AdROD: HyperNetwork-based Adversarially Robust Object Detection for Autonomous Driving

Yuting Wu, Dongfang Guo, Xiangzhong Luo, Qun Song, and Rui Tan

***Abstract*—Camera-based object detectors are vulnerable to physical adversarial attacks designed to suppress detections. While adversarial training and input purification offer some protection, they often overfit to specific attack distributions and fail on adaptive adversaries. This paper presents AdROD, an embedded, stochastic ensemble defense software designed for autonomous driving. AdROD employs *low-rank HyperNetworks*, which require only 1.6% of the parameter footprint of standard HyperNetworks, to generate diverse detectors at a per-frame rate, making it impractical for attackers to obtain the deployed detectors in time. To further improve adversarial robustness, AdROD incorporates a novel *functional diversity* mechanism, which couples stochastic weight updates with unique input-space transformations. We design two serving modes of AdROD that strike different trade-offs between robustness and runtime overhead: AdROD-I, a continuous protection mode for maximum resilience that leverages inter-detector disagreement to recover compromised detections, and AdROD-II, an on-demand mode triggered by kinematic discontinuities in object tracking. Through comprehensive evaluation with synthetic benchmarks, physically deployed adversarial patches, and end-to-end safety tests in the OpenCDA co-simulator, AdROD outperforms five baseline defenses and exhibits superior generalizability compared with the evaluated adversarial-training baselines, while maintaining real-time performance for safely stopping the vehicle at a stop sign instrumented with adversarial patches.**

***Index Terms*—Adversarial robustness, autonomous driving, HyperNetworks, object detection, physical adversarial attacks.**

I. Introduction

Camera-based object detectors built on deep neural networks (DNNs) are essential components of autonomous vehicle perception but are susceptible to physical adversarial attacks [1]–[4]. These attacks exploit model vulnerabilities and embed crafted perturbations into the scene to disturb predictions. In practice, adversaries often attach adversarial patches to critical road objects and suppress the victim vehicle's detection of these objects, posing severe safety risks [3]–[5]. To address these threats, existing defenses include *adversarial training*, which augments training data with perturbed samples [6], [7], and *input purification*, which disrupts the perturbations through techniques such as compression and masking [8], [9]. However, as these defenses are often designed for certain attack characteristics, such as those covered by the adversarial training samples, their robustness degrades on novel attacks or adaptive adversaries that actively optimize perturbations to bypass deployed defenses [10].

To address this limitation, we propose AdROD (Adversarially Robust Object Detection), a stochastic defense that employs *HyperNetworks* [11] to generate an ensemble of diverse detectors from random noise at the per-frame rate. The predictions of the generated ensemble are then fused to produce robust detection results. This design establishes a *moving target defense* [12], [13], in that the continuous variation of the detector ensemble renders the adversary's timely exploitation of the DNN gradients and decision boundaries impractical. Furthermore, AdROD is trained exclusively on benign data to remain agnostic to specific patch distributions, thereby reducing dependence on attack-specific training distributions.

However, standard HyperNetworks that generate the full weight space of the target DNN model scale poorly to support modern high-capacity object detectors. Specifically, the high dimension of the weight space leads to parameter explosion and destabilized training. AdROD resolves this by introducing *low-rank HyperNetworks* that produce compact, low-rank updates to a pretrained base detector rather than full weight replacements [14]. This design preserves knowledge contained in the factory model while drastically reducing memory overhead and promoting embedded deployments on resource-constrained platforms. Moreover, to mitigate attack effects that survive the *weight diversity* introduced by HyperNetworks, AdROD additionally enforces a novel *functional diversity* by coupling each weight update with a unique, noise-driven input transformation. This dual-layer randomization forces the ensemble's detectors to react to out-of-distribution patterns differently, thereby reducing the transferability of adversarial patches.

With the ensemble generation capability as the basis, the next question is how to fuse the ensemble's detection results. Existing fusion approaches typically rely on rigid bounding-box score averaging or voting [15], inherently trading benign accuracy for adversarial robustness. For instance, lowering the confidence thresholds may recover the objects missed due to attacks but inevitably increases the rate of benign false positives, as both attack-compromised true objects and spurious background predictions can receive low confidence scores. To navigate this trade-off, we design two serving modes. First, AdROD-I leverages inter-model disagreement to distinguish compromised detections from benign false positives. Second, to prioritize runtime efficiency, AdROD-II employs an on-

This work was supported in part by the National Research Foundation, Singapore, and DSO National Laboratories under the AI Singapore Programme (AISG Award No. AISG4-GC-2023-006-1B); in part by the Ministry of Education, Singapore, under its Academic Research Fund Tier 2 (Award No. T2EP20225-0007); in part by City University of Hong Kong (Project No. 9610753); and in part by the JC STEM Lab of Future Energy Systems (Project No. 2025-0039). (*Corresponding author: Rui Tan.*)

Yuting Wu, Dongfang Guo, Xiangzhong Luo, and Rui Tan were with Nanyang Technological University, Singapore. Qun Song was with the City University of Hong Kong, Hong Kong SAR, China.

demand serving strategy. Specifically, a low-overhead object tracker monitors the base detector under benign conditions and dynamically activates the ensemble defense upon abrupt object disappearances that signal potential attacks.

Our contributions are summarized as follows. **First**, we propose AdROD, a stochastic ensemble defense driven by parameter-efficient *low-rank HyperNetworks*. This design incurs only 1.6% of the parameter footprint of standard HyperNetworks and stabilizes training. **Second**, by exploiting the computational headroom arising from the low-rank design, we introduce *functional diversity*, a dual-layer randomization that couples weight diversity with input transformations for improved robustness. **Third**, we introduce two specialized serving modes: a continuous protection mode (AdROD-I) optimized for maximum robustness, and an adaptive on-demand mode (AdROD-II) tailored for runtime efficiency. **Fourth**, we conduct extensive evaluation from perception performance to end-to-end safety via the OpenCDA pipeline [16].[1] AdROD consistently outperforms five representative baselines [8], [9], [17]–[19] and exhibits superior generalizability over adversarial training, while its adaptive on-demand mode preserves the real-time throughput required for safe vehicle control in front of traffic signs instrumented with adversarial patches.

Paper organization: §II introduces preliminaries and reviews related work. §III states the research problem. §IV presents the design and training of AdROD. §V describes the two serving modes. §VI presents implementation details and evaluation results. §VII discusses the limitations and future work. §VIII concludes this paper.

II. Preliminaries and Related Work

A. Camera-based Object Detection and Adversarial Attacks

Camera-based object detection forms the perceptual foundation for autonomous vehicles and subsequent navigation tasks. Modern detectors are broadly categorized into one-stage architectures (e.g., YOLO [20]), which perform localization and classification simultaneously, and two-stage architectures (e.g., Faster R-CNN [21]), which rely on region proposal networks. Both approaches ultimately utilize Non-Maximum Suppression (NMS) to filter redundant bounding boxes. Specifically, NMS suppresses all candidate boxes with a confidence score below a threshold τ, and then iteratively removes overlapping boxes that exceed a predefined Intersection over Union (IoU) threshold γ, ensuring only the most prominent detection remains. We adopt a confidence threshold $\tau = 0.3$ and an IoU threshold $\gamma = 0.4$, which follow common settings in the literature (e.g., [9]). While this paper primarily adopts YOLO as the design basis for AdROD due to its high efficiency, AdROD can also adopt Faster R-CNN as evaluated in §VI.

Despite DNN-based object detectors' state-of-the-art accuracy under benign conditions, they are vulnerable to physical adversarial attacks, including regional solid patches or stickers affixed to objects [1], [22] and global semi-transparent overlays projected or placed on the objects [23]–[26]. Among these, *adversarial patches* have received wide attention due to their simplicity and low cost [5]. Formally, an adversarial patch $\mathbf{P}$ modifies a benign image $\mathcal{X}$ into an adversarial image $\mathcal{X}_{\text{adv}}$ via binary mask $\mathbf{M}$ and spatial transformation $\mathcal{A}$, yielding $\mathcal{X}_{\text{adv}} = \mathbf{M} \odot \mathcal{A}(\mathbf{P}) + (1 - \mathbf{M}) \odot \mathcal{X}$. The attack design typically adopts the loss function $\mathcal{L}_{\text{hide}} = \alpha_{\text{conf}} \cdot \mathcal{L}_{\text{conf}} + \mathcal{L}_{\text{colors}}$, where $\mathcal{L}_{\text{conf}}$ measures the reduction in detection confidence for the target object (e.g., objectness loss), $\mathcal{L}_{\text{colors}}$ enforces constraints on the patch's visual realism and printability, and α_{conf} is a weight. To improve attack robustness under varying environmental conditions, adversaries can use Expectation Over Transformation (EOT) [27], which optimizes the patch under randomized transformations to mimic real-world shifts in object scale, spatial placement, illumination, and background context. For instance, the RP_2 attack [1] adopts a uniform distribution of sign size; the SysAdv attack [3] adopts a distribution derived from real traces.

B. Adversarial Defenses

Existing defense approaches against adversarial attacks broadly fall into the following categories. A common solution is *adversarial training* [6], [28]. While adversarial training introduces no additional inference latency, this approach can overfit to attack-specific training distributions and exhibits limited generalization to novel threats. Furthermore, exhaustive anticipation of all potential attacks is not practical. *Input purification and patch removal* techniques preprocess the input to disrupt or remove adversarial perturbations. For example, Local Gradient Smoothing (LGS) [8] filters high-gradient regions, Jedi [19] localizes adversarial patches using entropy-based cues, and Segment-And-Complete (SAC) [9] segments and completes suspicious patch regions. Other recent works follow a similar localization-and-removal principle. Adversarial Pixel Masking [29] learns a MaskNet to mask out patch-like perturbations for pre-trained object detectors. PatchZero [30] detects adversarial pixels and zeros out the detected patch region by repainting it with mean pixel values. Attention-based defenses [31] use internal attention signals to track and mask malicious regions in multi-frame vision applications. These methods are effective when the adversarial regions can be reliably identified. However, they still rely on patch localization or masking rules, which may be challenged by adaptive attacks, low-frequency or smooth patches, irregular patch shapes, and patches that overlap heavily with the target object. For instance, as shown in §VI, customized smooth textures can bypass LGS's gradient-based filtering; visually realistic patches with irregular shapes can defeat SAC, which is primarily designed to address square patches with high-frequency textures. Overall, these defenses either depend on attack-specific training data or apply fixed preprocessing rules, making them difficult to generalize across diverse and adaptive physical attacks.

Beyond attack-specific or static defenses, another category of research improves robustness through model diversity or runtime randomness. *Ensemble-based defenses* use multiple models during training or inference and have shown effectiveness mainly in image classification settings [32]–[34]. However, many of them rely on adversarial training or fixed

[1]Demo videos are available at https://github.com/yutingwu-ntuiot/AdROD.

model sets, which may limit their robustness against white-box adaptive attackers. Directly applying such ensembles to complex object detection also raises practical issues, including high inference latency and difficulty in balancing robustness with benign accuracy, as discussed in §III-B. *Stochastic defenses* introduce randomness to disrupt attack transferability, such as using dropout [35] or randomly pruning activations [36] during inference. HyperNetwork-based dynamic ensembles [37], [38] combine *ensemble diversity with runtime stochasticity* by generating different model instances from random noise and are therefore the prior designs closest to AdROD. However, directly applying this approach to camera-based object detection remains challenging, as discussed next.

C. HyperNetworks and Their Limitations

Standard HyperNetworks [11] provide a fast way for dynamic parameter generation. They construct the weights of a target DNN model from random noise. Formally, a generator $\mathcal{G}$ is assigned to each layer of the target model, and maps a shared random vector $\mathbf{z} \in \mathbb{R}^{1\times d}$ to a new weight matrix $\mathbf{\Theta}'$, which then replaces the original weights $\mathbf{\Theta}$ of that layer.

Song et al. [37] utilized HyperNetworks to protect single-label image classifiers, while Xu et al. [38] extended this scheme to LiDAR-based object detection. However, directly adapting HyperNetworks to camera-based object detection faces two challenges. First, LiDAR attacks in [38] typically inject sparse, spurious points while largely preserving the victim vehicle's geometric structure. In contrast, physical attacks on camera-based object detection use intrusive patches that corrupt target-related visual features. Consequently, weight diversity alone in [38] is insufficient to deal with physical attacks on camera-based object detection, as shown in §VI. Second, scaling standard HyperNetworks to modern convolutional object detectors causes substantial parameter growth and overhead increase, as quantified in §III. The resulting larger optimization space also makes it harder to jointly balance ensemble diversity and benign detection accuracy.

III. Problem Statement

A. Threat Model

Attack objectives and scope. We consider an autonomous driving scenario in which a vehicle may encounter physical adversarial attacks. An attack is deemed successful if it subverts detection in any of the following ways: (1) the target object is not detected, meaning no predicted bounding box overlaps with the ground truth; (2) the object is misclassified; or (3) the object is poorly localized (i.e., low IoU). These criteria collectively reflect detection quality across object presence, classification, and localization. Accordingly, this work focuses on attacks that compromise the detection of existing objects.

Attacker capabilities. Following prior studies [1], [3], [4], we consider two attacker knowledge settings. Under a weak adversarial setting, the attack is optimized against an undefended vanilla model. Under a strong adversarial setting, the attacker possesses full white-box knowledge of the AdROD pipeline to optimize the adversarial pattern, with the exception of the runtime stochastic noise vectors. This is consistent with Shannon's maxim [39], under which the adversary is assumed to know the system except for private runtime randomness.

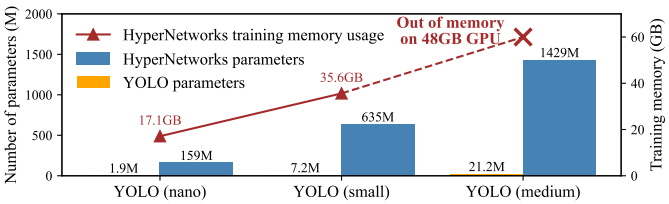


Fig. 1: Large parameter spaces and training memory requirements of HyperNetworks for three YOLO versions.

Fig. 2: Illustration of the trade-off between robustness and FPR under an adversarial attack on a stop sign. Raising the confidence threshold removes an undesired false positive (left) but hinders the recovery of the victim stop sign that would otherwise be detected at the lower threshold (right). Bounding boxes for other correct detections are omitted for readability.

Attack implementation. We consider physically plausible adversarial patterns that are visible to the victim camera. These include patterns attached to target objects (e.g., stop signs, persons, and vehicles), projected onto targets (e.g., stop signs), or placed near the target (e.g., traffic light). We assume that the attacker cannot modify the digital image stream or arbitrarily perturb every pixel.

B. Challenges of HyperNetwork-based Defense

We aim to design a HyperNetwork-based defense that generates an ensemble of diverse detectors at runtime and fuses their outputs to produce a reliable final result. If the attack designed against certain detectors transfers poorly to the generated detectors, robust object detection can be achieved. However, the defense design faces the following system challenges.

HyperNetworks scalability. Standard HyperNetworks exhibit a significant parameter overhead relative to the target model. As illustrated in Fig. 1, standard HyperNetworks for target models of YOLO-nano, YOLO-small, and YOLO-medium require 67×, 88×, and 84× more parameters than their respective target models. This leads to memory problems and poor training convergence. A new design for parameter-efficient HyperNetworks is required.

Robustness vs. false positive rate (FPR). An inherent trade-off exists between adversarial robustness and the benign FPR. Both victim objects and benign false positives (FPs) often exhibit low detection confidence. As illustrated in Fig. 2, naive mitigation strategies, such as lowering confirmation thresholds to recover a compromised stop sign, inevitably increase the retention of undesired FPs. Improving the trade-off requires a novel, discriminative metric to distinguish

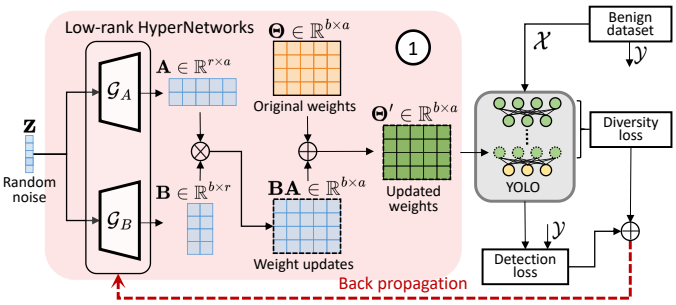


Fig. 3: AdROD training overview. The green circles in the YOLO architecture indicate weights updated by the low-rank HyperNetworks ($\mathbf{BA}$), while the yellow circles represent the original weights ($\mathbf{\Theta}$) retained from the base detector.

attack-related objects from benign FPs. This paper identifies *objectness variance* (cf. §V-A) as such a novel metric.

Serving efficiency. A continuous ensemble defense may incur high overhead. For instance, YOLO-small on a Jetson AGX Xavier achieves a throughput of 66.9 frames per second (FPS) at 640×640 resolution, whereas executing a continuous ensemble of 10 generated models drops to 5.4 FPS. Trading computational resources for safety is necessary, but unnecessary overhead should be avoided when the vehicle operation is safe. To this end, we seek a proactive yet low-overhead mechanism for detecting attack onset and activating ensemble defense on demand (cf. §V-B).

§IV and §V address these challenges through low-rank generation, functional diversity, and two deployment modes.

IV. A New HyperNetwork Design for AdROD

AdROD resolves scalability bottlenecks and reduces attack transferability via low-rank HyperNetworks (Fig. 3-①, §IV-A) and functional diversity (Fig. 3-②, §IV-B), respectively. Unlike adversarial training, the proposed architecture is optimized exclusively on benign images $\mathcal{X}$ and their object annotations $\mathcal{Y}$, driven jointly by an average detection loss and a weight diversity loss across the generated ensemble.

To illustrate the contribution of individual architectural components, we use the EOT-optimized SysAdv attack [3] on COCO [40] stop sign instances as a running example. Unless stated otherwise, ten generated models are fused *affirmatively* [15] (i.e., retaining any detection from any model to maximize robustness) and followed by NMS. We evaluate robustness via *attack success rate* (ASR), defined as the percentage of frames where the attack succeeds, and *benign detection accuracy* measured by mean average precision (mAP) on the attack-free COCO validation dataset. Paragraphs with a heading mark of "※" report results of the running example.

A. Low-rank HyperNetworks

AdROD integrates Low-Rank Adaptation (LoRA) [14] with HyperNetworks for scalability. While recent works have explored LoRA-HyperNetworks integrations for cross-task generalization in large language models and physics-informed networks [41], [42], AdROD specifically deploys this architecture for adversarial robustness in object detection. Instead of generating the weights of a target model directly, the proposed low-rank HyperNetworks generate updates to the weights of a factory-designed *base detector* that offers validated benign accuracy. Specifically, for a base detector's layer with pre-trained weights $\mathbf{\Theta} \in \mathbb{R}^{b\times a}$, the system employs two smaller generators, $\mathcal{G}_A$ and $\mathcal{G}_B$, to produce low-rank matrices $\mathbf{A} \in \mathbb{R}^{r\times a}$ and $\mathbf{B} \in \mathbb{R}^{b\times r}$, where the rank $r \ll \min\{a, b\}$. Their product, $\mathbf{BA} \in \mathbb{R}^{b\times a}$, updates the original weights as $\mathbf{\Theta}' = \mathbf{\Theta}+\mathbf{BA}$. This decomposition preserves the learned knowledge in $\mathbf{\Theta}$ while reducing the trainable parameter footprint by up to two orders of magnitude. The resulting training efficiency mitigates the optimization instability observed with standard HyperNetworks. In our implementation, we use the same rank r for all LoRA-updated layers to keep the design simple.

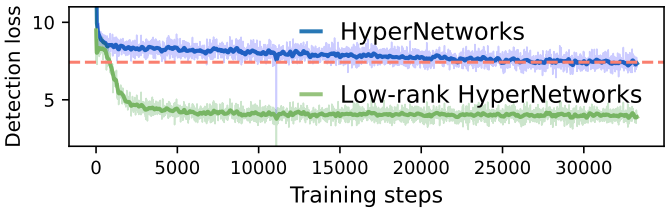


Fig. 4: Low-rank HyperNetworks achieve lower training detection loss compared with standard HyperNetworks.

※ **Impact of rank r.** From empirical evaluation across rank settings of $\{1, 4, 8, 16, 32, 64\}$, we observe that $r=1$ severely degrades mAP, whereas $r=64$ causes training instability. Although mAP peaks at $r=32$, increasing r degrades adversarial robustness. To achieve a satisfactory balance between accuracy and robustness, we adopt $r=4$. With this setting, the low-rank HyperNetworks for the YOLO-small base detector require only 1.6% of the parameters of standard HyperNetworks, reducing the parameter count from 635 million to about 10 million. They also achieve 45.9% lower detection loss than standard HyperNetworks, as shown in Fig. 4.

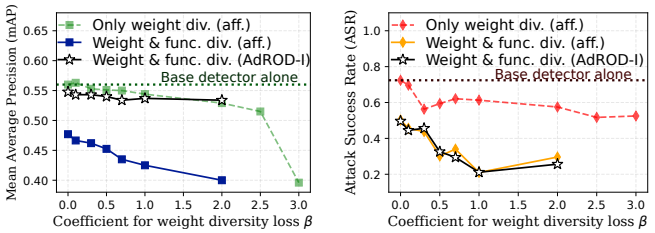
B. Diversifying Models for Robustness

To comprehensively disrupt attack transferability, AdROD establishes *functional diversity* by integrating *weight diversity* with input transformations. Let K denote the number of models in the ensemble (i.e., ensemble size), and let $\mathbf{\Theta}'_{k,n}$ denote the weight matrix of the n-th updated layer of the k-th model, where $1\leq k\leq K$ and $1\leq n\leq N$. We flatten $\mathbf{\Theta}'_{k,n}$ into a vector $\boldsymbol{\theta}'_{k,n} \in \mathbb{R}^{1\times ab}$. The weight diversity score γ_n across K models is defined via Pearson correlation:

$$\gamma_n = \frac{1}{\binom{K}{2}} \sum_{1\leq k_1<k_2\leq K} \left|\text{Pearson correlation}(\boldsymbol{\theta}'_{k_1,n}, \boldsymbol{\theta}'_{k_2,n})\right|.$$

The low-rank HyperNetworks are jointly optimized using the loss $\mathcal{L} = \frac{1}{K}\sum_{k=1}^{K}\mathcal{L}_{\text{det},k} + \frac{\beta}{N}\sum_{n=1}^{N}\gamma_n^2$, where $\mathcal{L}_{\text{det},k}$ is the primary detection loss and $\beta \geq 0$ dictates the diversity penalty. Note that the training ensemble used to compute the diversity loss is kept small to reduce GPU memory usage during joint optimization, which does not need to be matched with the serving ensemble size tuned for robust detection.

※ **Impact of coefficient β.** Fig. 5b shows that increasing β initially lowers ASR but saturates for $\beta \geq 2.5$. Meanwhile,



(a) Impact of β on mAP (b) Impact of β on ASR

Fig. 5: Comparison of ensemble detection performance under affirmative (aff.) fusion and the fusion of AdROD-I. Incorporating functional diversity (func. div.) improves robustness compared with using weight diversity only, while AdROD-I further mitigates benign detection accuracy degradation.

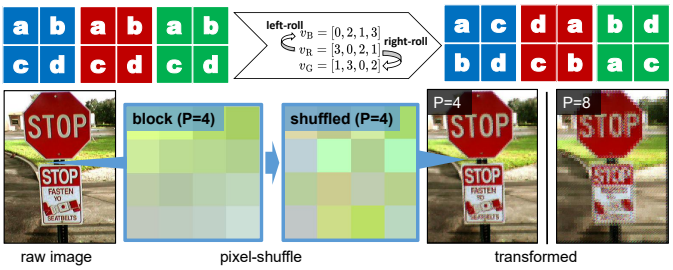


Fig. 6: Top: Permutation orders and permutation process for P=2. Bottom: Effects of pixel shuffling for P=4 and P=8.

Fig. 5a shows that the mAP drops sharply. Weight diversity alone is therefore insufficient, motivating an orthogonal, input-level countermeasure.

1) Functional Diversity: To further reduce attack transferability, AdROD introduces functional diversity by using the HyperNetwork's random noise vector $\mathbf{z} \in \mathbb{R}^{1\times d}$ to jointly generate the detector's weight updates and parameterize an input-space transformation $\mathbf{T}$ during both training and serving, i.e., $\mathcal{X}' = \mathbf{T}(\mathcal{X} \mid \mathbf{z})$. Thus, each generated detector acts as a "decoder" for its own "encoded" transformed input, reducing attack transferability across ensemble members.

Rather than relying on computationally heavy cryptographic operations, we adopt block-wise pixel shuffling [43], [44] to attain high-throughput spatial distortion. Specifically, the pipeline divides the input image $\mathcal{X}$ into several non-overlapping $P\times P$ blocks and shuffles each block following a *permutation order*. The permutation order of the red channel, $\mathbf{v}_R=(v_1,\ldots,v_{P^2})$, is derived by average-pooling $\mathbf{z}$ with a kernel of $\frac{d}{P^2}$, where $P^2 < d$, then ranking the pooled vector. To decorrelate the color channels, the permutation orders for green and blue channels, $\mathbf{v}_G$ and $\mathbf{v}_B$, are generated by right- and left-rolling $\mathbf{v}_R$, respectively. Fig. 6 illustrates this process.

※ **Effectiveness of functional diversity.** Evaluating the spatial block size $P \in \{1,2,4,5,8\}$ reveals a trade-off between patch distortion and feature preservation. Larger P values induce image distortion, which more strongly distorts the adversarial patch but degrades benign detection accuracy. With $P=4$, which balances robustness and benign accuracy, functional diversity brings an average reduction of 26.3 percentage points in ASR for $\beta \in [0, 2.0]$ compared with weight diversity alone, as shown in Fig. 5b. However, as shown in Fig. 5a, this robustness enhancement incurs an average mAP drop of 10.5 percentage points. This drop stems from the affirmative fusion rule, which accumulates FPs across the diverse ensemble, compounded by transformation-induced false negatives (FNs). In the next section, we devise an attack-aware fusion scheduler to recover benign detection accuracy without compromising the attained robustness.

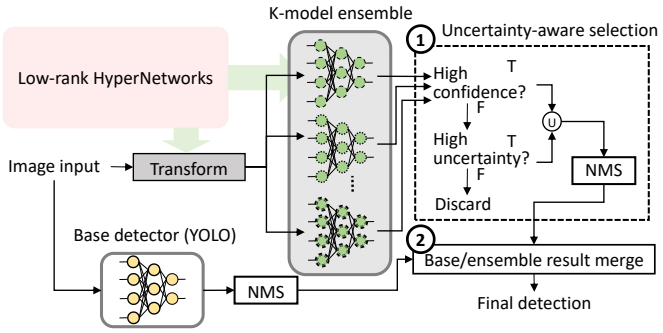


Fig. 7: AdROD-I workflow.

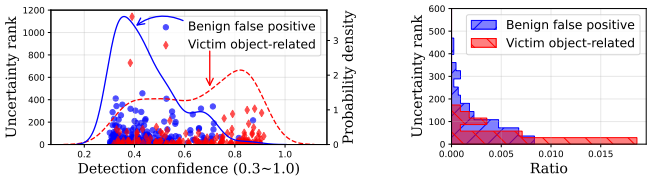
V. Two Serving Modes of AdROD

This section presents two serving modes: AdROD-I employs ensemble disagreement as an attack indicator to strike a better balance between adversarial robustness and benign detection accuracy; AdROD-II exploits the kinematic discontinuity of attack effectiveness observed during the victim vehicle's movement to reduce unnecessary ensemble execution.

A. AdROD-I

AdROD-I is built on two ideas to address the increased benign FPs and FNs caused by the ensemble defense. First, it uses an uncertainty indicator to distinguish benign FPs from predictions associated with victim objects (see Fig. 7 ①). Second, it merges the ensemble results with the base detector results to correct FNs caused by the image distortion introduced by functional diversity (see Fig. 7 ②). We continue to use the running example to illustrate the first idea.

To formalize the attack indicator, we first elaborate on some internal details of the object detector. Given an input image, the detector produces predictions for densely predefined spatial anchors across the image, each representing a candidate object hypothesis at a particular spatial location and scale. For each spatial anchor, the detector generates a prediction vector $\mathbf{b} = [x, y, w, h, o, \mathbf{p}]$, where (x, y, w, h) encodes the bounding-box geometry, o represents the scalar *objectness* score, and $\mathbf{p} = [p_1, \ldots, p_C] \in [0, 1]^C$ contains the per-class confidence scores over C detection classes. The final detection confidence is computed as $o{\cdot}\max_c p_c$. We hypothesize that, when adversarial transferability is reduced, physical adversarial patches cannot suppress all generated detectors consistently. As a result, predictions around an attacked genuine object exhibit higher disagreement across the ensemble, whereas benign FPs tend to appear more consistently. To quantify this difference, AdROD-I computes the variance of o across the K ensemble members for each spatially aligned anchor. For each image frame, it then sorts the aligned anchor predictions in descending order of *objectness variance* to obtain a localized *uncertainty rank*, where rank 1 indicates the highest disagreement.



(a) Scatter plot of uncertainty rank vs. detection confidence and their probability density functions (PDFs)

(b) Histogram of uncertainty rank (detection confidence $\in$ [0.3,0.5])

Fig. 8: Distributions of uncertainty rank and detection confidence for benign FPs and victim objects.

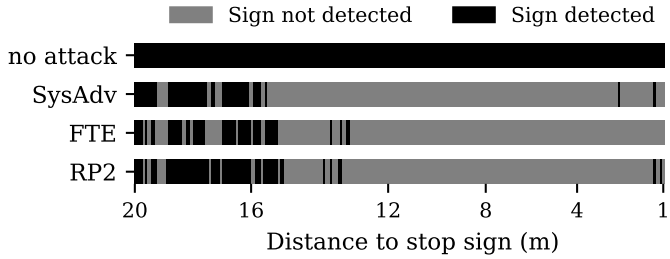


Fig. 9: Result trace of undefended YOLO when the vehicle approaches a stop sign with an adversarial patch. The patch is digitally applied to maximize attack effectiveness.

※ Fig. 8a shows the scatter plot of uncertainty rank versus detection confidence, where each point corresponds to a detection. This scatter plot merges results from all tested frames and distinguishes those corresponding to victim objects and benign FPs. It also shows the probability density functions (PDFs) of detection confidence for two cases. From the plot, high-confidence detections are rarely FPs, while low-confidence ones may be either. From Fig. 8b, for detections with confidence below 0.5, victim object-related detections are concentrated at low uncertainty ranks, showing that the attack likely increases objectness variance and disagreements among ensemble members. These observations motivate our design of AdROD-I as follows.

① **Uncertainty-aware selection.** First, from all anchors' box predictions $\mathcal{B}$ given by all detection models in the ensemble, we select the high-confidence predictions (denoted by $\mathcal{B}_{\text{confident}}$), which are likely benign and rarely false positives (cf. Fig. 8a), and the low-confidence but top-Q most uncertain predictions (denoted by $\mathcal{B}_{\text{victim}}$; cf. Fig. 8b), where Q is the uncertainty quota controlling the number of retained victim object candidates:

$$\mathcal{B}_{\text{confident}} = \{\mathbf{b} \in \mathcal{B} \mid \text{conf}(\mathbf{b}) \geq \tau_c\},$$
$$\mathcal{B}_{\text{victim}} = \{\mathbf{b} \in \mathcal{B} \mid \text{conf}(\mathbf{b}) < \tau_c \wedge \text{rank}_u(\mathbf{b}) \leq Q\},$$

where $\text{conf}(\mathbf{b})$ extracts $\mathbf{b}$'s detection confidence, and $\text{rank}_u(\mathbf{b})$ denotes the uncertainty rank. We combine the above two sets and apply NMS to form the ensemble's detection result $\mathcal{D}_{\text{ens}}$, i.e., $\mathcal{D}_{\text{ens}} = \text{NMS}\left(\mathcal{B}_{\text{confident}} \cup \mathcal{B}_{\text{victim}}\right)$. The NMS adopts its default confidence threshold τ. We set $\tau_c > \tau$, so that we impose a stricter requirement to confirm an anchor-box prediction as benign for joining $\mathcal{B}_{\text{confident}}$. This helps reduce benign FPs. A greater Q enhances victim recovery but also increases the risk of including benign FPs within $\mathcal{B}_{\text{victim}}$.

② **Base/ensemble result merge.** To resolve FNs caused by transformation distortion and the confidence threshold τ_c, we merge $\mathcal{D}_{\text{ens}}$ with the base detector's results on untransformed inputs $\mathcal{D}_{\text{base}}$. If two detections $\mathbf{d}_{\text{base}} \in \mathcal{D}_{\text{base}}$ and $\mathbf{d}_{\text{ens}} \in \mathcal{D}_{\text{ens}}$ lack *sufficient spatial overlap* (i.e., IoU < 0.6), both are retained to recover benign objects detected only by the base detector and victim objects recovered by the ensemble. For pairs with *substantial overlap*, the system applies the following resolution logic: (1) If predicted classes match, it retains only $\mathbf{d}_{\text{base}}$, prioritizing the base detector's superior bounding-box accuracy on untransformed inputs; (2) If classes diverge and $\mathbf{d}_{\text{ens}}$ possesses higher confidence, it retains $\mathbf{d}_{\text{ens}}$, treating it as a recovered victim object; (3) Otherwise, it conservatively retains both bounding boxes to prevent accidental suppression of ambiguous cases.

※ **Effectiveness.** We set $\tau_c = 0.5$ based on the results in Fig. 8, and $Q = 10$ based on the sensitivity analysis in §VI-C. By sequentially executing steps ① and ②, as shown in Fig. 5, AdROD-I reclaims the benign detection accuracy without compromising established adversarial robustness.

B. AdROD-II

During benign operation, the defense need not execute the ensemble continuously. AdROD-II triggers the ensemble defense only when an attack becomes effective. The system detects such cases via the inherent brittleness of adversarial patterns. In dynamic environments, changes in distance and perspective can hinder sustained patch effectiveness across the field of view [45]. As demonstrated in Fig. 9, when subjected to physical attacks (SysAdv [3], FTE [2], and RP2 [1]), an undefended YOLO detector consistently tracks the adversarially patched stop sign at a distance. Adversarial suppression takes effect after the vehicle is within a certain distance from the patch, and even then, the suppression is rarely absolute, still permitting intermittent successful detections. In practice, uncontrolled environmental variables, such as illumination changes and camera distortions, further degrade patch color fidelity and contrast, rendering continuous adversarial suppression practically unattainable.

AdROD-II exploits the kinematic discontinuity of attacked objects to detect adversarial interference. Specifically, once a vehicle enters the effective range of the patch, the detection trace of the attacked object exhibits a sustained and sudden loss of detection. In contrast, we observe that benign object disappearance typically occurs gradually: the bounding box shrinks as the object recedes, while occlusion or exiting the field of view progressively reduces its visible extent before disappearance. While benign FNs can also induce abrupt disappearances, such fluctuations are generally transient.

Kinematic tracking and defense activation. AdROD-II employs a Kalman Filter (KF) to track each object [46]. The KF takes detection results from the base detector to maintain a kinematic state $[x, y, w, h, \dot{x}, \dot{y}, \dot{w}, \dot{h}]$ for each candidate object, where (x, y) and (w, h) denote the center coordinates and bounding box dimensions, respectively. A disappearance event is classified as *abrupt* if the object vanishes without exhibiting the expected gradual scale reduction according to $\dot{w}$ and $\dot{h}$ typically observed during receding occlusion. Such objects

are added to a *watch set* $\mathbb{S}$, where their states continue to be extrapolated by the KF for an extrapolation window of ℓ frames despite the lack of new observations. If the base detector re-detects and associates a detection with an object in $\mathbb{S}$ within this window, the disappearance is dismissed as a benign fluctuation, and the object is restored to standard tracking. However, if the disappearance persists beyond ℓ frames, AdROD-II identifies a sustained adversarial suppression and activates the HyperNetwork-based defense. We empirically set the extrapolation window $\ell = 3$ based on a sensitivity analysis in §VI-C.

Sequential execution and early termination. Once the ensemble defense is activated, AdROD-II sequentially executes the generated models and applies early termination to minimize computational load. For each model, the system retains only detections that can be associated with the KF-extrapolated states in the watch set $\mathbb{S}$. These validated recoveries are then merged into the final output according to the resolution logic defined in §V-A. AdROD-II halts further model execution once all watched objects in $\mathbb{S}$ are recovered for the current frame. Consequently, the number of executed models, denoted by k, adapts to the scene, reducing redundant computation. If no watched object can be recovered for a sustained duration, the system issues a defense-failure warning and deactivates the ensemble to conserve resources. Such a warning may indicate either successful adversarial suppression or a false activation.

Scope of activation. AdROD-II activates only after an object has been detected and tracked at least once. This track may be established when the object first enters the field of view or later during approach, since physical attacks are often imperfect and may permit intermittent detections before sustained suppression. Once a tracked object subsequently disappears abruptly, AdROD-II activates the ensemble. False activations may arise from benign detector fluctuations, including transient FNs of genuine objects and the disappearance of short-lived FP tracks. For a transient FN, subsequent re-detection restores the genuine track and terminates ensemble execution. For a short-lived FP track, the ensemble typically fails to recover a corresponding object; persistent recovery failure eventually triggers the defense-failure warning and deactivation described above.

VI. Implementation and Evaluation

A. Implementation and Evaluation Methodology

Base detectors. We adopt a lightweight YOLO-small [20] as the primary base detector, and a two-stage Faster R-CNN [21] with a MobileNetV3 backbone to demonstrate architectural generalizability. We initialize both models with weights pretrained on the Microsoft COCO dataset [40].

Implementation details. (1) *Training:* We configure the low-rank HyperNetworks to update the base detector's backbone and neck modules, with rank $r = 4$, block size $P = 4$, and coefficient $\beta = 1$. We use the COCO dataset (640×640) for training, with a training period of 9 epochs, a batch size of 32, a training ensemble size of 3, and a learning rate controlled by cosine annealing (10^{-4} to 10^{-5}). The training takes about 10 hours on a Quadro 8000 GPU. (2) *Serving:*

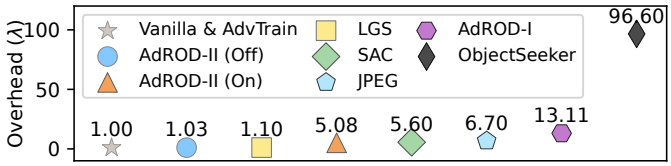


Fig. 10: Computational overhead of defenses. The execution time ratio is normalized relative to the vanilla model, with lower values indicating lower overhead.

Since each detector is independently generated from sampled random noise, the ensemble size used to compute the diversity loss during training does not need to match the ensemble size used for serving. For AdROD-I, we set a serving ensemble size to $K = 10$, $\tau_c = 0.5$, and $Q = 10$; for AdROD-II, we set maximum ensemble size to $K = 10$ and extrapolation window $\ell = 3$. §VI-C provides a sensitivity analysis on these settings.

Baselines. We benchmark AdROD against: an undefended Vanilla model, adversarial training (AdvTrain), and five representative defenses designed without considering the tested attacks, including JPEG [17], LGS [8], SAC [9], Jedi [19], and ObjectSeeker [18], using their original configurations. For AdvTrain, we adopt a naming convention of *AdvTrain-Class* to denote a model that is adversarially trained against a specific class of objects instrumented with regional patch attacks.

Metrics. We evaluate AdROD using the following three metrics: (1) *ASR* as defined in §IV; (2) *benign detection accuracy* measured by standard object detection metrics including precision, recall, and mAP. We report mAP at IoU 0.5 (mAP50) and the average mAP across thresholds from 0.5 to 0.95 (mAP50:95) on the COCO validation dataset; and (3) *runtime overhead* measured in the unit of λ, where λ denotes the per-image inference latency of the vanilla YOLO detector (including NMS). Expressing overhead in λ ensures a device-agnostic benchmark for computational cost.

B. Benign Detection Accuracy and Overhead

Benign detection accuracy. Table I summarizes the results on the COCO validation dataset. AdROD-II is excluded because COCO lacks the continuous video frames required for its tracking mechanism. Jedi is omitted because only a MATLAB implementation is available, and a direct overhead comparison would be unfair. From Table I, AdROD-I's benign detection accuracy is comparable to the vanilla model and superior to adversarial training baselines.

Baseline overhead. Fig. 10 presents computational overhead in λ. AdvTrain introduces no additional inference latency, but fails to generalize to unseen threats as shown in §VI-D3 shortly. While LGS is fast (1.1λ), it sacrifices accuracy due to global image distortion. Other baselines introduce significant overhead: SAC relies on a U-Net backbone (5.6λ), JPEG incurs substantial CPU encoding costs (6.7λ), and ObjectSeeker delivers the highest latency (96.6λ) due to exhaustive sliding-mask passes with numerous detector inferences.

AdROD overhead breakdown. Table II breaks down the overhead of the two AdROD variants. Although AdROD-I

TABLE I: Benign detection accuracy.

Method	Dep.	Precision	Recall	mAP50	mAP50:95
Vanilla	–	0.666	0.515	0.562	0.371
LGS [8]	–	0.613	0.436	0.472	0.301
JPEG [17]	–	0.656	0.492	0.537	0.348
ObjectSeeker [18]	–	0.662	0.520	0.557	0.369
SAC [9]	○	0.667	0.513	0.558	0.369
AdvTrain-Person	●	0.642	0.499	0.539	0.344
AdvTrain-Stopsign	●	0.634	0.503	0.539	0.346
AdROD-I	–	**0.646**	**0.508**	**0.541**	**0.358**

Dep. indicates training-time attack dependency: ○ denotes attack-pattern assumptions or patch-specific training; ● denotes attack-specific adversarial training; – denotes no attack-related training.

TABLE II: Overhead breakdown of AdROD-I and AdROD-II.

Component	Overhead ($\times\lambda$)		
	AdROD-I	AdROD-II (On)	AdROD-II (Off)
Weight generation	$0.169K$	$0.169k$	–
Input transform	$0.041K$	$0.041k$	–
Ensemble inference	$0.938K$	$1.000k$	–
Anomaly detection	–	0.03	0.03
Result merge	1.63	1.00	1.00
Total	**1.148K+1.63**	**1.21k+1.03**	**1.03**

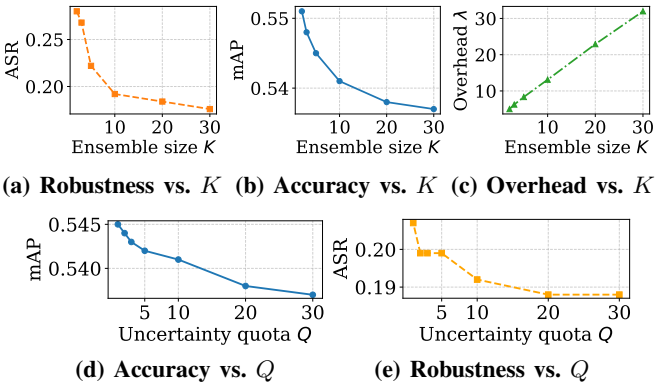
Note: Ensemble member inference in AdROD-I excludes NMS, whereas AdROD-II uses the full detection pipeline; $k \leq K$; "On"/"Off" indicates ensemble activation status.

($K=10$) achieves strong robustness (see §VI-D3), its continuous ensemble execution incurs a 13.11λ static overhead, limiting throughput to 5.4 FPS on an NVIDIA Jetson AGX Xavier. AdROD-II bypasses this bottleneck via an on-demand policy. In attack-free scenarios, it disables the ensemble and runs only the anomaly detector (1.03λ), achieving 68.7 FPS. When attacked, the activated ensemble executes sequentially and terminates early, thereby dynamically adjusting the subset size k ($k \leq K$). Consequently, even though the per-model overhead coefficient in AdROD-II is marginally higher, AdROD-II's aggregate overhead reduces to an average of 5.08λ (13.9 FPS), as detailed in §VI-D1 shortly. Furthermore, AdROD-II's average throughput of 13.9 FPS satisfies the representative 100-ms perception deadline reported in [47].

C. Sensitivity Analysis and Ablation Study

Impact of ensemble size K on AdROD-I. Fig. 11(a–c) shows the effect of varying $K \in \{2, 3, 4, 10, 20, 30\}$. ASR decreases with increasing ensemble size, but further gains diminish beyond $K=10$. Conversely, benign detection accuracy (mAP) shows a slight decline as the larger ensemble increases the statistical probability of introducing FPs. Meanwhile, computational overhead grows nearly linearly with K. Thus, selecting an appropriate K is essential.

Impact of uncertainty quota Q on AdROD-I. The parameter Q controls the number of highly uncertain bounding boxes retained during the ensemble merging process. As shown in Fig. 11(d–e), increasing Q effectively lowers the ASR but slightly degrades mAP. This suggests a trade-off between precision and defense strength. Because overall ASR and mAP are less sensitive to Q than to K, Q can be employed as the secondary tuning parameter.



(a) Robustness vs. K **(b) Accuracy vs. K** **(c) Overhead vs. K**

(d) Accuracy vs. Q **(e) Robustness vs. Q**

Fig. 11: (a–c) illustrate how increasing K affects accuracy, robustness, and runtime overhead, while (d–e) show how varying Q influences accuracy and robustness in AdROD-I.

TABLE III: Ablation study on weight and functional diversity against stop-sign-targeted SysAdv attack. Affirmative fusion, which retains any detection, is applied for the best robustness.

Defense Variant	ASR	mAP50
Training-free Baselines		
Vanilla	0.728	0.562
Vanilla + Image transform	0.834	0.132
Weight and Functional Diversity		
None	0.705	0.562
Functional diversity only (P=4)	0.498	0.476
Weight diversity only (β=1)	0.613	0.544
Weight + Functional diversity (β=1, P=4)	0.195	0.424

Impact of extrapolation window ℓ on AdROD-II. The parameter ℓ controls the sensitivity of the defense trigger. Because evaluating ℓ requires continuous video frames, we conduct this analysis using a physical outdoor testbed, which will be detailed in §VI-D1 shortly. An aggressive setting of $\ell=1$ leads to a marginal robustness gain (ASR 0.033 vs. 0.058 at $\ell=3$). However, it leads to high overhead, because the occasional benign FNs produced by the base detector trigger the ensemble defense at almost every momentary loss of an object. Conversely, a highly tolerant setting of $\ell=5$, which leads to an ASR of 0.079, introduces extra delays before classifying sustained disappearances as attacks, thereby increasing ASR undesirably. Consequently, $\ell=3$ strikes a satisfactory balance, leveraging kinematic memory to efficiently bridge benign FNs while ensuring prompt defense activation.

Ablation. In Table III, we isolate the contributions of each diversity component. First, we examine whether block-wise pixel shuffling, applied without retraining, provides robustness gains. As shown under *Training-free Baselines*, its direct application degrades detection accuracy while increasing ASR, indicating that the transform alone is ineffective as a defense. Then, we evaluate each diversity component by training the low-rank HyperNetworks under various configurations. Functional diversity and weight diversity each improve robustness when applied individually, but neither is sufficient to provide strong protection. When combined, the two components achieve substantially stronger robustness, demonstrating their complementary effects. Although this joint configuration reduces the mAP, AdROD-I recovers most of the performance

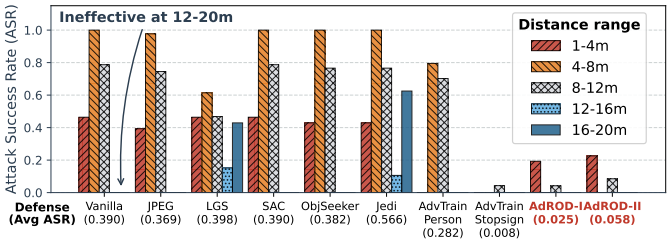


Fig. 12: ASR of defense methods against SysAdv attack.

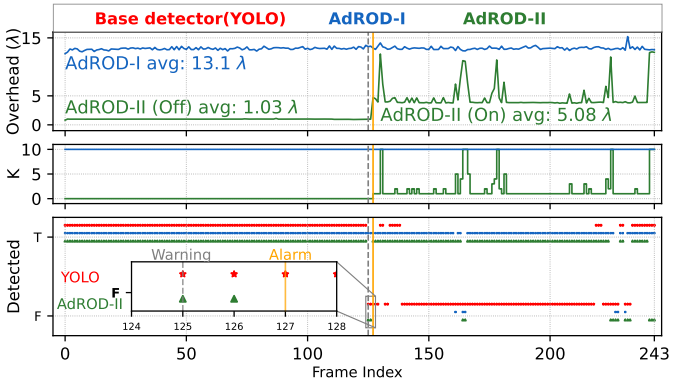
loss, as seen in Table I. Furthermore, AdROD-II avoids this degradation by selectively fusing only validated and recovered objects with the base detector outputs, thereby preserving accuracy on benign samples. Overall, these results show that both weight diversity and functional diversity are essential for effective defense in object detection.

D. Robustness Analysis at the Perception Level

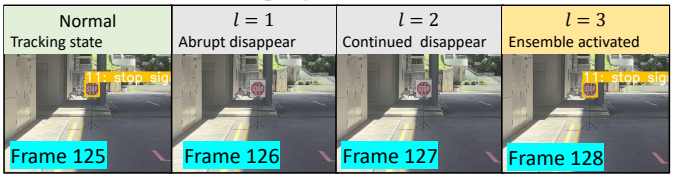
1) Physical Outdoor Evaluation: We evaluate AdROD-I and AdROD-II on an outdoor testbed using a physically deployed SysAdv [3] adversarial patch, which is selected due to its strong digital-to-physical transferability and system-level effectiveness. The patch is printed and attached to a stop sign placed on a restricted private road. A victim camera mounted on a vehicle approaches the stop sign from a distance of 20 m. Fig. 12 shows that on the vanilla model, the patch is ineffective at 20-12 m because its adversarial patterns cannot be sufficiently captured by the camera, reaches ASR 0.78 at 12-8 m and 1.00 at 8-4 m, then falls to 0.46 at 4-1 m because such close-ups are rare in patch optimization.

Effectiveness. Fig. 12 also compares the robustness of different defense methods. Both AdROD variants significantly reduce the ASR across all distance ranges. AdROD-I achieves the lowest average ASR of 0.025, while AdROD-II achieves an average ASR of 0.058. The slightly higher ASR of AdROD-II is expected because its defense is triggered only after the disappearance of the victim object is detected. Despite this delayed activation, AdROD-II remains the second-best attack-agnostic defense in this experiment. Moreover, AdROD achieves performance comparable to *AdvTrain-StopSign*, which is trained with attack-specific knowledge, and substantially outperforms *AdvTrain-Person*, which is trained on a different object class. These results further demonstrate the superior generalization ability of AdROD.

Runtime dynamics. Fig. 13a illustrates the execution trace of AdROD compared with the vanilla model, covering runtime overhead, ensemble size, and object recovery status. Before Frame 126, when the camera remains distant and the adversarial patch has no effect, AdROD-II maintains a near-baseline overhead of 1.03λ. As the attack manifests at Frame 126, AdROD-II leverages KF-extrapolated states to flag the anomaly and transitions the lost object to its watch set $\mathbb{S}$. By Frame 128, after three consecutive misses, the ensemble defense is activated with sequential ensemble execution and early termination to bound overhead. Fig. 13b visualizes this state transition: from normal tracking (Frame 125), to watching (Frames 126–127), and finally to defense activation (Frame



(a) Runtime behaviors of vanilla YOLO, AdROD-I, and AdROD-II. 'T' indicates recovered stop sign detection; 'F' indicates miss.



(b) Operational states of AdROD-II (Frames 125–128): transition from tracking to ensemble activation.

Fig. 13: Physical experiment against SysAdv. The attack becomes effective from Frame 126 to the vanilla YOLO.

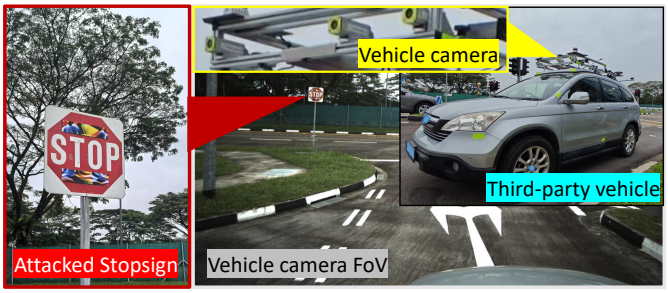


Fig. 14: Experiment in the vehicle security challenge.

128). Together, these results show how AdROD-II enables robust recovery with minimal ensemble execution at runtime.

2) Real-world Third-party Vehicle Evaluation: In a third-party vehicle security challenge, an Allied Vision Alvium G1-510C camera recorded a vehicle traveling at 20 km/h and 10 FPS. Its wide field of view (74.5° horizontal and 54° vertical) made the sign smaller than in §VI-D1, so we optimized SysAdv at 640×640 and 1280×1280 (LowRes and HighRes). Recorded data were released to us after collection. Fig. 14 illustrates the adversarial setup used for evaluation.

SysAdv-HighRes achieves an ASR of 0.759 on the vanilla model, which decreases to 0.310 and 0.345 with AdROD-I and AdROD-II, respectively. *SysAdv-LowRes* achieves an ASR of 0.719 on the vanilla model, which decreases to 0.219 and 0.344 with AdROD-I and AdROD-II, respectively. These results demonstrate AdROD's effectiveness and provide evidence of transfer across the evaluated camera setups and real-world conditions. Prior system-level evaluation of stop-sign attacks [3] suggests that traffic-rule violations require sustained detection failures over multiple frames. Therefore, §VI-E further evaluates whether AdROD preserves system-level safety in the full perception–control loop.

SysAdv-smooth SysAdv Rectangle Square Flush Wide Fixed Free

(a) Stop sign (COCO & self-collected) **(b) Person** (COCO) **(c) Traffic light** (self-collected) **(d) Car** (self-collected)

Fig. 15: Adversarial patch attacks with different attack targets and characteristics. The callouts show zoomed-in patches.

TABLE IV: ASR of attacks against baseline defense methods and AdROD-I.

			No Defense	**Baseline Defenses**						**Ours**
Object	**Data**	**Attack Type**	Vanilla	JPEG [17]	LGS [8]	SAC [9]	Jedi [19]	ObjSeeker [18]	AdvTrain-Stopsign	AdROD-I
Stop sign	COCO	RP_2	0.701	0.682	0.490	0.701	0.678	0.602	**0.153**	**0.180**
		FTE	0.851	0.808	0.644	0.851	0.789	0.759	**0.187**	**0.218**
		SysAdv	0.728	0.751	0.521	0.728	0.655	0.651	**0.190**	**0.195**
		SysAdv-smooth	0.375	0.360	0.414	0.375	0.483	0.287	**0.093**	**0.153**
Person	COCO	Rectangle	0.685	0.674	0.563	0.635	0.587	**0.524**	0.676	**0.172**
		Square	0.587	0.567	0.298	**0.124**	0.332	0.336	0.646	**0.088**
Traffic light	Self-collected	Flush	0.443	0.424	0.167	**0.057**	0.261	0.290	0.234	**0.096**
		Wide	0.729	0.765	0.351	**0.131**	0.547	0.512	0.483	**0.202**
Car	Self-collected	Fixed	0.913	0.895	0.648	0.644	0.882	0.913	**0.614**	**0.247**
		Free	0.868	0.881	0.667	0.534	0.873	0.873	**0.504**	**0.288**
Benign mAP@50			0.562	0.537	0.472	0.558	–	0.557	0.539	0.541

Note: **Bold and underlined** text indicates the lowest ASR (best), and **bold** text indicates the second lowest. "–" indicates unavailable benign mAP.

3) Diverse Synthetic Evaluation: We evaluate AdROD against a broad spectrum of physical-style adversarial attacks, varying in *shape* (rectangular vs. square), *size* (flush vs. wide), *placement* (fixed vs. free), and *target class*, as illustrated in Fig. 15. The evaluation is conducted on COCO validation subsets (256 stop signs, 426 persons) and a self-collected dataset comprising 244 frames of stop signs, 505 frames of traffic lights, and 218 frames of cars. Each frame contains a single attacked object with perturbations digitally applied. Because some datasets lack consecutive frames, this evaluation focuses on AdROD-I.

Effectiveness. Table IV shows that AdROD-I achieves the lowest or second-lowest ASR across all configurations. Conversely, baselines are configuration-sensitive. Specifically, *AdvTrain-StopSign* degrades on unseen classes and yields high (>0.5) ASRs on cars. *SAC* [9] performs well on the square-noise traffic-light attack but degrades on several other tested shapes/textures. *LGS*'s gradient-based filtering [8] and *Jedi*'s entropy-based localization [19] show degraded robustness against *SysAdv-smooth*'s smooth-textured patches. *ObjectSeeker* [18] is less effective against adversarial patches that overlap the target object. Finally, *JPEG* compression [17] is ineffective against visually prominent patches.

Generalization. To further examine AdROD's generalizability beyond regional patches, we extend our evaluation to semi-transparent global masking attacks targeting stop signs as shown in Fig. 16. We use two adversarial training configurations: *AdvTrain-StopSign*, which is based on the correct target class but an incorrect masking type (i.e., solid regional masking during training but semi-transparent global masking during testing), and *AdvTrain-Person*, which is based on an incorrect target class and solid regional masking type. Let η denote the attack opacity. When $\eta \leq 0.3$, AdROD outperforms all baselines, including *AdvTrain-StopSign*, which is otherwise

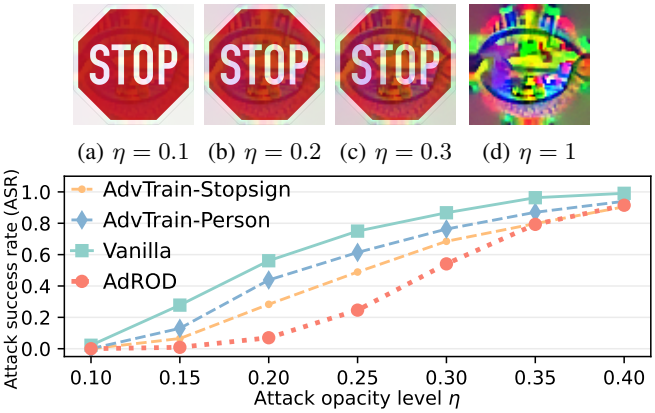


Fig. 16: ASR of global masking attacks targeting stop signs. Evaluated on the stop sign images from the outdoor testbed.

highly effective against solid patch attacks. Beyond this threshold, high image distortion impairs all detectors. Among the two adversarial training baselines, *AdvTrain-Person* performs noticeably worse, due to mismatches in both class and patch type. This reinforces the understanding of the limitations of adversarial training. That is, although it is effective when testing conditions match its training assumptions, its robustness declines when attack characteristics shift. In contrast, AdROD maintains robust performance across these masking variations, underscoring its better generalizability.

4) Architectural Generalizability: We extend AdROD to Faster R-CNN, a two-stage detector comprising a backbone with Feature Pyramid Networks (FPN), a Region Proposal Network (RPN), and a Region of Interest (RoI) head. We configure the low-rank HyperNetworks to update the weights of all convolutional layers in the backbone, FPN, and RPN, as well as the linear layers in the RoI head. This configuration yields a good balance between robustness and accuracy.

Unlike YOLO's single-stage architecture, Faster R-CNN

TABLE V: Performance (ASR and mAP50) of Faster R-CNN (FRCNN) and AdROD defense variants under SysAdv attack on outdoor testbed.

Metric	FRCNN	AdROD-I			AdROD-II		
		$K=2$	$K=3$	$K=4$	$K=2$	$K=3$	$K=4$
ASR	0.635	0.090	0.098	0.090	0.137	0.122	0.071
mAP50	0.252	0.223	0.222	0.222		–	

TABLE VI: ASR of component-aware adaptive attack.

Case	Attacker Knowledge			Vanilla ASR		AdROD ASR	
	Θ_0	Θ	$\mathcal{G}$	$S=2$	$S=10$	$S=2$	$S=10$
3	✓	✓	✓	0.628	0.234	0.245	0.169
2	✓		✓	0.621	0.222	0.257	0.153
1	✓	✓		0.625	0.326	0.241	0.184
0	✓			0.728		0.195	

Θ_0: base detector; Θ: a set of historically generated weights; $\mathcal{G}$: HyperNetwork generator and the random noise sampling distribution. S denotes the surrogate ensemble size, i.e., the number of generated models used by the adversary for patch optimization.

filters low-score proposals at the RPN stage before classification. Consequently, the number and spatial distribution of final detections may not align across diverse ensemble members. To compute uncertainty for AdROD-I, we align the outputs by extracting objectness scores directly from the intermediate RPN logits and inserting low-score dummy bounding boxes wherever an anchor lacks a corresponding detection in any ensemble member. AdROD-II, however, operates purely on post-NMS temporal tracking and can be applied seamlessly.

Table V shows that AdROD-I reduces ASR to about 0.09 across all settings with only a modest mAP drop, leaving little observable trend with respect to K. In contrast, AdROD-II becomes more robust as the serving ensemble size increases. At $K=4$, AdROD-II achieves the lowest ASR (0.071), surpassing AdROD-I. We attribute this improvement to AdROD-II's temporal dynamics-based attack detection, which enables more precise localization and successful recoveries from the ensemble than AdROD-I's uncertainty-based estimation. Overall, these results confirm that AdROD generalizes effectively to Faster R-CNN.

5) Adaptive Adversaries: We evaluate two adaptive attacks under a unified optimization protocol. The first, *component-aware adaptive attack*, uses the standard hiding objective $\mathcal{L}_{\text{hide}}$ (see §II-A), averaged over the vanilla detector and a surrogate ensemble constructed from exposed AdROD components. This averaging acts as a Monte Carlo approximation to an expectation over generated models and input transformations, while preserving the hiding effect on the vanilla detector. The second, *uncertainty-aware adaptive attack*, further targets AdROD-I's uncertainty-aware selection rule by adding a disagreement loss that suppresses objectness variance across the surrogate ensemble. For both attacks, optimization is performed on differentiable detector outputs before NMS. At each step, the adversary constructs the surrogate ensemble according to its knowledge setting and applies the corresponding input transformations to the patched image. We do not backpropagate through non-differentiable components, such as NMS or tracking. After optimization, the patch is evaluated on the full defended pipeline. The adversary knows the defense pipeline and sampling distribution, but not the runtime noise vectors used during AdROD's deployment.

TABLE VII: ASR under uncertainty-aware adaptive attack.

α_{dis}	0	0.1	0.5	1	5	10	100
Vanilla ASR	0.621	0.617	0.590	0.582	0.602	0.594	0.245
AdROD ASR	0.257	0.218	0.215	0.207	0.222	0.215	0.165

The attack uses **Case 2** with $S=2$. $\alpha_{\text{dis}}=0$ corresponds to the component-aware adaptive attack.

Component-aware adaptive attack. Table VI reports the results under increasing attacker knowledge. The cases range from access only to the base detector Θ_0 (**Case 0**, same as the setting in Table IV), to additional access to historical generated weights Θ (**Case 1**), to access to the HyperNetwork generator $\mathcal{G}$ for sampling surrogate runtime detectors (**Case 2**), and finally to full disclosure of all components (**Case 3**). We denote the surrogate ensemble size by S and consider two settings: $S=2$, the smallest setting that still forms an ensemble, and $S=10$, matching the serving ensemble size of AdROD-I ($K=10$). We draw two key observations from Table VI. First, increasing the surrogate ensemble size reduces attack effectiveness. Intuitively, a larger surrogate ensemble makes the optimization harder because the attacker must find one patch that hides the victim object across more generated models. Second, the optimized patches remain effective against the vanilla detector, especially when $S=2$. This shows that the adaptive attack can still produce valid hiding patches for the base model. However, the same patches transfer much less effectively to the full AdROD pipeline, indicating that AdROD's generated ensemble reduces attack transferability. Even in the worst case observed in this table (**Case 2**, $S=2$), the ASR only increases moderately compared with the non-adaptive setting. These results show that AdROD remains robust against component-aware adaptive adversaries.

Uncertainty-aware adaptive attack. We further evaluate a stronger white-box adversary that accounts for AdROD-I's uncertainty-aware selection rule. Specifically, we augment the original hiding loss $\mathcal{L}_{\text{hide}}$ with a disagreement loss $\alpha_{\text{dis}} \cdot \text{Var}(\{o_i\}_{i=1}^{S})$, where o_i denotes the victim object's objectness score from the i-th generated model in the surrogate ensemble, and α_{dis} controls the weight of the disagreement loss. Since the attack minimizes this objective, the added term reduces objectness variance across the surrogate ensemble, making the attacked object less likely to be selected into $\mathcal{B}_{\text{victim}}$ by AdROD-I. Based on Table VI, we use the strongest observed attack setting (**Case 2**, $S=2$), and sweep $\alpha_{\text{dis}} \in \{0, 0.1, 0.5, 1, 5, 10, 100\}$. The case $\alpha_{\text{dis}}=0$ corresponds to the component-aware adaptive attack under the same setting.

Table VII shows that reducing objectness variance of the attacked object across the ensemble does not make the attack stronger against AdROD-I. Across all nonzero α_{dis} values, the ASRs on AdROD are lower than in the $\alpha_{\text{dis}}=0$ case. The vanilla ASR also drops when the disagreement loss becomes large, suggesting that overemphasizing disagreement reduction weakens the patch's basic hiding ability. The conflicting

attack objectives, i.e., hiding the victim object and making the generated models respond similarly, impede optimization. These results further show that AdROD's diversity complicates adaptive patch optimization, even when the attacker accounts for AdROD-I's uncertainty-aware selection rule.

E. An End-to-End Safety Evaluation

We employ OpenCDA [16], a cooperative driving system integrated with the CARLA simulator [48], to investigate the end-to-end effects of attack and defense in a specific scenario where the vehicle should stop before the stop line in front of a stop sign instrumented with adversarial patches. We implement a stop procedure as follows. When a stop sign is detected in at least three frames within a short time window and two of these frames are consecutive, the vehicle starts the stop procedure of decelerating to 5 km/h and then engaging a longitudinal controller to halt before the stop line. The three-frame requirement handles transient FPs. There are three possible outcomes: (1) *Safe stop*, where the vehicle halts entirely before the stop line; (2) *Marginal stop*, where the vehicle halts on or slightly beyond the line; and (3) *Missed stop*, where the vehicle fails to confirm the stop sign and bypasses braking. For safe and marginal stops, we record the remaining distance to the stop line as the *safety margin*. A comprehensive safety analysis covering all possible scenarios and adversarial attacks is beyond the scope of this paper. However, our results here provide insights into understanding the interplay among attack strength, defense, compute constraints, and safety.

The simulated vehicle initially cruises at a constant speed and executes the perception-control loop at a rate of 20 Hz. Under no attack, with the vanilla model and the two AdROD variants, the vehicle achieves safe stops at all considered cruise speeds from 20 to 40 km/h. This range allows us to evaluate stopping safety under increasingly demanding approach speeds, while remaining within a low-to-moderate speed regime relevant to urban road safety [49].

We use a SysAdv adversarial patch. We adjust only the lightness channel (L) of the patch in the CIELAB color space, while keeping the chrominance channels unchanged to preserve adversarial color features. The light-adjusted patch is derived by: $L'_{\text{patch}} = (1-\zeta)\cdot L_{\text{patch}} + \zeta \cdot L_{\text{matched}}$, where L_{patch} is the original lightness of the patch, and L_{matched} is obtained by linearly scaling and shifting L_{patch} to match the mean and standard deviation of the ambient lightness. Thus, $\zeta = 0$ leaves the patch unchanged, while $\zeta = 1$ fully matches the ambient lightness statistics. We use $\zeta = 0.5$ by default.

1) Safety Under Attack: Fig. 17 shows that the undefended vanilla model's performance rapidly degrades as the vehicle speed increases. In contrast, both AdROD variants yield no missed stops up to 35 km/h. At 40 km/h, AdROD-I still yields no misses, whereas AdROD-II experiences a rare missed stop. This occurs because the high vehicle speed limits the number of frames containing the stop sign, impeding AdROD-II in establishing the kinematic tracking of the attacked sign for defense activation. Note that if we set $\zeta > 0.5$, the increased lightness matching will reduce the attack strength; however, the vanilla model still yields substantial missed stop rates at

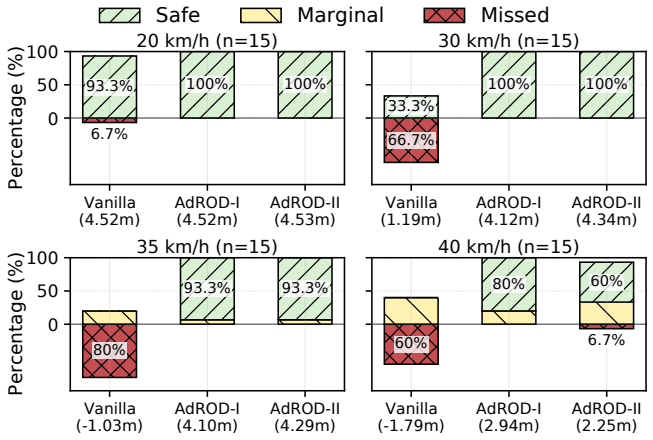


Fig. 17: Stop outcomes in front of an adversarial stop sign. Bars show the percentages of *safe*, *marginal*, and *missed* stops. Values below the x-axis give the mean signed distance to the stop line, where positive values indicate stopping before the line and negative values indicate stopping beyond it. n denotes the number of trials per bar.

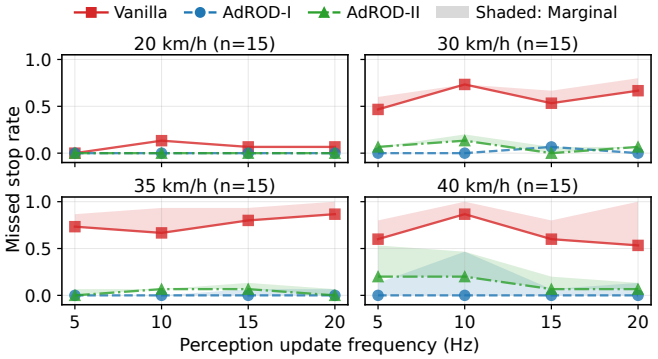


Fig. 18: Impact of reduced perception update frequency due to computational constraints on stopping outcomes. Solid lines denote the rate of missed stops; shaded bands represent the rate of marginal stops. n denotes the number of trials per data point.

40 km/h across all tested ζ values (e.g., over 50% at $\zeta = 0.6$ and above 10% even at $\zeta = 1.0$). In contrast, both AdROD variants achieve zero missed stops across these setups.

Fig. 17 also shows that both AdROD variants leave substantially larger safety margins than the vanilla model. In addition, they excel at different speeds. At moderate speeds (20–35 km/h), AdROD-II achieves slightly larger margins, as once the ensemble defense is triggered, it can precisely and continuously track the stop sign and invoke the stop procedure. As for AdROD-I, if the attack does not immediately induce a significant disagreement, stop sign recovery as well as the stop procedure may be delayed. However, at 40 km/h, this trend reverses. The delayed activation of AdROD-II, which confirms an adversarial scenario after ℓ frames of loss of the tracked sign, consumes greater braking distance at high speeds. In contrast, AdROD-I operates continuously and produces earlier braking in these experiments.

2) Robustness Under Compute Constraints: The earlier results are obtained at a perception-control rate of 20 Hz.

To understand how computational constraints, such as those caused by resource contention, affect the safety of the undefended and defended systems, we simulate the scenario where the perception (including object detection) is executed at lower rates but the vehicle control is still performed at 20 Hz. To deal with the difference between their execution rates, the latest perception results are used by the vehicle control.

Fig. 18 shows the rates of missed stop and marginal stop versus the perception rate under various vehicle cruise speeds. With the undefended vanilla model, the two rates are mainly affected by the vehicle speed, rather than the perception rate.

The AdROD variants effectively suppress missed stops. At 20–30 km/h, missed and marginal stop rates remain near zero across all perception rates. Degradation becomes pronounced only at 40 km/h, where AdROD-I maintains zero missed stops, while AdROD-II shows increased missed and marginal stops at 5–10 Hz. This is because high vehicle speeds combined with reduced perception rates weaken the temporal continuity available for object tracking, making it more difficult for AdROD-II to detect an abrupt disappearance and activate the defense in time. Nevertheless, AdROD-II's near-baseline overhead of 1.03λ leaves greater computational headroom for maintaining the perception rate, although the realized rate under resource contention remains platform-dependent.

VII. Discussion

AdROD focuses on hiding attacks against genuine objects in camera-based object detection. It does not address attacks that fabricate *fake objects*, such as phantom signs [50], which may even deceive human observers. Defending against object-creation attacks would require additional mechanisms, such as object existence verification or contextual reasoning. AdROD is also specialized for bounding-box object detectors. While the idea of stochastic model generation with functional diversity could be extended to other perception tasks, such as semantic segmentation or instance segmentation, doing so would require task-specific output fusion, uncertainty estimation for AdROD-I, and activation logic for AdROD-II. We leave these extensions as future work. Finally, AdROD-II may fail to activate if an attack-instrumented object first appears when the patch is already effective and remains undetected thereafter, leaving no prior track to trigger the ensemble. Although maintaining such continuous suppression in practice is difficult, we identify this as a limitation of AdROD-II. Periodic ensemble checks can serve as a practical fallback.

VIII. Conclusion

This paper presented AdROD, a stochastic ensemble defense for camera-based object detection in autonomous driving. It includes an efficient HyperNetwork training framework and introduces functional diversity to enhance defense performance. With two serving modes, AdROD supports different deployment goals: robust detection in AdROD-I and runtime-efficient defense in AdROD-II. Across diverse attack settings, AdROD outperforms representative baseline defenses and generalizes better than evaluated adversarial-training baselines across the tested shifts.

References

[1] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, F. Tramèr, A. Prakash, T. Kohno, and D. Song, "Physical adversarial examples for object detectors," in *Proc. USENIX WOOT*, 2018.

[2] W. Jia, Z. Lu, H. Zhang, Z. Liu, J. Wang, and G. Qu, "Fooling the eyes of autonomous vehicles: Robust physical adversarial examples against traffic sign recognition systems," in *Proc. NDSS*. The Internet Society, 2022.

[3] N. Wang, Y. Luo, T. Sato, K. Xu, and Q. A. Chen, "Does physical adversarial example really matter to autonomous driving? towards system-level effect of adversarial object evasion attack," in *Proc. IEEE/CVF ICCV*, 2023, pp. 4412–4423.

[4] Y. Zhao, H. Zhu, R. Liang, Q. Shen, S. Zhang, and K. Chen, "Seeing isn't believing: Towards more robust adversarial attack against real world object detectors," in *Proc. ACM CCS*. ACM, 2019, pp. 1989–2004.

[5] N. Wang, S. Xie, T. Sato, Y. Luo, K. Xu, and Q. A. Chen, "Revisiting physical-world adversarial attack on traffic sign recognition: A commercial systems perspective," in *Proc. NDSS*. The Internet Society, 2025.

[6] H. Zhang and J. Wang, "Towards adversarially robust object detection," in *Proc. IEEE/CVF ICCV*. IEEE, 2019, pp. 421–430.

[7] A. Amirkhani, M. P. Karimi, and A. Banitalebi-Dehkordi, "A survey on adversarial attacks and defenses for object detection and their applications in autonomous vehicles," *Vis. Comput.*, vol. 39, no. 11, pp. 5293–5307, 2023.

[8] M. Naseer, S. H. Khan, and F. Porikli, "Local gradients smoothing: Defense against localized adversarial attacks," in *Proc. WACV*. IEEE, 2019, pp. 1300–1307.

[9] J. Liu, A. Levine, C. P. Lau, R. Chellappa, and S. Feizi, "Segment and complete: Defending object detectors against adversarial patch attacks with robust patch detection," in *Proc. IEEE/CVF CVPR*. IEEE, 2022, pp. 14 973–14 982.

[10] F. Tramèr, N. Carlini, W. Brendel, and A. Madry, "On adaptive attacks to adversarial example defenses," in *Proc. NeurIPS*, 2020.

[11] D. Ha, A. M. Dai, and Q. V. Le, "Hypernetworks," in *Proc. ICLR*, 2017.

[12] S. Jajodia, A. K. Ghosh, V. Swarup, C. Wang, and X. S. Wang, "Moving target defense: creating asymmetric uncertainty for cyber threats," 2011.

[13] Q. Song, Z. Yan, and R. Tan, "Moving target defense for embedded deep visual sensing against adversarial examples," in *Proc. ACM SenSys*. ACM, 2019, pp. 124–137.

[14] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," in *Proc. ICLR*, 2022.

[15] Á. Casado-García and J. Heras, "Ensemble methods for object detection," in *Proc. ECAI*, ser. Frontiers in Artificial Intelligence and Applications, vol. 325, 2020.

[16] R. Xu, Y. Guo, X. Han, X. Xia, H. Xiang, and J. Ma, "Opencda: an open cooperative driving automation framework integrated with co-simulation," in *Proc. IEEE ITSC*. IEEE, 2021, pp. 1155–1162.

[17] C. Guo, M. Rana, M. Cissé, and L. van der Maaten, "Countering adversarial images using input transformations," in *Proc. ICLR*, 2018.

[18] C. Xiang, A. Valtchanov, S. Mahloujifar, and P. Mittal, "ObjectSeeker: Certifiably robust object detection against patch hiding attacks via patch-agnostic masking," in *Proc. IEEE S&P*. IEEE, 2023, pp. 1329–1347.

[19] B. Tarchoun, A. B. Khalifa, M. A. Mahjoub, N. B. Abu-Ghazaleh, and I. Alouani, "Jedi: Entropy-based localization and removal of adversarial patches," in *Proc. IEEE/CVF CVPR*. IEEE, 2023, pp. 4087–4095.

[20] G. Jocher, "Yolov5 by ultralytics," 2020, accessed: 2025-03-14.

[21] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017.

[22] C. He, X. Ma, B. B. Zhu, Y. Zeng, H. Hu, X. Bai, H. Jin, and D. Zhang, "Dorpatch: Distributed and occlusion-robust adversarial patch to evade certifiable defenses," in *Proc. NDSS*. The Internet Society, 2024.

[23] S. Chen, C. Cornelius, J. Martin, and D. H. P. Chau, "Shapeshifter: Robust physical adversarial attack on faster R-CNN object detector," in *Proc. ECML PKDD*, ser. Lecture Notes in Computer Science, vol. 11051. Springer, 2018, pp. 52–68.

[24] D. Guo, Y. Wu, Y. Dai, P. Zhou, X. Lou, and R. Tan, "Invisible optical adversarial stripes on traffic sign against autonomous vehicles," in *Proc. ACM MobiSys*. ACM, 2024, pp. 534–546.

[25] A. Zolfi, M. Kravchik, Y. Elovici, and A. Shabtai, "The translucent patch: A physical and universal attack on object detectors," in *Proc. IEEE/CVF CVPR*, 2021, pp. 15 232–15 241.

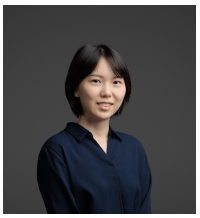
[26] T. Sato, S. H. V. Bhupathiraju, M. Clifford, T. Sugawara, Q. A. Chen, and S. Rampazzi, "Invisible reflections: Leveraging infrared laser reflections

to target traffic sign perception," in *Proc. NDSS*. The Internet Society, 2024.
[27] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok, "Synthesizing robust adversarial examples," in *Proc. ICML*, vol. 80. PMLR, 2018, pp. 284–293.
[28] J. Im Choi and Q. Tian, "Adversarial attack and defense of yolo detectors in autonomous driving scenarios," in *Proc. IEEE Intell. Veh. Symp. (IV)*. IEEE, 2022, pp. 1011–1017.
[29] P. Chiang, C. Chan, and S. Wu, "Adversarial pixel masking: A defense against physical attacks for pre-trained object detectors," in *Proc. ACM MM*. ACM, 2021, pp. 1856–1865.
[30] K. Xu, Y. Xiao, Z. Zheng, K. Cai, and R. Nevatia, "PatchZero: Defending against adversarial patch attacks by detecting and zeroing the patch," in *Proc. WACV*. IEEE, 2023, pp. 4632–4641.
[31] G. Rossolini, A. Biondi, and G. C. Buttazzo, "Attention-based real-time defenses for physical adversarial attacks in vision applications," in *Proc. ACM/IEEE ICCPS*. IEEE, 2024, pp. 23–32.
[32] F. Tramèr, A. Kurakin, N. Papernot, I. J. Goodfellow, D. Boneh, and P. D. McDaniel, "Ensemble adversarial training: Attacks and defenses," in *Proc. ICLR*, 2018.
[33] T. Pang, K. Xu, C. Du, N. Chen, and J. Zhu, "Improving adversarial robustness via promoting ensemble diversity," in *Proc. ICML*. PMLR, 2019, pp. 4970–4979.
[34] H. Yang, J. Zhang, H. Dong, N. Inkawhich, A. Gardner, A. Touchet, W. Wilkes, H. Berry, and H. Li, "DVERGE: diversifying vulnerabilities for enhanced robust generation of ensembles," in *Proc. NeurIPS*, 2020.
[35] S. Wang, X. Wang, P. Zhao, W. Wen, D. Kaeli, P. Chin, and X. Lin, "Defensive dropout for hardening deep neural networks under adversarial attacks," in *Proc. ACM/IEEE ICCAD*, 2018, pp. 1–8.
[36] G. S. Dhillon, K. Azizzadenesheli, Z. C. Lipton, J. Bernstein, J. Kossaifi, A. Khanna, and A. Anandkumar, "Stochastic activation pruning for robust adversarial defense," in *Proc. ICLR*, 2018.
[37] Q. Song, Z. Yan, W. Luo, and R. Tan, "Sardino: Ultra-fast dynamic ensemble for secure visual sensing at mobile edge," in *Proc. EWSN*. Junction Publishing / ACM, 2022, pp. 24–35.
[38] Y. Xu, D. Guo, Q. Song, Y. Lou, Y. Zhu, J. Wang, C. Qiao, and R. Tan, "Dynamic defense for car-borne lidar vehicle detection," in *Proc. ACM MobiSys*, 2025, pp. 431–444.
[39] C. E. Shannon, "Communication theory of secrecy systems," *Bell Syst. Tech. J.*, vol. 28, no. 4, pp. 656–715, 1949.
[40] T. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," in *Proc. ECCV*, ser. Lecture Notes in Computer Science, vol. 8693. Springer, 2014, pp. 740–755.
[41] C. Lv, L. Li, S. Zhang, G. Chen, F. Qi, N. Zhang, and H. Zheng, "Hyperlora: Efficient cross-task generalization via constrained low-rank adapters generation," in *Findings of EMNLP*. Association for Computational Linguistics, 2024, pp. 16 376–16 393.
[42] R. Majumdar, V. S. Jadhav, A. Deodhar, S. Karande, L. Vig, and V. Runkana, "Pihlora: Physics-informed hypernetworks for low-ranked adaptation," in *NeurIPS Workshop*, 2023.
[43] M. AprilPyone and H. Kiya, "Block-wise image transformation with secret key for adversarially robust defense," *IEEE Trans. Inf. Forensics Secur.*, vol. 16, pp. 2709–2723, 2021.
[44] A. P. M. Maung, I. Echizen, and H. Kiya, "Efficient key-based adversarial defense for ImageNet by using pre-trained model," *IEEE Open J. Signal Process.*, vol. 5, pp. 902–913, 2024.
[45] X. Han, H. Wang, K. Zhao, G. Deng, Y. Xu, H. Liu, H. Qiu, and T. Zhang, "VisionGuard: Secure and robust visual perception of autonomous vehicles in practice," in *Proc. ACM CCS*, 2024, pp. 1864–1878.
[46] W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, and T. Kim, "Multiple object tracking: A literature review," *Artif. Intell.*, vol. 293, p. 103448, 2021.
[47] S.-C. Lin, Y. Zhang, C.-H. Hsu, M. Skach, M. E. Haque, L. Tang, and J. Mars, "The architectural implications of autonomous driving: Constraints and acceleration," in *Proc. ACM ASPLOS*, 2018, pp. 751–766.
[48] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proc. CoRL*, 2017.
[49] E. T. Campolettano, K. D. Kusano, and T. Victor, "Potential safety benefits associated with speed limit compliance in San Francisco and Phoenix," *Traffic Inj. Prev.*, vol. 26, pp. S21–S30, 2025.
[50] B. Nassi, Y. Mirsky, D. Nassi, R. Ben-Netanel, O. Drokin, and Y. Elovici, "Phantom of the ADAS: Securing advanced driver-assistance systems from split-second phantom attacks," in *Proc. ACM CCS*, 2020, pp. 293–308.

Yuting Wu received the B.Eng. degree from Northeastern University, China, in 2019, and the M.Sc. degree from Nanyang Technological University (NTU), Singapore, in 2020. From 2020 to 2026, he was a Research Associate at NTU. He is currently pursuing the Ph.D. degree with the College of Computing and Data Science, NTU. His research interests include autonomous driving, reinforcement learning, adversarial attacks, and world models.

Dongfang Guo received the B.Eng. degree from the University of Electronic Science and Technology of China, China, and the M.Sc. and Ph.D. degrees from Nanyang Technological University, Singapore. He is currently a Research Fellow with the College of Computing and Data Science, NTU. His research interests include reliable and secure sensing for the Internet of Things and cyber-physical systems, with particular interests in robust perception for autonomous systems and resource-aware edge AI.

Xiangzhong Luo received the B.E. degree from Shanghai Jiao Tong University, China, in 2019, and the Ph.D. degree from Nanyang Technological University, Singapore, in 2023. From 2024 to 2025, he was a Research Fellow at Nanyang Technological University. He is now an Associate Professor at the School of Computer Science and Engineering, Southeast University, China. His research interests include embedded systems, neural architecture search, model compression, and LLM inference acceleration.



Qun Song received the B.Eng. degree from Nankai University, China, and the Ph.D. degree from Nanyang Technological University, Singapore. She is currently an Assistant Professor with the Department of Electrical Engineering, City University of Hong Kong. From 2022 to 2024, she was an Assistant Professor with the Embedded Systems Group, Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, the Netherlands, and from 2024 to 2025, with the Information Systems Technology and Design Pillar, Singapore University of Technology and Design. Her research interests include Artificial Intelligence of Things (AIoT), cyber-physical system robustness and resilience, and autonomous driving security and safety. Her honors include the 2023 MobiCom Best Community Contribution Award, the 2022 SenSys Best Paper Award Finalist, and the 2021 IPSN Best Artifact Award Runner-up.

Rui Tan is a Full Professor at College of Computing and Data Science, Nanyang Technological University, Singapore. Previously, he was a Research Scientist (2012-2015) and a Senior Research Scientist (2015) at Advanced Digital Sciences Center of University of Illinois at Urbana-Champaign, and a postdoctoral Research Associate (2010-2012) at Michigan State University. He received the Ph.D. (2010) degree in computer science from City University of Hong Kong, the B.S. (2004) and M.S. (2007) degrees from Shanghai Jiao Tong University. His research interests include cyber-physical systems and Internet of things. He is the recipient of Best Demo Award from SenSys'24, Best Paper Awards from ICCPS'23 and IPSN'17. He served as Associate Editor of IEEE Transactions on Mobile Computing and ACM Transactions on Sensor Networks, TPC Co-Chair of e-Energy'23, EWSN'24, SenSys'24, and General Co-Chair of e-Energy'24 and RTCSA'25. He received the Distinguished TPC Member recognition from SenSys in 2025 and from INFOCOM in 2017, 2020, and 2022.