

Article

Evaluating Perturbation Support and Image Restoration for Remote Sensing Object Detection

Yimang Li ¹, Wenkai Pan ^{1,*}, Jin Liu ² and Xilong Lu ²
¹ School of Mechanical Engineering and Rail Transit, Changzhou University, Changzhou 213164, China; liyimang@cczu.edu.cn

² Institute of Forensic Science of China, Beijing 100038, China

* Correspondence: s23050858034@smail.cczu.edu.cn

Abstract

We evaluate perturbation support and image-restoration controls on a locked remote sensing object-detection benchmark. On 100 FAIR1M images, a perturbation-free 640-pixel attack/export path loses 113 of 1247 original clean matches and gains 104 new matches, giving 9.06% target-conditioned ASR despite similar pooled Recall. Relative to the clean image passed through that same path, Global, Box, Polygon and SAM PGD-ZS yield 61.47%, 32.55%, 26.66% and 24.39% ASR. A separate equal-area comparison gives a source-referenced ASR of 28.39% for target-aligned SAM versus 11.73% averaged over random offsets; SAM minus target-aligned geometry is -0.88 percentage points (95% CI, -2.42 to 0.42), leaving a SAM-specific contour benefit unresolved. Five-fold held-out restoration and two additional repetitions separate resampling, learned correction and confidence selection. The correction-disabled 384-pixel path lowers envelope-IoU Recall from 42.37% to 32.89%; learned correction at gain 0.15 raises it to 36.40%. With the same confidence gate, correction-disabled and learned paths give 43.80% and 44.24% Recall. Incremental learned-versus-control Recall and native Ultralytics ProbIoU-based AP intervals span zero at both gains; gain 0.5 adds 48 false positives (95% CI, 1 to 103). Bilateral filtering improves native mAP50 by 2.75 points (0.78 to 4.82) and Recall by 1.60 points (0.47 to 2.84). Exact-forward BPDA gives pipeline Recall values of 23.40% and 26.86% at gains 0.15 and 0.5. These results support a controlled evaluation protocol while leaving contour-specific and incremental gated-learning benefits unestablished. Restoration uses paired-clean oracle masks, and intervals condition on the fixed detector, attacks and saved models.

Keywords: remote sensing object detection; adversarial examples; SAM semantic mask; YOLOv8-OBB; image restoration; attack success rate; image fidelity



Academic Editor: Francesco Zirilli

Received: 17 September 2026

Revised: 16 October 2026

Accepted: 19 October 2026

Published: 20 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 \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Satellite, unmanned-aerial and airborne images combine very large canvases, extreme object-scale variation, cluttered land-cover texture and arbitrary object orientation. DOTA1.5 contains dense rotated aircraft, ships and vehicles; DIOR broadens category and scene diversity; and FAIR1M provides fine-grained aircraft, ship and vehicle classes [1–3]. Horizontal boxes may include large sea or ground regions, while resizing attenuates the high-frequency boundaries of tiny targets. Deep detectors are sensitive to adversarial inputs. For source image x_{src} and detector f_{θ} , the clean attack tensor and optimized tensor are $x_{640} = R_{640}(x_{src})$, $x_{640}^{adv} = \text{clip}(x_{640} + \delta)$ subject to $\|\delta\|_{\infty} \leq \epsilon = \frac{8}{255}$. The budget and fixed support apply to the 640×640 tensor relative to its resized clean reference. The quantized tensor is then exported to source dimensions; even zero perturbation can alter source

pixels outside the corresponding support. This threat model follows adversarial-example, gradient-attack, and robust-optimization studies [4–7]. A global perturbation can obtain a high ASR by modifying the background, but it does not establish that the target itself was attacked. Box and polygon supports are more local but can still contain a background or depend on manual annotations. Patch and physical-attack studies further show that support area and transformation conditions change detector failure modes [8–11].

This study evaluates three questions on a locked benchmark: whether target-aligned semantic contours improve attacks beyond equal-area geometric controls; how resampling affects measured target losses and recovery; and whether learned correction adds task-metric benefit over a correction-disabled path with the same confidence gate. Gradient attacks and robust optimization define threat models [5–7], while localized and physical attack studies motivate explicit support and transformation assumptions [8–11]. Input transformations, feature denoising and representation-guided denoising provide different defense mechanisms [12–14]. The contribution is a target-conditioned paired evaluation protocol with explicit image paths and uncertainty intervals. Negative and inconclusive comparisons are reported as outcomes. The accompanying archive supplies implementation, image and fold lists, target identities, AP statistics and regeneration scripts.

The evaluation has four stages. First, frozen YOLOv8/YOLOv8-OBB detectors establish clean target matches. Second, FGSM, zero-start iterative PGD (BIM/PGD-ZS) and MIFGSM use a decoded class-probability suppression objective, with fixed Global, Box, Polygon and clean-prompted SAM supports. Equal-area translations and target-aligned geometric contours separate placement from contour effects; a post hoc equal-L1 control examines realized-energy sensitivity. Third, attacked images pass through classical transforms, oracle-assisted prototype repair or held-out learned SAM-guided Feature Restoration (SAM-FR), with correction-disabled and confidence-gated controls. Fourth, task metrics and paired intervals compare these branches on the same 100 FAIR1M images; a separate defense-aware PGD replay evaluates the raw and restored branches with the actual gate. SAM supplies a contour from a detector box prompt, while annotation-derived oriented boxes provide an independent locality proxy. This design evaluates detector utility and image fidelity without assuming semantic contours or reconstruction losses preserve object features.

Two-stage detectors obtain localization through proposals, whereas one-stage YOLO models optimize speed and dense multi-scale prediction [15]. Feature pyramids support small-object representation [16], and rotation-aware detectors reduce the background included by horizontal boxes [17,18]. YOLOv8/YOLOv8-OBB is selected because one PyTorch pipeline can cover horizontal and oriented annotations while exposing the gradients required by white-box attacks [19,20]. Defense assumptions are equally diverse. Input transformations, randomized smoothing, activation pruning, feature denoising, and representation-guided denoising each remove or suppress a different form of adversarial evidence [12–14,21,22]. Low-cost filtering can also remove aircraft wings and ship boundaries; end-to-end robust training is more principled but requires detector retraining; restoration leaves detector weights unchanged but risks reconstruction artifacts. The reported comparison therefore separates classical transformations, prototype repair, learned restoration, and confidence fallback.

2. Materials and Methods

2.1. Study Design and Evaluation Principles

We freeze detector weights, image lists, annotations and matching thresholds before attack generation. ASR uses objects matched on clean images, not all annotations or native validator Recall. The corrected gradient target is the mean decoded class probability in Section 3.1.1, distinct from the native detector training loss. FAIR1M training combines classification, rotated box regression, distribution focal loss and a separately logged angle

loss without separate objectness. Our corrected surrogate excludes angle and includes every class; error modes are assessed after attack generation. SAM prompts and support masks are precomputed from clean inputs and held fixed during optimization. The retrospective mask replay in Section 2.4 cannot recover original in-run fallback flags. The clean-reference effective repair mask, unlike the fixed attack support, is recomputed at every adaptive iteration.

Object scale and geometry affect restricted attacks. Scale groups use ground-truth axis-aligned envelope area divided by image area: small <1%, medium 1–5% and large $\geq 5\%$. Attack losses and clean-matched denominators are reported within each group, with aspect-ratio, orientation and crowding diagnostics separately identified. ASR measures target-match loss; Recall measures pooled task performance; Loc/Leak measure residual concentration against the annotation-derived oriented-box proxy; and fidelity metrics describe image changes. Under a common L-infinity limit, support area also changes editable pixels and realized energy. Equal-L1 normalization, equal-area translations and target-aligned geometry therefore test distinct factors. Restoration training uses only SAM-supported PGD-ZS pairs. Section 3.3.1 defines the 640-pixel attack and 384-pixel restoration paths; no additional tiling or tile-merging stage is used.

ASR first establishes clean target indices and then counts loss of those same class-consistent matches. A confidence change that preserves matching is not an ASR event; localization or classification changes count when the target no longer matches. Section 3.3 distinguishes the envelope matcher, polygon diagnostics and native validator metrics. Background leakage, support omission and competing gradients are possible explanations for support-dependent failures rather than measured causal mechanisms. Image stems, configurations and hashes connect lossless metric files to the records in the reproducibility archive (Supplementary Materials). Figure 1 summarizes the information flow; the adaptive runtime window is specified in Section 4.3.

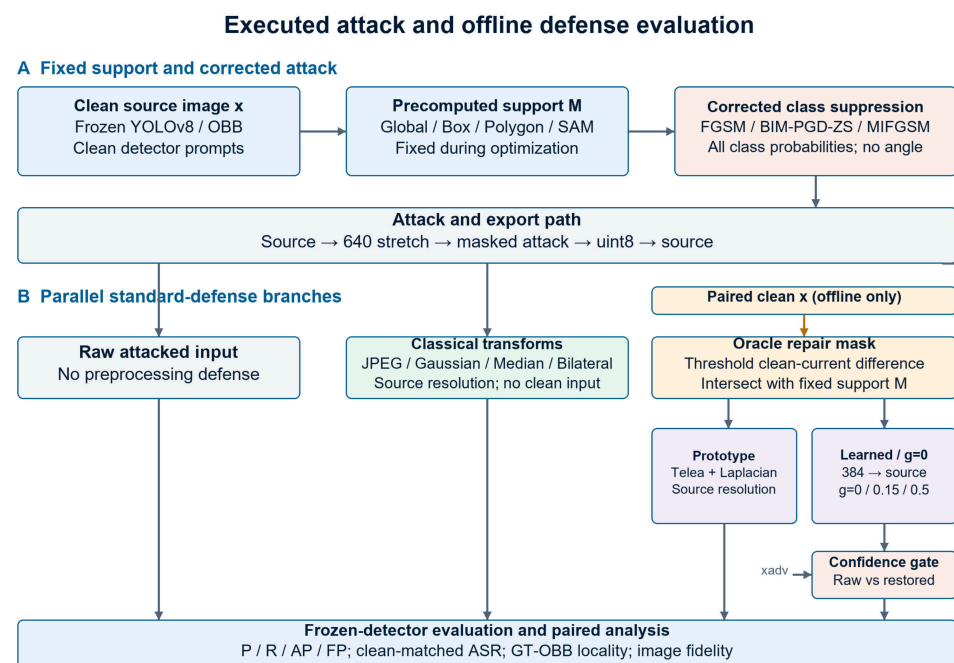


Figure 1. Executed standard-attack and offline defense evaluation. Saved supports remain fixed; BIM and PGD-ZS are one zero-start algorithm. Raw input, classical transforms and oracle-assisted restoration are parallel branches, not a serial filter-restorer pipeline. Only the oracle mask branch receives the paired clean reference. Prototype repair operates at source resolution, whereas the learned and correction-disabled controls pass through 384 pixels before source-resolution export. The confidence gate compares raw and learned-restored predictions. No feedback adjusts support or attack budget; the separate adaptive path is specified in Section 3.3.2.

2.2. Datasets, Annotations and Detector Baselines

DOTA1.5 uses eight-point oriented boxes, DIOR uses horizontal boxes and FAIR1M uses 37 oriented classes. Dataset integrity scripts found no missing images, labels or invalid annotation lines. The train/validation image counts are 1129/282, 5862/5863 and 16,488/8287, respectively. Annotation geometries are not interchangeable; oriented labels require consistent vertex ordering and angle conventions [1–3,17,18,23]. Clean detection baselines for the three datasets are listed in Table 1.

Table 1. Native Ultralytics validation baselines: ProbIoU-based OBB AP for DOTA1.5/FAIR1M and horizontal-IoU AP for DIOR; native macro P/R at the max-F1 operating point.

Dataset	Precision	Validator Recall	mAP50	mAP50-95
DOTA1.5	61.91%	46.28%	46.36%	32.38%
DIOR	74.86%	60.75%	63.97%	45.81%
FAIR1M	47.84%	38.98%	31.57%	23.91%

Note: Validator Recall in Table 1 and attack-protocol clean-matching Recall are different estimators. Table 1 records native class-macro validation metrics at a shared confidence chosen to maximize the smoothed mean class F1 curve, not pooled Recall at confidence 0.25. Section 3.3 specifies the overlap, matching and threshold selection. The attack protocol instead runs the frozen detector at confidence 0.25 and computes class-consistent, greedy one-to-one matches at axis-aligned envelope IoU 0.50, so $R_{\text{attack}} = N_{\text{match}}/N_{\text{GT}}$. Consequently, the DOTA1.5 values 46.28% and 20.58% are both based on the same 282 validation images but have different evaluators and operating points; they are not attributed to different subsets and are not used interchangeably.

The final FAIR1M detector is YOLOv8n-OBB with 640 input, batch size 8, AMP and 30 epochs. Its mAP50 is 31.57%. The 27.66% result from the discarded 10-epoch stage is reported as evidence of under-training and is excluded from final attack claims. This helps prevent an incompletely learned detector from being misinterpreted as robust. The benchmarks have distinct experimental roles. DOTA1.5 tests a full validation protocol with dense rotated instances; DIOR exposes the sensitivity of a horizontal-box detector to iterative attacks; FAIR1M supplies the fixed subset used for semantic-support and defense ablations. Cross-dataset ASR is not interpreted as a pure architecture effect because detector maturity, clean-matching denominator, object scale, category granularity and annotation geometry all vary. Images are not selected after observing attack outcomes. Each clean prediction is matched to ground truth first, and the identical image indices are then attacked and defended. This paired design makes the ASR denominator explicit and allows the per-image CSV records to be reaggregated without changing the sample population. Figure 2 shows the resulting clean detector baseline.

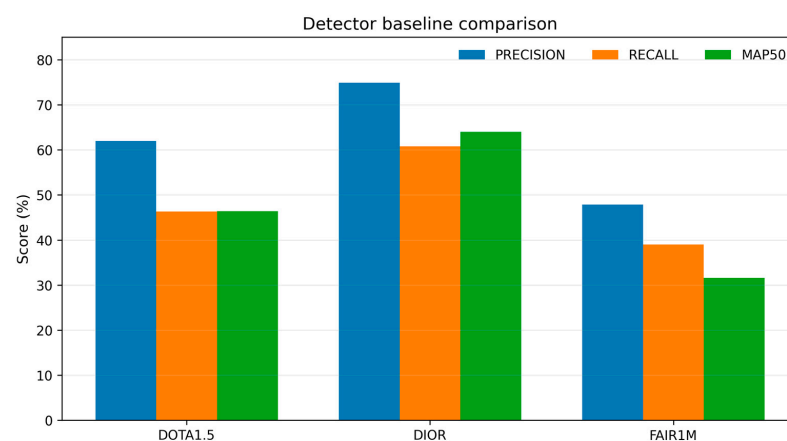


Figure 2. Native Ultralytics clean detector baselines; OBB AP uses ProbIoU, and horizontal DIOR AP uses ordinary IoU. Native P/R use the macro max-F1 estimator of Table 1.

2.3. YOLOv8-OBb and Multi-Scale Detection

YOLOv8 comprises a Backbone, Neck and Head. C2f and SPPF extract hierarchical features, top-down/bottom-up fusion combines scales, and the head predicts classes and boxes [15,16,19]. YOLOv8-OBb adds an angle and uses rotated NMS to reduce merging of crowded oriented objects. The archived detector training objective combines classification, rotated ProbIoU regression, distribution focal loss and a separately logged angle loss, without separate objectness. Attack gradients do not use that native composite loss: the corrected decoded class-probability surrogate is specified in Section 3.1.1. Localization and orientation outcomes are assessed afterward. Explicit attack preprocessing stretches source images to 640, restoration uses 384 and source-resolution detection adds native LetterBox; these paths are traced separately in Section 3.3.1 rather than described as one common 640 tensor. High-resolution features help represent small-object edges, while lower-resolution features provide scene context [16–18]. Figure 3 shows the detector architecture.

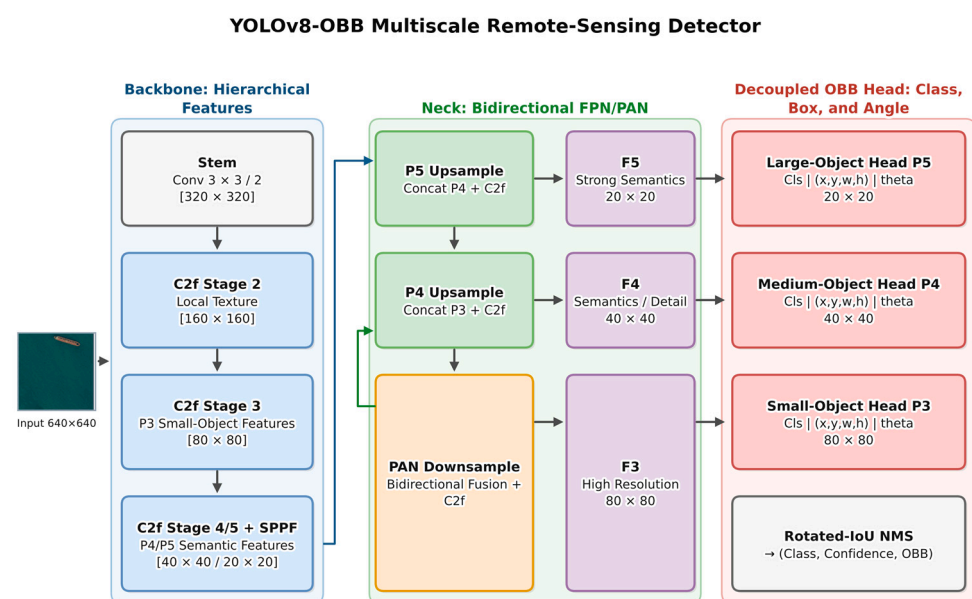


Figure 3. YOLOv8-OBb Backbone, bidirectional Neck, and multi-scale oriented heads.

The archived FAIR1M args.yaml specifies box = 7.5, cls = 0.5, dfl = 1.5 and angle = 1.0, and the training CSV records all four loss components. In Ultralytics 8.4.68, classification uses BCE, the rotated box term uses ProbIoU, DFL supervises distance distributions, and the separate angle term is an aspect-ratio-weighted periodic angular error. These settings describe this archived FAIR1M training run, not every YOLOv8-OBb release. The attack code does not call that native loss API: it differentiates the selected decoded-output values defined below. Its channel interpretation is therefore software- and head-layout-dependent; reproducing the gradient requires the same output layout and slice, rather than assuming release independence.

2.4. SAM Semantic Masks and Support Area

SAM contains an image encoder, prompt encoder and mask decoder [24]. In the executed pipeline, box prompts come from the frozen detector on the clean image, with annotations used only if no prediction is available. Returned masks are thresholded at 0.5 and combined by pixel-wise union. If inference fails or this union is empty, the same prompts supply an oriented-polygon or rectangular geometric fallback. Figure 4 shows this implemented support-generation path. Quality ranking, connected-component screening, geometry acceptance tests and scale-adaptive dilation were not executed and

are not depicted as parts of the evaluated method. The saved support is nearest-neighbor resized to attack resolution and held fixed throughout optimization; it is distinct from the clean-reference repair mask recomputed during adaptive attacks.

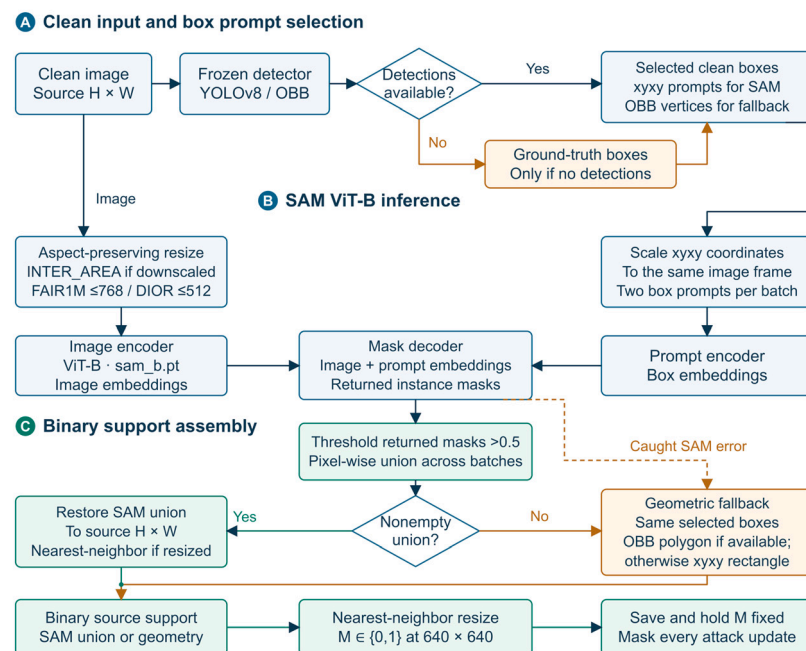


Figure 4. Detailed executed SAM support-generation pipeline: (A) clean-detector box selection with annotation fallback only when detections are absent; (B) coordinated image/prompt resizing and the SAM ViT-B image encoder, prompt encoder and mask decoder; (C) thresholding (>0.5), union, geometric failure fallback and nearest-neighbor mask mapping back to source size when needed and then to 640×640 . Resize limits refer to the longest side in pixels. The saved attack support M is held fixed and is distinct from the oracle repair mask in Equation (9). No ranking, geometry screening or dilation is performed.

The executed settings and reconstruction audit are as follows. The support is built from the released SAM ViT-B checkpoint (sam_b.pt) through the Ultralytics SAM interface, with detector boxes supplied as box prompts. For FAIR1M and DIOR, the input longest side is capped at 768 and 512 pixels, respectively; prompts are processed in batches of two to bound decoder memory. Returned masks are thresholded at probability 0.5 and combined by pixel-wise union into one binary support; instance count is handled by batching rather than by mask ranking. If SAM returns no mask or inference fails, the archived code falls back to a geometric mask made from the same prompts: an oriented polygon when the box carries OBB vertices, or a rectangle otherwise; prompts use clean detector predictions when available and annotations otherwise. The original attack runner did not save per-image fallback flags. We retrospectively replayed mask generation on the fixed FAIR1M and DIOR 100-image lists, preserving each archived detector, SAM checkpoint and prompt rule, batch size two, and longest-side limits of 768 and 512 pixels, respectively. SAM inference was moved to CPU to avoid current GPU memory errors. SAM returned masks for 100/100 FAIR1M images and 100/100 DIOR images, giving replay fallback counts of 0/100 and 0/100. Replayed supports matched the stored masks with mean IoU 0.99770 and 0.99981; 98/100 and 100/100, respectively, reached IoU at least 0.99. The geometric fallback exactly matched 0/100 FAIR1M and 0/100 DIOR stored masks. The two archived FAIR1M SAM runs had identical supports on all 100 images. These are retrospective reconstruction results, not original in-run logs. The reported configuration therefore contains neither candidate screening nor expansion: no quality, area-ratio or connected-component coefficients are

applied, and no morphological dilation or scale-adaptive expansion is performed. The support follows the SAM contour directly, so boundary pixels of thin structures are covered only when SAM itself includes them. This construction is more conservative and easier to reproduce than a tuned dilation rule, but the reported SAM results consequently inherit SAM's raw contour accuracy, including its failures on tiny, shadowed and crowded targets, and the absence of original per-image fallback flags remains a provenance limit despite the reconstruction audit.

2.5. Performance Indicators

Precision = $TP/(TP + FP)$, and Recall = $TP/(TP + FN)$. Fixed-threshold P/R use class-consistent one-to-one axis-aligned envelope IoU ≥ 0.50 at confidence 0.25. Native Ultralytics ProbIoU-based OBB mAP50 is computed at ProbIoU = 0.50; mAP50-95 averages ProbIoU thresholds from 0.50 to 0.95. Horizontal DIOR AP uses ordinary horizontal-box IoU; polygon-intersection IoU is reserved for rotated diagnostics. For a specified clean reference, target-conditioned ASR is $N(\text{cleanmatchedandadvunmatched})/N(\text{cleanmatched})$. The source-clean and identical attack/export clean references are defined in Section 3.3. Adv Recall and Recall drop are also reported so that a small clean denominator is not sufficient to inflate ASR. For fidelity, x and x_{adv} denote the paired clean and attacked tensors in the explicitly stated metric space. On normalized pixels, $\text{MAE} = \frac{1}{N} \sum |x_{\text{adv}} - x|$, $\text{RMSE} = \sqrt{\frac{1}{N} \sum (x_{\text{adv}} - x)^2}$ and $\text{PSNR} = 10\log_{10}(1/\text{MSE})$; SSIM closer to one indicates higher structural similarity [25,26]. L-infinity and changed-pixel ratio distinguish a few strong changes from widespread weak changes. Targets are grouped as small, medium and large by pixel area, and clean matches, adversarial unmatched objects and ASR are computed inside each group. The weighted sum of scale ASRs need not equal the macro average because the clean-matched denominators differ. Object-level counts are retained in the machine-readable output. Defense recovery is reported as $\text{Recovery} = (R_{\text{def}} - R_{\text{adv}})/(R_{\text{clean}} - R_{\text{adv}})$, where R_{clean} , R_{adv} and R_{def} are clean, adversarial and defended Recall. A negative value indicates secondary damage. Because the denominator becomes unstable when the attack causes little Recall loss, absolute Recall is shown alongside the normalized recovery. Here, N is the number of scalar RGB entries, $C \times H \times W$. MAE is the mean absolute error, not the unnormalized L1 norm, and RMSE is not the unnormalized L2 norm. Changed-pixel ratio counts spatial pixels with any changed channel. Fidelity summaries average per-image metrics rather than computing PSNR from a pooled MSE. Whole-image SSIM uses `skimage.metrics.structural_similarity` with `data_range = 1`, `channel_axis = 2` and its default 7×7 uniform window; local SSIM averages the channel-mean full SSIM map over the stated region, whose neighborhoods can extend beyond the mask.

3. Methodology and Experimental Protocol

3.1. Multiple Attack Methods

FGSM uses one sign-gradient step; BIM and PGD-ZS use repeated projected updates from zero perturbation. Canonical random-start PGD instead begins with a masked random perturbation. MIFGSM accumulates normalized historical gradients [4,6,27,28]. The implementation minimizes the detector-response surrogate J_{sup} or equivalently maximizes $L_{\text{attack}} = -J_{\text{sup}}$. The support-constrained ascent form is $\delta_{t+1} = \Pi_{[-\epsilon, \epsilon]}(\delta_t + \alpha \text{sign}(\nabla_{\delta} L_{\text{attack}}) \odot M)$. Thus, the negative update applied to J_{sup} in the code and the positive update written for L_{attack} are the same operation. Global has no spatial restriction; Box, Polygon and SAM use the corresponding support [8–11].

3.1.1. Threat Model and Objective

Let x_{src} denote the normalized source-resolution RGB image. R_{640} applies the implemented uint8 INTER_LINEAR stretch and normalization, giving the clean attack tensor $x_{640} = R_{640}(x_{\text{src}})$. The optimized tensor is $x_{640}^{\text{adv}} = \text{clip}_{[0,1]}(x_{640} + \delta)$. Q rounds it to uint8, and R_{src} applies INTER_LINEAR export to the original dimensions, giving $x_{\text{exp}} = R_{\text{src}}(Q(x_{640}^{\text{adv}}))$; the perturbation-free matched-path reference is $x_{0\text{exp}} = R_{\text{src}}(Q(x_{640}))$. These symbols denote different images. The 8/255 budget is relative to x_{640} , and the fixed support M_{640} constrains δ only in attack space. Even at $\delta = 0$, export can change source pixels outside the corresponding source support. Equations (1)–(3) optimize the frozen detector in this 640×640 space. P_{cls} contains decoded probabilities for all classes at every candidate; $c_M = \max(\text{mean}(M_{640}), 10^{-6})$ is a fixed positive area factor. We minimize J_{sup} , equivalently maximize L_{attack} , without annotations or another sigmoid. Removing c_M gives one common response for comparing feasible support sets; their inclusion relation describes ideal optima, not finite-step ASR.

The decoded non-end-to-end head has four box channels; n_c class-probability channels; and, for OBB, one angle channel. The corrected selector outputs[:,4:4+n_c,:] uses FAIR1M channels 4:41 of 42 ($n_c = 37$). Channel-selection tests verify nonzero gradients for every class, zero box/angle gradients and direct use of decoded probabilities. An archived-response control uses outputs[:,n_c,:].sigmoid(), selecting 5:42 and including angle while omitting class 0. Sections 4.2 and 4.3 compare these objectives with the same detector, supports and preprocessing. Primary attacks use the corrected objective; the training-response $\times$ test-response control retains separately labeled archived and corrected model distributions.

$$\delta^* = \operatorname{argmax}_{\delta \in \Omega_M} L_{\text{attack}}(\text{clip}(x_{640} + \delta); M_{640}) \quad (1)$$

$$J_{\text{sup}}(z; M_{640}) = c_M \operatorname{mean}(P_{\text{cls}}(z)), \quad L_{\text{attack}} = -J_{\text{sup}}, \quad z \in [0, 1]^{3 \times 640 \times 640} \quad (2)$$

$$\Omega_M = \left\{ \delta : \|\delta\|_{\infty} \leq \varepsilon = \frac{8}{255}, \delta = \delta \odot M_{640} \right\} \subseteq \Omega_{\text{Global}} \quad (3)$$

3.1.2. Derivation of Iterative Attacks

In Equations (4)–(7), x abbreviates the clean attack tensor x_{640} , M denotes M_{640} and each iterate remains in 640×640 space. Writing $L = L_{\text{attack}}$, the first-order approximation is $L(x + \delta) \approx L(x) + \delta^T \nabla_x L$. Maximization over an L-infinity ball gives $\delta_{\text{FGSM}} = \varepsilon \operatorname{sign}(\nabla_x L) \odot M$. BIM applies T clipped steps from $x_0 = x$. The main PGD-ZS results retain $x_0 = x$ and project after every update; the canonical random-start PGD condition samples $\delta_0 \sim U(-\varepsilon, \varepsilon)$ within the masked L-infinity ball before projected updates. MIFGSM uses $v_{t+1} = \mu v_t + \frac{d_t}{\max(\operatorname{mean}_{\text{CHW}}(|d_t|), 10^{-8})}$, where $d_t = \nabla_x J_{\text{sup}}(x_t; M)$. Momentum starts from zero with $\mu = 0.9$. The mean is computed separately for each image over all $C \times H \times W$ gradient entries, including entries outside M , and the denominator is floored at 10^{-8} . The descent step is $\tilde{x} = x_t - \alpha \operatorname{sign}(v_{t+1}) \odot M$, followed by projection onto the masked L-infinity ball centered at the original clean image x and clipping to $[0, 1]$. The mask multiplies the signed update, not the accumulated gradient. Since $L_{\text{attack}} = -J_{\text{sup}}$, the negative J_{sup} update corresponds to maximizing L_{attack} . Equation (6) specifies the canonical random-start branch. The reported PGD-ZS results follow Equation (5) with $x_0 = x$ and no random start, so their projection set is $B_{\infty}(x, \varepsilon)$. For $N = C \times H \times W$ elements, $\|d_t\|_1 = N \cdot \operatorname{mean}_{\text{CHW}}(|d_t|)$. L1-sum and mean normalization therefore produce different momentum magnitudes. Their signed updates coincide in exact arithmetic only under fixed N , zero initial momentum, identical gradient histories, inactive denominator floors and corresponding ascent/descent

sign conventions. Equation (7) and the MIFGSM results here use the mean-normalized implementation, not L1-sum normalization.

$$\delta_{\text{FGSM}} = \epsilon \text{sign}(\nabla_x L_{\text{attack}}) \odot M \quad (4)$$

$$x_{t+1} = \Pi_{B_\infty(x, \epsilon)}[x_t + \alpha \text{sign}(\nabla_x L_{\text{attack}}) \odot M] \quad (5)$$

$$x_0 = x + U(-\epsilon, \epsilon) \odot M; \quad x_{t+1} = \Pi_{B_\infty(x, \epsilon)}[x_t + \alpha \text{sign}(\nabla_x L_{\text{attack}}) \odot M] \quad (6)$$

$$v_{t+1} = \mu v_t + \frac{d_t}{\max(\text{mean}_{\text{CHW}}(|d_t|), 10^{-8})} \quad (7)$$

3.1.3. Locality and Leakage

Locality and leakage need an independent reference to be meaningful. If a perturbation is constrained to a support M and its locality is then measured with that same M , the ratio is identically one for every nonzero perturbation. Attack locality is therefore referred to the independent annotation-polygon proxy M_{GT} : the union of all four-corner FAIR1M ground-truth-oriented boxes, rasterized at 640×640 . $\text{Loc}_{\text{GT}} = \frac{\text{sum}(|\delta| \odot M_{\text{GT}})}{\text{sum}(|\delta|)}$, and $\text{Leak} = 1 - \text{Loc}_{\text{GT}}$; the ratio is computed per image and then averaged. The implementation first averages absolute RGB residuals across channels, sums this energy inside and outside the reference, and divides by the strictly positive total energy without an additive stabilizer. A zero-total residual triggers an error rather than being assigned locality one; none of the evaluated residuals has zero total energy. In Section 4.4, this M_{GT} is applied to Global, Box, Polygon and SAM on identical images. Target-region PSNR and SSIM there retain the separately identified common clean-image SAM region, so they are not annotation-region fidelity estimates. For restoration, Section 4.4 reports local PSNR/SSIM inside the effective repair mask M_δ , residual-error locality and leakage, and correction confinement. Because M_δ is an explicit restorer input, correction confinement near one is an architectural consistency check rather than independent evidence of mask quality. The same section additionally evaluates local PSNR/SSIM inside the common GT-OBB region; the original SAM-region fidelity remains separately identified in the same table.

$$\text{Loc}_{\text{GT}} = \frac{\text{sum}(|\delta| \odot M_{\text{GT}})}{\text{sum}(|\delta|)}, \quad \text{Leak} = 1 - \text{Loc}_{\text{GT}} \quad (8)$$

3.2. Image-Restoration Defenses

JPEG, Gaussian, Median and Bilateral are the classical input-transformation baselines evaluated here [13]; the guided filter is a closely related edge-preserving smoother [29]. In the paired offline benchmark, prototype SAM-FR forms an oracle repair region from the clean–adversarial difference, applies Telea inpainting, and adds local edge processing [30,31]. The learned SAM-FR oracle-assisted offline experiment uses a lightweight U-Net-like restorer [32,33]; ViT, MAE and hierarchical window-attention encoders are related designs, not components of this implementation [34–36]. Its training objective combines pixel, local, edge, frequency and residual terms as $\mathcal{L} = \mathcal{L}_{\text{pix}} + 2\mathcal{L}_{\text{local}} + 0.5\mathcal{L}_{\text{edge}} + 0.15\mathcal{L}_{\text{spec}} + 0.25\mathcal{L}_{\text{res}}$. $L_{\text{spec}} = \text{mean}(H \odot \text{abs}(|F_r(G(y))| - |F_r(G(x))|))$ is an asymmetric spectral-magnitude discrepancy, not a validated high-frequency preservation mechanism. Feature and representation-guided denoising [12,14] motivate evaluating detector utility separately from pixel reconstruction; they do not validate this particular spectral selector.

Exact classical configurations. All four transforms take the lossless source-resolution OpenCV BGR uint8 array, before detector LetterBox resizing, and return the same source dimensions. JPEG uses `cv2.imencode` followed by `cv2.imdecode(IMREAD_COLOR)`, with `IMWRITE_JPEG_QUALITY = 55`; no additional subsampling option is passed. Gaus-

sian uses GaussianBlur with a 3×3 kernel, sigmaX = 0.8 and default sigmaY = 0.8 and BORDER_DEFAULT. Median uses medianBlur with kernel size 3, whose edge handling is BORDER_REPLICATE. Bilateral uses bilateralFilter with diameter 5, sigmaColor = 35 in uint8 intensity units, sigmaSpace = 35 pixels and BORDER_DEFAULT. Clean and attacked inputs use these identical settings at source resolution; no 384×384 restoration stretch is used by these four transforms. Their outputs undergo the same native detector preprocessing as their unfiltered comparator. The saved preprocessing times exclude detector inference.

Prototype configuration. The prototype row of the defense comparison takes the source-resolution BGR uint8 attacked image and the effective oracle mask from Equation (9), binarized to 0/255. OpenCV inpaint uses INPAINT_TELEA with a radius of 3 pixels. The repaired image is converted to grayscale; cast to float32; and processed by cv2.Laplacian with CV_32F, kernel size 3 and BORDER_DEFAULT. A detail gain of 0.18 multiplies this same Laplacian before it is added to each of the three BGR channels; the result is clipped to [0, 255] and converted to uint8 by truncation. Only nonzero mask pixels are replaced, leaving all other source pixels byte-identical to the attacked input. An empty mask returns an image copy. This prototype has no 384-pixel stretch, learned checkpoint or confidence fallback in the defense comparison. Its detail gain is distinct from the learned residual gain g and does not establish preservation of object features.

Learned restorer: architecture and training protocol. The restorer is a small encoder-decoder with two downsampling stages and two skip connections. Each block uses two 3×3 convolutions, GroupNorm over four groups, and SiLU activations. The four-channel input concatenates the adversarial RGB image with the effective repair mask; the 96-channel bottleneck is decoded through 64- and 32-channel stages to a three-channel output head with tanh-bounded residuals, and correction is confined to the mask. Training uses 384×384 crops, AdamW [37,38] at 2×10^{-4} with betas = (0.9,0.999), eps = 1×10^{-8} and weight_decay = 0.01, batch size one, five epochs, eight deterministic augmented views per source pair (640 training entries per fold per epoch), horizontal and vertical flips, and a crop scale s drawn uniformly in [0.65,1.0], with crop height/width clamped between 384 pixels and the corresponding source dimension before INTER_AREA resizing to 384×384 ; the eight views are fixed by fold seed and sample index and reused across epochs, while the loader order is shuffled; these augmentations are applied directly to the image arrays rather than through a dedicated augmentation library such as Albumentations [39]. The 100-image FAIR1M list is shuffled once with split seed 42 and divided into five non-overlapping 20-image folds. Five models are independently initialized with seeds 42, 123, 2026, 31,415 and 27,182; each model trains on 80 distinct clean/PGD-ZS pairs and predicts only its 20-image held-out fold. The epoch-5 checkpoint is used in each fold; validation loss is logged but is not used for checkpoint selection. Concatenating the five held-out predictions therefore gives an out-of-fold result for all 100 images, with no image evaluated by a model that trained on it. The tanh-bounded residual is multiplied by gain g in Equation (9). Training uses $g = 0.15$, with held-out evaluation at 0.15 and the historic 0.5 workpoint. All five primary models are freshly initialized and trained on the regenerated corrected-objective SAM/PGD-ZS pairs, using exactly the original five-fold partitions, initialization seeds, augmentation order and hyperparameters. Historic checkpoints are retained as controls. A factorial control crosses the training objective and test objective at both gains, instead of assuming that changing the attack leaves the training distribution unaffected.

3.2.1. Oracle-Assisted Mask Purification and Reconstruction Loss

The candidate repair region is $M_\delta = \mathbb{I}(\text{gray}(|x_{\text{adv}} - x|) \geq \tau) \odot M$. The restorer receives $[x_{\text{adv}}, M_\delta]$ and predicts residual r_ϕ , producing $y = \text{clip}(x_{\text{adv}} + g \cdot M_\delta \odot r_\phi)$. Pixel, local,

edge, and asymmetric spectral-magnitude terms constrain color, contaminated regions, boundaries, and Fourier magnitudes; a masked residual penalty limits unnecessary correction. SSIM is an evaluation metric, not a training-loss term. The final loss is $\mathcal{L}_{\text{rec}} = \mathcal{L}_{\text{pix}} + 2\mathcal{L}_{\text{local}} + 0.5\mathcal{L}_{\text{edge}} + 0.15\mathcal{L}_{\text{spec}} + 0.25\mathcal{L}_{\text{res}}$. Here, $\mathcal{L}_{\text{res}}$ is the mean absolute masked correction relative to the adversarial input; the implemented loss weights are 1, 2, 0.5, 0.15 and 0.25, respectively. For the unnormalized real two-dimensional Fourier transform F_r and binary frequency selector $H(u,v)$, $\mathcal{L}_{\text{spec}} = \text{mean}(H \odot \text{abs}(|F_r(G(y))| - |F_r(G(x))|))$. Fourier-magnitude agreement does not preserve the phase or establish the identity of reconstructed edges. The archived selector is therefore not interpreted as selectively preserving high-frequency object structure. Here, $\tau = 2/255$. No ranking, dilation or other morphological post-processing is applied. Evaluation and the adaptive replay use Equation (9) even when it yields an empty mask. The training/validation loader alone substitutes the saved support M if the source-resolution thresholded mask is empty, before augmentation. A read-only audit finds no such cases in either the 100 historic or 100 corrected SAM/PGD-ZS source pairs (0/100 each); their directly resized 384 masks are also nonempty. This training safeguard is not an inference-time mask estimator. The weights in Equation (12) are the implemented values. The FFT is unnormalized and the reductions differ across loss terms; these coefficients are implementation settings, not calibrated relative importances. Pixel, edge and frequency terms use means rather than unnormalized L1 sums. In Equation (9), $r_\phi = \tanh(h_\phi([x_{\text{adv}}, M_\delta]))$, where h_ϕ denotes the encoder-decoder through its output head; r_ϕ is bounded elementwise between -1 and 1 . The nonnegative scalar g multiplies this residual before masking and clipping to $[0, 1]$. The reported training setting is $g = 0.15$, and the original detector-evaluation setting is $g = 0.5$. This g is distinct from the fallback gate γ in Equation (14).

Implementation-specific loss definitions. Let $E = |y - x|$, $G(z) = \text{mean}_C(z)$, and $e = G(y) - G(x)$. G is the arithmetic RGB-channel mean, not luminance-weighted grayscale. The difference operators with subscripts h and v take forward horizontal and vertical differences without boundary padding. Each mean or sum reduces every entry of the expression supplied to it. At training batch size one, the local numerator sums the RGB error after broadcasting the single-channel mask, while its denominator counts mask pixels without RGB replication and is clamped below at one. It is therefore not an RGB error averaged per masked scalar entry. The residual penalty is $\mathcal{L}_{\text{res}} = \text{mean}(M_\delta \odot |y - x_{\text{adv}}|)$. The real Fourier operator is `torch.fft.rfft2` applied to G , with the default unnormalized forward transform and no `fftshift`. For zero-based output indices u and v and spectral tensor dimensions (H_f, W_f) , the archived binary selector is $H(u,v) = \mathbb{I}[u/H_f > 0.12 \vee v/W_f > 0.12]$. This is asymmetric spectral weighting: indices near the end of the vertical dimension include low-absolute-frequency negative components that are selected while their positive counterparts can be excluded. It is neither a radial cutoff nor a sign-symmetric high-frequency selector. The frequency mean includes zero-weighted entries rather than dividing by the number of selected entries. Equations (10) and (11) retain these exact reductions because replacing a mean by an L1 sum changes the relative loss scaling.

$$M_\delta = \mathbb{I}(\text{gray}(|x_{\text{adv}} - x|) \geq \tau) \odot M; \quad y = \text{clip}(x_{\text{adv}} + g \cdot M_\delta \odot r_\phi) \quad (9)$$

$$\begin{aligned} \mathcal{L}_{\text{pix}} &= \text{mean}(E) \\ \mathcal{L}_{\text{local}} &= \frac{\text{sum}(M_\delta \odot E)}{\max(\text{sum}(M_\delta), 1)} \end{aligned} \quad (10)$$

$$\begin{aligned} \mathcal{L}_{\text{edge}} &= \text{mean}(|D_h e|) + \text{mean}(|D_v e|) \\ \mathcal{L}_{\text{spec}} &= \text{mean}(H \odot \text{abs}(|F_r(G(y))| - |F_r(G(x))|)) \end{aligned} \quad (11)$$

$$\mathcal{L}_{\text{rec}} = \mathcal{L}_{\text{pix}} + 2\mathcal{L}_{\text{local}} + 0.5\mathcal{L}_{\text{edge}} + 0.15\mathcal{L}_{\text{spec}} + 0.25\mathcal{L}_{\text{res}} \quad (12)$$

Frequency-coordinate control. We separately retrain five models with a sign-symmetric selector built from `torch.fft.fftfreq`(Himg) and `torch.fft.rfftfreq`(Wimg), in cycles per pixel. The control selects a bin when either absolute axis frequency exceeds 0.06 cycles/pixel, equivalently 0.12 of the Nyquist frequency. This excludes a square low-frequency region rather than a radial disk. At 384×384 , both selectors retain 98.478% of rfft2 bins, but place their excluded bins differently. All folds, initialization seeds, image pairs, augmentation order, optimizer, five-epoch schedule, loss coefficients, unnormalized FFT and full-tensor mean reduction are held fixed. Both gains are evaluated out of fold under the corrected ordinary attack. The corresponding comparison reports this isolated selector control; the primary comparison and the adaptive-attack variant retain the explicitly named asymmetric models and are not silently reassigned to the new models.

$$H_{\text{phys}}(u,v) = I[\max(2|f_y(u)|, 2f_x(v)) > 0.12] \quad (13)$$

The mask in Equation (9) is therefore an oracle mask rather than an inference-time mask. It is computable only in the paired offline benchmark, where both the clean image x and the adversarial image x_{adv} are retained. The reported prototype and learned SAM-FR values quantify reconstruction behavior when the contaminated support is known and should be interpreted as oracle-assisted offline benchmarks, not formal performance upper bounds for input-only methods. The current implementation cannot construct this mask during ordinary inference without x . A deployment-oriented variant would have to replace Equation (9) with an input-only estimator based on anomalous residual cues or detector/SAM proposals and then retrain and evaluate the complete pipeline without clean references; that variant is not evaluated in this study.

3.2.2. Detection-Conditioned Fallback

Let $S(z)$ aggregate post-NMS prediction confidence for image z . The gate $\gamma = I[S(y) \geq S(x_{\text{adv}}) + m]$ outputs $z = \gamma y + (1 - \gamma)x_{\text{adv}}$. It is an engineering risk control, not a certified defense; randomized smoothing offers a different probabilistic guarantee in the literature [21] and is not evaluated here.

In the implementation, $D(z)$ is the set of predictions retained after the fixed confidence threshold and non-maximum suppression, and $S(z)$ is the sum of their confidence values; $S(z)$ is set to zero when $D(z)$ is empty. The scores for the restored image y and attacked image x_{adv} are computed independently. The gate uses neither ground-truth boxes, clean-image detections, clean pixels, nor cross-branch object matching. It is therefore label-free, although the preceding repair-mask construction remains oracle-assisted. Because summed confidence may also increase through false positives, this gate is treated only as a heuristic for limiting restoration-induced damage, and its output is evaluated afterward with labeled detection metrics.

The gate uses the restored image when $S(y) - S(x_{\text{adv}}) \geq m$, where S is summed post-NMS confidence at 0.25 and m is an additive confidence-sum margin. The primary value is $m = 0$. The margin control applies five prespecified margins to the same corrected-attack inputs and freshly trained out-of-fold $g = 0.5$ predictions; no labels tune the margin. An empty prediction set has sum zero and a tie at $m = 0$ selects restoration. This label-free gate cannot distinguish additional true detections from false positives. The preceding repair mask remains clean-reference assisted. We additionally execute the identical gate with $g = 0$ on the corrected attacked inputs, retaining the network invocation, quantization, 384-down/source-up path and native confidence calculations. This isolates the combined resampling-and-selection control; it is not an identity-path gate. For clean-input evaluation, the oracle reference equals the input, so the effective repair mask is zero. All 100 held-out images are passed through their corresponding primary model at both gains; all 200

outputs are byte-identical to the clean 384-down/source-up image. The actual confidence gate is then evaluated on those outputs without labels. Thus, the two gains share one verified clean fallback result, while an input-only mask estimator and its false alarms remain unevaluated.

Post hoc gate-choice audit. We join the retained per-image confidence scores and selection decisions to the fixed-threshold TP/FP/FN records by image identifier. At margin $m = 0$, we verify that each selected output has exactly the counts of its chosen branch. For the clean zero-mask path and the attacked paths at $g = 0, 0.15$ and 0.5 , we count selected scenes with increased or decreased TP and FP relative to their own raw input. Image counts and pooled object-count changes are reported separately; TP-loss and FP-increase categories can overlap. Labels are used only for this retrospective assessment, never to change the gate. These descriptive counts add no new attack, training run or prediction pass.

$$\gamma = I[S(y) \geq S(x_{\text{adv}}) + m]; z = \gamma y + (1 - \gamma)x_{\text{adv}} \quad (14)$$

3.3. Settings and Confidence Fallback

The corrected FAIR1M replay holds the 100-image list, detector weights, labels, $\varepsilon = 8/255$, $\alpha = 2/255$, ten steps and IoU = 0.50 fixed. Ordinary attacks use a zero start unless explicitly labeled random-start. Native Ultralytics ProbIoU-based OBB AP uses confidence 0.001 and full AP-curve calculation; fixed-threshold counts use confidence 0.25. Five newly trained models give held-out predictions at each gain, with historic models retained in the factorial control. Paired image-level percentile intervals [40,41] use 10,000 re-samples and seed 20261012; native Ultralytics ProbIoU-based OBB AP is recalculated from per-image statistics for every draw. The primary comparison includes a complete correction-disabled path control; the adaptive-attack variant uses independently generated exact-forward adaptive attacks at each gain, with current repair masks and the actual confidence gate.

The saved configurations retained by the corrected replay in Table 2 use $\varepsilon = 8/255$, normalized float32 inputs in $[0, 1]$, confidence threshold 0.25, IoU threshold 0.50, and seed 42. DOTA1.5 uses Polygon support, $\alpha = 1.6/255$, $T = 5$, and 282 validation images; DIOR uses Box support, $\alpha = 2/255$, $T = 5$, and 100 validation images; and FAIR1M uses SAM support, $\alpha = 2/255$, $T = 10$, and 100 validation images. These dataset-specific values are read from the saved run configurations rather than inferred from library defaults. The archived FAIR1M training log records Ultralytics 8.4.68, Python 3.12.10, PyTorch 2.11.0 + cu128 and an NVIDIA RTX 5060 Ti under CUDA 12.8. The attack runner did not independently snapshot package versions. The SAM weights are Ultralytics sam_b.pt (ViT-B; SHA-256 EC2DF62732614E57411CDCF32A23FFDF28910380D03139EE0F4FCBE91EB8C912). The new replay independently verifies Python 3.12.10, Ultralytics 8.4.68, PyTorch 2.11.0 + cu128, CUDA 12.8, OpenCV 4.13.0, NumPy 2.4.6 and RTX 5060 Ti. The attack loop, masks and detector weights are otherwise held fixed within each archived-versus-corrected comparison.

The method-specific settings are as follows. FGSM performs one update with step magnitude ε and does not use α or momentum. BIM starts from $x_0 = x$ and performs T signed steps of size α . PGD-ZS uses the same zero initialization, α and T as BIM and projects the perturbation into the L-infinity ball after every step; the reported branch uses no random start and no restart. MIFGSM also starts from $x_0 = x$, uses the same α and T , normalizes each gradient by its mean absolute magnitude and accumulates momentum with $\mu = 0.9$. Every branch multiplies the update by the fixed spatial support, clips δ to $[-\varepsilon, \varepsilon]$, clips the adversarial image to $[0, 1]$ and keeps detector parameters frozen. The implementation is the project's explicit PyTorch attack loop; no unreported torch attack defaults are used.

Table 2. Source-referenced complete-path target-conditioned ASR under three dataset-specific attack protocols; class-consistent envelope-IoU matching at confidence 0.25. BIM and PGD-ZS are one algorithm.

Dataset	Support	FGSM ASR	BIM/PGD-ZS ASR	MIFGSM ASR	Clean Matched
DOTA1.5	Polygon	17.81% 1649/9261	19.87% 1840/9261	22.17% 2053/9261	9261
DIOR	Box	40.74% 132/324	86.11% 279/324	88.58% 287/324	324
FAIR1M	SAM	25.02% 312/1247	28.23% 352/1247	30.87% 385/1247	1247

Note: DOTA1.5 uses Polygon support, $\alpha = 1.6/255$ and five steps; DIOR uses Box support, $\alpha = 2/255$ and five steps; and FAIR1M uses SAM support, $\alpha = 2/255$ and ten steps. All retain $\epsilon \approx 8/255$, seed 42 and saved masks/weights. Each cell includes successful/clean-matched objects. The corrected selector includes every decoded class probability and excludes the OBB angle, without a second sigmoid. BIM and PGD-ZS share zero initialization and the same projected update; one column replaces the two historic aliases.

Clean-reference sets are protocol-specific. Source-referenced attacks in Table 2 use 9261/44,996 clean matches for DOTA1.5, 324/482 for DIOR and 1247/2308 for FAIR1M. FAIR1M support ASRs and scale counts instead use the 1238/2308 targets matched after the perturbation-free 640-down/source-up attack/export path. The support comparison retains both references. The independent equal-area and archived-objective comparisons retain their stated source-reference sets, while the separate 640-space equal-energy control uses 1218/2308 matches. Native validation Recall in Table 1 uses another estimator.

Main attack and fixed-threshold evaluator. Ground-truth OBB vertices are scaled into source-image pixels, and their minimum/maximum coordinates define horizontal envelopes; predicted OBBs use the xxyy envelope returned by the detector. Horizontal DIOR boxes use their xxyy coordinates directly. Overlap is intersection area divided by union area of these axis-aligned rectangles, without a pixel-inclusive +1; nonpositive union yields zero. Predictions are retained in post-NMS output order at confidence 0.25, with NMS threshold 0.50. Iterate targets in label-file order; among unused predictions of the same class, select $\max(\text{IoU}, \text{prediction index})$. Accept $\text{IoU} \geq 0.50$, and consume that prediction. Exact IoU ties therefore select the larger prediction index; target conflicts favor the earlier label row. ASR counts clean-matched target indices absent from the attacked matched set, divided by the clean set size. Precision and Recall pool TP/FP/FN across images, rather than averaging class Recalls. This rule gives 1247/2308 FAIR1M source-clean matches and 1238/2308 matches for the perturbation-free attack/export path. Primary SAM ASR is 302/1238 with the matched-path reference and 352/1247 with the source reference.

Rotated diagnostic evaluator. It retains four-point OBB vertices in source pixels, converts each to a float32 OpenCV convex hull and uses $\text{intersectConvexConvex}$ area divided by $\max(\text{union area}, 1 \times 10^{-9})$; nonpositive polygon area or an intersection exception yields zero. Clean matching uses the same label-order, class constraint, unused-prediction rule, ≥ 0.50 threshold and larger-index IoU tie rule as above, replacing only the overlap. In the independently cached diagnostic pass, predictions generated at confidence 0.05 are re-thresholded at 0.25. The axis matcher gives 1247 targets and the pure polygon matcher gives 1226; the error-mode loop uses their intersection, 1216, not all 1226 rotated matches. A fresh per-image audit reproduces all three counts. On these 1216 targets, the failure classifier examines attacked predictions independently per target, so predictions can be reused across diagnostic labels. Maximum polygon overlap is checked first regardless of class; overlap ≥ 0.50 means preserved or class flip. Otherwise, it checks same-class envelope overlap, target aspect ratio and long-edge angle for angle/localization errors,

and then nearby other-class detections or complete miss. Classifier score ties retain the first prediction in list order, unlike the clean matcher tuple tie rule; these labels are target diagnostics, not another one-to-one TP count.

Failure-mode thresholds and representation details. After the ≥ 0.50 polygon-overlap test, angle error requires target aspect ratio ≥ 1.5 , same-class envelope IoU ≥ 0.50 and long-edge angle error $\geq 15^\circ$ modulo 180° . Otherwise, same-class envelope IoU ≥ 0.10 or center distance $\leq$ one target envelope diagonal gives localization error; absence of that proximity gives complete miss. If there is no same-class prediction, any-class IoU ≥ 0.10 or the same center proximity gives other-class confusion. The native OBB validator instead converts annotation quadrilaterals to xywhr through OpenCV minAreaRect and compares boxes in its common resized/letterboxed pixel frame. It is not the OpenCV convex-polygon intersection used by these diagnostics.

Native validation estimator. OBB validation represents each box as (cx, cy, width, height, angle) and uses batch_probiou, not polygon-intersection IoU. It models each rect-angle as a Gaussian with rotated covariance $\text{diag}(\text{width}^2/12, \text{height}^2/12)$, uses the Bhattacharyya distance clipped to $[1 \times 10^{-7}, 100]$ and returns $1 - \sqrt{1 - \exp(-\text{distance})} + 1 \times 10^{-7}$. Horizontal DIOR validation uses ordinary xxyy IoU. Classes must agree. For each overlap threshold 0.50:0.05:0.95, the default matcher enumerates eligible label-prediction pairs, sorts by decreasing overlap using NumPy argsort, and then retains the first pair per prediction index and the first per label index via np.unique; no Hungarian assignment is used. Equal-overlap ordering follows that versioned NumPy routine rather than the main matcher's larger-index rule. For AP, detections are sorted by decreasing confidence with np.argsort(-conf); no secondary confidence-tie key is imposed. New replay statistics are concatenated in locked lexical image order, and every bootstrap uses the same ordering rule. Archived Table 1 retains its original validator ordering.

For native Precision and Recall, each represented ground-truth class has Precision/Recall curves at the first overlap threshold (0.50), interpolated on 1000 common confidence points from 0 to 1. The common operating point is the first argmax of $\text{smooth}(\text{mean_class}(\text{F1_curve}), 0.1)$; per-class Precision and Recall at that point are then averaged equally over classes present in the labels, including zero-detection classes. This is macro averaging, not pooled target matching and not confidence 0.25. The archived baseline configurations use $\text{imgsz} = 640$, $\text{max_det} = 300$, $\text{confidence} = \text{None}$ (validator candidate floor 0.01 for OBB DOTA1.5/FAIR1M and 0.001 for horizontal DIOR in the checked implementation) and NMS overlap threshold 0.70. Table 1's exact maximizing confidence values were not separately archived; the selection algorithm is specified rather than inventing those values. The new AP replays explicitly use confidence floor 0.001 and NMS 0.50, with the native Ultralytics 101-point interpolated Precision-envelope integration at each overlap threshold. Native AP, native max-F1 Recall, envelope Recall@0.25 and polygon diagnostic labels are therefore four distinct estimators.

Additional controlled attack protocol. The fixed-area non-semantic control cyclically translates each 640×640 SAM mask among 64 deterministic offsets and selects the offset with minimum overlap with the original mask. Translation preserves the exact number of editable pixels and the mask topology modulo boundary wrap. Canonical PGD draws the initial perturbation uniformly from $[-\epsilon, +\epsilon]$ only inside the selected support, clips it to the valid image range, and then applies the same projected sign updates as PGD-ZS. Budget sensitivity uses ϵ in $\{2, 4, 8, 16\}/255$ with step size $\epsilon/4$ and 10 iterations. Iteration sensitivity uses $\{1, 5, 10, 20\}$ steps at $\epsilon = 8/255$ and step size $2/255$. A second fixed-area control draws one of the same 64 offsets uniformly per image without ranking overlap. Three prespecified draws use separate seeds while keeping the zero start, $8/255$ budget, $2/255$ step, 10 iterations and the same 100-image set. A further target-aligned geometric control uses the archived

clean-prompt Polygon support from the same 30-epoch detector and 100 images. Its signed-distance ranking contracts or expands the Polygon region to exactly the SAM pixel count in each 640×640 image; this uses SAM area but not its contour for placement. A separately regenerated paired replay applies the same zero-start PGD-ZS budget, step size, iterations, low-threshold prediction cache, and clean-matching rule to both this geometry and SAM. Deterministic PyTorch settings are enabled for the paired replay. The corrected paired replay records one execution; the exact repeat check retained in the historic archive is not a second execution of this corrected experiment.

Diagnostic protocol. Confidence curves cache post-NMS predictions at 0.05 and re-threshold them at 0.05, 0.10, 0.25 and 0.50. Their 0.25 clean denominator is 1247; rotated-IoU error modes instead use the 1216-target intersection of the 1247 envelope matches and 1226 polygon matches specified above. Aspect-ratio bins are <2 , $2-4$ and ≥ 4 ; orientation is folded to $0-90^\circ$ with boundaries 30° and 60° ; and crowding counts other GT centers within twice the target diagonal and uses 0, 1–2 and ≥ 3 neighbors. Source-resolution attack/defense results, 640-model-space energy controls and 384-restorer-space fidelity must not be pooled. The direct tensor normalization used by the paired primary attack replay can differ by one float32 ULP from the historic adapter; both controlled objective branches use the same new conversion. The extended replay retains the earlier adapter and prediction-cache path and is independently labeled.

3.3.1. Exact Image Paths and Correction Disabled Control

Ordinary attack path: source-resolution lossless BGR uint8 $\rightarrow$ RGB $\rightarrow$ OpenCV INTER_LINEAR stretch to $640 \times 640 \rightarrow$ float32/255 $\rightarrow$ masked attack $\rightarrow$ NumPy rounding to uint8 $\rightarrow$ INTER_LINEAR resize to the source dimensions. Restoration path: source attacked BGR uint8 $\rightarrow$ INTER_LINEAR stretch to $384 \times 384 \rightarrow$ RGB/255 using CPU float32 normalization $\rightarrow$ four-channel restorer $\rightarrow$ NumPy rounding to uint8 $\rightarrow$ BGR $\rightarrow$ INTER_LINEAR resize to source dimensions. No padding is applied in these two explicit stretches. Native detector prediction adds its own LetterBox preprocessing. Training crop resizing uses INTER_AREA, while restoration evaluation uses INTER_LINEAR. Source sizes include 600×600 , 600×800 , 800×600 , 800×800 and 1000×1000 ; thus, the two metric spaces and source images are not interchangeable. The 8/255 constraint applies to the 640-space attack tensor relative to its resized clean tensor. Source export includes resampling, so differences from the original source clean image are not guaranteed to satisfy that same bound. Source-referenced ASR measures the complete attack/export path. Matched-path ASR instead uses the perturbation-free clean image passed through the identical 640 attack/export preprocessing, with detections still evaluated at source dimensions. The two references define different matched target sets.

At source resolution, the effective mask thresholds OpenCV grayscale of absolute clean-current-attack differences at integer value 2 and intersects the fixed saved SAM support. It is nearest-neighbor resized for restoration. This mask can include interpolation-induced differences, including before the first adaptive update. We execute the complete restorer forward with $g = 0$ and verify, for every image/model/test combination, exact byte equality to the attacked-image 384-down/source-up path. This control retains normalization, network invocation, quantization, resizing and detector evaluation but removes learned correction. The same path is compared at $g > 0$; the historic discrepancy is traced using hashed source files and their paired resized clean references. The actual source/model filenames and retained full-hash manifests link both historical and corrected file groups.

3.3.2. Explicit Adaptive Forward and Backward Paths

The adaptive replay starts at the clean 640 tensor with zero perturbation; uses $\varepsilon = 8/255$, $\alpha = 2/255$ and ten signed descent steps; and has no random restart or label-based candidate

selection. The attack support remains the saved SAM contour. At each iteration, exact uint8 quantization and OpenCV source resizing produce the attacked image; the effective clean-reference repair mask is recomputed on that current image, nearest-resized to 384 and detached. The real restorer and final confidence sums use the same forward path as evaluation. Detector input uses native LetterBox(auto = True, stride = 32), including rectangular inputs. Following the BPDA approach [42,43], quantization uses identity straight-through derivatives; OpenCV resizing and LetterBox use torch bilinear (align_corners = False) derivatives with the forward values overwritten by exact uint8 results. The hard threshold mask has no derivative. We differentiate through the continuous restorer with the current detached mask; no differentiable input-only mask estimator is assumed.

Let J_r and J_y be the corrected mean class responses of the raw and restored branches, and let γ be the actual confidence gate in Equation (14). The BPDA response has the exact selected-branch value in the forward path and an equal-branch derivative in the backward path. NMS and hard confidence selection are executed in the forward path but are not differentiated; their backward path is replaced by this explicit response approximation. The gate uses no labels. After the tenth update, the mask, restoration and actual gate are recomputed once more to obtain the evaluated pipeline image. All final restoration outputs are checked byte-for-byte against the standalone evaluation routine. Stepwise masks, gate decisions and gradient norms are retained. This approximation is empirical and does not guarantee the strongest adaptive attack.

$$J_{BPDA} = \text{stopgrad}((1 - \gamma)J_r + \gamma J_y) + (J_r + J_y)/2 - \text{stopgrad}((J_r + J_y)/2). \quad (15)$$

Pseudocode: initialize $z \leftarrow \text{clean640}$; repeat ten times: form exact source image $A(z)$; recompute detached $M_\delta(\text{clean}, A(z))$; form exact restored image $B(A(z), M_\delta, g)$; compute actual γ from post-NMS confidence sums; construct the BPDA response above; obtain its gradient with respect to z ; update z by $-\alpha \text{sign}(\text{gradient})$ inside fixed SAM support; project around clean640 to $[-\varepsilon, +\varepsilon]$; and clip to $[0, 1]$. Finally, repeat A , M_δ , B and γ exactly and evaluate the selected source image. The first term is detached only for differentiation; its numeric value remains the actual selected response. Both response branches use $\text{cM} = 1$ in this adaptive approximation.

3.3.3. Same-Partition Training-Seed Replication

To separate training randomness from the scene-composition variation between folds, we repeat the complete five-fold training twice with the identical train/held-out image lists and corrected SAM/PGD-ZS pairs [44]. The two new per-fold seed vectors are the original vector plus 100,000 and plus 200,000. We verify exact partition equality and absence of train/held-out overlap for all ten additional models. Training gain remains 0.15; both gains are evaluated with and without confidence fallback on each held-out fold. The three full out-of-fold repetitions are summarized separately from the primary paired image-bootstrap intervals. Neither attacks nor detector weights are retrained or resampled.

4. Results and Discussion

4.1. Multi-Attack Comparison

Table 2 reports source-referenced complete-path outcomes from the same primary attack records used in the support comparison. FAIR1M, shown in Table 3, changes the clean reference to the identical attack/export path; DOTA1.5 and DIOR retain their dataset-specific source references. Scale groups use axis-aligned object-envelope area divided by image area: small <1%, medium 1–5% and large $\geq 5\%$. Each scale cell gives lost/reference-matched targets; reference sets are shared across supports within a protocol.

Table 3. PGD-ZS support and scale comparison using envelope-IoU matching at confidence 0.25. FAIR1M ASR uses the identical attack/export clean reference (1238 targets); DOTA1.5 and DIOR use source-clean references. a. Corrected FAIR1M PGD-ZS support area and realized model-space perturbation energy (mean $\pm$ sample SD; $n = 100$). b. Exact per-image L1 normalization of corrected FAIR1M residuals in 640-space ($n = 100$). P/R and FP use envelope-IoU matching at confidence 0.25; AP uses native Ultralytics ProbIoU-based OBB evaluation. c. Fixed-area support and PGD-initialization control on 100 FAIR1M images. ASR is source-referenced, with envelope-IoU matching at confidence 0.25. d. Separately regenerated deterministic paired SAM-versus-geometric control at exactly equal support area on the same 100 FAIR1M images. ASR is source-referenced, with envelope-IoU matching at confidence 0.25. e. Controlled archived-versus-corrected response replay on the same FAIR1M images, masks, preprocessing and detector. ASR is source-referenced, with envelope-IoU matching at confidence 0.25. f. Post hoc target-identity audit on the same 100 FAIR1M images. Source reference: 1247 targets; matched 640 attack/export reference: 1238 targets. Envelope-IoU ≥ 0.50 at confidence 0.25.

Dataset	Support	Adv Recall	ASR	Small	Medium	Large
DOTA1.5	Polygon	17.15%	19.87% 1840/9261	20.31% 1763/8681	10.75% 56/521	35.59% 21/59
DIOR	Box	9.34%	86.11% 279/324	86.17% 162/188	81.93% 68/83	92.45% 49/53
FAIR1M	Global	21.97%	61.47% 761/1238	61.73% 708/1147	57.65% 49/85	66.67% 4/6
FAIR1M	Box	37.95%	32.55% 403/1238	32.26% 370/1147	35.29% 30/85	50.00% 3/6
FAIR1M	Polygon	41.12%	26.66% 330/1238	26.68% 306/1147	24.71% 21/85	50.00% 3/6
FAIR1M	SAM	42.37%	24.39% 302/1238	24.76% 284/1147	16.47% 14/85	66.67% 4/6

(a)

Support	Area Ratio	MAE	RMSE	Changed Ratio	N
Global	1.0000 $\pm$ 0.0000	0.016018 $\pm$ 0.000345	0.019528 $\pm$ 0.000307	0.9764 $\pm$ 0.0052	100
Box	0.0526 $\pm$ 0.0612	0.000951 $\pm$ 0.001048	0.004386 $\pm$ 0.002223	0.0516 $\pm$ 0.0599	100
Polygon	0.0315 $\pm$ 0.0359	0.000597 $\pm$ 0.000643	0.003515 $\pm$ 0.001799	0.0309 $\pm$ 0.0351	100
SAM	0.0238 $\pm$ 0.0382	0.000473 $\pm$ 0.000686	0.003039 $\pm$ 0.001834	0.0235 $\pm$ 0.0374	100

Note: All four rows use identical image indices, detector weights, loss, scaling path, $\epsilon = 8/255$, 10 PGD-ZS steps, seed 42, and lossless model-space PNGs. MAE and RMSE are computed on normalized RGB values in [0,1]. The table reports realized perturbation magnitude; it does not equalize energy across supports.

(b)

Branch	N	Mean Scale	P@0.25	R@0.25	mAP50	mAP50-95
Clean	100	—	54.99%	52.77%	42.05%	31.57%
Global	100	0.0279	57.53%	49.83%	41.80%	31.24%
Box	100	0.4760	54.43%	17.03%	30.47%	21.58%
Polygon	100	0.7523	52.48%	16.03%	23.01%	15.92%
SAM	100	0.9852	51.79%	14.43%	22.62%	15.70%

Note: Each image's target L1 is the minimum realized integer L1 of its four corrected residuals. Sign-preserving scale-down with largest-remainder integer allocation gives exact equality for all branches; no attack is re-optimized under an L1 constraint. The mean Global scale is 0.0279, and SAM scale 0.9852. All detections use these 640-space normalized images, whose clean Recall is 1218/2308 = 52.77%, not the source-space Table 4 baseline.

(c)

Support and Initialization	Clean Matched	Successful	ASR	Adv Recall
Minimum-overlap translation, zero start	1247	137	10.99%	52.25%
Random offset, zero start (draw 1)	1247	151	12.11%	51.52%
Random offset, zero start (draw 2)	1247	154	12.35%	51.65%
Random offset, zero start (draw 3)	1247	134	10.75%	52.21%
SAM, zero start	1247	354	28.39%	42.37%
SAM, random start	1247	351	28.15%	42.37%

Note: $\epsilon = 8/255$, step size = 2/255, 10 iterations, confidence threshold 0.25, and envelope-IoU threshold 0.50; source-clean target reference. All rows use the auxiliary low-threshold prediction cache described in Section 3.3. The shifted control retains exactly the same mask pixels as the semantic support on every image. The random-offset control reports three independent offset draws per image; the 100 clean-matched sets are shared rather than treated as 300 independent images. Paired 95% ASR-difference intervals are computed by 10,000 image-cluster bootstrap resamples, recomputing successes divided by the pooled clean-matched denominator for each branch; the three random offsets remain repeated outcomes on the same 100 images rather than 300 independent samples.

Table 3. Cont.

(d)					
Support and Initialization		Clean Matched	Successful	ASR	Adv Recall
Corrected SAM, zero start		1247	354	28.39%	42.37%
Corrected target-aligned geometric contour		1247	365	29.27%	41.98%
Note: Signed-distance contraction/expansion of clean-prompt Polygon gives the exact SAM pixel count per image; 19 masks require expansion. The geometric placement does not use the SAM contour. Both rows use corrected PGD-ZS, the same deterministic prediction-cache path and 1247 source-clean envelope-IoU matches. SAM has 354 successes and geometry 365, a difference of -0.88 percentage points (95% CI, -2.42 to 0.42 ; paired image-level bootstrap with 10,000 resamples of the same 100 paired images). These corrected rows come from a single execution with deterministic settings.					

(e)				
Support	Clean Matched	Archived-Response ASR	Corrected ASR	Corrected — Archived Points (95% CI)
Global	1247	41.94%	61.59%	19.65 (14.12 to 26.37)
Box	1247	24.06%	35.93%	11.87 (7.02 to 18.51)
Polygon	1247	19.97%	30.71%	10.75 (5.93 to 16.98)
SAM	1247	18.60%	28.23%	9.62 (5.32 to 15.06)

(f)						
Path	Source ASR	Matched-Path ASR	Retained Source	New vs. Source	Retained Path	New vs. Path
Resize only	9.06% (113/1247)	0.00% (0/1238)	1134	104	1238	0
Global	61.59% (768/1247)	61.47% (761/1238)	479	28	477	30
Box	35.93% (448/1247)	32.55% (403/1238)	799	77	835	41
Polygon	30.71% (383/1247)	26.66% (330/1238)	864	85	908	41
SAM	28.23% (352/1247)	24.39% (302/1238)	895	83	936	42
Note: Retained counts are intersections with the named clean target set; new counts are matches outside that set. Retained plus new equals total TP for either reference. The resize-only matched-path value is zero by construction. Saved image replay reproduces every primary per-image source/resize/SAM TP, FP and FN, and all four source-referenced support success counts. These target-identity analyses are post hoc; no attacks were regenerated. Source-referenced scale counts remain in the accompanying records.						

Source-referenced PGD-ZS ASR is 19.87% for DOTA1.5 Polygon, 86.11% for DIOR Box and 28.23% for FAIR1M SAM. DIOR loses 279/324 clean-matched targets under PGD-ZS and 287/324 under MIFGSM (88.58%). These dataset-specific detectors, supports and schedules do not rank intrinsic dataset vulnerability. The controlled archived-response FAIR1M SAM ASR is 18.60%, versus 28.23% for the corrected response, showing a material objective effect under a shared source reference. Figure 5 displays these complete-path outcomes.

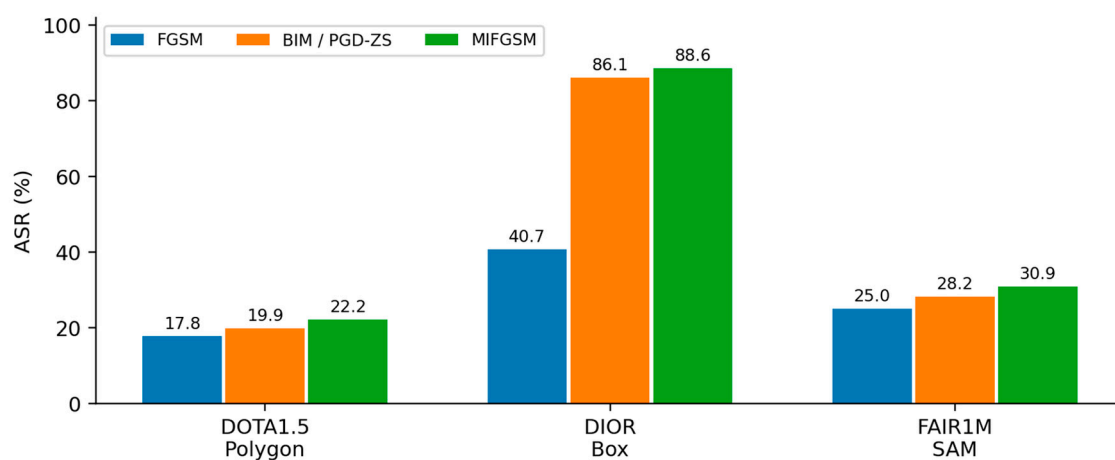


Figure 5. Source-referenced complete-path ASR under the dataset-specific supports and schedules of Table 2; envelope-IoU matching at confidence 0.25. Cross-dataset bars are descriptive.

Table 4. Corrected PGD-ZS defense evaluation with out-of-fold models and a full-path correction-disabled control ($n = 100$). P/R and FP use envelope-IoU matching at confidence 0.25; AP uses native Ultralytics ProbIoU-based OBB evaluation. a. Clean-input transforms, restoration paths and false-positive counts on FAIR1M ($n = 100$). P/R and FP use envelope-IoU matching at confidence 0.25; AP uses native Ultralytics ProbIoU-based OBB evaluation. b. Exact-forward BPDA under fixed SAM support with current repair-mask recomputation and the real confidence gate ($n = 100$ per row). P/R and FP use envelope-IoU matching at confidence 0.25; AP uses native Ultralytics ProbIoU-based OBB evaluation. b (continued). Measured adaptive replay cost on RTX 5060 Ti; 100 images per response/gain combination. c. Corrected-attack confidence-margin sensitivity with newly trained $g = 0.5$ out-of-fold predictions ($n = 100$). P/R and FP use envelope-IoU matching at confidence 0.25; AP uses native Ultralytics ProbIoU-based OBB evaluation. d. Ungated held-out training-response $\times$ test-response $\times$ gain control; the corresponding fallback records are also retained. P/R and FP use envelope-IoU matching at confidence 0.25; AP uses native Ultralytics ProbIoU-based OBB evaluation. e. Physical-frequency selector control on the corrected ordinary SAM attack, compared with the asymmetric-selector rows of Table 4 ($n = 100$; out-of-fold). P/R and FP use envelope-IoU matching at confidence 0.25; AP uses native Ultralytics ProbIoU-based OBB evaluation. f. Paired effects on task metrics and false positives: estimate (95% interval). P/R and FP use envelope-IoU matching at confidence 0.25; AP uses native Ultralytics ProbIoU-based OBB evaluation. g. Same-partition training-seed replication: mean $\pm$ sample SD across three complete out-of-fold repetitions. P/R and FP use envelope-IoU matching at confidence 0.25; AP uses native Ultralytics ProbIoU-based OBB evaluation. h. Paired gate controls on the same 100 FAIR1M images: estimate (95% interval). P/R and FP use envelope-IoU matching at confidence 0.25; AP uses native Ultralytics ProbIoU-based OBB evaluation.

Condition	N	P@0.25	R@0.25 (95% CI)	mAP50	mAP50-95	FP@0.25	Fold R Mean $\pm$ SD
Clean	100	56.79%	54.03% (45.87–63.63)	43.85%	34.61%	949	52.49 $\pm$ 9.83%
Corrected PGD-ZS/none	100	60.86%	42.37% (32.50–53.95)	36.22%	28.31%	629	41.35 $\pm$ 11.69%
Jpeg	100	58.88%	44.37% (34.75–55.88)	38.72%	29.76%	715	43.08 $\pm$ 11.41%
Gaussian	100	58.62%	36.70% (26.22–49.54)	36.57%	28.16%	598	35.51 $\pm$ 10.56%
Median	100	58.37%	37.78% (27.60–50.24)	37.60%	29.16%	622	36.71 $\pm$ 10.77%
Bilateral	100	58.81%	43.98% (34.44–55.34)	38.97%	30.32%	711	42.71 $\pm$ 10.66%
Prototype	100	58.32%	20.80% (12.92–30.87)	27.33%	19.44%	343	19.91 $\pm$ 6.58%
Correction disabled $g = 0$ (full path)	100	59.67%	32.89% (21.34–47.01)	35.15%	26.62%	513	31.37 $\pm$ 11.36%
Correction disabled + fallback $g = 0$	100	59.37%	43.80% (34.46–54.80)	37.18%	28.35%	692	42.70 $\pm$ 11.65%
Learned $g = 0.15$	100	60.17%	36.40% (24.72–50.00)	36.87%	28.06%	556	35.16 $\pm$ 12.85%
Learned + fallback $g = 0.15$	100	59.36%	44.24% (34.91–55.20)	37.82%	28.78%	699	43.08 $\pm$ 11.84%
Learned $g = 0.5$	100	58.52%	39.12% (27.08–52.68)	36.21%	27.82%	640	37.57 $\pm$ 14.03%
Learned + fallback $g = 0.5$	100	58.43%	45.06% (35.25–56.51)	38.43%	29.56%	740	43.74 $\pm$ 12.27%

Note: Fixed-threshold P/R/FP use source-resolution envelope-IoU matching at confidence 0.25 and IoU 0.50. AP uses native Ultralytics ProbIoU-based OBB per-image statistics with the full curve at confidence 0.001; all primary stats are locked to lexical image order, and native-validator values are also retained. Recall intervals use paired image resampling, 10,000 draws, seed 20261012. Fold SD combines scene composition and initialization, not pure same-split seed variance. The $g = 0$ output is byte-identical to the full 384-down/source-up resize control. Gains 0.15 and 0.5 otherwise share attacks, masks and held-out newly trained models. The $g = 0$ fallback row uses the same actual confidence gate as the learned rows; its fold mean/SD follows the original partition. It is a preprocessing-and-selection control, not learned correction. Additional paired contrasts are reported in Table 4h.

(a)

Transform	P@0.25	R@0.25	mAP50	mAP50-95	FP@0.25	ΔR (Points)	Preprocess (ms/image)
Clean	56.79%	54.03%	43.85%	34.61%	949	0.00	0.49
Clean 640 down/source up	56.35%	53.64%	44.68%	34.98%	959	−0.39	—
Clean 384 down/source up	57.79%	42.59%	43.48%	33.85%	718	−11.44	—
Clean learned + fallback ($g = 0.15/0.5$)	56.37%	54.25%	43.99%	34.79%	969	+0.22	—
JPEG	58.24%	51.00%	40.96%	31.92%	844	−3.03	3.20
Gaussian	55.80%	51.52%	44.31%	34.91%	942	−2.51	0.90
Median	55.17%	51.13%	44.32%	34.50%	959	−2.90	0.66
Bilateral	55.94%	53.68%	43.23%	34.26%	976	−0.35	4.84

Note: The same source-resolution clean_raw.png images, labels, detector, envelope-IoU matching at confidence 0.25 and IoU 0.50, and native Ultralytics ProbIoU-based OBB AP are used. Delta R is in percentage points versus clean. Preprocessing time excludes detector inference; the clean-row value is array-copy overhead. The two resize-only paths contain no attack or learned correction and use INTER_LINEAR in both resize directions. Their preprocessing times were not measured (—), rather than zero. The clean learned-plus-fallback row is verified for both gains under an identically zero oracle mask. Its preprocessing cost was not profiled. It selects the 384-path image for 33 scenes and the source image for 67; the resulting false-positive count is 969 versus 949 for clean source inputs. Clean Prototype repair returns the source input exactly when this same mask is empty. FP is the total false-positive count across the same 100 images under the fixed envelope-based matcher at confidence 0.25 and IoU 0.50. The clean 384-path row also represents the executed ungated learned outputs at both gains under the verified zero repair mask; the forward-equivalence check covers every held-out image at each gain. It does not represent source-image identity or an input-only localization test. The clean 640 attack/export row has source-referenced target-conditioned ASR 113/1247 = 9.06%; Table 3f gives retained and newly matched targets.

Table 4. Cont.

(b)						
Adaptive Branch	N	P@0.25	R@0.25	FP@0.25	mAP50	mAP50-95
Corrected response; g = 0.15; attack	100	56.32%	21.23%	380	26.33%	20.33%
Corrected response; g = 0.15; pipeline	100	57.14%	23.40%	405	26.86%	20.88%
Corrected response; g = 0.5; attack	100	57.86%	24.39%	410	29.36%	23.14%
Corrected response; g = 0.5; pipeline	100	58.33%	26.86%	443	29.91%	23.18%
Archived response; g = 0.15; attack	100	54.63%	40.64%	779	34.04%	25.06%
Archived response; g = 0.15; pipeline	100	53.79%	41.16%	816	32.93%	24.19%
Archived response; g = 0.5; attack	100	55.07%	41.90%	789	35.60%	26.65%
Archived response; g = 0.5; pipeline	100	53.72%	42.85%	852	34.10%	25.26%
All rows use the same five corrected-attack restorers, fixed image/support lists and zero-start ten-step budget. The corrected and archived controls differ only in the decoded response and both use exact-forward BPDA with current repair-mask updates. At gains 0.15/0.5, corrected pipeline Recall is 23.40%/26.86%, versus 41.16%/42.85% for the archived response; mAP50 is 26.86%/29.91%, versus 32.93%/34.10%. Section 3.3.2 specifies the backward approximations. The different optimization targets preclude a universal cross-table attack-strength ordering.						
(b) continued						
Response/Gain	N	Mean s/image	Median s/image	P95 s/image	Max Allocated MiB	Max Reserved MiB
Archived/0.15	100	1.942	1.700	2.459	826.85	850.00
Archived/0.5	100	1.937	1.704	2.461	826.85	850.00
Corrected/0.15	100	1.922	1.695	2.443	826.04	850.00
Corrected/0.5	100	1.927	1.693	2.451	826.04	850.00
Timing starts after source image/mask loading and model loading, with CUDA synchronized and peak memory counters reset. It covers ten gradient updates, repeated exact branch/mask/gate evaluations, final standalone-forward equivalence checking, output saving and final synchronization; the retained implementation also repeats its fixed-list bookkeeping within this window. It excludes subsequent aggregate validator runs and bootstrap calculation. Memory is PyTorch’s peak allocated/reserved CUDA memory, including the resident models, not total device or process memory. Per-image peaks are maximized within each row; reserved memory is allocator-dependent. These are measured offline replay costs, not defense-only inference latency or service throughput.						
(c)						
Margin	Restore/Fallback		P@0.25	R@0.25	F1@0.25	
−0.10	71/29		58.29%	44.93%	50.75%	
−0.05	66/34		58.39%	45.06%	50.87%	
0.00	59/41		58.43%	45.06%	50.88%	
0.05	55/45		58.43%	45.06%	50.88%	
0.10	51/49		58.48%	45.10%	50.93%	
Note: The restored image is used when restored confidence sum minus attacked confidence sum is at least the stated margin. Counts and metrics use confidence 0.25 and IoU 0.50.						
(d)						
Training Response	Test Response	Gain	R@0.25	mAP50	mAP50-95	FP@0.25
Archived	Archived	0.15	39.64%	38.17%	29.63%	649
Archived	Archived	0.5	40.73%	37.22%	28.83%	718
Archived	Corrected	0.15	36.48%	36.58%	28.07%	561
Archived	Corrected	0.5	39.47%	35.85%	27.49%	654
Corrected	Archived	0.15	39.51%	38.42%	29.76%	657
Corrected	Archived	0.5	40.90%	38.30%	29.32%	714
Corrected	Corrected	0.15	36.40%	36.87%	28.06%	556
Corrected	Corrected	0.5	39.12%	36.21%	27.82%	640
(e)						
Selector/Gain	Branch	R@0.25	mAP50	mAP50-95	FP@0.25	ΔmAP50 Points (95% CI)
Physical/0.15	Restored	36.35%	36.89%	28.06%	555	0.02 (−0.16 to 0.30)
Physical/0.15	Fallback	44.24%	37.77%	28.74%	698	−0.05 (−0.25 to 0.21)
Physical/0.5	Restored	39.12%	36.31%	27.97%	650	0.11 (−0.18 to 0.38)
Physical/0.5	Fallback	44.97%	38.44%	29.55%	746	0.01 (−0.51 to 0.32)
Physical-minus-asymmetric intervals use the paired bootstrap protocol in Section 3.3. Both selectors are trained at gain 0.15; the gain-0.5 rows evaluate a held-out workpoint with the same models. This control changes the frequency selector while retaining the FFT loss, so it cannot identify the spectral term’s causal effect on false positives.						

Table 4. Cont.

(f)					
Contrast	ΔP (points)	ΔR (points)	$\Delta mAP50$ (points)	$\Delta mAP50-95$ (points)	ΔFP (count)
Clean 640 path – clean	−0.44 (−2.11, 1.32)	−0.39 (−1.97, 1.07)	+0.83 (−1.99, 2.84)	+0.37 (−1.87, 2.07)	+10 (−41, 66)
Clean 384 path – clean	+1.00 (−1.59, 3.38)	−11.44 (−17.26, −6.06)	−0.38 (−4.74, 1.80)	−0.76 (−4.63, 1.17)	−231 (−370, −120)
g = 0 path – attack	−1.19 (−5.35, 2.28)	−9.49 (−14.38, −4.31)	−1.08 (−3.97, 1.04)	−1.69 (−3.96, 0.03)	−116 (−249, 0)
Bilateral – attack	−2.05 (−3.23, −0.93)	+1.60 (0.47, 2.84)	+2.75 (0.78, 4.82)	+2.01 (0.48, 3.64)	+82 (53, 115)
g = 0.15 restored – g = 0	+0.50 (−1.32, 3.07)	+3.51 (1.80, 5.52)	+1.73 (0.25, 3.10)	+1.44 (0.30, 2.59)	+43 (8, 82)
g = 0.5 restored – g = 0	−1.15 (−3.35, 1.89)	+6.24 (3.49, 9.33)	+1.06 (−1.74, 3.69)	+1.20 (−0.81, 3.19)	+127 (62, 199)
g = 0.15 restored – attack	−0.69 (−3.60, 2.02)	−5.98 (−10.24, −1.03)	+0.65 (−2.33, 2.66)	−0.25 (−2.52, 1.32)	−73 (−188, 26)
g = 0.5 restored – attack	−2.34 (−5.75, 0.74)	−3.25 (−7.76, 1.55)	−0.02 (−3.39, 2.46)	−0.49 (−2.92, 1.42)	+11 (−101, 113)
g = 0.15 fallback – attack	−1.50 (−2.90, 0.12)	+1.86 (0.38, 3.99)	+1.60 (−1.32, 3.44)	+0.47 (−1.63, 1.84)	+70 (34, 112)
g = 0.5 fallback – attack	−2.43 (−4.65, −0.35)	+2.69 (1.00, 5.10)	+2.21 (−0.87, 4.55)	+1.25 (−0.80, 2.88)	+111 (54, 179)
g = 0.15 new – old training	+0.16 (−0.48, 0.65)	−0.09 (−0.44, 0.27)	+0.30 (−0.43, 0.74)	−0.01 (−0.60, 0.45)	−5 (−16, 5)
g = 0.5 new – old training	+0.31 (−0.69, 1.35)	−0.35 (−0.86, 0.19)	+0.35 (−0.41, 1.12)	+0.33 (−0.33, 1.01)	−14 (−41, 10)
Adaptive g = 0.15 pipeline – own attack	+0.82 (−1.70, 3.04)	+2.17 (0.23, 4.80)	+0.53 (−2.10, 2.57)	+0.55 (−1.69, 2.37)	+25 (3, 47)
Adaptive g = 0.5 pipeline – own attack	+0.46 (−2.43, 2.78)	+2.47 (0.24, 5.38)	+0.55 (−1.28, 2.57)	+0.05 (−1.42, 1.73)	+33 (0, 68)
Note: All contrasts use paired image-level percentile bootstrap with 10,000 resamples, seed 20261012, with the full native Ultralytics ProBioU-based OBB AP curve recalculated for each draw. Standard contrasts use the primary corrected records; adaptive contrasts use each pipeline's own gain-specific raw attacked branch. New-minus-old training holds the corrected test attacks fixed and has no fallback. Intervals are unadjusted marginal 95% intervals conditional on saved primary models and attacks, not simultaneous or training-seed uncertainty intervals. Precision/Recall use confidence 0.25; ΔFP is a difference in total count on the same 100 images. Rounded values are computed from unrounded records.					
(g)					
Gain/Branch	P@0.25 (%)	R@0.25 (%)	mAP50 (%)	mAP50-95 (%)	FP@0.25
g = 0.15/restored	60.39 ± 0.20	36.51 ± 0.11	36.80 ± 0.06	27.89 ± 0.15	552.67 ± 3.06
g = 0.15/fallback	59.47 ± 0.14	44.40 ± 0.15	37.67 ± 0.16	28.63 ± 0.18	698.33 ± 2.08
g = 0.5/restored	57.75 ± 0.72	38.63 ± 0.52	35.95 ± 0.23	27.73 ± 0.08	652.33 ± 16.44
g = 0.5/fallback	57.83 ± 0.56	44.83 ± 0.20	38.14 ± 0.53	29.42 ± 0.17	754.67 ± 14.50
Note: Every repetition uses the identical five train/held-out partitions and all 100 corrected SAM/PGD-ZS images. The original per-fold seeds are 42, 123, 2026, 31415 and 27182; repetition 1 adds 100000 and repetition 2 adds 200000 to each corresponding seed. Ten additional models are freshly trained, giving three seeds per fold and 15 models in total. A changed training seed affects initialization, the eight fixed augmented views and loader order; this is training-randomness sensitivity, not isolated initialization variance. All other hyperparameters and epoch-5 selection remain unchanged. Each reported metric is computed separately on a complete 100-image out-of-fold repetition before taking its mean/SD. The same images are reused, not treated as 300 independent images; SD across three repetitions is descriptive and is not a 95% confidence interval. Primary Table 4 and adaptive Table 4b are unchanged; no adaptive evaluation of the additional seed models is claimed.					
(h)					
Contrast	ΔP (points)	ΔR (points)	$\Delta mAP50$ (points)	$\Delta mAP50-95$ (points)	ΔFP (count)
g = 0 + gate – attacked	−1.49 (−2.86 to −0.23)	+1.43 (0.20 to 3.18)	+0.95 (−1.31 to 2.59)	+0.04 (−1.60 to 1.35)	+63 (29 to 101)
g = 0.15 + gate – g = 0 + gate	−0.01 (−1.05 to 1.22)	+0.43 (−0.26 to 1.20)	+0.64 (−0.92 to 1.80)	+0.43 (−0.86 to 1.35)	+7 (−16 to 33)
g = 0.5 + gate – g = 0 + gate	−0.94 (−2.88 to 0.86)	+1.26 (−0.06 to 2.55)	+1.25 (−1.22 to 3.61)	+1.22 (−0.61 to 2.92)	+48 (1 to 103)
Clean + gate – clean source	−0.41 (−1.08 to 0.28)	+0.22 (−0.39 to 0.90)	+0.14 (−1.17 to 1.25)	+0.18 (−0.98 to 1.16)	+20 (1 to 43)
Note: These four gate contrasts are post hoc and were not prespecified in the primary analysis. They apply the paired bootstrap procedure and metric definitions in Section 3.3 to the same 100 images in locked order. Clean contrasts condition on the verified zero oracle repair mask.					

4.2. Semantic-Support Ablation

With the identical perturbation-free attack/export path as a clean reference, FAIR1M Global, Box, Polygon and SAM lose 761, 403, 330 and 302 of 1238 matched targets, giving ASRs of 61.47%, 32.55%, 26.66% and 24.39%. Their source-referenced complete-path ASRs remain 61.59%, 35.93%, 30.71% and 28.23% (Table 3f). The resize-only path itself loses 113/1247 original matches (9.06%) but gains 104, leaving 1238 total matches; similar pooled Recall therefore does not imply preservation of target identities. For SAM, the source reference retains 895 matches and adds 83, whereas the matched-path reference retains 936 and adds 42; both give the same 978 attacked matches. ASR differences between references are not additive components of resampling and attack damage because the target sets differ. Table 3e separately controls the attack objective. Box is the axis-aligned rectangle, Polygon its oriented contour, and SAM the clean-prompted contour.

Support area is unchanged by the objective correction. We recompute realized MAE, RMSE and changed-pixel ratio from the new corrected lossless 640-space images; these same files serve both fidelity and annotation-referenced locality. Lower editable area remains associated with less whole-image distortion, not necessarily better preservation of object features.

The locked FAIR1M subset provides oriented-box annotations but no manual pixel-level instance masks. Their four-corner polygons permit an annotation-derived, SAM-independent reference region for the attack-residual Loc/Leak measurement. This oriented-box proxy is not an object-segmentation mask: mask IoU, Dice, and boundary F-score against manual pixel masks still cannot be reported, and locality cannot establish SAM contour accuracy. The 100-image post hoc exact-L1 control matches realized energy by rescaling saved perturbations, but it does not re-optimize an attack under an L1 constraint and cannot by itself remove support-area or semantic-geometry confounding. The demonstrated role of SAM remains automatic, instance-adaptive contour generation and reduced background modification at the observed support size.

Post hoc equal-energy Global remains close to the 640-space clean baseline, while localized residuals remain more damaging. This result describes normalized corrected residuals, not an optimized equal-L1 attack or evidence of SAM segmentation accuracy.

The near-clean Global result in Table 3b must be interpreted in light of amplitude attenuation. Before integer allocation, a saved residual δ is multiplied by $s = \frac{B}{\|\delta\|_1}$, where B is that image's common target L1 energy, giving $\delta' = s\delta$ and $\|\delta'\|_\infty = s\|\delta\|_\infty$. Thus, the continuous L-infinity amplitude changes linearly with the imposed scale, not exponentially. For a residual spread over K active scalar image entries, its mean absolute magnitude at fixed L1 is $\frac{B}{K}$; inverse-support attenuation of the maximum additionally requires a diffuse, approximately uniform amplitude pattern. The mean Global scale of 0.0279 shows that this post hoc protocol severely attenuates the corrected Global residual; the subsequent exact 8-bit allocation also changes which entries remain nonzero. L1 and L-infinity constraints are mathematically compatible, but equal total L1 across very different support sizes does not preserve the same local amplitude. This support-amplitude coupling limits cross-scale interpretation. The weak rescaled Global branch neither proves that an equal-L1 Global attack cannot affect detector features nor represents a Global attack re-optimized under the common L1 constraint.

Figure 6 uses the corrected lossless 640-space Global and SAM attack outputs for the same v_0 ship scene, with the saved SAM support and annotation OBB shown separately. This single illustrative scene does not estimate segmentation accuracy or average attack performance. Figure 7 compares old and corrected response replays and reports support area and corrected locality against ground-truth oriented-box proxies.

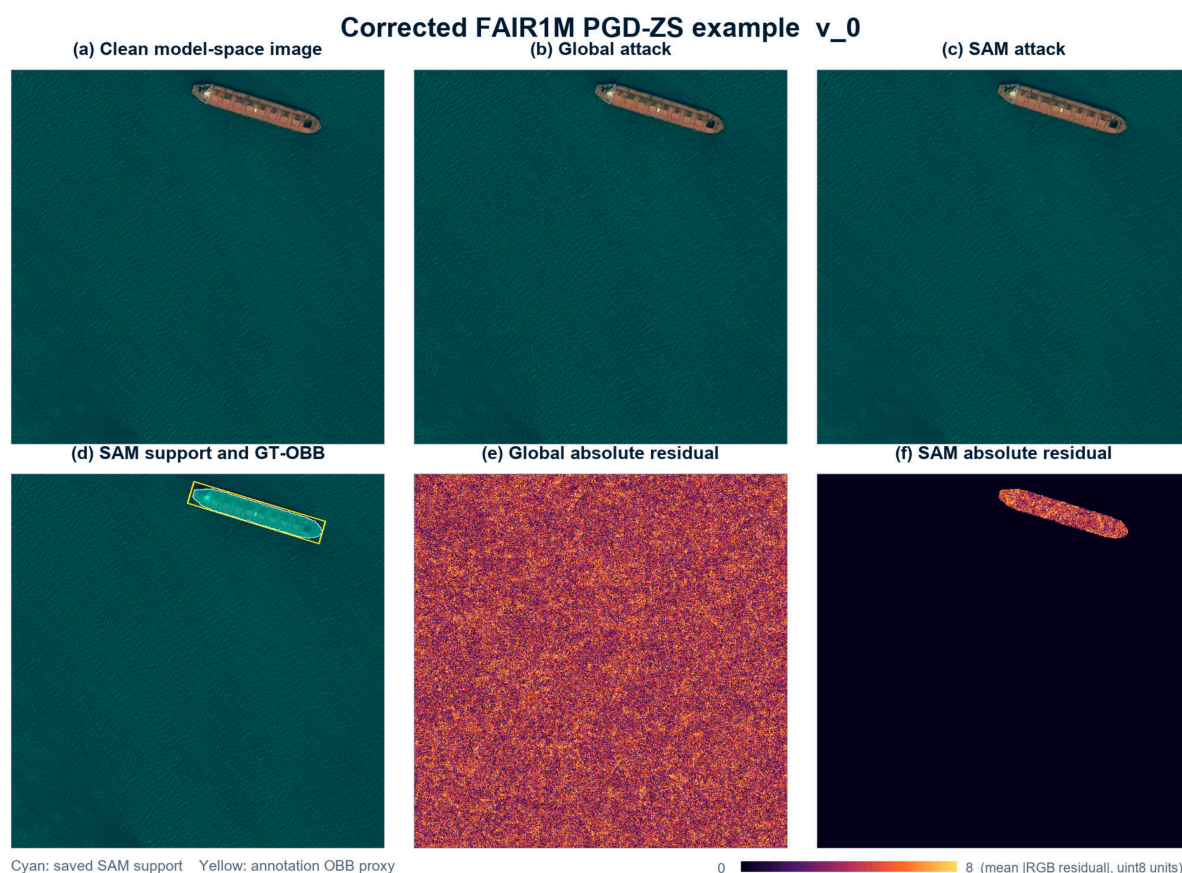


Figure 6. Corrected PGD-ZS example on FAIR1M image v_0: clean, Global and SAM attack images; saved SAM support (cyan, with a light boundary) and ground-truth oriented-box outline (yellow); and mean absolute RGB residuals. All panels use paired lossless 640×640 model-space images. Residual heatmaps share the same 0–8 scale in uint8 intensity units and are not independently autoscaled or computed from source-resized display files. The annotation box is an independent geometric proxy, not a manual pixel-level mask; this one scene does not establish SAM segmentation accuracy.

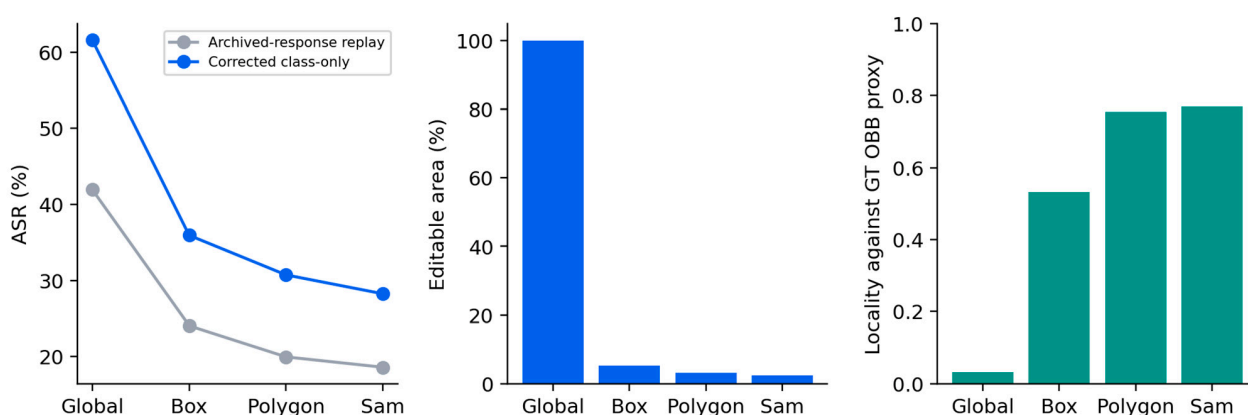


Figure 7. Source-referenced complete-path archived/corrected objective comparison, fixed support area and annotation-referenced locality on 100 FAIR1M images. Matched-path ASRs are reported separately in Tables 3 and 3f.

4.2.1. Fixed-Area Control and PGD Sensitivity

The equal-area and initialization controls use an independently generated auxiliary replay with predictions cached at confidence 0.05 and re-thresholded at 0.25. Its source-referenced target sets and images are kept separate from the primary matched-path support comparison.

At equal area and shape, source-referenced zero-start SAM ASR is 28.39% versus 10.99% for minimum-overlap translation. Random offsets average 11.73%; SAM minus random mean is 16.65 points (95% CI, 10.70 to 24.11), with the three offsets kept together within each resampled image. Random-start SAM gives 28.15%. This auxiliary replay has 354 zero-start successes versus the primary source-referenced 352 and uses a distinct prediction-cache path. Table 3d tests contour geometry at matched target placement.

The geometric control gives 29.27% ASR versus 28.39% for SAM. The paired interval contains zero, leaving a SAM-specific contour advantage unresolved. Mean mask IoU remains 0.6100 because support construction is unchanged; neither contour is checked against manual pixel segmentation.

Corrected random-start PGD ASRs at $\epsilon = \{2,4,8,16\}/255$ are 13.47%, 18.60%, 28.15% and 37.13%; at $\{1,5,10,20\}$ steps, they are 12.99%, 25.50%, 28.15% and 27.91%. Confidence thresholds $\{0.05,0.10,0.25,0.50\}$ give 16.77%, 20.25%, 28.15% and 34.74%. Clean-matched denominators change with threshold, so that curve describes operating-point sensitivity rather than one invariant target set. Figure 8a is regenerated from these corrected records.

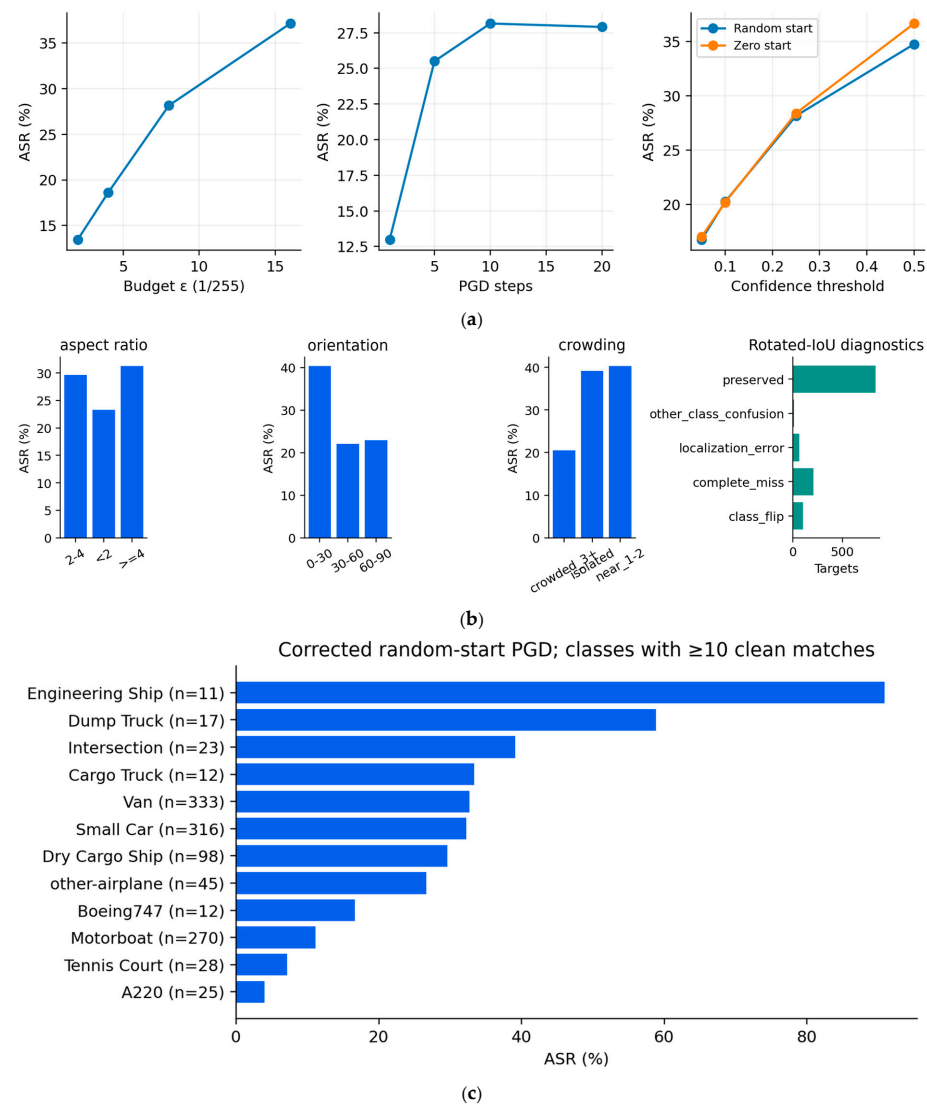


Figure 8. (a) FAIR1M random-start PGD sensitivity to perturbation budget and iteration count, with random-start and zero-start confidence-threshold curves. ASR uses each curve's source-clean envelope-IoU target set. (b) Corrected random-start PGD target strata and polygon-intersection-IoU diagnostic error modes; fixed-area placement controls are in Table 3c,d. (c) Corrected random-start PGD per-class ASR with source-clean envelope-IoU matched counts for classes with at least ten matches.

4.2.2. Target and Class Diagnostics

Among the 1216 clean targets matched by both envelope and polygon IoU, corrected random-start PGD preserves 835 (68.67%), causes 102 class flips (8.39%), 207 complete misses (17.02%), 63 localization errors (5.18%) and nine nearby other-class confusions (0.74%); zero targets meet the prespecified angle-error rule. Aspect-ratio, orientation and crowding ASRs are recomputed in Figure 8b and retain their actual 1247-target axis-matching denominators. The rotated-IoU diagnostic denominator and axis-matching ASR denominator are not interchangeable.

The corrected replay represents 28 classes with clean matches; 16 have fewer than ten. Figure 8c reports the 12 better-supported classes with their clean-matched counts. These point estimates are descriptive and do not establish stable rankings of sparsely represented classes.

4.3. Defense Comparison

SAM PGD-ZS lowers envelope-IoU Recall from 54.03% to 42.37% and native ProbIoU-based mAP50 from 43.85% to 36.22%. Bilateral filtering increases mAP50 by 2.75 points (95% CI, 0.78 to 4.82), mAP50-95 by 2.01 (0.48 to 3.64) and Recall by 1.60 (0.47 to 2.84), while Precision falls and false positives rise from 629 to 711. Figure 9 displays the fixed-threshold Recall comparison.

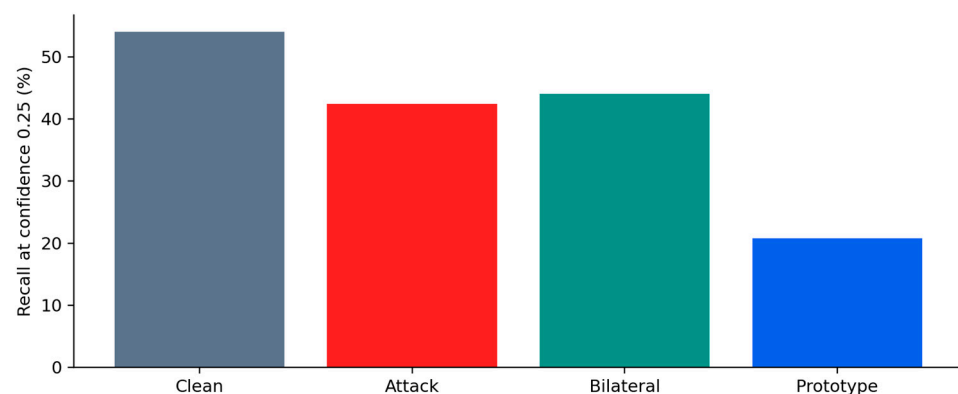


Figure 9. Corrected standard-attack fixed-threshold Recall; complete metrics and preprocessing controls appear in Table 4. The additional $g = 0$ -plus-fallback control is reported in Tables 4 and 4h. P/R and FP use envelope-IoU matching at confidence 0.25; AP uses native Ultralytics ProbIoU-based OBB evaluation.

Native ProbIoU-based AP integrates ranking and overlap across confidence thresholds, whereas envelope-IoU Recall@0.25 measures one operating point. Bilateral filtering has positive paired AP and Recall intervals but lower Precision. Relative ranking or localization changes may contribute; these aggregates do not identify a causal feature mechanism. Uniform monotonic score rescaling alone would not improve AP.

The complete $g = 0$ restoration path lowers Recall from 42.37% to 32.89%, a paired change of -9.49 points (95% CI, -14.38 to -4.31). Learned correction at $g = 0.15$ improves that control by 3.51 Recall points (1.80 to 5.52) and 1.73 mAP50 points (0.25 to 3.10). Against the raw attacked image, however, its ungated Recall remains lower by 5.98 points (95% CI, -10.24 to -1.03); its AP difference spans zero. At $g = 0.5$, ungated Recall is 39.12%, also above the $g = 0$ control, but its paired AP improvement over that control is uncertain. Thus, learned correction and full-pipeline preprocessing must be distinguished.

With confidence fallback, $g = 0.15$ gives 44.24% Recall, 37.82% mAP50 and 699 false positives; $g = 0.5$ gives 45.06%, 38.43% and 740. Relative to the 629-FP raw attacked input, Recall gains are +1.86 points (95% CI, 0.38 to 3.99) and +2.69 (1.00 to 5.10). Their mAP50

differences, +1.60 (−1.32 to 3.44) and +2.21 (−0.87 to 4.55), have intervals spanning zero. We report these positive Recall effects and false-positive costs without claiming joint recovery of clean performance. Paired FP differences are +70 (95% CI, 34 to 112) and +111 (54 to 179), respectively. At $g = 0.5$, the paired Precision difference is −2.43 points (−4.65 to −0.35). Table 4f reports these adverse effects alongside the positive Recall and uncertain AP effects using the same resampling method.

The clean 384-down/source-up path gives 42.59% envelope-IoU Recall versus 54.03% for the source input, a paired loss of 11.44 points (95% CI, −17.26 to −6.06). The perturbation-free 640 attack/export path gives 53.64% Recall and 44.68%/34.98% native ProbIoU-based mAP50/mAP50-95. Its Recall difference is −0.39 points (−1.97 to 1.07) and mAP50 difference +0.83 (−1.99 to 2.84); these intervals do not establish equivalence. Its target-conditioned ASR is $113/1247 = 9.06\%$ (conditional image-bootstrap 95% interval, 6.03% to 12.55%), with 104 newly matched targets offsetting most losses (Table 3f). The 384-path and clean gated controls have zero oracle repair masks, as defined in Section 3.2.1; their measured effects therefore describe resampling and selection.

JPEG, Gaussian, Median and Bilateral require no clean reference, whereas Prototype and learned SAM-FR use the paired-clean repair mask in Section 3.2.1. All branches share the 100-image evaluation, and each learned output is held out from its model's training fold. This pairing does not equalize the branches' information assumptions.

At $g = 0.15$, the corrected adaptive pipeline minus its own attacked branch changes Recall by 2.17 points (95% CI, 0.23 to 4.80), mAP50 by 0.53 (−2.10 to 2.57) and mAP50-95 by 0.55 (−1.69 to 2.37); false positives increase from 380 to 405. Recall intervals exclude zero, but AP intervals span zero and absolute pipeline performance remains far below the shared clean baseline.

At $g = 0.5$, the corrected adaptive pipeline minus its own attacked branch changes Recall by 2.47 points (95% CI, 0.24 to 5.38), mAP50 by 0.55 (−1.28 to 2.57) and mAP50-95 by 0.05 (−1.42 to 1.73); false positives increase from 410 to 443. Recall intervals exclude zero, but AP intervals span zero and absolute pipeline performance remains far below the shared clean baseline.

Across the prespecified margins, Recall varies by 0.17 points. Confidence-sum change is associated with per-image Recall, F1 and false-positive changes (Spearman $\rho = 0.607$, 0.427 and 0.678; all $p < 0.001$). These observational associations do not make confidence sum a calibrated measure of semantic quality.

Per-image gate outcomes expose failures that aggregate confidence correlations can conceal. On the ordinary corrected attacks, $g = 0$ plus the gate selects the resampled branch for 59/100 scenes; ten selected scenes have fewer TP and 25 have more FP than their raw attacked input, despite strictly larger confidence sums. Its pooled changes are +33 TP and +63 FP. At $g = 0.15$, 66/100 scenes select restoration, including six score ties: TP counts increase on 24 selected scenes, decrease on ten and remain unchanged on 32; FP counts increase on 31. At $g = 0.5$, 59/100 scenes select restoration, including three ties: TP counts increase on 29, decrease on seven and remain unchanged on 23; FP counts increase on 35. All TP-decrease and FP-increase selections at both gains have strictly larger confidence sums, so these adverse choices are not explained by tie handling. Positive and negative per-image TP-count changes sum to +63 and −20 at $g = 0.15$, and +76 and −14 at $g = 0.5$, giving the net +43 and +62 TP reported by the pooled Recall changes. Net FP increases are +70 and +111, consistent with Table 4. These are count differences, not identities of individually recovered targets or additional independent samples.

The training-response control uses identical fold assignments and seeds. On corrected test inputs, corrected minus archived training changes ungated mAP50 by +0.30 points (95% CI, −0.43 to 0.74) and Recall by −0.09 (−0.44 to 0.27) at $g = 0.15$; at $g = 0.5$, the

differences are +0.35 (−0.41 to 1.12) and −0.35 (−0.86 to 0.19). These paired intervals leave a training-distribution benefit unestablished.

Selector sensitivity. Physical minus asymmetric differences in percentage points (95% CI) are $g = 0.15$ restored: Recall −0.04 (−0.33 to 0.20), mAP50-95 0.00 (−0.18 to 0.20); $g = 0.15$ fallback: Recall 0.00 (−0.13 to 0.13), mAP50-95 −0.03 (−0.28 to 0.14); $g = 0.5$ restored: Recall 0.00 (−0.34 to 0.34), mAP50-95 0.15 (−0.14 to 0.38); $g = 0.5$ fallback: Recall −0.09 (−0.32 to 0.11), mAP50-95 −0.01 (−0.49 to 0.24). The frequency-coordinate control changes the meaning of the selector but does not establish a detection advantage on this locked set. Its 384-space PSNR is 36.52/34.87 dB at gains 0.15/0.5; both inputs and resized clean references follow the same path. These results do not validate selective preservation of object boundaries.

False-positive selector contrasts are also retained: physical minus asymmetric counts are −1 (95% CI, −5 to 3) at $g = 0.15$ without fallback, −1 (−5 to 3) at $g = 0.15$ with fallback, +10 (2 to 19) at $g = 0.5$ without fallback, and +6 (0 to 13) at $g = 0.5$ with fallback. Thus, the frequency-coordinate control is not uniformly favorable; the positive ungated $g = 0.5$ FP interval is a conditional adverse effect, not proof of feature hallucination or a particular FFT mechanism.

Across the three repetitions, restored/fallback Recall ranges are 36.40–36.61%/44.24–44.54% at $g = 0.15$ and 38.08–39.12%/44.67–45.06% at $g = 0.5$. These observed ranges complement the primary image-bootstrap intervals; Table 4g reports their descriptive scope.

For both gains in all three repetitions, ungated Recall is below the raw attacked-input 42.37%, while fallback Recall is above it and false positives exceed the raw input's 629.

The correction-disabled path plus the actual confidence gate selects its 384-path branch for 59/100 attacked scenes, giving 43.80% Recall, 37.18% mAP50 and 692 false positives. Relative to raw attacked input, its Recall effect is +1.43 (0.20 to 3.18) points and its FP effect is +63 (29 to 101). Learned-plus-gate minus $g = 0$ -plus-gate gives Recall effects of +0.43 (−0.26 to 1.20) and +1.26 (−0.06 to 2.55) points at gains 0.15 and 0.5; the corresponding mAP50 effects are +0.64 (−0.92 to 1.80) and +1.25 (−1.22 to 3.61) points. These contrasts attribute only the between-pipeline difference to enabling learned correction with the gate active; improvements over raw input also include resampling and confidence selection. They do not identify one causal feature mechanism. At both gains, the incremental Recall and AP intervals span zero; they do not establish an added task-metric benefit of learned correction with the gate active on this locked set. At $g = 0.5$, the incremental FP effect is +48 (95% CI, 1 to 103).

On unmodified clean images, the zero-mask learned path with fallback gives 54.25% Recall and 43.99% mAP50. Compared with clean source input, its Recall effect is +0.22 (−0.39 to 0.90) points, mAP50 is +0.14 (−1.17 to 1.25) points, and FP is +20 (1 to 43). This quantifies the complete clean oracle-assisted path, including selection, rather than assuming it is identity because the correction mask is empty. The clean Recall and AP difference intervals span zero, whereas the FP difference interval is strictly positive on this locked set. These conditional results do not establish clean-performance equivalence or preservation of Precision.

The clean-input gate audit likewise identifies individual adverse selections. Among the 33/100 scenes for which the 384-path branch is selected, TP counts increase on eight, decrease on four and remain unchanged on 21; FP counts increase on 14. All four TP-decrease scenes also have increased FP and strictly larger confidence sums. Positive and negative per-image TP changes sum to +13 and −8, yielding the pooled +5 TP (1247 to 1252), while FP increases by 20 (949 to 969). The same result applies to both gains because the zero-mask candidate images are byte-identical. This clean-reference control demonstrates

that a small average Recall change can coexist with scene-specific damage; it does not estimate false alarms of an input-only repair-mask estimator.

4.4. Image Fidelity

The corrected replay preserves area-related whole-image fidelity, but all values in Tables 3a and 5a are recomputed rather than inherited. GT-OBB locality uses independently rasterized annotation polygons for every support, including SAM; target PSNR/SSIM still use the separately specified common SAM region. Neither is evidence of manual pixel-level segmentation quality. Target-aligned geometry remains an unresolved alternative to the SAM contour.

Table 5. Corrected PGD-ZS model-space fidelity under dataset-specific protocols. a. Corrected FAIR1M model-space support fidelity against the same clean reference and independent GT-OBB locality proxy (mean $\pm$ sample SD). b. Corrected four-support fidelity inside the common independent GT-OBB region (mean $\pm$ sample SD).

Dataset	Method/Support	N	MAE	RMSE	PSNR (dB)	SSIM	Max L-inf (Post-Quant.)	Changed Ratio
DOTA1.5	PGD-ZS/Polygon	282	0.001227	0.004438	49.84	0.9953	0.031373	0.0727
DIOR	PGD-ZS/Box	100	0.003631	0.007324	44.38	0.9799	0.031373	0.2550
FAIR1M	PGD-ZS/SAM	100	0.000473	0.003039	51.70	0.9981	0.031373	0.0235

Note: These corrected-response metrics use lossless 640-space attack outputs and the paired 640-space clean image. DOTA1.5 has 282 images, DIOR and FAIR1M 100 each. The imposed L-infinity limit is checked on float tensors; the reported post-quantization maximum is 8/255. Dataset rows retain different supports, detectors and schedules and are not causal cross-dataset comparisons. Source-resolution and 384-space fidelity are reported separately.

(a)

Support	Area Ratio	PSNR (dB)	SSIM	Target PSNR	Target SSIM	GT-OBB Loc	GT-OBB Leak	N
Global	1.0000 $\pm$ 0.0000	34.19 $\pm$ 0.14	0.8626 $\pm$ 0.0773	33.94 $\pm$ 0.28	0.9575 $\pm$ 0.0434	0.0317 $\pm$ 0.0329	0.9683 $\pm$ 0.0329	100
Box	0.0526 $\pm$ 0.0612	48.29 $\pm$ 4.64	0.9944 $\pm$ 0.0086	33.15 $\pm$ 0.90	0.9551 $\pm$ 0.0432	0.5328 $\pm$ 0.1932	0.4672 $\pm$ 0.1932	100
Polygon	0.0315 $\pm$ 0.0359	50.24 $\pm$ 4.71	0.9971 $\pm$ 0.0054	33.22 $\pm$ 0.96	0.9576 $\pm$ 0.0401	0.7545 $\pm$ 0.2038	0.2455 $\pm$ 0.2038	100
SAM	0.0238 $\pm$ 0.0382	51.70 $\pm$ 4.90	0.9981 $\pm$ 0.0063	32.55 $\pm$ 0.98	0.9552 $\pm$ 0.0449	0.7696 $\pm$ 0.2212	0.2304 $\pm$ 0.2212	100

Note: All methods use identical image indices, detector weights, PGD-ZS parameters ($\epsilon = 8/255$, step = 2/255, 10 steps, seed 42), and lossless 640 $\times$ 640 model-space PNGs. Loc/Leak use one common reference M_{GT} : the union of all four-point FAIR1M ground-truth-oriented polygons, rasterized to each 640 $\times$ 640 image before evaluating the absolute RGB residual. The reference is independent of the saved SAM contour, though it is an oriented-box proxy rather than a manual pixel-level segmentation. Global, Box, Polygon, and SAM are all scored against this same region; in particular, the SAM row is no longer a self-reference consistency check. The annotation region covers 0.0308 $\pm$ 0.0321 of the image (mean $\pm$ sample SD). Target PSNR/SSIM, unlike Loc/Leak, retain the common clean-image SAM target region used in the original fidelity comparison. Box is the corrected axis-aligned xxyy support, and Polygon is the four-point OBB support. Attack supports arise from clean detector predictions where available; the generator can fall back to annotations if none are present, so this proxy should not be presented as an independent pixel-level contour truth. The reference region itself covers 0.0308 $\pm$ 0.0321 of each image, whereas the GT-OBB Loc column reports the fraction of the absolute RGB residual that falls inside that region (0.0317 $\pm$ 0.0329 for the corrected Global row); the two are different quantities, not a discrepancy.

(b)

Support	GT-Region PSNR (dB)	GT-Region SSIM	N
Global	33.98 $\pm$ 0.25	0.9385 $\pm$ 0.0445	100
Box	34.16 $\pm$ 1.47	0.9436 $\pm$ 0.0425	100
Polygon	34.47 $\pm$ 1.70	0.9479 $\pm$ 0.0408	100
SAM	35.90 $\pm$ 2.35	0.9726 $\pm$ 0.0333	100

Note: These metrics use the same 100 clean/attacked lossless 640-space pairs and the same annotation-polygon union as Table 5a Loc/Leak. PSNR is computed from mean squared RGB error within that union per image; local SSIM is the channel-mean full default 7 $\times$ 7 SSIM map averaged over the union, so neighborhoods may cross its boundary. The region is a bounding-box proxy, not manual pixel segmentation. It is common to every support and does not use the SAM contour for evaluation. No new attack is generated. The original SAM-region fidelity is retained in Table 5a rather than silently replaced.

The GT-region SAM result is 35.90 dB/0.9726, whereas its original common-SAM-region result is 32.55 dB/0.9552. The difference reflects the evaluated region, not inconsistent image pairs. Higher GT-region fidelity may partly reflect unmodified pixels inside the

larger annotation-box union; support areas and realized energies are still unequal. These values do not establish SAM contour accuracy or a SAM-specific geometry advantage.

Corrected-attack restoration fidelity uses newly trained out-of-fold models and a paired clean image resized directly to 384×384 (Table 6). The attack input has PSNR 36.39 dB, SSIM 0.9681 and outside-repair-mask error share 93.81%. The actual $g = 0$ network forward is identical to that input within 384 space, but its complete source-resolution path is not identity. At $g = 0.15$, mean PSNR is 36.51 dB and local PSNR 30.58 dB; at $g = 0.5$, these are 34.81 and 21.79 dB. Gain mismatch still affects fidelity, but the source-resolution resampling controls must be included when attributing detector changes to the network.

Table 6. Corrected-attack out-of-fold fidelity in the 384×384 checkpoint space, including $g = 0$ (mean $\pm$ sample SD; $n = 100$). a. Exact-path audit of archived 640- and 384-space fidelity measurements ($n = 100$; not substituted for corrected primary fidelity). b. Contents of the accompanying reproducibility archive MDPI_reproducibility_20261015.zip.

Stage	N	MAE	RMSE	PSNR (dB)	SSIM	Local PSNR	Local SSIM	Error Loc	Error Leak
Corrected PGD input	100	0.010415 $\pm$ 0.006751	0.018274 $\pm$ 0.010586	36.39 $\pm$ 5.71	0.9681 $\pm$ 0.0238	29.30 $\pm$ 3.63	0.9501 $\pm$ 0.0373	0.0619 $\pm$ 0.0841	0.9381 $\pm$ 0.0841
Network forward $g = 0$	100	0.010415 $\pm$ 0.006751	0.018274 $\pm$ 0.010586	36.39 $\pm$ 5.71	0.9681 $\pm$ 0.0238	29.30 $\pm$ 3.63	0.9501 $\pm$ 0.0373	0.0619 $\pm$ 0.0841	0.9381 $\pm$ 0.0841
New OOF restorer $g = 0.15$	100	0.010369 $\pm$ 0.006732	0.018066 $\pm$ 0.010525	36.51 $\pm$ 5.74	0.9682 $\pm$ 0.0239	30.58 $\pm$ 3.10	0.9575 $\pm$ 0.0390	0.0576 $\pm$ 0.0794	0.9424 $\pm$ 0.0794
New OOF restorer $g = 0.5$	100	0.011244 $\pm$ 0.007166	0.021160 $\pm$ 0.011217	34.81 $\pm$ 5.06	0.9641 $\pm$ 0.0283	21.79 $\pm$ 2.86	0.8647 $\pm$ 0.0750	0.1292 $\pm$ 0.1358	0.8708 $\pm$ 0.1358

Note: All rows share the corrected attacks, labels, 384-space paired references and nearest-resized effective masks. Normalized RGB floating tensors are measured before export quantization; source-space Table 4 follows the exact uint8 export/resize path. $g = 0$ equals the attack tensor here but still passes through resampling at source resolution. Correction confinement at 384 is an architectural check and does not imply source-resolution background identity.

(a)			
Audit Condition	MAE	PSNR (dB)	Error Outside Effective Mask
Historic saved attack vs. clean640	0.000499	51.26	—
Historic source attack vs. source clean	0.014301	33.95	93.18%
Clean 640 down/up only vs. source clean	0.014155	34.10	94.49%
Historic attack384 vs. direct clean384	0.010423	36.37	93.48%
Source attack vs. 640 down/up clean	0.000376	53.59	8.95%
Historic full $g = 0$ source output vs. source clean	0.024267	29.39	94.09%
Clean 384 down/up vs. source clean	0.021595	30.26	94.40%

(b)		
Artifact	Archive Path	Role
Implementation and tests	scripts/corrected_attack.py;	Corrected channel selection, attacks,
	scripts/author_code/	preprocessing, matching and restoration
Image and fold lists	records/image_names.json;	100 locked images, five
	records/folds.json	training/validation partitions and seeds
Target identities	records/target_identity_records.json;	Per-target source/matched-path
	records/dual_reference_asr.json	membership and predictions
Task and AP records	records/metrics/; ap_statistics/	Per-image envelope counts and native
		ProbIoU AP arrays for 55 conditions
Table and interval scripts	scripts/regenerate.py; records/intervals/	Regenerate principal task tables and
		paired 10,000-draw intervals
Models and fixed supports	checkpoints/; masks640/	Author-trained detector/restorers and
		frozen 640-space support masks
Provenance and hashes	MANIFEST_SHA256.json;	Archive checksums, source identities and
	records/*manifest*.json	retained path audit

Note: README.md maps the executable scripts to the reported protocols and specifies which computations were rerun. Principal task-table values are reconstructed from author-generated counts and AP statistics; per-target matches can also be reconstructed from saved predictions. The new source/matched-path audit includes full input hashes. Historical path_audit.json and CLOSEOUT-20261013 source_file_manifest.json remain in records/primary/audit/and records/closeout/; the latter retains SHA-256 9b51612e11874b6c35c68cc3ed68034d5f995856dfdbeb4fc95373636f75708b. Archive readers use non-object NPZ AP arrays. Archived runners retain original paths for provenance, while portable entry points accept explicit input roots. Third-party source images and pretrained weights are obtained separately; the archive supplies author-generated supports, records and author-trained models.

Paired image-level bootstrap with 10,000 resamples from the same 100 paired images for $g = 0.15$ minus corrected attack input gives mean PSNR +0.12 dB (95% CI, 0.08 to 0.16) and SSIM +0.00015 (−0.00019 to 0.00041). These checkpoint-space fidelity effects do not erase the full-path source-resolution Recall cost measured against $g = 0$ and raw input in Table 4. Figure 10 shows the corrected cross-dataset fidelity values, and Figure 11 the corrected same-image defense metrics.

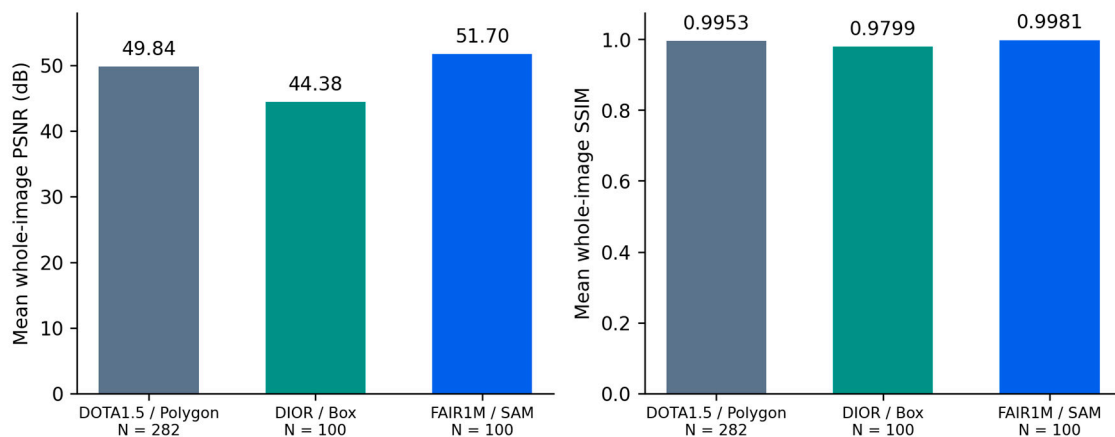


Figure 10. Final corrected PGD-ZS cross-dataset fidelity from the same records as Table 5: DOTA1.5 Polygon $n = 282$, DIOR Box $n = 100$ and FAIR1M SAM $n = 100$. PSNR/SSIM are means of per-image 640-space metrics; dataset-specific detectors, supports and schedules differ. This is not the four-support comparison in Table 5a or a causal cross-dataset ranking.

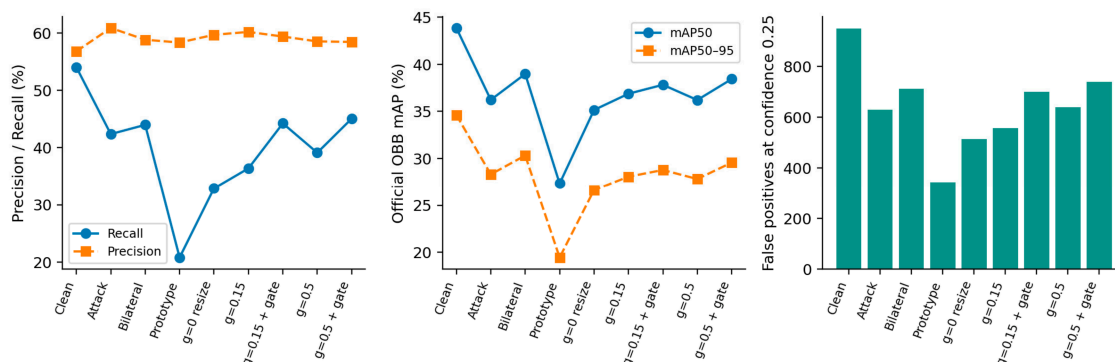


Figure 11. Corrected same-image defense metrics including the full-path $g = 0$ control, newly trained gains 0.15/0.5 and confidence fallback. Paired difference intervals appear in Section 4.3; plotted points do not replace those intervals. Tables 4 and 4h additionally report the $g = 0$ -plus-fallback control and its paired contrasts. P/R and FP use envelope-IoU matching at confidence 0.25; AP uses native Ultralytics ProbIoU-based OBB evaluation.

The historic discrepancy is reproducible: 640-space MAE/PSNR 0.000499/51.26 becomes 0.010423/36.37 after source export followed by restoration resizing, with 93.48% error outside the effective repair mask. Table 6a traces the exact paired references and resampling stages. A clean-image 640-down/source-up path already has MAE 0.014155, close to the historic source attack's 0.014301; relative to the same-preprocessed clean image, attack MAE is only 0.000376. This identifies a substantial resampling contribution, not an additive causal percentage of total MAE. Since whole-image resampling changes pixels outside M_δ , masked learned correction at 384 cannot confine the full source-resolution path. The $g = 0$ detection/fidelity controls prevent attributing all full-path losses to learned artifacts.

4.5. Limitations and Discussion Boundaries

The 100-image source baseline has envelope-IoU Recall 54.03% and native ProbIoU-based mAP50 43.85%; ordinary SAM attacks give 42.37% and 36.22%. Detector-training error, optimized perturbation effects and image-path effects are distinct. FAIR1M full-validation mAP50 remains 31.57%, and the 640-space energy and 384-space fidelity baselines retain their own references.

The matched-path Global/Box/Polygon/SAM ASRs are 61.47/32.55/26.66/24.39%; source-referenced values and the 9.06% resize-only ASR describe complete-path effects. Equal-area translation supports a target-proximity effect, while the target-aligned geometric comparison leaves a SAM-specific contour advantage unresolved. Exact-L1 normalization is post hoc rather than attack re-optimization. The annotation-derived locality proxy is independent of SAM construction but remains an oriented-box region.

Tables 4 and 4h separate resampling, learned correction and confidence selection. At the training-matched gain, learned correction improves the correction-disabled path, but ungated Recall remains below raw attacked input. With the same gate active, incremental learned-versus- $g = 0$ Recall and AP intervals span zero at both gains, while $g = 0.5$ increases false positives by 48 (95% CI, 1 to 103). Improvements over raw input therefore cannot be assigned solely to learned correction. The specified adaptive pipeline also has much lower absolute Recall, despite some recovery relative to its own attacked branch.

False positives can arise from reconstruction/detection objective mismatch, resampling and confidence selection. The restorer has no detector classification/localization supervision, and Fourier-magnitude matching does not determine phase or edge identity. Texture enhancement or object-like artifacts remain plausible mechanisms; $g = 0$ also produces damage without learned correction. The selector control retains the FFT term, so the current evidence cannot attribute false positives to that loss. A causal assessment would require a loss-on/off comparison and a spatial unmatched-box audit.

Confidence selection can improve pooled Recall while selecting scenes with fewer true positives or more false positives despite a higher confidence sum (Section 4.3). Such choices occur in $g = 0$ and clean zero-mask controls as well as learned branches. Their retrospective scene counts are descriptive findings on the locked sample; a deployment selection rule would require independent calibration of missed-target and false-positive costs.

Statistical scope. Image-bootstrap intervals condition on the fixed detector, generated attacks and saved restorers; draws do not retrain models or regenerate attacks. They exclude detector-training uncertainty and are unadjusted marginal 95% intervals rather than simultaneous family-wise intervals. Fold SD in Table 4 mixes scene composition and training randomness, while Table 4g gives descriptive SD over three complete out-of-fold repetitions on identical partitions. These summaries do not replace the conditional intervals. Tables 3f and 4h contain post hoc controls, and the clean gate contrast conditions on a zero oracle mask.

Dataset-specific supports and schedules, fixed prompts and frozen detectors define the evaluated scope. Confidence sensitivity changes its clean target set, and small class/geometry bins give limited evidence. The physical-frequency models are standard-attack selector controls; the primary standard and adaptive comparisons retain asymmetric weighting. They do not measure spectral-loss removal or phase-sensitive reconstruction.

Figure 11 summarizes the corrected standard same-image defense comparison, including both gains and the full-path zero-correction control.

The evaluation supports finite-sample findings about target placement, image paths and confidence selection. It leaves SAM-specific contour superiority and incremental gated learning benefits unestablished. Paired-clean repair masks make restoration an oracle-assisted benchmark. Manual pixel segmentation, input-only localization, other detector

checkpoints and physical-weather conditions remain outside the evaluation. The specified finite-step BPDA replay does not cover random restarts or joint prompt attacks, and its measured offline runtime is not defense-only latency or service throughput.

5. Conclusions

On the locked FAIR1M benchmark, matched-path PGD-ZS ASRs for Global, Box, Polygon and SAM are 61.47%, 32.55%, 26.66% and 24.39%. The perturbation-free attack/export path already loses 9.06% of source-clean target matches. Equal-area controls establish a placement effect, while the paired target-aligned SAM-minus-geometry interval leaves contour-specific superiority unresolved. The restoration controls separate 384-pixel re-sampling, learned correction and confidence selection: correction at $g = 0.15$ improves the ungated $g = 0$ path, yet its Recall remains below the raw attacked input. With the gate active, $g = 0$ already reaches 43.80% Recall; incremental learned-versus- $g = 0$ Recall and AP intervals span zero at both gains, and $g = 0.5$ adds 48 false positives (95% CI, 1 to 103). Bilateral filtering has positive AP and Recall effects on this fixed ordinary-attack set. These comparisons, including their negative and inconclusive outcomes, are the contribution of the evaluation.

The exact-forward BPDA replay gives a pipeline Recall of 23.40%/26.86% at gains 0.15/0.5. Its finite-step outcomes and conditional intervals do not certify worst-case or distribution-wide robustness. Fold variation and three same-partition training repetitions describe additional variability under fixed detector weights and attacks.

Restoration uses the paired clean image to construct the repair mask in Equation (9). These offline oracle-assisted results do not validate a deployable blind defense or a formal performance upper bound. The reproducibility archive makes the implementation, target identities and task-statistic comparisons inspectable.

Supplementary Materials: The following supporting information can be downloaded at <https://www.mdpi.com/article/10.3390/app162010392/s1>. The reproducibility archive MDPI_reproducibility_20261015.zip accompanies this article and contains: File S1: corrected attack code, including the corrected decoded class-channel objective, channel-selection tests, zero-start PGD/FGSM/MIFGSM attacks, explicit 640-pixel preprocessing/export and saved-image replay; File S2: lossless input preparation and reconstruction of the original annotation order; File S3: table-regeneration and paired image-bootstrap scripts; File S4: first-party matching, detector-adaptor, restoration and training code, together with the original experiment runners retained for provenance; File S5: the 100 FAIR1M image stems, five disjoint held-out folds, training seeds and cross-dataset lists; File S6: annotation geometries and matched target identities for all evaluated conditions; File S7: per-image fixed-threshold records and native OBB AP statistics for 55 conditions; File S8: primary, gate, closeout, spectral and interval records; File S9: the four fixed 640-pixel support masks for each FAIR1M image; File S10: 27 author-trained detector and restorer checkpoints with role and SHA-256 index; File S11: the regeneration outputs used to check the reported values; File S12: README.md, requirements.txt and MANIFEST_SHA256.json, giving the execution commands, software versions and file hashes. Third-party DOTA1.5, DIOR and FAIR1M source images and SAM pretrained weights are not redistributed; their provider identifiers and retained provenance are listed in the accompanying records.

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

Funding: This research was funded by the Key Laboratory of Trace Science and Technology, Ministry of Public Security, grant number: 2018FMKFKT05.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The accompanying archive MDPI_reproducibility_20261015.zip contains the corrected attack code and channel-selection tests; preprocessing and one-to-one matching routines; fixed 640-pixel support masks; the 100-image and five-fold lists; per-image target identities and predictions; per-image fixed-threshold records and native OBB AP statistics for 55 conditions; scripts that regenerate principal task tables and paired image-bootstrap intervals; and author-trained detector and restorer checkpoints. README.md gives executable commands, artifact roles, software versions and the scope of verified calculations. File identities and SHA–256 hashes are supplied in MANIFEST_SHA256.json and the input/checkpoint manifests. Source DOTA1.5, DIOR and FAIR1M images, SAM weights and third-party detector weights are obtained from their official providers and are excluded from the archive. Their identifiers and retained provenance are supplied with the author-generated records. No external repository DOI or public upload is asserted here.

Acknowledgments: The authors thank the Institute of Forensic Science of China and the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, for their institutional support and collaboration.

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

Abbreviations

The following abbreviations are used in this manuscript:

ASR	Attack Success Rate
BIM	Basic Iterative Method
BPDA	Backward Pass Differentiable Approximation
DIOR	Detection in Optical Remote Sensing Images
DOTA	Dataset for Object Detection in Aerial images
FAIR1M	Fine-Grained Object Recognition in High-Resolution Remote Sensing Imagery
FFT	Fast Fourier Transform
FGSM	Fast Gradient Sign Method
FN	False Negative
FP	False Positive
IoU	Intersection over Union
MAE	Mean Absolute Error
mAP	Mean Average Precision
MIFGSM	Momentum Iterative Fast Gradient Sign Method
NMS	Non-Maximum Suppression
OBB	Oriented Bounding Box
OOF	Out-of-Fold
PGD-ZS	Projected Gradient Descent with Zero Initialization (Reported Branch)
PSNR	Peak Signal-to-Noise Ratio
RMSE	Root Mean Square Error
SAM	Segment Anything Model
SAM-FR	SAM-Guided Feature Restoration
SSIM	Structural Similarity Index Measure
TP	True Positive
YOLO	You Only Look Once

References

1. Xia, G.-S.; Bai, X.; Ding, J.; Zhu, Z.; Belongie, S.; Luo, J.; Datcu, M.; Pelillo, M.; Zhang, L. DOTA: A Large-Scale Dataset for Object Detection in Aerial Images. In *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018*; IEEE: New York, NY, USA, 2018; pp. 3974–3983. [\[CrossRef\]](#)

2. Sun, X.; Wang, P.; Yan, Z.; Xu, F.; Wang, R.; Diao, W.; Chen, J.; Li, J.; Feng, Y.; Xu, T.; et al. FAIR1M: A Benchmark Dataset for Fine-Grained Object Recognition in High-Resolution Remote Sensing Imagery. *ISPRS J. Photogramm. Remote Sens.* **2022**, *184*, 116–130. [[CrossRef](#)]
3. Li, K.; Wan, G.; Cheng, G.; Meng, L.; Han, J. Object Detection in Optical Remote Sensing Images: A Survey and a New Benchmark. *ISPRS J. Photogramm. Remote Sens.* **2020**, *159*, 296–307. [[CrossRef](#)]
4. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and Harnessing Adversarial Examples. In Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015), San Diego, CA, USA, 7–9 May 2015.
5. Kurakin, A.; Goodfellow, I.J.; Bengio, S. Adversarial Examples in the Physical World. In Proceedings of the 5th International Conference on Learning Representations (ICLR 2017), Toulon, France, 24–26 April 2017. Workshop Track Proceedings.
6. Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; Vladu, A. Towards Deep Learning Models Resistant to Adversarial Attacks. In Proceedings of the 6th International Conference on Learning Representations (ICLR 2018), Vancouver, BC, Canada, 30 April–3 May 2018.
7. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing Properties of Neural Networks. In Proceedings of the 2nd International Conference on Learning Representations (ICLR 2014), Banff, AB, Canada, 14–16 April 2014.
8. Brown, T.B.; Mané, D.; Roy, A.; Abadi, M.; Gilmer, J. Adversarial Patch. *arXiv* **2017**, arXiv:1712.09665.
9. Liu, X.; Yang, H.; Liu, Z.; Song, L.; Li, H.; Chen, Y. DPatch: An Adversarial Patch Attack on Object Detectors. *arXiv* **2018**, arXiv:1806.02299.
10. Eykholt, K.; Evtimov, I.; Fernandes, E.; Li, B.; Rahmati, A.; Xiao, C.; Prakash, A.; Kohno, T.; Song, D. Robust Physical-World Attacks on Deep Learning Visual Classification. In *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018*; IEEE: New York, NY, USA, 2018; pp. 1625–1634. [[CrossRef](#)]
11. Thys, S.; Van Ranst, W.; Goedemé, T. Fooling Automated Surveillance Cameras: Adversarial Patches to Attack Person Detection. In *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Long Beach, CA, USA, 16–20 June 2019*; IEEE: New York, NY, USA, 2019; pp. 49–55. [[CrossRef](#)]
12. Xie, C.; Wu, Y.; van der Maaten, L.; Yuille, A.L.; He, K. Feature Denoising for Improving Adversarial Robustness. In *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019*; IEEE: New York, NY, USA, 2019; pp. 501–509. [[CrossRef](#)]
13. Guo, C.; Rana, M.; Cissé, M.; van der Maaten, L. Countering Adversarial Images Using Input Transformations. In Proceedings of the 6th International Conference on Learning Representations (ICLR 2018), Vancouver, BC, Canada, 30 April–3 May 2018.
14. Liao, F.; Liang, M.; Dong, Y.; Pang, T.; Hu, X.; Zhu, J. Defense Against Adversarial Attacks Using High-Level Representation Guided Denoiser. In *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018*; IEEE: New York, NY, USA, 2018; pp. 1778–1787. [[CrossRef](#)]
15. 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 28 (NIPS 2015), Montreal, QC, Canada, 7–12 December 2015*; Curran Associates, Inc.: Red Hook, NY, USA, 2015; pp. 91–99.
16. Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017*; IEEE: New York, NY, USA, 2017; pp. 936–944. [[CrossRef](#)]
17. Han, J.; Ding, J.; Xue, N.; Xia, G.-S. ReDet: A Rotation-Equivariant Detector for Aerial Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021*; IEEE: New York, NY, USA, 2021; pp. 2785–2794. [[CrossRef](#)]
18. Yang, X.; Yang, J.; Yan, J.; Zhang, Y.; Zhang, T.; Guo, Z.; Sun, X.; Fu, K. SCRDet: Towards More Robust Detection for Small, Cluttered and Rotated Objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019*; IEEE: New York, NY, USA, 2019; pp. 8231–8240. [[CrossRef](#)]
19. Jocher, G.; Chaurasia, A.; Qiu, J. Ultralytics YOLOv8. Available online: <https://github.com/ultralytics/ultralytics> (accessed on 8 September 2026).
20. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Proceedings of the Advances in Neural Information Processing Systems 32 (NeurIPS 2019), Vancouver, BC, Canada, 8–14 December 2019*; Curran Associates, Inc.: Red Hook, NY, USA, 2019; pp. 8024–8035.
21. Cohen, J.; Rosenfeld, E.; Kolter, J.Z. Certified Adversarial Robustness via Randomized Smoothing. In *Proceedings of the 36th International Conference on Machine Learning (ICML 2019), Long Beach, CA, USA, 9–15 June 2019*; PMLR: New York, NY, USA, 2019; Volume 97, pp. 1310–1320.
22. Dhillon, G.S.; Azizadenesheli, K.; Lipton, Z.C.; Bernstein, J.; Kossai, J.; Khanna, A.; Anandkumar, A. Stochastic Activation Pruning for Robust Adversarial Defense. In Proceedings of the 6th International Conference on Learning Representations (ICLR 2018), Vancouver, BC, Canada, 30 April–3 May 2018.

23. Ding, J.; Xue, N.; Xia, G.-S.; Bai, X.; Yang, W.; Yang, M.Y.; Belongie, S.; Luo, J.; Datcu, M.; Pelillo, M.; et al. Object Detection in Aerial Images: A Large-Scale Benchmark and Challenges. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 7778–7796. [[CrossRef](#)] [[PubMed](#)]
24. Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A.C.; Lo, W.-Y.; et al. Segment Anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 2–6 October 2023*; IEEE: New York, NY, USA, 2023; pp. 3992–4003. [[CrossRef](#)]
25. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [[CrossRef](#)] [[PubMed](#)]
26. Hore, A.; Ziou, D. Image Quality Metrics: PSNR vs. SSIM. In *Proceedings of the 2010 20th International Conference on Pattern Recognition (ICPR), Istanbul, Turkey, 23–26 August 2010*; IEEE: New York, NY, USA, 2010; pp. 2366–2369. [[CrossRef](#)]
27. Dong, Y.; Liao, F.; Pang, T.; Su, H.; Zhu, J.; Hu, X.; Li, J. Boosting Adversarial Attacks with Momentum. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018*; IEEE: New York, NY, USA, 2018; pp. 9185–9193. [[CrossRef](#)]
28. Kurakin, A.; Goodfellow, I.; Bengio, S. Adversarial Machine Learning at Scale. In *Proceedings of the International Conference on Learning Representations (ICLR), Toulon, France, 24–26 April 2017*.
29. He, K.; Sun, J.; Tang, X. Guided Image Filtering. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1397–1409. [[CrossRef](#)] [[PubMed](#)]
30. Telea, A. An Image Inpainting Technique Based on the Fast Marching Method. *J. Graph. Tools* **2004**, *9*, 23–34. [[CrossRef](#)]
31. Canny, J. A Computational Approach to Edge Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1986**, *8*, 679–698. [[CrossRef](#)]
32. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Proceedings of the Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015, Munich, Germany, 5–9 October 2015*; Springer: Cham, Switzerland, 2015; pp. 234–241. [[CrossRef](#)]
33. Zhang, K.; Zuo, W.; Chen, Y.; Meng, D.; Zhang, L. Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising. *IEEE Trans. Image Process.* **2017**, *26*, 3142–3155. [[CrossRef](#)] [[PubMed](#)]
34. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *Proceedings of the 9th International Conference on Learning Representations (ICLR 2021), Virtual Conference, 3–7 May 2021*.
35. He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; Girshick, R. Masked Autoencoders Are Scalable Vision Learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022*; IEEE: New York, NY, USA, 2022; pp. 15979–15988. [[CrossRef](#)]
36. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021*; IEEE: New York, NY, USA, 2021; pp. 9992–10002. [[CrossRef](#)]
37. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015), San Diego, CA, USA, 7–9 May 2015*.
38. Loshchilov, I.; Hutter, F. Decoupled Weight Decay Regularization. In *Proceedings of the 7th International Conference on Learning Representations (ICLR 2019), New Orleans, LA, USA, 6–9 May 2019*.
39. Buslaev, A.; Iglovikov, V.I.; Khvedchenya, E.; Parinov, A.; Druzhinin, M.; Kalinin, A.A. Albumentations: Fast and Flexible Image Augmentations. *Information* **2020**, *11*, 125. [[CrossRef](#)]
40. Efron, B. Bootstrap Methods: Another Look at the Jackknife. *Ann. Stat.* **1979**, *7*, 1–26. [[CrossRef](#)]
41. DiCiccio, T.J.; Efron, B. Bootstrap Confidence Intervals. *Stat. Sci.* **1996**, *11*, 189–228. [[CrossRef](#)]
42. Athalye, A.; Carlini, N.; Wagner, D. Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples. In *Proceedings of the 35th International Conference on Machine Learning (ICML 2018), Stockholm, Sweden, 10–15 July 2018*; PMLR: New York, NY, USA, 2018; Volume 80, pp. 274–283.
43. Tramèr, F.; Carlini, N.; Brendel, W.; Madry, A. On Adaptive Attacks to Adversarial Example Defenses. In *Proceedings of the Advances in Neural Information Processing Systems 33 (NeurIPS 2020), Virtual, 6–12 December 2020*; Curran Associates, Inc.: Red Hook, NY, USA, 2020; pp. 1633–1645.
44. Bouthillier, X.; Delaunay, P.; Bronzi, M.; Trofimov, A.; Nichyporuk, B.; Szeto, J.; Mohammadi Sepahvand, N.; Raff, E.; Madan, K.; Voleti, V.; et al. Accounting for Variance in Machine Learning Benchmarks. In *Proceedings of the 4th Conference on Machine Learning and Systems (MLSys 2021), Virtual, 5–9 April 2021*; MLSys: Virtual, 2021; pp. 747–769.

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.