applied sciences

MDPI

Article

DHR-OD: Dynamic Heterogeneous Redundancy for Robust Object Detection

Fengrui Tian [1] and Xun Chen [1,2,*]

[1] School of Information and Communication Engineering, Hainan University, Haikou 570228, China; 20243007015@hainanu.edu.cn
[2] Hainan Seed Industry Laboratory, Sanya 572024, China
* Correspondence: chenxun@hainanu.edu.cn

Abstract

Object detection underpins safety-critical applications such as autonomous driving and intelligent surveillance, yet the inherent vulnerability of object detection models exposes them to the dual threat of adversarial and backdoor attacks. Existing defense methods are mostly passive, relying on prior knowledge of known attacks and thus failing to defend against unknown attacks. To address these problems, we introduce Dynamic Heterogeneous Redundancy (DHR) into object detection and build the Dynamic Heterogeneous Redundancy for robust Object Detection (DHR-OD) framework. DHR-OD defends against both threats through four modules, namely a heterogeneous executor pool, dynamic scheduling, multi-model arbitration, and closed-loop feedback, of which the multi-model arbitration module is the core. It fuses the box-level outputs of multiple heterogeneous detectors and identifies anomalies, turning the output differences among heterogeneous models into a defense. Experiments on COCO and PASCAL VOC show that DHR-OD retains 88.9% accuracy under PGD-20 attack and reduces the BadDet backdoor attack success rate from about 100% to about 35% without a dedicated detection module, making it a defense framework that covers both adversarial and backdoor threats while not relying on prior knowledge of specific attack forms.

Keywords: object detection; adversarial attack; dynamic heterogeneous redundancy; multi-model arbitration; dynamic scheduling

1. Introduction

Deep learning is profoundly reshaping human production and daily life [1], and object detection, as a core task of computer vision based on deep learning, underpins safety-critical applications such as autonomous driving, intelligent surveillance, and industrial inspection [2]. However, object detection models are vulnerable to adversarial and backdoor attacks [3,4]. Adversarial attacks inject imperceptible perturbations at test time to induce target disappearance, false targets, or class misclassification [5], while backdoor attacks embed concealed triggers during training to output attacker-specified malicious results [6]. In scenarios such as autonomous driving, such attacks may directly endanger human life and property safety [7,8], making a robust object detection defense system an urgent need.

To counter these threats, researchers have proposed defenses along routes such as adversarial training, input preprocessing, ensemble defense, and backdoor detection [9]. Adversarial training [10,11] improves robustness only against the attack types seen during training, so its accuracy degrades on unseen attack forms. Input preprocessing [12] is

Check for updates

Academic Editor: Andrea Prati

Received: 11 October 2026
Revised: 29 October 2026
Accepted: 29 October 2026
Published: 30 October 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.

often bypassed by adaptive attacks, because its static deployment lets the attacker adapt perturbations to the fixed defense. Ensemble defense [13,14] fuses outputs of models sharing similar detection paradigms, so a perturbation that transfers across same-paradigm models still breaks the ensemble. Backdoor detection [15–17] relies on specific assumptions about trigger patterns or anomaly statistics and thus generalizes poorly to unseen triggers. In each case, the defense presupposes the attack form it protects against, which limits its robustness to attacks that deviate from that assumption.

To address this problem, we introduce Dynamic Heterogeneous Redundancy (DHR) [18] into object detection defense and propose the DHR-OD framework, whose core is a multi-model arbitration module. Because object detectors output a variable number of boxes with positional deviation, this module fuses their outputs by consensus, turning the differences among heterogeneous models into a defense. The framework does not rely on prior knowledge of specific attack forms, so its defense is not tailored to any attack seen in advance.

The main contributions of this paper are as follows:

- We introduce Dynamic Heterogeneous Redundancy into object detection defense and propose the DHR-OD framework, which defends against attacks without relying on attack prior knowledge.
- We design a three-stage box-level multi-model arbitration mechanism for the differences among multi-model outputs, including inconsistent box counts, positional deviation, and isolated detections, achieving output fusion and compromised-executor identification through IoU matching, class voting, confidence-weighted fusion, existence penalty, and anomaly scoring.
- We systematically evaluate the defense capability of DHR-OD under a gray-box threat model. Cross-model immunity analysis and ensemble adaptive attacks verify the effectiveness of heterogeneous redundancy, with a retention rate of 88.9% under PGD-20 attack on COCO and PASCAL VOC, and a reduction in the BadDet backdoor attack success rate from about 100% to about 35% without a dedicated detection module.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 details the DHR-OD method. Section 4 presents the results. Section 5 discusses the conclusions and future work.

2. Related Work

Object detection models face increasingly severe security threats, among which adversarial and backdoor attacks are particularly prominent. Existing defenses, such as adversarial training, input preprocessing, noise detection, and ensemble defense, generally rely on a single model architecture and degrade under adaptive attacks, lacking a systematic exploitation of heterogeneous redundancy and dynamic defense mechanisms.

2.1. Deep-Learning-Based Object Detection

Current detectors fall into two paradigms. Two-stage detectors, such as Faster R-CNN [19] and its variants [20,21], first generate candidate regions and then classify and regress them, achieving high accuracy but slow inference. One-stage detectors perform dense prediction directly on feature maps. The YOLO series [22,23] is fast, while Transformer-based detectors, including DETR [24], RT-DETR [25], and DINO [26], output objects end-to-end via learnable queries.

2.2. Adversarial Attacks

Early adversarial attacks were gradient-driven white-box methods such as FGSM [27] and PGD [5], which rely on model information and are mostly effective only in the digital

domain. For black-box scenarios, genetic-algorithm-based suppressive attacks [28] restrict perturbations to the target box. Physical-world attacks, such as the AdvReal framework [7], IPatch [29], and multi-view closed-box attacks [8], improve the success rate in real scenes. Improved transferable targeted attacks [30] and adaptive retention attacks (ARAs) [31] further enhance transferability and stealth. New frameworks such as LSDM [32] and TFA [33] continue to emerge.

2.3. Backdoor Attacks

Unlike adversarial attacks, backdoor attacks embed malicious behavior during training. BadNets [6] established the basic data-poisoning paradigm. Subsequent work focused on trigger concealment, including fixed-patch frequency triggers [34], FBAO [35], wavelet-embedding attacks [36], and twin-trigger generative networks [37]. At the detection-task level, BadDet [15] defines four types of detection backdoors and designs the Detector Cleanse defense, while BTPSA [38] optimizes trigger positions. SSL-OTA [39] further extends backdoor attacks to self-supervised object detection.

2.4. Defense Strategies

Defenses against adversarial and backdoor attacks can be classified into preprocessing, adversarial training, noise detection, architecture modification, ensemble defense, and certified defense [9]. AIAD [40] introduces an adversarial-intensity-awareness mechanism for multi-intensity attacks. RPU-PVB [12] combines perturbation filtering with Circle-Loss-driven robust optimization. PaSED [13] uses hypernetworks and block shuffling as a moving-target mechanism. For backdoors, Detector Cleanse [15] identifies poisoned samples through entropy detection.

2.5. Dynamic Heterogeneous Redundancy

Dynamic heterogeneous redundancy (DHR) [18] originates from intrinsic-security theory and provides a system-level defense paradigm. It deploys multiple heterogeneous executors that are functionally equivalent but implemented differently so that attacks targeting a single executor are difficult to transfer across the ensemble. The DHR architecture typically consists of a heterogeneous executor pool, a multi-model arbitration module, dynamic scheduling, and closed-loop feedback. The executor pool provides diverse attack surfaces, the arbitration module fuses executor outputs and identifies anomalies, and dynamic scheduling and closed-loop feedback rotate and repair the online executors. Unlike existing passive defenses, DHR does not rely on prior knowledge of specific attack types and is therefore not limited to attacks known in advance.

In summary, existing object detection defense methods are essentially passive defenses against known attacks, relying on prior knowledge of attack types. Their effectiveness degrades significantly when facing unknown or adaptive attacks.

3. Materials and Methods

To address the shortcomings of existing passive defenses discussed above, we design the DHR-OD framework. As shown in Figure 1, at inference time, the input image first undergoes preprocessing, and the system then selects k online detectors from the executor pool to perform parallel inference. DHR-OD consists of four modules:

- Heterogeneous executor pool, which deploys multiple detectors with mutually heterogeneous detection paradigms to provide diverse attack surfaces;
- Multi-model arbitration, which fuses the box-level outputs of online executors and identifies anomalies. Since object detection outputs a variable number of boxes with

positional deviation, this module uses IoU matching, majority voting, confidence-weighted fusion, and an existence penalty rather than simply averaging or voting;
- Dynamic scheduling, which rotates the online combination and replaces anomalous executors;
- Closed-loop feedback, which cleans and restores compromised executors.

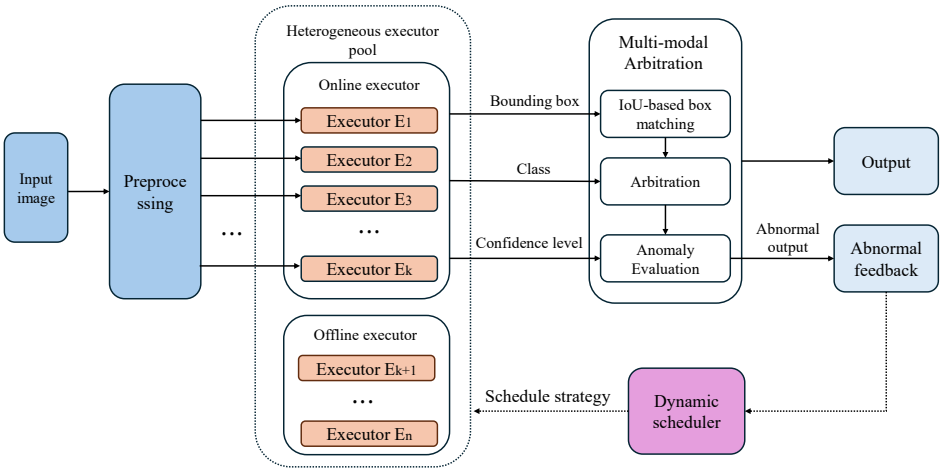


Figure 1. Overall architecture of the DHR-OD defense framework.

Threat Model.

DHR-OD targets a gray-box threat scenario, in which the attacker holds partial rather than full information about the defense system. This setting is closer to real threats than a pure black-box assumption while retaining the defender's information advantage over a pure white-box assumption [18]. Separately, to bound the defense in the most unfavorable case, we additionally evaluate in Section 4.4 a worst-case adaptiveattacker who is granted the architectures and weights of all executors. This stress test is deliberately stronger than the gray-box baseline and does not redefine it. The attacker's knowledge, goals, and capabilities are defined as follows and summarized in Table 1.

- The attacker knows that the system adopts the DHR architecture, understands that it consists of heterogeneous executors, dynamic scheduling, and multi-model arbitration, and can access some models in the executor pool through open-source model weights, literature disclosure, or model stealing, thereby generating adversarial examples or implanting backdoors. However, the attacker cannot learn in real time the current online executor combination, the scheduling rotation period, or the parameter configuration of multi-model arbitration. This invisible information constitutes the premise for the effectiveness of DHR defense.
- The first attack goal is adversarial attack, in which the attacker injects perturbations constrained by the L_∞ norm into input images at test time, inducing target disappearance, false targets, or class misclassification. The second is backdoor attack, in which the attacker implants concealed triggers with a small number of poisoned samples during training, causing the implanted executor to output malicious results when the trigger pattern appears.
- The attacker's capability is not unlimited: adversarial perturbations must satisfy the imperceptibility constraint, the poisoning rate is usually controlled at 1–5% to avoid exposure, and the number of queries is limited. These constraints jointly define the design space of the defense method.

Table 1. Threat-model conditions considered in this paper and their relation to the gray-box assumption.

Condition	Attacker Knowledge	Role
Black box	Only input–output queries; no model or defense knowledge	Lower-bound reference
Gray box	DHR architecture known; some executor weights obtainable; no online set or rotation information	Baseline (Section 3); first-mode attack
White box	All executor architectures and weights	Upper-bound reference
Adaptive	All executor weights; optimized against arbitration and scheduling	Stress test (Section 4.4); second-mode attack

Heterogeneous Executor Design.

Heterogeneity is the cornerstone of the DHR architecture, and its strength directly determines the difficulty of resisting attacks. In this paper, we construct heterogeneity at the detection-paradigm level rather than at the backbone or model scale. Object detectors follow fundamentally different design routes in target representation and positive-negative sample matching, such as anchor-based dense regression, region-proposal-based two-stage refinement, and learnable-query-based set prediction. These routes are mutually heterogeneous, so a perturbation or trigger must take effect across paradigms simultaneously, which is far harder than transferring among same-paradigm models. Accordingly, we select multiple mutually heterogeneous detection paradigms to construct the executor pool; the specific paradigm selection and model configuration are detailed in the Section 4

Multi-Model Arbitration.

Unlike image classification, the output of object detection is a variable number of boxes with positional deviation, so simply averaging or voting on model outputs is not applicable. The multi-model arbitration module addresses this challenge by fusing reliable results and identifying possibly compromised executors when box counts vary across executors, the same target has pixel-level deviation, and attacks may introduce false targets or miss real targets. The arbitration process is divided into three stages.

The first stage is IoU box matching, which determines which output boxes from different executors correspond to the same target. The intersection-over-union of two boxes is computed as

$$\mathrm{IoU}(B_i, B_j) = \frac{\mathrm{Area}(B_i \cap B_j)}{\mathrm{Area}(B_i \cup B_j)} \tag{1}$$

When the overlap is no lower than the threshold $\theta_{\mathrm{IoU}} = 0.5$, the two boxes are judged to point to the same target. All boxes are then grouped into connected components by a union-find procedure over the pairwise IoU matches so that a box overlapping any member of a component joins that component. This transitive closure assigns each box to exactly one component rather than greedy one-to-one pairing or Hungarian matching. Each component therefore corresponds to one target. Each executor has already performed non-maximum suppression (NMS) within its own inference pipeline, so it typically contributes at most one box per target. The arbitration groups the boxes of multiple executors for the same target by IoU and determines consensus by the number of distinct executors after de-duplication so that a duplicate box from a single executor is not counted multiple times as support. Components containing at least $\lceil k/2 \rceil$ executors are classified as consensus targets, and those containing fewer are classified as isolated targets. The threshold $\lceil k/2 \rceil$ is the vote count adopted in this paper rather than a strict majority: it equals the strict

majority $\lfloor k/2 \rfloor + 1$ for odd k and is one vote below it for even k. This classification underlies subsequent anomaly detection.

The second stage is per-target arbitration, which fuses the class and position of each consensus target. Because heterogeneous detectors produce confidence scores with different calibrations, the arbitration uses each executor's raw confidence without additional cross-model calibration. This is acceptable because the consensus classification depends primarily on the number of supporting executors, and the confidence-weighted fusion and anomaly scoring rely on relative comparisons among executors rather than on absolute confidence values. Class arbitration uses majority voting with the highest-confidence class prevailing in a tie. Box coordinates are weighted by each executor's confidence

$$\hat{B} = \frac{\sum_{i \in \mathcal{M}} c_i \cdot B_i}{\sum_{i \in \mathcal{M}} c_i} \tag{2}$$

so that more confident executors contribute more to the final position. For isolated targets, an existence penalty factor $\alpha = 0.5$ attenuates their confidence before comparison with the output threshold $\theta_{\text{conf}} = 0.3$. Isolated detections below the threshold are discarded as false targets. This follows the majority-consensus principle: a unique detection is more likely an attack-induced false output or a model misdetection. Ablating $\alpha \in \{0.3, 0.4, 0.5, 0.6, 0.7\}$ under PGD-20 targeting FCOS on COCO with the $k = 3$ online set consisting of FCOS, YOLOv8, and Faster R-CNN yields attacked AP_{50} of 0.381, 0.398, 0.407, 0.408, and 0.403, peaking near $\alpha = 0.5$–0.6. This indicates that $\alpha = 0.5$ is a near-optimal choice under this specific ensemble composition; the robustness of this choice across other compositions is not separately assessed.

The third stage is anomaly scoring, which evaluates the degree to which each online executor deviates from consensus along four dimensions, namely miss rate, isolated-box ratio, class-conflict rate, and confidence deviation, which are combined into an anomaly score by weighted summation

$$S_i = w_1 \cdot R^{(i)}_{\text{miss}} + w_2 \cdot R^{(i)}_{\text{isolated}} + w_3 \cdot R^{(i)}_{\text{conflict}} + w_4 \cdot \Delta^{(i)}_{\text{conf}} \tag{3}$$

where $R^{(i)}_{\text{miss}}$, $R^{(i)}_{\text{isolated}}$, $R^{(i)}_{\text{conflict}}$, and $\Delta^{(i)}_{\text{conf}}$ are the missed-consensus-target proportion, isolated-box ratio, class-conflict rate, and confidence deviation, respectively. The weights are set uniformly here and can be adjusted per scenario: the miss-rate weight can be increased in safety-critical settings to reduce missed detection, and the isolated-box ratio weight can be increased in false-alarm-sensitive settings. When the anomaly score exceeds a threshold (set to the 95th percentile of the anomaly-score distribution of normal executors on a validation set), the executor is judged suspicious and handed to the dynamic scheduling and closed-loop feedback mechanisms.

Dynamic Scheduling.

The dynamic scheduling mechanism manages the online/offline state of executors for two purposes. First, it overcomes the inherent defect of static deployment, in which attackers who figure out the fixed online combination can accumulate targeted information. Second, it takes offline a suspicious executor at the earliest opportunity after an anomaly signal, preventing its output from polluting the arbitration results. To this end, dynamic scheduling combines periodic rotation and event-driven replacement.

- Periodic rotation: At each fixed period, k models are randomly selected from the executor pool to go online, making the online combination change over time and unpredictable; the rotation period needs to balance service continuity and security.

- Event-driven replacement: It acts immediately when the arbitration module reports an anomaly, taking the abnormal executor offline and substituting a new model from the standby pool, ensuring that loss is stopped at the first moment the attack is detected.
- Hybrid strategy: Periodic rotation maintains dynamism in normal operation, while event-driven replacement quickly stops loss in anomalies, balancing scheduling unpredictability with the timeliness of anomaly isolation.

Let the executor pool size be n with online set $\mathcal{E}_{\text{on}} = \{E_1, \ldots, E_k\}$ and standby set $\mathcal{E}_{\text{off}} = \{E_{k+1}, \ldots, E_n\}$. When the arbitration module judges the online executor E_i abnormal, event-driven replacement takes it offline and selects E_{k+1} from the standby pool, i.e.,

$$\mathcal{E}_{\text{on}} \leftarrow (\mathcal{E}_{\text{on}} \setminus \{E_i\}) \cup \{E_{k+1}\}, \qquad \mathcal{E}_{\text{off}} \leftarrow (\mathcal{E}_{\text{off}} \setminus \{E_{k+1}\}) \cup \{E_i\}. \tag{4}$$

To ensure rotation does not interrupt service, the new executor is preloaded into memory before replacement, using a soft-switching add-before-delete approach to avoid inference interruption. Executors taken offline enter the cleaning queue for processing by the closed-loop feedback mechanism.

Closed-Loop Feedback.

The closed-loop feedback mechanism couples arbitration and scheduling into an adaptive defense loop, giving the system self-repair capability under sustained attacks. When an executor's anomaly score exceeds the threshold, the closed-loop process is initiated.

- Anomaly determination: The arbitration module marks the executor as suspicious and issues a replacement request to the scheduling module.
- Offline isolation: The scheduling module removes the executor from the online set to prevent its output from continuing to affect arbitration.
- Cleaning and recovery: The offline executor reloads the initial weights or is fine-tuned on a clean validation set. This removes the backdoor introduced by fine-tuning on the poisoned data, because the trusted weights contain no trigger. It does not, however, remove adversarial perturbations, which are applied to incoming images at test time rather than stored in the weights; such perturbations are instead handled by the arbitration module during inference. This paper assumes that the pretrained weights come from a trusted source, so the attacker can only inject a backdoor through data poisoning in the downstream fine-tuning stage and cannot tamper with the original weight files. Supply-chain attacks, in which the pretrained weights themselves are compromised, are outside the scope of this threat model.
- Verification and re-pooling: The cleaned executor is re-evaluated on the validation set. If its anomaly score returns to normal and detection accuracy shows no significant degradation, it rejoins the standby pool. Otherwise, it remains isolated and triggers manual review.

The threshold must balance recalling anomalies against false-alarming normal executors. We compute the anomaly-score distribution of normal executors on the validation set and set the threshold to its 95th percentile so that at most 5% of normal executors are falsely flagged.

4. Results

4.1. Experimental Setup

Datasets and Metrics.

Experiments are conducted on two standard object detection datasets, MS-COCO 2017 [41] and PASCAL VOC 2007 [42]. The former contains 118K training images and

5K validation images across 80 object classes, while the latter contains about 5K training images and 5K test images across 20 object classes.

Object detection performance is measured by Average Precision (AP). The area under the precision–recall curve is taken as the AP of that class, and the mean over all classes yields mean Average Precision (mAP). Different IoU thresholds for determining correct detection yield different AP values. This paper uses two variants: the first is AP_{50}: AP computed only at IoU threshold $\geq$0.5, which places relatively loose requirements on localization accuracy and is the official evaluation metric of PASCAL VOC; the second is $AP_{0.5:0.95}$: AP computed and averaged over IoU thresholds 0.5–0.95 (step 0.05, 10 levels in total), which is more sensitive to bounding-box localization quality and is the official primary metric of MS-COCO. This paper follows the official protocol of each dataset to ensure direct comparability with reported results, so AP (i.e., $AP_{0.5:0.95}$) is adopted on COCO, with AP_{50} also shown for a more complete picture. AP_{50} is adopted on VOC. This is because VOC has fewer classes and relatively coarse annotation boxes, so a single IoU threshold suffices to distinguish models, whereas COCO has more classes, a larger range of target scales, and finer annotations, so a single threshold cannot fully reflect localization quality, and multi-threshold averaging is needed to comprehensively characterize detection performance.

For adversarial defense metrics, let mAP_{clean} and mAP_{adv} denote the mAP under clean and attacked inputs, respectively. We adopt the absolute accuracy under attack mAP_{adv}, the absolute drop

$$\Delta mAP = mAP_{clean} - mAP_{adv}, \tag{5}$$

the relative retention rate

$$Ret. = \frac{mAP_{adv}}{mAP_{clean}} \times 100\%, \tag{6}$$

the multi-strength robust accuracy RA (the mean of mAP_{adv} across PGD step budgets), and the overall assessment index

$$OA = \frac{mAP_{clean} + RA}{2}. \tag{7}$$

For backdoor defense metrics, the Attack Success Rate (ASR) is defined as the proportion of trigger-containing test samples on which malicious behavior is executed:

$$ASR_{bd} = \frac{\text{triggered samples successfully attacked}}{\text{total triggered samples}} \times 100\%, \tag{8}$$

and the Defense Rate (DR) is defined as

$$DR = \frac{ASR_{bd}^{victim} - ASR_{bd}^{DHR}}{ASR_{bd}^{victim}} \times 100\%, \tag{9}$$

where ASR_{bd}^{victim} is the ASR of the backdoored victim executor running alone, and ASR_{bd}^{DHR} is the ASR after DHR-OD overall defense.

For dynamic defense metrics, the Dynamic Rotation Rate (DRR) is used, which is defined as the proportion of image batches that trigger at least one executor rotation, to quantify the activity of the dynamic scheduling mechanism.

Executor Configuration.

DHR-OD configures $n = 6$ heterogeneous executors covering five mutually heterogeneous detection paradigms. The specific model correspondence of each paradigm is as follows.

- Anchor-based dense regression paradigm, represented by E_4 (RetinaNet [43]), which regresses bounding boxes cell-by-cell on feature maps through preset anchors.
- Anchor-free dense regression paradigm, represented by E_1 (FCOS [44]) and E_2 (YOLOv8 [45]), which remove anchors and regress bounding boxes directly without preset anchors.
- Region-proposal refinement paradigm, represented by E_3 (Faster R-CNN R50-FPN [19]), which first proposes candidate regions and then performs classification and regression in two stages.
- Query-based set prediction paradigm, represented by E_6 (DETR [24]), which outputs the target set end-to-end via learnable queries and bipartite matching.
- Hybrid-encoding query paradigm, represented by E_5 (RT-DETR [25]), which enhances query prediction through hybrid encoding of CNN cross-scale features and Transformers.

Although E_1 (FCOS) and E_2 (YOLOv8) both belong to anchor-free dense regression, they differ in positive–negative sample assignment. FCOS uses center sampling with a center-ness branch, whereas YOLOv8 uses a task-aligned assigner with a decoupled head, providing further diversity within the paradigm. By default, $k = 3$ executors are online and $n - k = 3$ are on standby.

Implementation Details.

All six executors load official pretrained weights. YOLOv8 uses the ultralytics `yolov8l.pt` checkpoint (about 44 M parameters) and RT-DETR uses `rtdetr-l.pt` (about 32 M parameters); these two models are the backdoor attack targets (target class "person") and are fine-tuned on the poisoned data. The remaining four models are used directly for inference without fine-tuning as clean executors: Faster R-CNN uses torchvision's `fasterrcnn_resnet50_fpn_v2` (`weights=DEFAULT`, 43.7 M parameters), RetinaNet uses `retinanet_resnet50_fpn_v2` (`weights=DEFAULT`, 38.2 M parameters), FCOS uses `fcos_resnet50_fpn` (`weights=DEFAULT`, 32.3 M parameters), and DETR uses the HuggingFace `facebook/detr-resnet-50` checkpoint (41.5 M parameters).

For the backdoor fine-tuning, the backdoored models are trained for five epochs with SGD (momentum 0.937, weight decay 5×10^{-4}), a cosine learning-rate schedule without warmup from 10^{-4} to 10^{-5}, a batch size of 8, and the first 10 layers (part of the backbone) frozen.

For preprocessing, YOLOv8 and RT-DETR use the built-in ultralytics preprocessing (letterbox to 640×640, normalized to $[0, 1]$). Faster R-CNN, RetinaNet, and FCOS use torchvision's `F.to_tensor` normalization to $[0, 1]$, with ImageNet mean/std normalization performed inside the model transform, and DETR uses the HuggingFace `DetrImageProcessor` (ImageNet mean/std normalization).

The input resolution is 640×640 for YOLOv8 and RT-DETR, and the shorter side is rescaled to 800 with the longer side capped at 1333 for Faster R-CNN, RetinaNet, FCOS, and DETR. Data augmentation applies only to the backdoor fine-tuning of YOLOv8 and RT-DETR, which uses the ultralytics defaults (mosaic, random horizontal flip, multi-scale, HSV perturbation, etc.), and the other four models are inference-only with no augmentation.

All experiments run on a single NVIDIA RTX 4090 (24GB) with PyTorch 2.5.1 (CUDA 12.4) and Python 3.12, using ultralytics 8.4.92 (YOLOv8, RT-DETR), torchvision 0.20.1 (Faster R-CNN, RetinaNet, FCOS), and HuggingFace transformers 5.13.0 (DETR).

Attack Settings.

All adversarial attacks in this paper are based on PGD-20 with $\epsilon = 8/255$, $\alpha = 2/255$, and the L_∞ constraint. FGSM ($\epsilon = 8/255$) is also evaluated as a weak baseline, along with a multi-strength scan over step budgets $\{2, 4, 8, 16, 20\}$. The perturbation budget

$\epsilon = 8/255$ means allowing a perturbation magnitude of at most $8/255$ on pixel values normalized to $[0, 1]$. This value is a standard default in the adversarial attack field: it ensures that perturbations are visually imperceptible while being sufficient to cause significant performance degradation in undefended detectors, and it is a widely adopted convention in object detection robustness research [12,40], so adopting it guarantees direct comparability with reported results. The step size $\alpha = 2/255$, a quarter of ϵ, allows 20 iterations to fully explore the feasible perturbation space within the ϵ-ball while avoiding the attack crossing the decision boundary too early with an overly large step or insufficient convergence with an overly small step. Depending on the attacker's model knowledge, we set the following two attack modes.

In the first mode, the attacker generates perturbations targeting only FCOS, which is the most vulnerable executor in the pool. This mode examines two points, namely the destructiveness of the perturbation to FCOS itself and its transferability to other heterogeneous executors. If the perturbation is nearly ineffective against other paradigms, the consensus mechanism of multi-model arbitration can filter it out. This corresponds to the PGD comparison in Section 4.2 and the cross-model immunity experiments in Section 4.3.

In the second mode, the attacker holds the architectures and weights of all $n = 6$ executors but does not know the real-time online combination. The attack therefore targets one executor at a time, which is equivalent to taking the expectation over the uniform distribution of the online set:

$$\max_{\delta} \ \mathbb{E}_{i\sim\mathrm{Uniform}(\mathcal{E}_{\mathrm{on}})} \ \mathcal{L}_{\mathrm{suppress}}\big(f_i(x+\delta)\big), \quad \text{s.t. } \|\delta\|_\infty \le \epsilon, \tag{10}$$

where $\mathcal{L}_{\mathrm{suppress}}$ is the detection-suppression loss. Although an ideal adaptive attack would optimize the fused arbitration output end-to-end, i.e., $\min_\delta \ \mathbb{E}_{\mathcal{E}_{\mathrm{on}}} \ \mathcal{L}_{\mathrm{det}}\big(\mathcal{A}(\{f_i(x+\delta)\}), y^*\big)$, the IoU matching and class voting inside the arbitration are non-differentiable operators, so the joint gradient degenerates almost everywhere. Moreover, the adversarial transferability among heterogeneous executors is low, and a bounded perturbation can hardly suppress the majority of executors simultaneously. We therefore adopt the single-executor white-box PGD (a suppression attack against a randomly selected executor) as the attack baseline, whose random selection is equivalent to taking the expectation over the uniform distribution of the online set, i.e., the attacker does not know the real-time online combination and does not perform end-to-end joint optimization against the multi-model arbitration. This corresponds to the adaptive attack experiments in Section 4.4. This second mode is a worst-case white-box stress test that deliberately exceeds the gray-box assumption defined above in order to verify that DHR-OD remains effective even when the attacker holds full knowledge of every executor.

Regarding the defense mechanism, DHR-OD's defense does not depend on which executor the attack targets. Multi-model arbitration performs a consistency comparison on the outputs of online executors, and targets detected by only a few executors are attenuated by the existence penalty, so false outputs induced by single-victim-model attacks are naturally filtered out. For attacks against all models, the periodic rotation and event-driven replacement of dynamic scheduling continuously change the online combination, and the closed-loop feedback takes offline and cleans the corresponding executor when the anomaly score exceeds the threshold, making it difficult for attackers to keep perturbations continuously hitting a fixed online combination.

Regarding the backdoor attack settings, BadDet OGA [15] (checkerboard trigger, 5% poisoning) and BadNets [6] (corner-patch trigger, 1%/5% poisoning) are used, with backdoors implanted into a single executor separately, to examine whether DHR-OD can suppress malicious outputs through heterogeneous arbitration alone without relying on

backdoor-specific detection modules. All attacks are implemented through the ARES library [46].

Baselines.

This paper compares DHR-OD with three categories of existing defense methods and two ablation configurations.

- Single-model adversarial defenses, RPU-PVB [12], RobustDet (with the CFR module) [11], AIAD [40], CWAT [10], and MTD [47], which are used to verify the advantage of DHR-OD over model-level robustness optimization.
- Ensemble defense, HeteRobust [14], which is used to verify the incremental contribution of dynamic scheduling within heterogeneous ensembles.
- Backdoor defenses, Detector Cleanse [15], TRACE [16], and ODSCAN [17], which are used to verify the general defense capability of DHR-OD without a dedicated backdoor detection module.
- Ablation configurations, Static Ensemble (fixed online combination, no dynamic scheduling) and Single Best (the best single executor under attack), which are used to decompose the respective contributions of heterogeneity and dynamic scheduling.

All comparisons adopt matched attack settings, and the baseline values from the literature are taken from the original papers' reports [10–12,14,40,47].

4.2. PGD Attack Defense Comparison

Table 2 and Figure 2 show the comparison under PGD-20 attack on COCO. DHR-OD achieves an AP of 0.367 under attack, 0.108 higher than the 0.259 of RPU-PVB, which is the best single-model method. All single-model defense methods have retention rates below 72%, whereas DHR-OD, relying on consensus arbitration among heterogeneous executors, confines the damage of the attack to individual models within a range tolerable to the overall system. This shows that the robustness gain brought by heterogeneous redundancy exceeds the upper bound achievable by any single model's adversarial training. At the same time, DHR-OD's clean AP is higher than RPU-PVB's, indicating that the consensus arbitration maintains a clean detection of real targets while filtering out attack-induced false outputs.

Table 2. Comparison with existing mainstream methods under PGD-20 attack on COCO (FCOS as the victim detector, $\epsilon = 8/255$). † Literature values are the mean of 3–4 PGD variants reported in [12].

Method	Arch.	Clean AP	Attack AP †	Drop (%)	Ret. (%)
DHR-OD (Ours)	6 Het. (CNN+TF), 3 on	0.413	0.367	−11.1	88.9
RPU-PVB [12]	SSD (ResNet-18)	0.362	0.259	−28.5	71.5
RobustDet [11]	SSD + discriminator	0.367	0.213	−42.0	58.0
CWAT [10]	SSD	0.237	0.130	−45.1	54.9
MTD [47]	SSD	0.242	0.114	−52.9	47.1

Table 3 and Figure 3 show the comparison under PGD attack on VOC. DHR-OD's multi-strength robust accuracy RA is 0.618, which is slightly better than RPU-PVB's 0.605. Notably, RPU-PVB's clean AP_{50} is higher than DHR-OD's, which is the inherent accuracy advantage of a dedicated SSD single detector on clean data, but DHR-OD reverses the gap under attack by virtue of heterogeneous redundancy, showing that its defense capability does not depend on the absolute accuracy of a single model but comes from the consistency fusion of outputs from multiple heterogeneous models. This result is consistent with COCO, verifying that DHR-OD's robustness gain holds stably on both datasets.

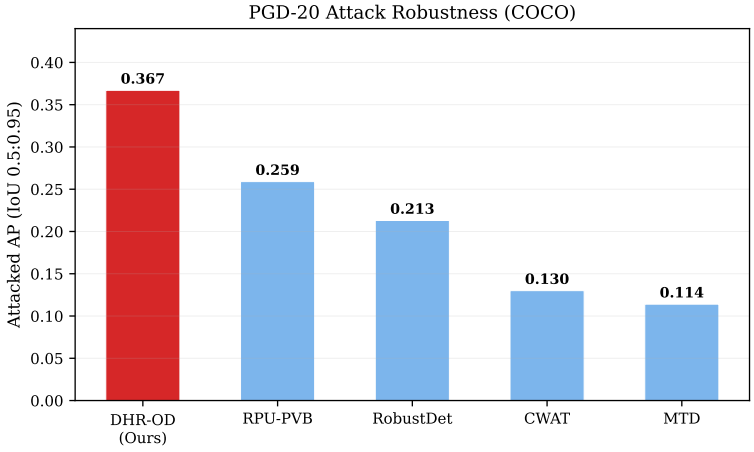


Figure 2. PGD-20 attack robustness comparison on COCO (AP, IoU 0.5:0.95). DHR-OD (red) compared with four single-detector defenses.

Table 3. Comparison with SOTA under PGD attack on VOC. DHR-OD RA is the mean over step budgets 2/4/8/16/20. † Literature RA is the mean of PGD variants in [12,40].

Method	Arch.	Clean AP_{50}	RA †	Ret. (%)	OA
DHR-OD (Ours)	6 Het. (CNN+TF), 3 on	0.701	0.618	88.1	0.660
RPU-PVB [12]	SSD (ResNet-18)	0.730	0.605	82.9	0.668
RobustDet (CFR) [11]	SSD + CFR	0.748	0.499	66.7	0.624
RobustDet [11]	SSD + discriminator	0.754	0.453	60.1	0.604
AIAD [40]	SSD (VGG-16)	0.764	0.497	65.1	0.631
CWAT [10]	SSD	0.513	0.323	63.0	0.418
MTD [47]	SSD	0.480	0.269	56.0	0.375

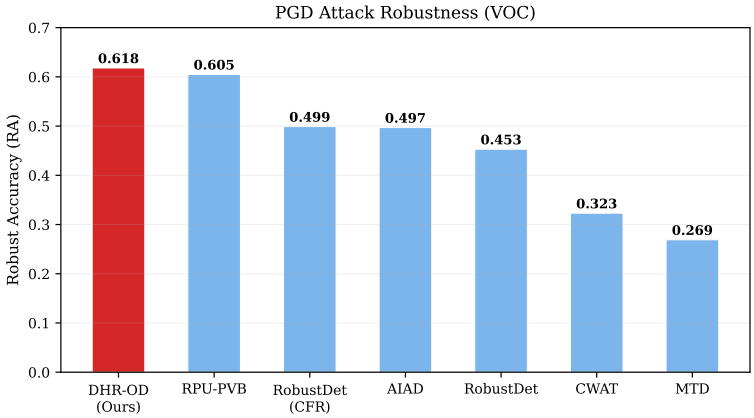


Figure 3. PGD attack robustness comparison on VOC.

4.3. Cross-Model Heterogeneous Immunity Analysis

Table 4 and Figures 4 and 5 show the transfer effect of PGD-20 attack with FCOS as the victim model on detectors of different architectures, which is used to directly verify the premise of the heterogeneous redundancy principle. FCOS itself drops 40.2% on COCO, while YOLOv8 drops only 1.3% and RT-DETR only 8.8%, forming a difference of nearly 40 percentage points. This difference shows that perturbations optimized for a certain detection paradigm cannot effectively transfer to executors of different paradigms, and the attacker's destructive power is naturally confined to a few executors. This pattern is highly consistent across datasets, with a Pearson correlation coefficient of $r = 0.94$ among the drop values of different models, proving that heterogeneous immunity is not an accidental phenomenon of a particular dataset but is determined by the structural differences of

detection paradigms. The transfer vulnerability ranking (FCOS $\gg$ DETR $>$ RT-DETR $>$ RetinaNet $>$ Faster R-CNN $>$ YOLOv8) further shows that the transfer between FCOS and YOLOv8, both CNN one-stage detectors, is still negligible, indicating that the immunity difference comes from the heterogeneous design of detection mechanisms rather than the backbone network type. It is precisely this transfer gap that enables DHR-OD's multi-model arbitration to take effect, buffering the victim model's degradation from 40.2% to 11.3% through the consensus mechanism: that is, the overall robustness of the system does not depend on the most vulnerable executor but rather on the collective resilience of the heterogeneous ensemble.

Table 4. Heterogeneous immunity analysis: cross-model transfer of PGD-20 attack. The attack is generated specifically against FCOS.

	COCO AP_{50}			VOC AP_{50}		
Detector	**Clean**	**Attack**	**Drop (%)**	**Clean**	**Attack**	**Drop (%)**
FCOS (victim)	0.597	0.357	−40.2	0.753	0.464	−38.4
YOLOv8	0.596	0.588	−1.3	0.806	0.799	−0.8
RT-DETR	0.681	0.621	−8.8	0.838	0.830	−1.0
Faster R-CNN	0.690	0.664	−3.8	0.836	0.805	−3.6
RetinaNet	0.630	0.595	−5.6	0.796	0.748	−6.1
DETR	0.632	0.555	−12.1	0.793	0.708	−10.8
DHR-OD (Ours)	0.576	0.512	−11.3	0.701	0.613	−12.6

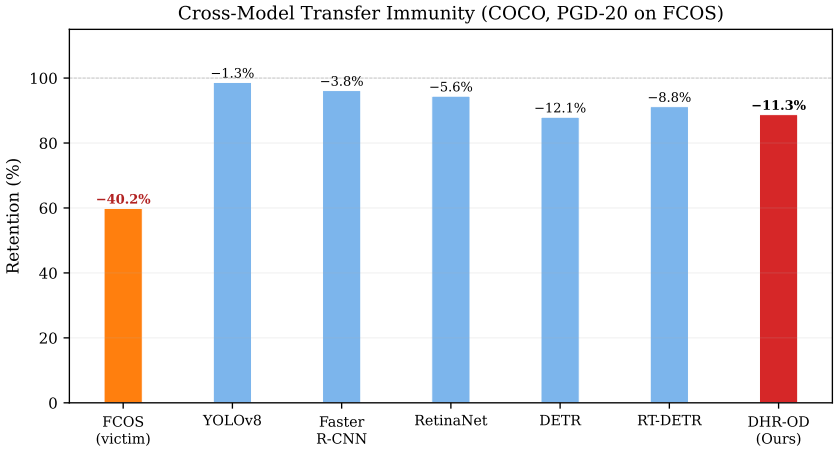


Figure 4. Cross-model transfer immunity on COCO (PGD-20 targeting FCOS). Label = drop (%). The red bold value marks the victim detector (FCOS), which shows the largest drop.

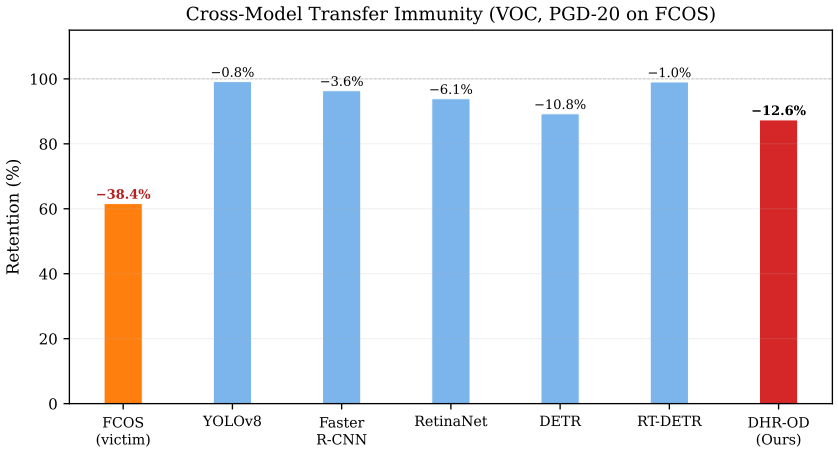


Figure 5. Cross-model transfer immunity on VOC (PGD-20 targeting FCOS). The red bold value marks the victim detector (FCOS), which shows the largest drop.

4.4. Adaptive and Backdoor Attacks

Figure 6 shows the first round of the adaptive attack, where the attacker holds the weights of all six executors and optimizes perturbations against a single executor. Across the full five rounds of the adaptive attack, DHR-OD and Static Ensemble perform comparably, each winning in different rounds. DHR-OD's advantage over Static Ensemble is therefore not a consistently higher single-round AP but rather the dynamic closed loop it enables. The anomaly score identifies the compromised executor, the scheduler isolates it, and the feedback mechanism recovers clean weights, capabilities that Static Ensemble, which only fuses outputs, does not possess. Because robustness depends on the fraction of high-transfer models in the online set, the scheduler can also rotate out a jointly attacked high-transfer model, whereas the fixed set of the static ensemble cannot.

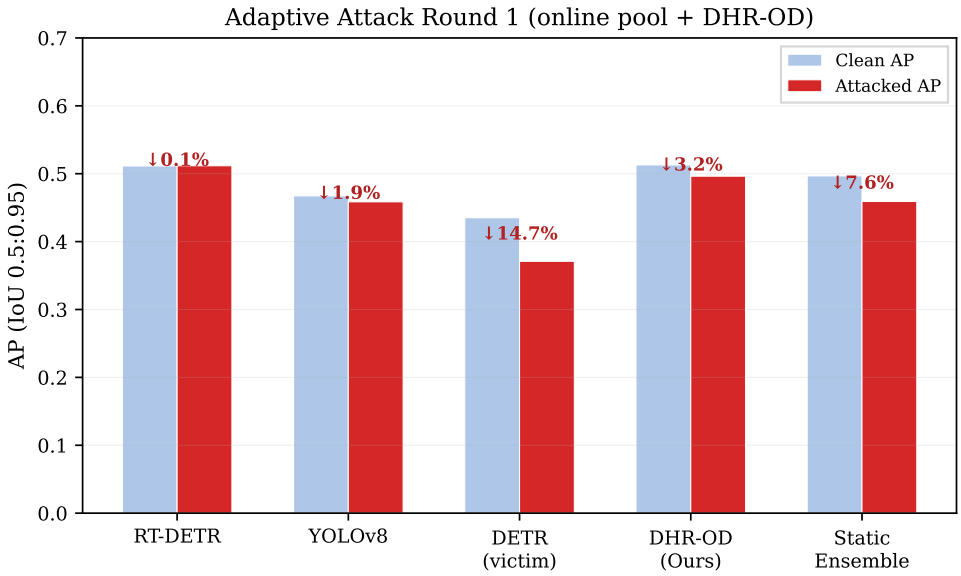


Figure 6. One of five rounds of the adaptive attack on COCO (online pool: RT-DETR, YOLOv8, DETR; victim: DETR). DHR-OD mitigates the victim model's vulnerability through dynamic arbitration, while Static Ensemble benefits from heterogeneity but lacks dynamic scheduling. The downward arrow (↓) annotation between each pair of bars marks the percentage drop from clean to attacked AP.

Across the five rounds, the drop of DHR-OD and the Static Ensemble are −3.2% and −7.6% (round 1), −10.1% and −2.5% (round 2), −12.5% and −6.3% (round 3), −31.6% and −28.1% (round 4), and −22.6% and −16.5% (round 5). The dynamic scheduler performed four switches in total, namely two anomaly-driven replacements that both correctly identified the attacked executor (rounds 2 and 3) and two periodic rotations (rounds 1 and 4) with no erroneous substitution.

Compared with two alternative fusion methods on the same executor pool under the same PGD-20 attack targeting FCOS, the attacked AP_{50} of the arbitration, simple averaging, and weighted voting are 0.407, 0.435, and 0.377. Simple averaging is slightly higher than the arbitration in this single-frame snapshot; the contribution of the arbitration is therefore not a higher single-frame AP but the anomaly score it produces, which enables compromised-executor identification and closed-loop recovery, whereas the alternative fusion methods do not provide such a signal.

Figure 7 shows the backdoor defense results with YOLOv8 and RT-DETR as the attack targets. DHR-OD reduces the BadDet OGA attack success rate from about 100% to about 35%, with defense rates of 65.1% (YOLOv8) and 63.6% (RT-DETR), and the impact on clean AP is $< 1\%$. Although this reduction is substantial, a residual ASR of about 35% is still high for safety-critical systems, indicating that DHR-OD partially mitigates rather than

fully eliminates the backdoor threat in the considered scenario. The key to this result is that the defense effect comes entirely from the general arbitration mechanism rather than any dedicated backdoor detection module: the false targets produced by the backdoored executor on trigger-containing inputs are supported by only a single executor and are attenuated below the threshold by the existence penalty. Unlike Detector Cleanse [15], which relies on entropy detection, TRACE [16], which relies on activation statistics, and ODSCAN [17], which relies on weight scanning, DHR-OD does not need to know the form of the backdoor in advance, showing that heterogeneous redundancy provides general defense capability across threat types. The same consensus arbitration mechanism can handle adversarial perturbations at test time and suppress backdoors implanted during training, which is something single-threat-specific defense methods do not possess.

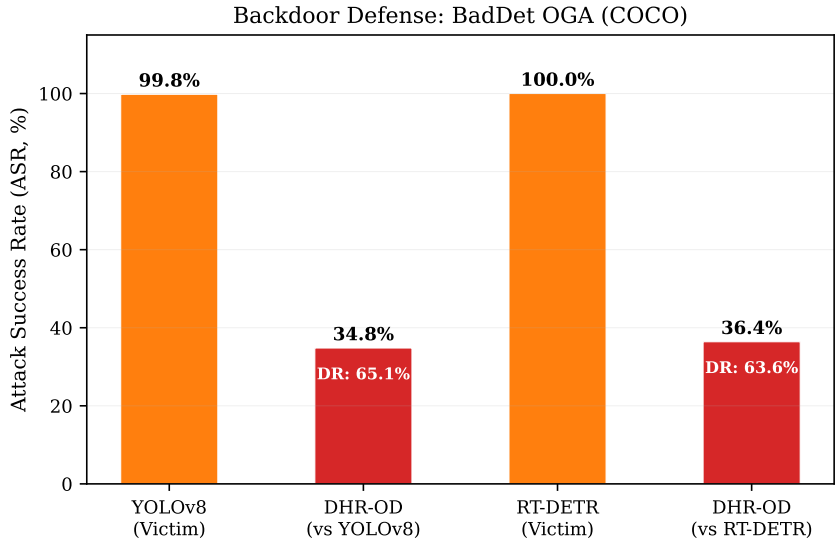


Figure 7. BadDet OGA backdoor defense on COCO.

Number of online executors k. Under PGD-20 targeting FCOS on COCO (Figure 8), the clean/attacked AP_{50} for $k = 3, 4, 5, 6$ are 0.578/0.407, 0.504/0.349, 0.491/0.312, and 0.518/0.381. The online sets are built by adding executors one by one, namely FCOS, YOLOv8, and Faster R-CNN for $k = 3$, then RetinaNet for $k = 4$, then DETR for $k = 5$, and finally RT-DETR for $k = 6$. The clean AP_{50} therefore varies with the composition of the online set rather than with the threshold alone: it decreases when a weaker or more divergent executor is added, and it recovers at $k = 6$ when the strong RT-DETR joins the pool. The default $k = 3$ attains the highest clean AP_{50}, which is the best accuracy–redundancy trade-off. Table 4 is a heterogeneous-immunity analysis that exhibits the cross-model transfer gap, and its DHR-OD row reports the consensus over the full pool of all six executors; this evaluation was conducted before the online-executor count k was made a tunable parameter, so the consensus threshold retained the then-default value $\lceil 3/2 \rceil = 2$. The k-scan here is a separate, controlled study of k in which the threshold follows $\lceil k/2 \rceil$ for each k. The attacked AP_{50} of 0.512 in Table 4 therefore reflects the full-pool consensus under the default threshold, whereas the $k = 3$ value of 0.407 reflects three online executors (FCOS, YOLOv8, Faster R-CNN) under the same threshold, and the $k = 6$ value of 0.381 reflects all six executors under the threshold $\lceil 6/2 \rceil = 3$; the three numbers thus come from different arbitration configurations serving different analyses rather than from one inconsistent setting.

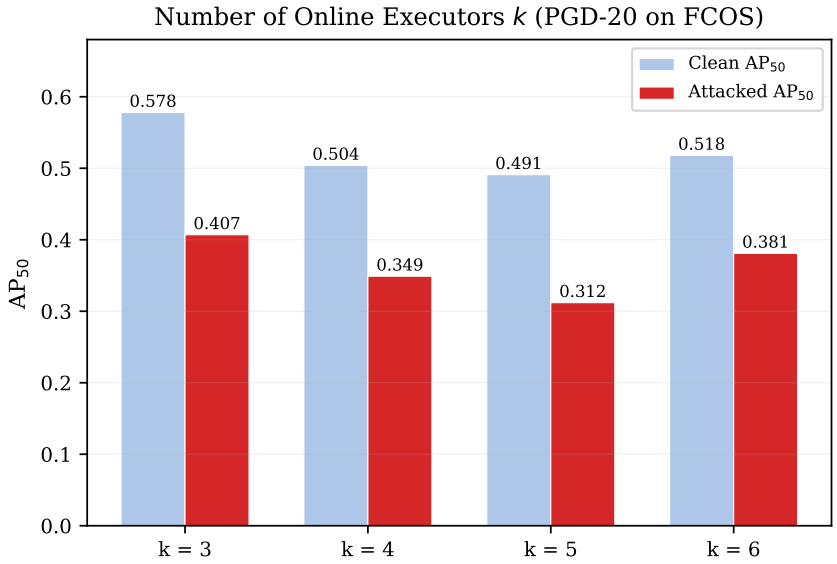


Figure 8. Number of online executors k: clean and attacked AP_{50} under PGD-20 targeting FCOS on COCO. The default $k = 3$ attains the highest clean AP_{50}.

Uncertainty. Across all $\binom{6}{3} = 20$ online sets of $k = 3$ executors under PGD-20 targeting FCOS on COCO (Figure 9), the attacked AP_{50} ranges from 0.119 to 0.533 (mean $\pm$ std $= 0.374 \pm 0.115$). This variance is structural rather than random noise: sets containing the victim FCOS (mean 0.323) or the high-transfer DETR (mean 0.322) attain markedly lower AP_{50} than sets without them (mean 0.424 and 0.426), which is precisely why the dynamic scheduler should rotate out such models.

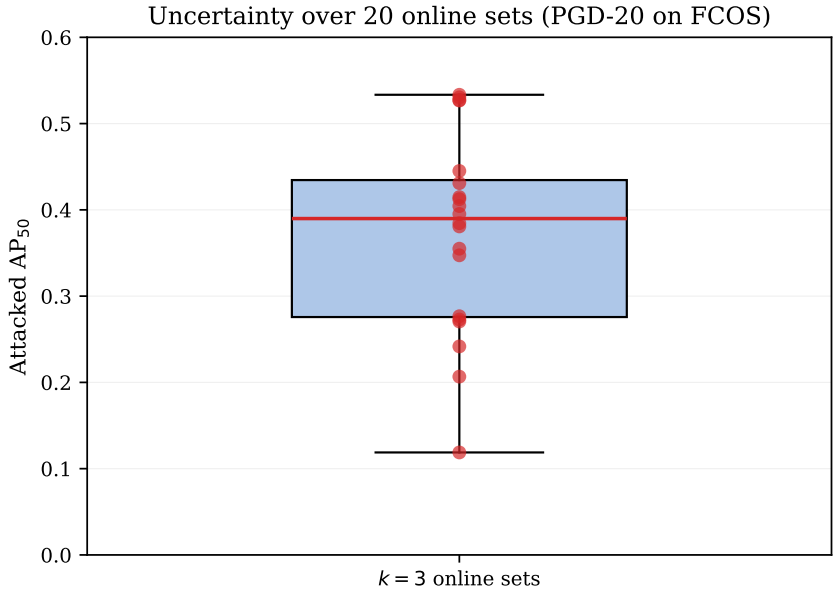


Figure 9. Attacked AP_{50} over all $\binom{6}{3} = 20$ online sets of $k = 3$ executors under PGD-20 targeting FCOS on COCO. The spread (0.119 to 0.533) is structural rather than random, determined by the fraction of compromised and high-transfer models in the online set, which motivates dynamic scheduling.

Table 5 compares the defense coverage of each method across multiple dimensions. DHR-OD is the only method covering both adversarial and backdoor threats, with the largest heterogeneous detector pool and unique multi-strength evaluation, covering six of the seven evaluation dimensions, while RPU-PVB, ranked second, covers only three. This comparison shows that DHR-OD's defense coverage is not a result of the evaluation design but a direct reflection of its architectural characteristics, because DHR-OD is not optimized

for any specific attack type, and the threat types it can defend against are determined by the diversity of the executor pool rather than by the training data, which is precisely the advantage of system-level defense over model-level defense in terms of generalization.

Table 5. Cross-threat defense capability comparison. DHR-OD is the only method covering both adversarial and backdoor threats. Multi-strength evaluation is unique to DHR-OD. A check mark and a dash indicate supported and unsupported capabilities, respectively.

Method	Adv. Defense	Backdoor	Datasets	Het. Detectors	Dynamic	RA/OA
DHR-OD (Ours)	✓	✓	2	6	✓	✓
RPU-PVB [12]	✓	–	2	1	–	✓
HeteRobust [14]	✓	–	2	6	–	–
Backbone Imp. [48]	✓	–	2	3	–	–
TRACE [16]	–	✓	3	3	–	–
ODSCAN [17]	–	✓	3	4	–	–

5. Discussion

This paper addresses the vulnerability of object detection models under adversarial and backdoor attacks and the fact that existing defense methods are mostly passive and fail to defend against unknown attacks. We introduce Dynamic Heterogeneous Redundancy into object detection and propose the DHR-OD framework. Its core contribution is a three-stage multi-model arbitration module that fuses the box-level outputs of heterogeneous detectors and identifies anomalies, turning the output differences among heterogeneous models into a defense. DHR-OD therefore does not rely on attack-specific prior knowledge, so its defense is not restricted to the attack forms used in the evaluation. Experiments on COCO and PASCAL VOC show that DHR-OD retains 88.9% accuracy under PGD-20 attack and reduces the BadDet backdoor attack success rate from about 100% to about 35% without a dedicated detection module. We emphasize, however, that the residual backdoor ASR of about 35% remains non-negligible, so DHR-OD mitigates but does not eliminate backdoor threats in the studied setting.

Although the experiments evaluate specific attack forms (PGD-20, FGSM, BadDet, BadNets), the independence from attack-specific prior knowledge is evidenced indirectly. The cross-model immunity analysis shows that an attack optimized against one executor transfers poorly to executors never targeted by it, and the same arbitration mechanism, configured without any attack knowledge, simultaneously handles test-time adversarial perturbations and training-time backdoors. We therefore claim not defense against every possible unknown attack but only that the defense does not depend on the attack forms used in the evaluation.

Future work can proceed along two directions. First, explore the defense effectiveness of the DHR architecture under physical-world attacks and more diverse backdoor trigger forms. Second, introduce more diverse sources of heterogeneity such as segmentation models and reduce the deployment overhead of multiple executors through model quantization and shared feature extraction, bringing DHR-OD closer to the real-time and resource constraints of practical deployment.

Author Contributions: Conceptualization, F.T. and X.C.; methodology, F.T.; software, F.T.; validation, F.T.; formal analysis, F.T.; investigation, F.T.; resources, X.C.; data curation, F.T.; writing—original draft preparation, F.T.; writing—review and editing, X.C.; visualization, F.T.; supervision, X.C.; project administration, X.C.; funding acquisition, X.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Hainan Seed Industry Laboratory, grant number B24H10035.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available in the MS-COCO repository at https://cocodataset.org/ (accessed on 30 September 2026) and the PASCAL VOC repository at http://host.robots.ox.ac.uk/pascal/VOC/ (accessed on 30 September 2026). These data were derived from the following resources available in the public domain: MS-COCO (https://cocodataset.org/, accessed on 30 September 2026) and PASCAL VOC (http://host.robots.ox.ac.uk/pascal/VOC/, accessed on 30 September 2026).

Acknowledgments: During the preparation of this manuscript, the authors used a large language model for language polishing and code review. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

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

References

1. Tian, C.; Cheng, T.; Peng, Z.; Zuo, W.; Tian, Y.; Zhang, Q.; Wang, F.Y.; Zhang, D. A survey on deep learning fundamentals. *Artif. Intell. Rev.* **2025**, *58*, 381. https://doi.org/10.1007/s10462-025-11368-7.
2. Zou, Z.; Chen, K.; Shi, Z.; Guo, Y.; Ye, J. Object Detection in 20 Years: A Survey. *Proc. IEEE* **2023**, *111*, 257–276. https://doi.org/10.1109/JPROC.2023.3238524.
3. Buckner, C. Understanding adversarial examples requires a theory of artefacts for deep learning. *Nat. Mach. Intell.* **2020**, *2*, 731–736. https://doi.org/10.1038/s42256-020-00266-y.
4. Woods, W.; Chen, J.; Teuscher, C. Adversarial explanations for understanding image classification decisions and improved neural network robustness. *Nat. Mach. Intell.* **2019**, *1*, 508–516. https://doi.org/10.1038/s42256-019-0104-6.
5. Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; Vladu, A. Towards deep learning models resistant to adversarial attacks. *arXiv* **2017**, arXiv:1706.06083.
6. Gu, T.; Dolan-Gavitt, B.; Garg, S. BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain. *arXiv* **2017**, arXiv:1708.06733.
7. Huang, Y.; Ren, Y.; Wang, J.; Huo, L.; Bai, X.; Zhang, J.; Yu, H. AdvReal: Physical adversarial patch generation framework for security evaluation of object detection systems. *Expert Syst. Appl.* **2026**, *296*, 128967.
8. Zhang, Y.; Yu, Z.; Yin, Z.; Ye, O.; Cong, X.; Song, H.H. Closed-Box Adversarial Attack Method for Object Detection Under Multiview Conditions. *IEEE Internet Things J.* **2025**, *12*, 27886–27900.
9. Thunuguntla, A.; Tadepalli, P.; Raffa, G. Defenses Against Adversarial Attacks on Object Detection: Methods and Future Directions. *Information* **2025**, *16*, 1003.
10. Chen, P.C.; Kung, B.H.; Chen, J.C. Class-Aware Robust Adversarial Training for Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 10420–10429.
11. Dong, Z.; Wei, P.; Lin, L. Adversarially-Aware Robust Object Detector. In Proceedings of the European Conference on Computer Vision (ECCV), Tel Aviv, Israel, 23–27 October 2022; pp. 297–313.
12. Yang, H.; Wang, X.; Chen, Y.; Dou, H.; Zhang, Y. RPU-PVB: robust object detection based on a unified metric perspective with bilinear interpolation. *J. Cloud Comput.* **2023**, *12*, 169.
13. Wu, Y.; Guo, D.; Luo, X.; Song, Q.; Tan, R. Demo Abstract: Parameterized Stochastic Ensemble Defense for Object Detection. In Proceedings of the 23rd ACM Conference on Embedded Networked Sensor Systems, Irvine, CA, USA, 6–9 May 2025; pp. 710–711.
14. Wu, Y.; Chow, K.H.; Wei, W.; Liu, L. Exploring Model Learning Heterogeneity for Boosting Ensemble Robustness. In Proceedings of the IEEE International Conference on Data Mining (ICDM), Shanghai, China, 1–4 December 2023.
15. Chan, S.; Dong, Y.; Zhu, J.; Zhang, X.; Zhou, J. BadDet: Backdoor Attacks on Object Detection. In Proceedings of the European Conference on Computer Vision (ECCV) Workshops, Tel Aviv, Israel, 23–27 October 2022; pp. 396–412.
16. Zhang, H.; Wang, Y.; Yan, S.; Zhu, C.; Zhou, Z.; Hou, L.; Hu, S.; Li, M.; Zhang, Y.; Zhang, L.Y. TRACE: Test-Time Backdoor Detection for Object Detection Models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 11–15 June 2025.
17. Cheng, S.; Shen, G.; Tao, G.; Zhang, K.; Zhang, Z.; An, S.; Xu, X.; Liu, Y.; Ma, S.; Zhang, X. ODSCAN: Backdoor Scanning for Object Detection Models. In Proceedings of the IEEE Symposium on Security and Privacy (S&P), San Francisco, CA, USA, 20–23 May 2024.
18. Zhang, F.; Chen, X.; Huang, W.; Wu, J.; Zhang, Z.; Zhou, C.; Li, J.; Peng, Z.; Guo, W.; Yang, G.; et al. Harnessing dynamic heterogeneous redundancy to empower deep learning safety and security. *Secur. Saf.* **2024**, *3*, 2024011.
19. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015.

20. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22-29 October 2017; pp. 2961–2969.
21. Cai, Z.; Vasconcelos, N. Cascade R-CNN: Delving into High Quality Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 6154–6162.
22. Wang, C.Y.; Yeh, I.H.; Mark Liao, H.Y. Yolov9: Learning what you want to learn using programmable gradient information. In Proceedings of the European Conference on Computer Vision, Milan, Italy, 29 September–4 October 2024; pp. 1–21.
23. Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J.; Ding, G. Yolov10: Real-time end-to-end object detection. *Adv. Neural Inf. Process. Syst.* **2024**, 107984–108011.
24. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 213–229.
25. Zhao, Y.; Lv, W.; Xu, S.; Wei, J.; Wang, G.; Dang, Q.; Liu, Y.; Chen, J. RT-DETR: DETRs Beat YOLOs on Real-Time Object Detection. *arXiv* **2023**, arXiv:2304.08069.
26. 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* **2022**, arXiv:2203.03605.
27. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples. *arXiv* **2014**, arXiv:1412.6572.
28. Chan, K.H.; Cheng, B.H. Evoattack: suppressive adversarial attacks against object detection models using evolutionary search. *Autom. Softw. Eng.* **2025**, *32*, 3.
29. Mirsky, Y. IPatch: a remote adversarial patch. *Cybersecurity* **2023**, *6*, 18. https://doi.org/10.1186/s42400-023-00145-0.
30. Ding, Z.; Sun, L.; Mao, X.; Dai, L.; Ding, R. Improving transferable targeted adversarial attack for object detection using RCEN framework and logit loss optimization. *Comput. Mater. Contin.* **2024**, *80*, 4387.
31. Gao, X.; Jin, X.; Miao, S.; Jiang, Q.; Dong, Y.; Zhou, W.; Yao, S. A Security Study of Multimodel Artificial Intelligence System: Adaptive Retention Attack for Object Detection System with Multifocus Image Fusion Model. *Adv. Intell. Syst.* **2024**, *6*, 2300771.
32. Wang, W.; Qi, H.; Huang, Z.; Yin, B.; Wang, P. Latent-space diffusion models for stealthy and transferable adversarial attacks on object detection. *Neurocomputing* **2025**, *656*, 131456.
33. Zhang, X.; Sun, C.; Han, H.; Wang, H.; Sun, H.; Zheng, N. Object-fabrication targeted attack for object detection. *Neurocomputing* **2025**, *627*, 129561.
34. Xue, S.; Du, J.; Huang, W.; Yang, Q. A Backdoor Attack via Fixed Patch Triggers in Frequency Domain. In Proceedings of the MultiMedia Modeling, Prague, Czech Republic, 29–31 January 2026; pp. 31–44.
35. Wang, Q.; Shen, H.; Wang, L.; Yuan, L.; Ren, Y.; Jia, X.; Sun, S.; Meng, W. FBAO: backdoor attack against object detection via frequency noise injection. *Appl. Intell.* **2026**, *56*, 134.
36. Shen, M.; Huang, R. Backdoor Attacks with Wavelet Embedding: Revealing and enhancing the insights of vulnerabilities in visual object detection models on transformers within digital twin systems. *Adv. Eng. Inform.* **2024**, *60*, 102355.
37. Li, Z.; Liu, Z.; Geng, G.; Gowda, S.N.; Lin, S.; Weng, J.; Jin, X. Twin trigger generative networks for backdoor attacks against real-time object detection. *Appl. Soft Comput.* **2026**, *186*, 114200.
38. Jo, S.; Cho, Y. An enhanced backdoor attack using a backdoor trigger position searching algorithm for avoiding deep learning-based object detection systems. *Cybersecurity* **2026**, *9*, 128.
39. Wang, Q.; Yin, C.; Fang, L.; Zhou, L.; Liu, Z.; Wang, R.; Lin, C. SSL-OTA: Unveiling Backdoor Threats in Self-Supervised Learning for Object Detection. *arXiv* **2024**, arXiv:2401.00137.
40. Cheng, J.; Huang, B.; Fang, Y.; Han, Z.; Wang, Z. Adversarial intensity awareness for robust object detection. *Comput. Vis. Image Underst.* **2025**, *251*, 104252.
41. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In Proceedings of the European Conference on Computer Vision (ECCV), Zurich, Switzerland, 6–12 September 2014; pp. 740–755.
42. Everingham, M.; Van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes (VOC) Challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338.
43. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988.
44. 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), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9627–9636.
45. Jocher, G.; Chaurasia, A.; Qiu, J. Ultralytics YOLOv8. 2023. Available online: https://github.com/ultralytics/ultralytics (accessed on 30 September 2026).
46. Dong, Y.; Fu, Q.A.; Yang, X.; Pang, T.; Su, H.; Xiao, Z.; Zhu, J. Benchmarking Adversarial Robustness on Image Classification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 321–331.

47. Zhang, H.; Wang, J. Towards Adversarially Robust Object Detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 421–430.
48. Li, X.; Chen, H.; Hu, X. On the Importance of Backbone to the Adversarial Robustness of Object Detectors. *IEEE Trans. Inf. Forensics Secur. (TIFS)* **2025**, *20*, 2387–2398.

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.