducible and statistically-rigorous approach, J. Intell. Manuf. (2025) 1–18.
[19] Z. Jia, M. Wang, S. Zhao, A review of deep learning-based approaches for defect detection in smart manufacturing, J. Opt. 53 (2) (2024) 1345–1351.

[20] J. Zhao, F. Wei, C. Xu, Hybrid proposal refiner: Revisiting DETR series from the faster R-CNN perspective, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 17416–17426.
[21] F. Yan, X. Jiang, Y. Lu, et al., Wavelet and prototype augmented query-based transformer for pixel-level surface defect detection, in: Proceedings of the Computer Vision and Pattern Recognition Conference, 2025, pp. 23860–23869.
[22] H. Sun, R. Wang, Y. Li, et al., SET: Spectral enhancement for tiny object detection, in: Proceedings of the Computer Vision and Pattern Recognition Conference, 2025, pp. 4713–4723.
[23] K. Dong, Q. Shen, C. Wang, et al., Improved swin transformer-based defect detection method for transmission line patrol inspection images, Evol. Intell. 17 (1) (2024) 549–558.
[24] Y. Tang, R. Liu, S. Wang, YOLO-SUMAS: Improved printed circuit board defect detection and identification research based on YOLOv8, Micromachines 16 (5) (2025) 509.
[25] Z. Zeng, H. Wang, C. Yao, et al., Optimizing surface defect detection with YOLOv9: The role of advanced backbone models, Front. Artif. Intell. 8 (2025) 1675154.
[26] J. Wang, J. Su, Z. Wen, et al., Enhanced YOLOv10 for small object detection with context-aware and adaptive modules, Int. J. Multimed. Inf. Retr. 14 (3) (2025) 22.
[27] P. Chen, Y.G. Luo, J. Zhang, et al., PRD-YOLOv11: Efficient and accurate textile tow defect detection via progressive representation distillation, Signal Image Video Process. 19 (8) (2025) 671.
[28] Z. Chen, B. Liu, A high-accuracy PCB defect detection algorithm based on improved YOLOv12, Symmetry 17 (7) (2025) 978.
[29] Y. Gao, W. Luo, S. Zhang, et al., Benchmarking YOLOv8 to YOLOv13 for robust hand gesture recognition in human–robot interaction, Sci. Rep. 15 (1) (2025) 40043.
[30] R. Sapkota, M. Flores-Calero, R. Qureshi, et al., YOLO advances to its genesis: A decadal and comprehensive review of the YOLO series, Artif. Intell. Rev. 58 (2025) 274.
[31] A. Wong, T. B"ack, A.V. Kononova, et al., Deep multi-agent reinforcement learning: Challenges and directions, Artif. Intell. Rev. 56 (6) (2023) 5023–5056.
[32] S. Guicheng, W. Yang, Review on dec-pomdp model for marl algorithms, in: Smart Communications, Intelligent Algorithms and Interactive Methods: Proceedings of 4th International Conference on Wireless Communications and Applications, ICWCA 2020, Springer Nature Singapore, Singapore, 2022, pp. 29–35.
[33] A. Oroojlooy, D. Hajinezhad, A review of cooperative multi-agent deep reinforcement learning, Appl. Intell. 53 (11) (2023) 13677–13722.
[34] M.A. Hady, S. Hu, M. Pratama, et al., Multi-agent reinforcement learning for resources allocation optimization: A survey, Artif. Intell. Rev. 58 (11) (2025) 354.
[35] X. Li, T. Zhang, C. Liu, B. Xu, Theory of mind inspired large reasoning language model improved multi-agent reinforcement learning algorithm for robust and adaptive partner modelling, Mach. Intell. Res. (2025).
[36] L. Yang, R.Y. Zhang, L. Li, et al., SimAM: A simple, parameter-free attention module for convolutional neural networks, in: International Conference on Machine Learning, PMLR, 2021, pp. 11863–11874.
[37] W. Lee, H. Chang, J. Moon, J. Lee, M. Kim, ABBSPO: Adaptive bounding box scaling and symmetric prior based orientation prediction for detecting aerial image objects, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2025, pp. 8848–8858.
[38] C. Xiong, T. Zayed, X. Jiang, G. Alfalah, E.M. Abelkader, A novel model for instance segmentation and quantification of bridge surface cracks—The YOLOv8-AFPN-MPD-IoU, Sensors 24 (13) (2024) 4288.
[39] G. Li, R. Wang, Y. Zhang, C. Xu, X. Fan, Z. Zhou, P. Lv, Z. Ruan, LR-Net: Lossless feature fusion and revised SIoU for small object detection, Comput. Mater. Contin. 85 (2) (2025) 3267–3288.

Yuhan Shao received the B.Eng. degree in Computer Science from Nanjing University of Information Science and Technology in 2024 as a member of the Tencent Experimental Class. He is currently pursuing the M.Eng. degree in Software Engineering with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics (NUAA), and is also conducting joint research at the Institute of Semiconductors, Chinese Academy of Sciences. He has participated in multiple interdisciplinary AI projects spanning hardware systems, aerospace applications, and industrial deployment. His research interests include data mining, applied AI agents, reinforcement learning, and edge computing.

Yu Zhou received the B.Sc. and Ph.D. degrees in Computer Science from Nanjing University, China, in 2004 and 2009, respectively. He is an Associate Professor with the College of Computer Science and Technology, NUAA. Before joining NUAA in 2011, he conducted postdoctoral research in software engineering at Politecnico di Milano, Italy. From 2015 to 2016, he was a Visiting Researcher with the SEAL Lab, University of Zurich, Switzerland, where he is also an Adjunct Researcher. His research interests include software evolution analysis, mining software repositories, software architecture, and reliability analysis. He has led and participated in several national research programs in China. More information: https://csyuzhou.github.io/.

Taolue Chen received the Bachelor and Master degrees from Nanjing University, China, both in Computer Science. He was a junior researcher (OiO) at the Centrum Wiskunde & Informatica (CWI) and obtained the Ph.D. degree from the Vrije Universiteit Amsterdam, The Netherlands. He is currently a Lecturer at the School of Computing and Mathematical Sciences, Birkbeck, University of London. He had been a postdoctoral researcher at the University of Oxford (UK) and the University of Twente (NL). His research spans software engineering, programming languages, verification and machine learning, focusing on trustworthy ML and data-driven software development. He has published over 140 papers in venues including POPL, LICS, CAV, OOPSLA, ICSE, FSE, ASE, TACAS, ESOP, NeurIPS, ICLR, IJCAI, AAAI, EMNLP, TSE, TOSEM, TOCL, and LMCS, and received awards such as SETTA20 Best Paper and the 1st Prize in the CCF Software Prototype Competition 2022. More information: https://chentaolue.github.io/.

Zixuan Li received the B.Sc. degree from the College of Chemistry and Chemical Engineering, Xiamen University, China, through the Strong Foundation Program. He is currently pursuing a Ph.D. degree at Xiamen University, affiliated with the State Key Laboratory of Physical Chemistry of Solids. His research focuses on advanced battery materials and interdisciplinary applications of artificial intelligence in energy storage, where he leverages deep learning, neural network models, and LLMs to develop intelligent analysis frameworks that bridge microscopic physicochemical mechanisms with macroscopic performance evaluation.