



Self-adaptive deployment of synthetic-trained object detectors via test-time adaptation

Zahra Elmi¹ · Furkan Ali Turan² · Emir Tuna² · Abdullatif Akkaya² · Soheila Elmi³

Received: 6 January 2026 / Revised: 16 April 2026 / Accepted: 24 May 2026 / Published online: 6 June 2026
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2026

Abstract

Deploying synthetic-trained object detectors in real environments is challenging under single-image, batch=1 inference, where pseudo-label noise and unstable normalization can undermine online adaptation. We present a deployment-oriented test-time adaptation framework for synthetic-to-real object detection that requires neither source data nor target labels. The framework combines entropy-guided adaptation triggering, IoU-filtered pseudo-labeling with augmentation consistency and greedy matching, and architecture-aware normalization updates for both CNN- and Transformer-based detectors. Trained on LookSIM synthetic data and evaluated on 3,500 real defense images spanning six target classes, the method improves AP@[.5:.95] from 51.7 to 56.5 for RF-DETR and from 49.7 to 54.9 for YOLOv12-L while preserving real-time deployment on RTX 3090 and Jetson AGX Orin. Ablation results show that IoU filtering reduces pseudo-label false positives from 40% to 15%, memory-augmented normalization limits batch=1 degradation, and entropy-guided selection provides additional gains. Additional robustness analyses under severe occlusion and illumination stress show that source-side synthetic stress augmentation improves pseudo-label stability and reduces the loss of adaptation benefit in harder night and low-visibility regimes. Results on a held-out target condition and on the public RarePlanes benchmark further indicate that the proposed framework remains effective beyond both the dominant deployment subset and the specific LookSIM-to-defense setting. These findings strengthen the case for test-time adaptation as a deployment-oriented correction mechanism for synthetic-trained detectors when labeled real data remain limited.

Keywords Test-time adaptation · Real-time object detection · Synthetic data · Domain adaptation · Edge deployment

1 Introduction

Deep learning has substantially improved object detection and tracking when models can be trained on large, diverse, and operationally representative datasets. In defense-oriented applications such as UAV, armored-vehicle, and threat detection, however, acquiring such data is often constrained by security restrictions, ethical concerns, collection cost, and limited access to operational environments [1]. These constraints have increased interest in synthetic data, which offers lower acquisition cost, greater controllability, and stronger auditability than fully real-data pipelines. Synthetic imagery generated with game engines and computer-graphics workflows has therefore become increasingly relevant across autonomous driving, remote sensing, and defense-oriented perception [2]. Early work such as *Playing for Data* demonstrated that virtual environments can support dense perception tasks including detection and segmentation [3], while later benchmarks such as Virtual

✉ Zahra Elmi
zehraelmi@beykoz.edu.tr
Furkan Ali Turan
030721021@std.izu.edu.tr
Emir Tuna
030721024@std.izu.edu.tr
Abdullatif Akkaya
akkaya.abdullatif@std.izu.edu.tr
Soheila Elmi
selmi24@ku.edu.tr

¹ Department of Software Engineering, Beykoz University, Muhtar Sokak No: 3, Istanbul 34010, Kavacak, Turkey
² Department of Computer Engineering, Istanbul Sabahattin Zaim University, Halkalı Cad., Istanbul 34303, Küçükçekmece, Turkey
³ Department of Electrical and Electronics Engineering, Koç University, Rumelifeneri Yolu, Istanbul 34450, Sarıyer, Turkey

KITTI, SYNTHIA, and RarePlanes further established the value of synthetic imagery for synthetic–real transfer.

Motivated by these developments, we introduce **LookSIM**, a multimodal synthetic-data generation tool implemented in Unreal Engine 5.3. LookSIM produces spatially aligned RGB, thermal IR, segmentation masks, and surface normals for each scene (Fig. 2), while its parametric and procedural design provides explicit control over environmental conditions, illumination, viewpoints, target categories, and object placement. This enables large-scale scene generation with dense supervision and limited manual intervention. Example outputs are shown in Fig. 1 for thermal and depth channels and in Fig. 3 for segmentation masks. Using this pipeline, we study *synthetic-to-real* generalization under a strictly *synthetic-only* training regime.

We compare two detectors with distinct inductive biases: **YOLOv12-L**, a CNN-based one-stage detector optimized for real-time regression, and **RF-DETR**, a Transformer-based detector designed to exploit global contextual reasoning. The YOLO family is widely used in real-time and edge-oriented detection settings [4], whereas RF-DETR adopts a DINOv2-based backbone with an optimized Transformer architecture and has been reported to achieve competitive COCO accuracy at real-time operating points [5]. RF-DETR-large is also substantially larger than YOLOv12-L, with approximately 128 M parameters versus approximately 40–50 M parameters, making the comparison informative not only across detector families but also across deployment trade-offs [6].

The utility of synthetic data for real-world perception has been studied extensively. Prior work has reported improvements of up to approximately 10% in defense target detection using Unreal Engine CGI and diffusion-generated imagery combined with real vehicle photographs [7]. In many settings, mixed synthetic–real training yields the strongest results, whereas synthetic-only or real-only training tends to be less robust [8]. In the present setting, however, access to

labeled real data is restricted. We therefore train exclusively on synthetic data and evaluate zero-shot on real imagery. To improve pre-deployment robustness, LookSIM incorporates domain randomization through variations in illumination, material properties, pose, and background, together with image-level augmentations such as rotation, blur, scaling, and brightness perturbation [9].

Despite these advances, deployment reliability under synthetic-to-real shift remains a central challenge [10]. Synthetic-trained detectors often degrade substantially when transferred to real imagery, with prior studies reporting losses on the order of 10–20% in defense-oriented settings [11]. Existing mitigation strategies, including domain adaptation [12], mixed synthetic–real training [6], and sim-to-real translation through GAN-based approaches [13], typically assume access to target-domain data during training or require substantial retraining and redesign. These assumptions are often impractical in restricted defense scenarios, where target data are scarce, unavailable, or operationally mismatched with the final deployment environment.

Test-time adaptation (TTA) offers a complementary alternative by allowing a model to adapt online using only unlabeled target samples during deployment [14]. Recent TTA studies in image classification have shown promising gains under distribution shift [15], but extending this idea to object detection is nontrivial. Unlike classification, detection requires structured region-level predictions, and pseudo-label quality depends jointly on classification confidence and localization accuracy. In addition, the small batch sizes typical of real deployment, especially on edge hardware, make normalization updates unstable [16]. Under these conditions, localization errors can contaminate pseudo-labels and accumulate over time, ultimately undermining adaptation reliability [17].

These limitations become more severe in the deployment scenario studied here, which combines four constraints: synthetic-to-real transfer, no access to source training data

Fig. 1 Example thermal (infrared) image and depth map produced by LookSIM. Only RGB/IR are used for training; depth is retained for diagnostics



at adaptation time, online single-image inference, and batch = 1 execution on edge-oriented hardware. Under this regime, localization-driven pseudo-label noise can accumulate rapidly, while conventional normalization updates become unreliable because batch statistics are poorly estimated. The objective of this work is therefore not merely to apply an existing TTA method to synthetic-trained detectors, but to design a detection-specific adaptation framework that remains stable and effective under these deployment constraints.

To this end, we propose a deployment-oriented test-time adaptation framework for synthetic-to-real object detection under single-image, batch = 1 inference. The framework combines uncertainty-triggered adaptation, noise-controlled pseudo-label generation, and architecture-specific normalization updates to support stable online adaptation for both CNN- and Transformer-based detectors. To broaden the empirical scope of the study despite restricted defense-data availability, we evaluate the framework not only on the main real deployment set, but also on a held-out night-imagery target condition and on the public RarePlanes synthetic-to-real benchmark. Our main contributions are as follows:

1. We introduce a detection-specific pseudo-labeling mechanism that combines confidence filtering, augmentation consistency, and greedy IoU matching to suppress localization-driven pseudo-label noise during online adaptation, reducing pseudo-label false positives from 40% to 15%.
2. We propose an uncertainty-triggered adaptation strategy based on confidence-weighted detection entropy, enabling selective updates on domain-shifted samples and improving adaptation efficiency.
3. We develop architecture-specific normalization update rules for batch = 1 deployment: memory-augmented batch-normalization statistics with affine updates for YOLOv12-L, and affine-only layer-normalization adaptation for RF-DETR, yielding stable adaptation under single-image inference.
4. We provide a batch-size sensitivity analysis showing that the proposed framework remains stable at batch = 1, where conventional test-time adaptation becomes substantially less reliable.
5. We evaluate the framework on a held-out night-imagery target condition within the real deployment set and on the public RarePlanes benchmark, showing that adaptation gains remain positive beyond the dominant deployment regime.
6. We present a deployment study on both RTX 3090 and NVIDIA Jetson AGX Orin platforms, showing that the proposed method improves detection accuracy while preserving real-time feasibility, with end-to-end latency

overheads of only +10.8 ms for YOLOv12-L and +14.2 ms for RF-DETR.

2 Related work

Training modern vision models on real imagery typically requires large, diverse, and densely annotated datasets, yet acquiring such data is expensive, time-consuming, and often operationally constrained [18]. These limitations have motivated growing interest in synthetic-data generation pipelines, in which simulation engines produce large volumes of annotated imagery together with precise ground truth for tasks such as segmentation and object detection [19]. In autonomous driving, datasets such as Virtual KITTI and SYNTHIA have become canonical references for studying synthetic data generation and sim-to-real transfer [20, 21]. More broadly, recent surveys have emphasized the scalability, controllability, and reproducibility of synthetic-data pipelines across computer-vision domains [18]. Similar trends have emerged outside urban driving. In remote sensing and geospatial analysis, for example, synthetic Sentinel-2 imagery has been used to reduce annotation burden while maintaining competitive land-cover classification performance [22]. Toolkits such as Unity Perception further show that large-scale, well-annotated synthetic datasets can be generated efficiently and reproducibly [19]. In some settings, detectors trained on synthetic imagery have approached or exceeded real-data baselines [23]. Despite this progress, the central unresolved issue remains the simulator-to-real distribution mismatch, commonly referred to as the *domain gap*.

A substantial body of work has therefore focused on reducing the synthetic-to-real gap. One major direction is *domain randomization*, in which nuisance factors such as background, texture, material, illumination, and viewpoint are deliberately varied to encourage invariance, including more recent content-aware randomization strategies [1, 9, 24]. A second direction is *appearance transformation*, where GAN- or diffusion-based sim-to-real pipelines attempt to make synthetic imagery visually closer to real data [25, 26]. These methods can be effective, particularly at scale and under synthetic-only training regimes, but improvements in image realism metrics such as FID or IS do not necessarily translate monotonically into better downstream task performance [26, 27]. A third and often practically strong direction is *mixed or incremental training*, where synthetic and real data are combined jointly or sequentially, for example through synthetic pre-training followed by real-domain fine-tuning [6, 28]. RarePlanes is a representative example, combining synthetic and real satellite imagery to reduce dependence on large labeled real datasets while improving

detection performance [6]. In defense-oriented detection, synthetic-only training has also yielded meaningful gains, although mixed synthetic–real regimes generally remain stronger when labeled real data are available [23].

Beyond synthetic-data generation itself, recent detection research has also improved performance in specialized sensing domains through task-specific model and loss design. In remote sensing, Xu et al. proposed a SAR ship detection method based on distribution-based anchor assignment, comprehensive score voting, and Distance-Penalty IoU loss, improving localization quality and robustness in challenging SAR imagery [31]. More recently, Xu et al. introduced *DiffSARShipInst*, a diffusion-based framework for SAR ship instance segmentation that formulates instance prediction from a generative denoising perspective [32]. These studies highlight the continued importance of architecture and prediction design for domain-specific perception. Their emphasis, however, is on improving detector or dense-prediction quality within a given sensing domain rather than addressing source-free online adaptation from synthetic training data to real deployment data.

2.1 Test-time adaptation

Test-time adaptation (TTA) addresses distribution shift from a different direction by allowing a model to update during deployment using only unlabeled target samples, without revisiting the original source training data [14]. Early TTA methods were developed primarily for image classification and typically relied on entropy minimization, self-training, or adaptation of batch-normalization statistics [16]. More recent work has studied continual adaptation under evolving target distributions and has emphasized the sensitivity of TTA to batch size, update stability, and catastrophic drift [15]. These observations are especially relevant in deployment settings where samples arrive sequentially and updates must remain stable under strict computational and memory constraints.

Extending TTA from classification to object detection is substantially more difficult. Unlike classification, detection requires structured spatial predictions, and pseudo-label reliability depends not only on class confidence but also on localization quality. This makes online self-training more fragile under distribution shift. Ruan et al. [14] introduced IoU-based indicators for pseudo-label filtering and showed that such filtering can improve adaptation across multiple object-detection datasets. Wang et al. [15] proposed sensitivity-guided channel pruning for continual test-time adaptation, reducing computational overhead while preserving adaptation gains. These studies show that object-detection TTA is feasible and increasingly relevant, but they mainly address generic domain-shift scenarios rather than the more

constrained synthetic-to-real deployment regime considered here.

The present work lies at the intersection of these two research threads: synthetic-data-based detection and test-time adaptation. Relative to prior synthetic-data literature, our setting assumes synthetic-only training and no labeled real data for model tuning. Relative to prior TTA literature, our setting imposes a stricter combination of constraints, namely source-free adaptation, synthetic-to-real transfer, online single-image inference, and batch = 1 execution under deployment-oriented compute limits. Under these conditions, two problems become particularly severe: localization-driven pseudo-label errors can accumulate rapidly, and conventional normalization updates become unreliable because the effective batch statistics are weak or unstable. This motivates a detection-specific adaptation strategy tailored to stable synthetic-to-real deployment rather than a direct reuse of generic TTA pipelines.

3 Proposed framework and experimental protocol

This section describes the LookSIM synthetic generation environment, the detector architectures, the unified training and domain-randomization protocol, the deployment-oriented test-time adaptation (TTA) framework, and the evaluation and runtime protocol. Unless otherwise stated, all main results are reported on real imagery, whereas synthetic-set experiments are used only for protocol verification and controlled analysis.

3.1 LookSIM environment, dataset construction, and detector training

LookSIM synthetic environment. LookSIM is a fully automated multimodal simulation platform implemented in Unreal Engine 5.3 using Blueprints and C++. The environment procedurally constructs scenes within predefined parameter ranges in order to reduce manual intervention while increasing appearance diversity. Scene generation randomizes terrain categories such as desert, forest, urban, and open-sea environments, together with environmental assets including buildings, trees, and rocks, as well as atmospheric factors such as fog, rain, and time-of-day.

Each capture yields spatially aligned RGB, thermal infrared (IR), segmentation masks, surface normals, and approximate depth. In the present study, only RGB and IR channels are used for detector training and evaluation. Surface normals and depth are retained solely for diagnostic purposes, including visual verification of alignment quality and inspection of occlusion plausibility. Segmentation

Fig. 2 Representative LookSIM samples showing RGB, thermal IR, surface normals, and approximate depth. Only RGB/IR are used for training and evaluation; normals and depth are used for diagnostics



Fig. 3 Sample segmentation output produced with LookSIM



masks and normals are produced through deferred rendering and G-buffer extraction, while semantic labels are assigned using class-specific materials whose rendered colors are read directly as ground truth. Thermal IR imagery is synthesized through a shader-based approximation of radiometric contrast using ambient temperature, object-specific heat attributes, and solar irradiance and is exported in 8-bit format. For every object instance, the simulator records class identity, 2D bounding box, and pose metadata. Annotations are stored in YOLO-compatible format, with additional JSON/YAML metadata. In video scenarios, persistent track identifiers are also retained to support multi-object tracking analysis.

The generated supervisory signals are validated qualitatively. Figure 1 illustrates aligned thermal and depth outputs, Fig. 2 presents representative multimodal samples, Fig. 3 shows segmentation outputs, and Fig. 4 visualizes the normals channel. These additional modalities are not used by the detection models and serve only as simulator diagnostics.

Synthetic dataset construction. The dataset models six defense-relevant target categories: UAV, fixed-wing aircraft, main battle tank, armored personnel carrier, personnel, and military truck. For each class, approximately ten 3D assets are used, together with intra-class variation in scale, camouflage, and platform subtype. All images are rendered at a resolution of 1920×1080 . In total, more than 10,000 synthetic images are generated and partitioned into training, validation, and synthetic-test splits using an 80/10/10 ratio. Final performance evaluation is conducted on a separate real-image test set described in Sect. 3.3.

To support the robustness analysis under harder deployment regimes, we additionally construct a separate severe-shift synthetic subset generated under the SSA protocol. Unlike the base synthetic set, this subset is intentionally biased toward conditions that are expected to reduce pseudo-label reliability during deployment, including severe low-light and night illumination, strong local illumination imbalance, visibility degradation, partial target occlusion, and small-object stress caused by longer stand-off distances. This subset is used only for the targeted robustness ablation

Fig. 4 Example surface normal map generated with LookSIM; normal vectors are color-coded



reported in Section 4.3 and does not alter the main evaluation protocol.

Detector architectures. We evaluate two complementary detection paradigms. YOLOv12-L is a single-stage CNN-based detector with multi-scale feature aggregation and lightweight attention modules. Inputs are resized to 640×640 , predictions are generated at $8\times$, $16\times$, and $32\times$ strides, and inference uses non-maximum suppression (NMS). Class outputs are represented by per-anchor sigmoid scores, corresponding to independent binary predictions across classes. RF-DETR is a DETR-style encoder–decoder detector with a CNN/DINOv2-derived backbone, deformable attention, and learnable object queries. In contrast to YOLOv12-L, RF-DETR produces final object predictions without NMS through bipartite matching. Its classification head outputs a softmax distribution over $C + 1$ categories, including the background/no-object class. We use the *large* RF-DETR variant, comprising approximately 128 M parameters.

Unified optimization and domain randomization. To ensure that architectural differences rather than training mismatches drive the comparison, both detectors are trained under a shared optimization schedule. Specifically, we use AdamW with an initial learning rate of 1×10^{-4} , weight decay 1×10^{-4} , batch size 16, a maximum of 200 epochs, 5 warm-up epochs with linear scheduling followed by cosine decay, and mixed-precision training (FP16). For YOLOv12-L, momentum is set to 0.937. Exponential moving average (EMA) is enabled, and the final checkpoint is selected according to validation $AP@[.5:.95]$ with early stopping using patience = 30 epochs. External augmentations include scale jitter, horizontal and vertical flips, small-angle rotation ($\pm 5^\circ$), mild blur, and brightness/contrast perturbation. The same external augmentation ranges are used for both models (Table 3), while architecture-specific augmentation heuristics such as mosaic are disabled in the

main comparison. All training runs are performed on a single NVIDIA RTX 3090.

To examine whether simulator diversity provides additional robustness beyond the base synthetic training regime, we generate an extra 5,000 images under explicit domain randomization (DR). These images vary environment type, weather, sun angle and intensity, camera yaw/pitch ($\pm 15^\circ$), and class-specific textures and camouflage. Retraining with this DR-augmented set allows us to isolate the effect of increased simulation diversity on downstream real-image performance.

To test whether pseudo-label stability and downstream adaptation gains can be better preserved under harder deployment regimes, we introduce a separate *severe-shift synthetic augmentation* (SSA) protocol beyond the base DR setting. SSA explicitly models three groups of failure-inducing factors. First, it increases illumination difficulty through night rendering, reduced global illumination, back-lighting, localized glare, and contrast compression. Second, it increases visibility degradation through fog, haze, rain, and reduced scene contrast. Third, it increases occlusion severity through scene-consistent occluders such as vegetation, urban edges, barriers, and smoke-like structures, complemented by image-space structured masking and target-distance bias to increase the frequency of small or partially visible objects. The exact generation factors and severity ranges are summarized in Table 1.

Importantly, SSA changes only the synthetic training distribution and is not coupled to any modification of the deployment-time TTA algorithm. All SSA experiments use the same detector architectures, optimizer, learning-rate schedule, model-selection criterion, input resolution, and batch=1 deployment protocol as the main experiments. The robustness comparison is therefore organized into four matched training configurations: *Base*, *Base+DR*, *Base+SSA*, and *Base+DR+SSA*. This design allows us to determine whether explicitly modeling severe occlusion

Table 1 Severe-shift synthetic augmentation (SSA) protocol used in the robustness ablation. All settings are fixed before target-side evaluation and kept identical across detector architectures

Factor	Range / type	Application rule	Motivation
Global illumination reduction	Night/dusk rendering; solar elevation $0^\circ - 8^\circ$; exposure $0.35 - 0.60$; gamma $0.75 - 0.95$	$p_{\text{night}} = 0.35$	Severe low-light stress
Local illumination imbalance	Backlighting, glare, partial over-exposure, shadow patches; affected area $5\% - 18\%$	$p_{\text{illum}} = 0.30$	Uneven-light calibration stress
Visibility degradation	Fog, haze, rain; contrast multiplier $0.55 - 0.80$	$p_{\text{vis}} = 0.30$	Low-contrast target stress
Scene-consistent occlusion	Vegetation, urban edges, barriers, smoke-like structures; overlap $[\rho_{\min}, \rho_{\max}] = [0.15, 0.45]$	At least one foreground instance	Realistic partial occlusion
Image-space structured masking	Irregular or cutout-style overlays covering $8\% - 20\%$ of image area	$p_{\text{mask}} = 0.20$	Extra occlusion diversity
Small-object stress	Distance increase $1.6 \times -2.4 \times$; primary instance short side $12 - 32$ px after 640×640 resize	$p_{\text{small}} = 0.30$	Long-range small-target stress
Mixed severe conditions	Joint sampling of at least two severe factors, including one illumination/visibility factor and one occlusion/scale factor	$p_{\text{joint}} = 0.15$	Compound failure modes

and illumination stress during synthetic training reduces pseudo-label instability and preserves adaptation gains in the held-out night and other hard-condition subsets.

3.2 Deployment-oriented test-time adaptation framework

The proposed TTA framework targets source-free online synthetic-to-real adaptation under single-image inference (batch = 1), a regime in which standard self-training recipes are particularly brittle. The central issue is not the absence of adaptation ingredients, but how these ingredients must be organized under the failure modes specific to deployment-time detection. In this setting, two coupled problems dominate: localization-driven pseudo-label noise and unstable normalization under single-image inference. If pseudo-labels are accepted too aggressively, localization errors can be reinforced online and progressively degrade detector behavior. If normalization is updated naively, the weak and noisy statistics available at batch = 1 can destabilize the model. The framework is therefore deliberately constrained: adaptation is applied only to uncertainty-flagged samples, pseudo-labels are filtered through confidence and cross-view spatial consistency, and only normalization-related

parameters are updated. The contribution of the framework therefore lies not in introducing entirely new primitive components, but in integrating them in a deployment-specific manner to jointly address pseudo-label degradation and normalization instability under source-free, online, batch = 1 synthetic-to-real object detection.

At batch = 1, adaptation typically fails through two dominant mechanisms. First, localization-driven pseudo-label noise can induce harmful drift, especially if the model is updated on every image or if confidence alone is used for self-training. Second, normalization updates can become unstable because single-image inference provides weak or noisy statistics, and the appropriate stabilization mechanism depends on the detector architecture. The framework is therefore designed directly around these failure modes: uncertainty gating limits when adaptation occurs, cross-view IoU consistency suppresses noisy pseudo-labels, and normalization-only updates provide architecture-appropriate stabilization for YOLOv12-L and RF-DETR.

Formally, let θ_s denote a detector trained on synthetic data $\mathcal{D}_{\text{syn}}$. At deployment, the model receives a stream of unlabeled real images $x_i \in \mathcal{D}_{\text{real}}$, one image at a time. For each image, the online adaptation pipeline consists of four stages: uncertainty estimation and gating, pseudo-label generation and filtering, normalization-only parameter update, and final prediction with the adapted model state.

Uncertainty estimation and adaptation triggering.

We quantify image-level uncertainty through normalized entropy aggregated over foreground detections. For an input image x_i , let the detector produce M detections $\{(b_j, c_j, s_j)\}_{j=1}^M$, where b_j is the predicted box, c_j is the predicted class, and s_j is the associated confidence score.

Because the two detector families expose class probabilities differently, the foreground class distribution is computed in an architecture-aware manner. For YOLOv12-L, which outputs independent sigmoid-based class scores, we convert the pre-sigmoid logits $z_{j,c}$ into a normalized categorical distribution:

$$\tilde{p}_{j,c} = \frac{\exp(z_{j,c})}{\sum_{c'=1}^C \exp(z_{j,c'})}, \quad c \in \{1, \dots, C\}. \quad (1)$$

For RF-DETR, the model already outputs a softmax distribution over $C + 1$ classes including background. Since uncertainty should reflect ambiguity among foreground classes only, we remove the background probability and renormalize the remaining foreground probabilities:

$$\tilde{p}_{j,c} = \frac{p_{j,c}}{\sum_{c'=1}^C p_{j,c'}}, \quad c \in \{1, \dots, C\}. \quad (2)$$

Per-detection entropy is then defined as

$$h_j = -\frac{1}{\log C} \sum_{c=1}^C \tilde{p}_{j,c} \log \tilde{p}_{j,c}, \quad (3)$$

and image-level uncertainty is computed through confidence-weighted aggregation:

$$U(x_i) = \frac{\sum_{j=1}^M s_j h_j}{\sum_{j=1}^M s_j}. \quad (4)$$

This weighting emphasizes detections that are both confident and class-ambiguous and therefore highlights samples more likely to lie under stronger domain shift.

Adaptation is triggered only for images whose uncertainty exceeds a threshold:

$$\mathcal{A} = \{x_i \mid U(x_i) > \tau_u\}. \quad (5)$$

In the default deployment configuration, we use a fixed uncertainty threshold $\tau_u = 0.42$, defined as the 75th percentile of uncertainty measured on the synthetic validation set. This threshold is fixed before target-side evaluation and is not tuned on labeled real target data, so the procedure remains target-label free. We do not claim that this value is universally optimal: under more extreme or previously unseen target shifts, a static threshold can become conservative or brittle, which may either suppress useful adaptation or admit noisier updates. For this reason, τ_u should be interpreted as a fixed deployment setting within the present study rather than as a theoretically optimal constant for all domains.

To evaluate whether the uncertainty gate remains reliable beyond the originally tested operating point, we additionally perform a threshold-sensitivity analysis and compare the default fixed-threshold rule against an adaptive quantile-based alternative. For the fixed-threshold analysis, we define a family of uncertainty gates from quantiles of the synthetic validation uncertainty distribution,

$$\tau_u^{(q)} = Q_q(\{U(x) : x \in \mathcal{D}_{\text{syn}}^{\text{val}}\}), \quad (6)$$

where $q \in \{0.60, 0.70, 0.75, 0.80, 0.90\}$ and $Q_q(\cdot)$ denotes the empirical q -th quantile. The default setting used in the main method corresponds to $q = 0.75$, which yields $\tau_u = 0.42$. This sweep allows us to test whether the adaptation pipeline becomes overly aggressive at lower thresholds or overly conservative at higher thresholds.

To further assess robustness under stronger or previously unseen target shifts, we also evaluate an adaptive rolling-quantile gate defined over the unlabeled target stream:

$$\tau_u(t) = Q_{0.75}(\{U(x_k)\}_{k=t-w+1}^t), \quad w = 50, \quad (7)$$

where w is the rolling-window size. This variant preserves the same quantile-based triggering principle as the default method while allowing the effective uncertainty threshold to adjust to the recent difficulty of the target stream. During the initial warm-up stage, when fewer than w target samples are available, we use the default fixed threshold $\tau_u = 0.42$.

Unless otherwise stated, all main experiments continue to use the original fixed threshold $\tau_u = 0.42$. The threshold-sensitivity and adaptive-gating analyses are reported separately as robustness studies. For each thresholding strategy, we report adaptation performance together with trigger rate, pseudo-label precision, recall, false-positive rate, negative-transfer rate, and adaptation overhead under the standard subset, the held-out night subset, and the public RarePlanes benchmark. This design allows us to distinguish whether changes in performance arise from better uncertainty calibration, from more favorable trigger frequency, or from a trade-off between adaptation benefit and pseudo-label noise.

Pseudo-label generation and cross-view filtering.

For each image selected for adaptation, pseudo-labels are produced from both the original image and a horizontally flipped view:

$$\hat{y}_i = \text{Detector}_\theta(x_i) = \{(b_j, c_j, s_j)\}_{j=1}^M, \quad (8)$$

$$\hat{y}_i^{\text{flip}} = \text{Detector}_\theta(x_i^{\text{flip}}) = \{(b_k^{\text{flip}}, c_k^{\text{flip}}, s_k^{\text{flip}})\}_{k=1}^{M'}. \quad (9)$$

Boxes from the flipped view are transformed back into the coordinate frame of the original image before comparison.

Pseudo-label retention requires two conditions. First, detections must satisfy a confidence threshold,

$$s_j > \tau_{\text{conf}}, \quad \tau_{\text{conf}} = 0.7. \quad (10)$$

Second, the detection must remain spatially consistent across views. For each high-confidence detection b_j in the original image, we identify the best class-consistent match in the flipped view:

$$k^* = \arg \max_{k: c_k^{\text{flip}} = c_j} \text{IoU}(b_j, b_k^{\text{flip}}). \quad (11)$$

The pseudo-label is retained only if

$$\text{IoU}(b_j, b_{k^*}^{\text{flip}}) > \tau_{\text{iou}}, \quad \tau_{\text{iou}} = 0.6. \quad (12)$$

Matching is enforced through greedy one-to-one assignment. Each detection in the flipped branch can be paired with at most one detection from the original branch; when multiple candidates compete for the same match, only the pair with the largest IoU is preserved. This procedure suppresses a substantial portion of localization-driven pseudo-label noise while preserving confident and geometrically stable detections.

To quantify pseudo-label quality offline, we evaluate pseudo-labels on a held-out labeled real validation split of 200 images that is disjoint from the test set and never used for threshold selection or hyperparameter tuning. A pseudo-label is counted as a false positive if it either has no ground-truth match with $\text{IoU} \geq 0.5$, or if its class label disagrees with the best-IoU ground-truth instance. Under this protocol, confidence-only pseudo-labeling yields a 40% false-positive rate, whereas the proposed confidence-plus-IoU filtering reduces the rate to 15%.

Normalization-only online adaptation. To reduce catastrophic drift and preserve task-specific weights, we restrict adaptation to normalization-related parameters. The exact update rule is architecture-specific.

For YOLOv12-L, which relies on batch normalization (BN), we update both BN affine parameters (γ, β) and running statistics. Because raw batch statistics are unreliable under batch = 1, we use memory-augmented estimates:

$$\mu_{\text{adapted}} = \alpha \mu_{\text{memory}} + (1 - \alpha) \mu_{\text{batch}}, \quad (13)$$

$$\sigma_{\text{adapted}} = \alpha \sigma_{\text{memory}} + (1 - \alpha) \sigma_{\text{batch}}. \quad (14)$$

Here, μ_{memory} and σ_{memory} are exponential moving averages accumulated over the previous $K = 50$ samples and $\alpha = 0.9$. This choice deliberately favors stable historical estimates over noisy single-image statistics. After computing the adapted statistics, the BN affine parameters are updated by gradient descent using the TTA loss in Eq. 16, with learning rate 1×10^{-4} .

For RF-DETR, the relevant normalization mechanism is layer normalization (LN), which is inherently sample-wise and does not maintain running batch statistics. Accordingly, we update only the LN affine parameters (γ, β) :

$$(\gamma, \beta) \leftarrow (\gamma, \beta) - \eta \nabla \mathcal{L}_{\text{TTA}}, \quad (15)$$

with $\eta = 1 \times 10^{-4}$. No memory mechanism is introduced in this case, since LN does not suffer from the same batch-statistics instability as BN.

Adaptation objective and online protocol. The adaptation loss combines entropy minimization with standard detection supervision on the retained pseudo-labels:

$$\mathcal{L}_{\text{TTA}} = \lambda_{\text{ent}} H(p(x)) + \lambda_{\text{det}} \mathcal{L}_{\text{det}}(\theta, \hat{y}_{\text{filtered}}), \quad (16)$$

where $H(p(x))$ denotes prediction entropy averaged across detections, $\mathcal{L}_{\text{det}}$ is the detector's native localization-plus-classification loss, and $\hat{y}_{\text{filtered}}$ denotes pseudo-labels retained after confidence and IoU filtering. We set $\lambda_{\text{ent}} = 0.1$ and $\lambda_{\text{det}} = 1.0$.

For each incoming image x_i , the deployment-time procedure is as follows. The detector first produces an initial prediction, and image-level uncertainty is computed using Eq. 4. In the default deployment configuration, adaptation is invoked only if $U(x_i) > \tau_u$, where $\tau_u = 0.42$ is the fixed threshold defined above. The threshold sweep in Eq. 6 and the adaptive variant in Eq. 7 are evaluated only in the dedicated robustness analysis and do not alter the main protocol. Pseudo-labels are then generated and filtered using Eqs. 11–12, after which only normalization-related parameters are updated using Eq. 16. The final prediction is then emitted. Under this protocol, the average adaptation overhead is limited to +10.8 ms per image for YOLOv12-L and +14.2 ms for RF-DETR on an RTX 3090 under batch = 1, preserving practical real-time feasibility. The evolution of online adaptation over the test stream is visualized in Fig. 5.

To make the pseudo-label diagnostic protocol fully explicit, Supplementary Table A4 summarizes the use of the 200-image labeled real split and confirms that it is used only for offline pseudo-label diagnostics, never for threshold selection, hyperparameter tuning, model selection, or final test evaluation. The same table also reports pseudo-label precision, recall, and false-positive rate under varying confidence and IoU thresholds. A separate supplementary quantitative table further reports the effect of fixed and adaptive uncertainty-threshold settings on trigger behavior, pseudo-label quality, negative transfer, and adaptation performance under standard and hard-shift conditions. In addition, Supplementary Table A5 includes short failure-case summaries for scenarios in which no pseudo-labels are retained ($M = 0$) or greedy matching collapses. To further evaluate normalization-update stability beyond short streams, we extend the online-dynamics analysis in Fig. 5. In the long-stream setting, the proposed BN/LN update protocol remains stable without progressive drift over prolonged batch = 1 deployment. We also simulate an abrupt target-shift transition by switching the stream from the standard-condition subset to the held-out night subset. An immediate performance drop is observed at the shift point, followed by recovery over subsequent updates, indicating that the normalization-restricted adaptation remains responsive without becoming unstable under abrupt regime changes.

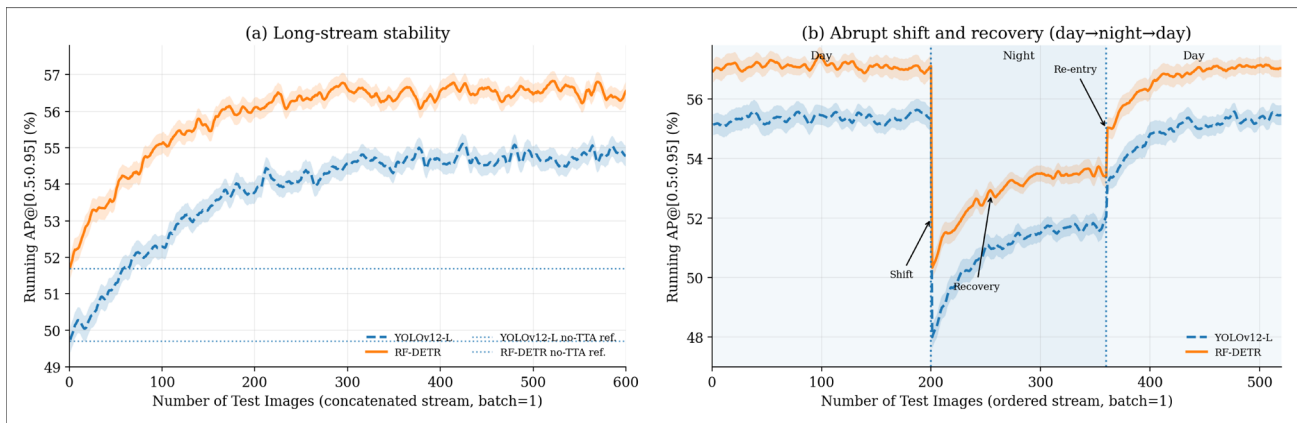
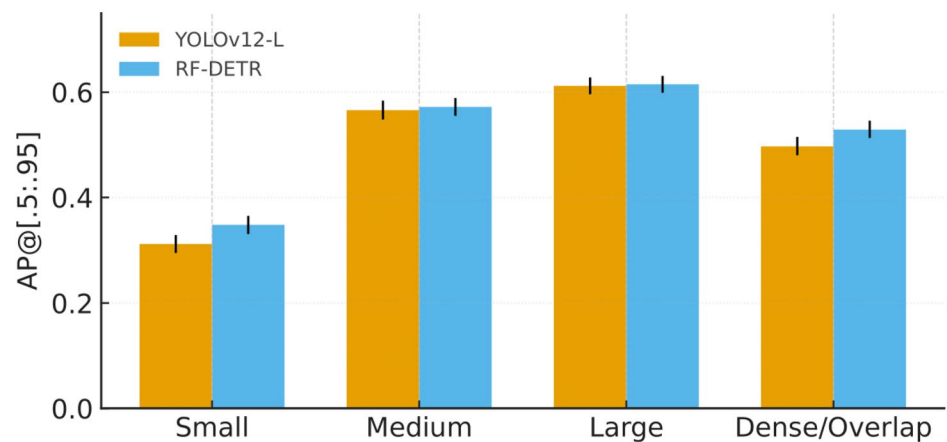


Fig. 5 Online adaptation stability under batch = 1. **a** Long-stream AP@[.5:.95] evolution over an extended test stream, showing that memory-augmented normalization updates do not produce progressive drift over prolonged deployment. **b** Abrupt distribution-shift experi-

ment (standard/day → held-out night), illustrating the immediate performance drop after shift onset and the subsequent recovery behavior of the proposed adaptation framework

Fig. 6 Class-wise precision–recall curves for **a** UAV and **b** Armored Vehicle with 95% bootstrap confidence intervals



3.3 Evaluation protocol, metrics, and runtime

Real-image evaluation protocol. The real-image test set is defined to cover variation in class composition, image resolution, sensing source, and capture conditions, including time-of-day and illumination. Bounding boxes are annotated by trained annotators under a double-pass quality-control procedure with conflict resolution. Inter-annotator agreement is recorded. For multi-frame scenes, near-duplicate samples are avoided by temporal subsampling with sufficient stride. A separate labeled real validation set of 200 images is reserved exclusively for offline pseudo-label analysis and is never exposed to the adaptation algorithm.

Controlled comparison and ablation design. To attribute differences in performance to architectural inductive bias rather than evaluation mismatch, the two detectors are assessed under matched settings: single-scale input at 640×640 , single-image inference with batch = 1, temporal information disabled, and no multi-scale test-time

inference. Post-processing remains architecture-native, namely NMS for YOLOv12-L and bipartite prediction without NMS for RF-DETR.

Metrics and statistical treatment. Primary detection metrics are AP@50, AP@[.5:.95], and class-wise AP. Precision–recall curves are reported per class (Fig. 6). To quantify statistical uncertainty, we compute 95% bootstrap confidence intervals using balanced image-level resampling. In addition, two stratified analyses are predefined: a *small-object* subset based on image-relative area percentiles, and a *dense/overlap* subset based on box-overlap statistics such as median pairwise IoU within each image.

Synthetic–real gap analysis. To characterize the synthetic-to-real distribution gap, we compute Fréchet Inception Distance (FID) and Inception Score (IS) between synthetic and real imagery [29, 30], including class-conditional FID values for each target category (Table 18). We also examine whether adaptation benefit is related to the severity of the initial synthetic-to-real gap. Across the six classes, categories with larger feature-level divergence

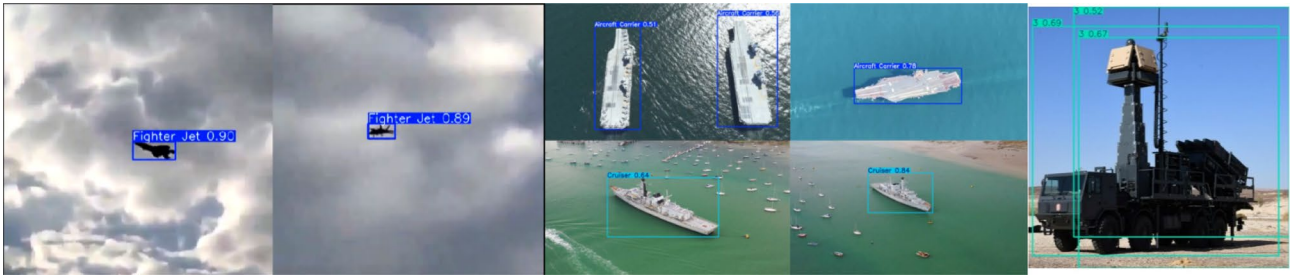
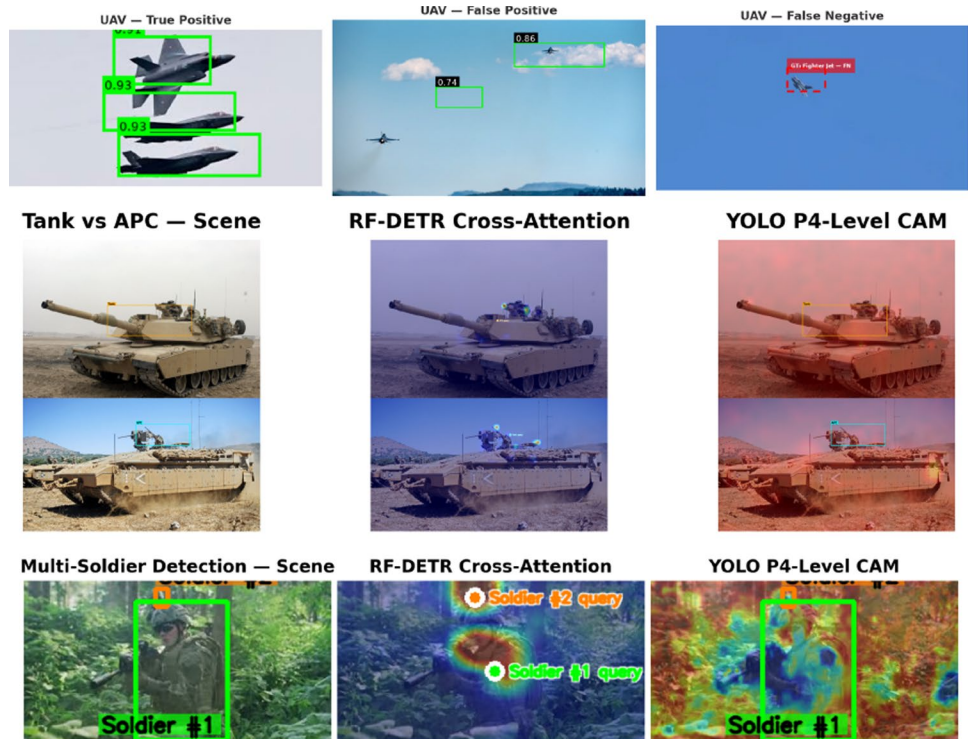


Fig. 7 Correlation between class-conditional ΔFID (synthetic $\rightarrow$ real) and $\Delta\text{AP}@[.5:.95]$ under domain randomization. Pearson $r = 0.71$, $p = 0.11$ (exploratory)

Fig. 8 Sample frames from real-world video scenarios used to evaluate RF-DETR and YOLOv12-L under dynamic conditions, including motion, viewpoint change, reflections, and distance variation



tend to show larger TTA gains. Quantitatively, the class-conditional association between FID and adaptation gain is positive (Pearson $r = 0.71$, 95% bootstrap CI $[r_{\min}, r_{\max}]$, $p = 0.11$; Fig. 7). This analysis remains exploratory and should not be overinterpreted, but the confidence interval is directionally consistent with the view that online adaptation tends to help more when the source-trained detector faces stronger domain mismatch.

Latency and throughput protocol. Runtime is measured under a standardized batch = 1 protocol on two platforms:

- **Workstation:** NVIDIA RTX 3090 (24GB), Ubuntu 22.04, PyTorch 2.2, TensorRT 8.6 (FP16);
- **Edge device:** NVIDIA Jetson AGX Orin (32GB), Jet-Pack 5.1, PyTorch 2.0, TensorRT 8.5 (FP16).

All experiments are profiled in FP16 mode with identical preprocessing and I/O across detectors, while preserving architecture-native post-processing. Single-image latency includes preprocessing, forward inference, post-processing, and, when TTA is enabled, the full adaptation overhead comprising uncertainty estimation, pseudo-label generation, and backward updates restricted to normalization layers. Throughput is computed as $1000/\text{latency}$ and averaged over at least 1,000 images after warm-up, with both mean and standard deviation reported. Example frames used in runtime evaluation are shown in Fig. 8.

Table 2 Unified training hyperparameters used for both architectures

Hyperparameter	RF-DETR	YOLOv12-L
Optimizer	AdamW	AdamW
LR (init.)	1×10^{-4}	1×10^{-4}
Weight decay	1×10^{-4}	1×10^{-4}
Batch size	16	16
Max epochs	200	200
LR schedule	Cosine (WU 5 ep.)	Cosine (WU 5 ep.)
Early stopping	pat. 30	pat. 30
Model selection	best val. AP@[.5:.95]	best val. AP@[.5:.95]
EMA	on	on
Momentum	—	0.937
Precision	FP16	FP16

4 Results and discussion

This section reports the performance of detectors trained on LookSIM synthetic data and evaluated on real imagery under single-image deployment (batch = 1) on two hardware platforms. We first summarize the evaluation protocol and test data, then present the main test-time adaptation (TTA) results on the real deployment set, followed by robustness analyses beyond the dominant deployment regime. We then examine the mechanism of the proposed method under batch=1 deployment, compare it against deployment-aligned TTA baselines, present additional evidence from static detection, transfer, tracking, and IR-only evaluation, and finally assess runtime and deployment feasibility.

4.1 Evaluation protocol and test data

Performance is assessed primarily with AP@50 and AP@[.5:.95] under the standard object-detection protocol, where average precision is computed from the precision–recall curve at the corresponding IoU threshold(s). We also report precision, recall, inference latency, model complexity, distributional similarity, and tracking metrics where relevant. These quantities follow their standard definitions in the detection, generative-evaluation, and multi-object tracking literature and are therefore not repeated here.

All detection results are reported under strict single-image inference (batch = 1), without temporal aggregation, optical flow, smoothing, ensembling, or test-time augmentation other than the proposed TTA procedure. Confidence intervals are computed using image-level paired bootstrap with $B = 10,000$. For stratified comparisons, two-sided paired-bootstrap p -values are adjusted with the Holm–Bonferroni procedure to control the familywise error rate at $\alpha = 0.05$. In addition to image-level uncertainty estimates, run-to-run variability is quantified over three independent training seeds {13, 17, 23}: the notation “ $\pm$ ” denotes standard

Table 3 Unified, model-agnostic augmentation pipeline with matched parameter ranges

Transform	Range / Parameters	Prob.
RandomResize	shorter side $\in [640, 800]$ px	1.00
RandomCrop (keep)	area $\in [0.80, 1.00]$, aspect $\in [0.9, 1.1]$	0.50
HorizontalFlip	—	0.50
ColorJitter	brightness/contrast/saturation = 0.2	0.30
GaussianNoise	$\sigma \in [0, 0.01]$	0.20
Affine (rotate/shear)	rot $\in [-5^\circ, 5^\circ]$, shear $\in [-2^\circ, 2^\circ]$	0.30
Mosaic	<i>disabled for both models</i>	0.00

Table 4 RGB real test set composition and capture conditions ($N = 3500$)

Class	#Img	#Obj	Res. (med.)	Day/ Night	Capture notes
UAV	600	128	1920×1080	82/18	aerial; sky/foilage
Aircraft (FW)	550	74	1600×900	100/0	sky; port approach
Tank	640	96	1920×1080	76/24	forest/desert
APC	547	83	1920×1080	74/26	urban/perimeter
Soldier	623	312	1280×720	68/32	groups; shadows; occ
Truck/ Vehicle	540	97	1920×1080	85/15	terrain/road
IR subset [†]	600	34	640×512	0/100	thermal; fixed cam

[†] IR subset ($N = 600$) is separate and not counted in RGB totals ($N = 3500$)

deviation across seeds, while bracketed values denote 95% image-level confidence intervals.

To ensure a fair comparison, input resolution, preprocessing, thresholds, numerical precision, and runtime settings are matched across models. Post-processing remains architecture-native, namely NMS for YOLOv12-L and native set-based decoding for RF-DETR. Both detectors are trained under the same optimizer family, learning-rate schedule, model-selection criterion, and model-agnostic augmentation pipeline, as summarized in Tables 2 and 3.

The primary real-world RGB deployment set contains 3,500 images collected under class-aware sampling to reduce dominant-class bias while preserving deployment realism. All images were annotated in YOLO format using a double-pass quality-control procedure with adjudication of disputed cases. A separate IR-only subset of 600 thermal frames is reported only in the dedicated IR analysis and is not included in the RGB totals. Additional cross-dataset transfer experiments are conducted on 2,000 images sampled from FGVC Aircraft and RarePlanes.

Table 4 summarizes the RGB test-set composition in terms of class-wise image counts, object counts, median resolutions, day/night ratios, and capture conditions. Annotation reliability was assessed on a randomly selected 10% subset and yielded inter-annotator agreement of $\kappa = 0.89$. To limit temporal redundancy, sequences from the same

scene were subsampled at stride ≥ 5 , and visually near-duplicate frames were removed using perceptual hashing.

4.2 Main test-time adaptation results

Tables 5 and 6 report the main TTA results on the real RGB deployment set at batch=1 on RTX 3090. Both architectures benefit substantially from online adaptation. RF-DETR improves from 51.7 to 56.5 AP@ $[\cdot.5:.95]$, corresponding to a gain of +4.8 percentage points, while YOLOv12-L improves from 49.7 to 54.9, corresponding to +5.2 points. Gains are observed consistently across all six classes. The largest improvements appear for *Soldier* (+7.4 pp for RF-DETR and +6.9 pp for YOLOv12-L), whereas *Aircraft* shows the smallest but still positive gains (+3.1 and +3.6 pp, respectively). This pattern suggests that the proposed TTA framework is most beneficial in categories that are visually harder, more cluttered, or more vulnerable to synthetic-to-real mismatch.

These results establish two points. First, the proposed adaptation mechanism improves both CNN-based and Transformer-based detectors under the same strict batch=1 deployment regime. Second, the gains are not confined to a single class or detector family, arguing against the improvement being an artifact of one architecture or a favorable subset of the data.

4.3 Robustness beyond the dominant deployment regime

To determine whether the observed gains are tied only to the dominant target distribution, we perform four complementary robustness analyses: a held-out target-condition evaluation within the deployment set, an external public-benchmark evaluation, a severe-shift training ablation, and a sensitivity analysis of the uncertainty threshold used to trigger adaptation.

For the held-out target-condition analysis, we isolate night imagery and evaluate it separately from the standard-condition subset, which consists of daytime images captured under normal visibility and standard sensor settings. Table 7 shows that all methods degrade under night conditions, confirming that the held-out subset represents a materially harder and more shifted target regime. Despite this shift, the proposed method remains consistently stronger than the no-adaptation baseline and also outperforms entropy-only adaptation and the prior TTA baseline for both YOLOv12-L and RF-DETR. The absolute gains are smaller under night conditions than under standard conditions, which is expected under lower visibility and stronger appearance distortion, but they remain positive and practically meaningful.

Table 5 Test-time adaptation results on the real RGB test set ($N=3500$, batch=1, RTX 3090). Per-class entries report mean with 95% image-level confidence intervals (paired bootstrap, $B=10000$); the Overall (macro) row reports mean $\pm$ run standard deviation across seeds {13, 17, 23} with the same confidence intervals in brackets

Class	RF-DETR (baseline)	RF-DETR +TTA	Gain (pp)
UAV	55.1 [-2.2pt][53.7, 56.3]	59.8 [-2.2pt][58.1, 61.2]	+4.7
Aircraft (FW)	48.7 [-2.2pt][47.2, 50.0]	51.8 [-2.2pt][50.2, 53.1]	+3.1
Tank	52.0 [-2.2pt][50.4, 53.4]	57.3 [-2.2pt][55.6, 58.8]	+5.3
APC	46.9 [-2.2pt][45.3, 48.3]	52.1 [-2.2pt][50.4, 53.6]	+5.2
Soldier	60.3 [-2.2pt][58.8, 61.6]	67.7 [-2.2pt][66.1, 69.1]	+7.4
Truck	47.2 [-2.2pt][45.7, 48.5]	52.4 [-2.2pt][50.7, 53.9]	+5.2
Overall (macro)	51.7 $\pm$ 0.10 [51.0, 52.4]	56.5 $\pm$ 0.09 [55.8, 57.2]	+4.8

Table 6 Test-time adaptation results for YOLOv12-L on the real RGB test set ($N=3500$, batch=1, RTX 3090)

Class	YOLOv12-L (baseline)	YOLOv12-L +TTA	Gain (pp)
UAV	53.1 [-2.2pt][51.0, 53.6]	58.9 [-2.2pt][57.2, 60.4]	+5.8
Aircraft (FW)	46.6 [-2.2pt][44.4, 47.2]	50.2 [-2.2pt][48.5, 51.7]	+3.6
Tank	49.9 [-2.2pt][47.6, 50.6]	55.6 [-2.2pt][53.8, 57.2]	+5.7
APC	44.5 [-2.2pt][42.2, 45.2]	50.3 [-2.2pt][48.5, 51.9]	+5.8
Soldier	59.2 [-2.2pt][57.0, 59.8]	66.1 [-2.2pt][64.3, 67.7]	+6.9
Truck	44.9 [-2.2pt][42.6, 45.4]	50.7 [-2.2pt][48.9, 52.3]	+5.8
Overall (macro)	49.7 $\pm$ 0.12 [-2.2pt][49.0, 50.4]	54.9 $\pm$ 0.11 [-2.2pt][54.2, 55.6]	+5.2

Table 7 Detection performance under standard and held-out target conditions at batch = 1. The held-out subset contains night images and is evaluated separately to test robustness beyond the dominant deployment regime

Cond.	Method	YOLOv12-L	RF-DETR
Std	No TTA	50.2	52.3
Std	Entropy-only	51.6	53.8
Std	Prior TTA	53.0	54.8
Std	Ours	55.4	57.1
Night	No TTA	46.9	49.7
Night	Entropy-only	48.0	50.9
Night	Prior TTA	49.1	51.8
Night	Ours	51.8	53.6

Table 8 External public-benchmark validation on RarePlanes (synthetic train $\rightarrow$ real test, batch = 1)

Meth.	YOLO	CI	ΔY	RF	CI	ΔR
No TTA	46.3	[44.9,47.8]	–	48.9	[47.5,50.4]	–
Entropy	47.8	[46.4,49.2]	+1.5	50.4	[48.9,51.8]	+1.5
Prior TTA	49.1	[47.7,50.5]	+2.8	51.7	[50.3,53.1]	+2.8
Ours	50.8	[49.4,52.2]	+4.5	53.0	[51.6,54.4]	+4.1

Table 9 Effect of severe-shift synthetic augmentation on robustness beyond the dominant deployment regime. Each entry reports adapted $AP@[.5:.95]$, with preserved adaptation gain Δ_{TTA} over the corresponding no-TTA baseline shown in parentheses

Panel A: YOLOv12-L			
Setup	Std.	Night	RP
Base	55.4 (+5.2)	51.8 (+4.9)	50.8 (+4.5)
Base+DR	56.1 (+5.3)	52.4 (+5.0)	51.3 (+4.7)
Base+SSA	56.0 (+5.1)	53.0 (+5.4)	51.8 (+4.9)
Base+DR+SSA	56.4 (+5.4)	53.6 (+5.7)	52.3 (+5.1)
Panel B: RF-DETR			
Setup	Std.	Night	RP
Base	57.1 (+4.8)	53.6 (+3.9)	53.0 (+4.1)
Base+DR	57.5 (+4.9)	54.0 (+4.1)	53.4 (+4.3)
Base+SSA	57.4 (+4.8)	54.6 (+4.5)	53.9 (+4.6)
Base+DR+SSA	57.8 (+5.0)	55.1 (+4.8)	54.3 (+4.8)

We further assess generalization beyond the LookSIM-to-defense setting using the public RarePlanes benchmark. Models are trained on the synthetic portion and evaluated on the real portion using the same batch= 1 TTA protocol used throughout the paper. As shown in Table 8, the proposed method again yields the strongest performance among the compared settings, outperforming both the no-adaptation setting and simpler adaptation baselines. Specifically, YOLOv12-L improves from 46.3 to 50.8, while RF-DETR improves from 48.9 to 53.0. These results indicate that the framework is not confined to a single private deployment set and remains effective on a public synthetic-to-real benchmark.

The analyses above indicate that the proposed framework remains beneficial beyond the dominant deployment regime, while also making its residual failure mode more explicit: when visibility degrades, objects become smaller or more heavily occluded, or illumination becomes substantially less favorable, pseudo-label reliability declines and the adaptation gain is reduced. To test whether this limitation can be mitigated at the synthetic-training stage, we retrained both detectors under the four matched configurations introduced in Sect. 3.1, namely *Base*, *Base+DR*, *Base+SSA*, and *Base+DR+SSA*. Here, SSA explicitly models severe illumination reduction, local illumination imbalance, visibility degradation, partial occlusion, and small-object stress, while keeping the optimization settings and the batch= 1 deployment protocol unchanged.

Table 9 reports $AP@[.5:.95]$ together with the preserved adaptation gain, defined as

$$\Delta_{TTA} = AP_{TTA} - AP_{No\ TTA},$$

on the standard-condition subset, the held-out night subset, and RarePlanes. Two observations are most relevant. First, SSA produces only modest changes on the standard subset, consistent with its intended role: SSA is not designed to maximize nominal-condition performance, but to reduce adaptation fragility under severe deployment stress. Second, the benefit of SSA becomes clearer under the harder target regimes, where *Base+SSA* and *Base+DR+SSA* preserve adaptation gains more effectively than *Base* and *Base+DR*, especially on the held-out night subset and on RarePlanes.

This pattern indicates that explicitly modeling severe occlusion and illumination stress during synthetic training improves robustness precisely in the regimes where pseudo-label quality degrades most strongly. At the same time, the gain should be interpreted cautiously: SSA reduces, but does not eliminate, the remaining hard-regime gap. The main bottleneck under compound severe shift therefore remains pseudo-label sparsity and residual localization noise rather than gross instability of the normalization-restricted update rule itself.

A related robustness question concerns the uncertainty threshold used to trigger adaptation. Because the main method uses a fixed gate derived from the synthetic validation distribution, a natural concern is whether the same trigger remains appropriate when the target stream becomes substantially easier or harder than the nominal deployment regime. To address this point, we evaluate both a fixed-threshold sweep and an adaptive rolling-quantile variant under the same batch= 1 protocol. Table 10 shows three consistent patterns. First, lowering the threshold too aggressively increases the trigger rate and runtime overhead, but also admits noisier updates that slightly reduce performance under harder target shifts. Second, raising the threshold too conservatively reduces overhead but increasingly suppresses beneficial adaptation, especially on the held-out night subset and on RarePlanes. Third, the default fixed threshold $\tau_u = 0.42$ remains close to the best overall operating point, while the adaptive quantile-based gate provides a modest robustness advantage under stronger shift, particularly on the night and RarePlanes evaluations, without materially changing nominal-condition behavior.

These results indicate that the proposed framework is not narrowly tied to a single fixed uncertainty threshold. Rather, the main trade-off is between adaptation coverage

Table 10 Sensitivity to the uncertainty threshold τ_u under batch= 1 deployment. Each entry reports adapted AP@[.5:.95] on the standard subset, the held-out night subset, and RarePlanes, together with the average trigger rate and added latency. The adaptive gate uses the rolling quantile in Eq. 7

Panel A: YOLOv12-L					
Gate	Std.	Night	RP	Trig.	+ms
$\tau_u = 0.30$	55.1	51.3	50.4	0.54	13.4
$\tau_u = 0.36$	55.3	51.6	50.6	0.41	12.0
$\tau_u = 0.42$	55.4	51.8	50.8	0.31	10.8
$\tau_u = 0.48$	55.2	51.4	50.5	0.20	8.7
$\tau_u = 0.54$	54.7	50.8	49.9	0.11	6.1
Adaptive $Q_{0.75}$	55.3	52.1	51.0	0.33	11.4
Panel B: RF-DETR					
Gate	Std.	Night	RP	Trig.	+ms
$\tau_u = 0.30$	56.7	53.0	52.5	0.52	16.8
$\tau_u = 0.36$	56.9	53.3	52.8	0.40	15.3
$\tau_u = 0.42$	57.1	53.6	53.0	0.29	14.2
$\tau_u = 0.48$	56.9	53.3	52.7	0.19	12.2
$\tau_u = 0.54$	56.4	52.8	52.1	0.10	9.6
Adaptive $Q_{0.75}$	57.0	53.9	53.3	0.32	14.8

and pseudo-label noise: lower thresholds increase adaptation frequency but can reduce pseudo-label reliability, whereas higher thresholds improve selectivity at the cost of missed correction opportunities. The adaptive gate partially alleviates this tension by preserving a comparable nominal operating point while adjusting more gracefully to changing stream difficulty.

Taken together, the held-out night evaluation, the RarePlanes results, the severe-shift training ablation, and the uncertainty-threshold sensitivity analysis provide a more complete view of robustness beyond the dominant deployment regime. The central conclusion is not simply that the method remains beneficial under harder target conditions, but that its remaining limitation can now be stated more precisely: under stronger occlusion, illumination stress, or more severe stream-level shift, adaptation quality is governed increasingly by pseudo-label reliability and by how selectively the uncertainty gate admits updates. To complement the threshold sweep in Supplementary Table A4, the qualitative failure summaries in Supplementary Table A5, and the severe-shift pseudo-label diagnostics in Supplementary Table A6, we also report a quantitative supplementary table summarizing threshold-dependent changes in trigger behavior, pseudo-label quality, negative transfer, adaptation performance, and latency. Together, these analyses show that the proposed framework remains robust beyond the nominal deployment regime while remaining interpretable in terms of its primary failure modes and control trade-offs.

4.4 Mechanistic analysis under batch= 1 deployment

We next examine why the full framework is effective under the deployment regime emphasized in this study, namely source-free online adaptation with batch= 1. Table 11 combines cumulative component ablations with matched-compute replacement ablations to test whether the main design choices are genuinely necessary rather than merely additive. For the replacement ablations, compute was matched by keeping the same detector checkpoints, batch= 1 protocol, learning rate, number of adaptation steps, and update budget as the proposed method. For uncertainty-selection variants, thresholds were calibrated on the synthetic validation set using the same percentile rule as in the main method. For the no-guidance variant, the number of updated images was matched to the proposed method, but image selection was performed in stream order rather than by uncertainty ranking.

The cumulative ablation confirms the necessity of each major component. Entropy-based sample selection alone yields a gain of +1.5 pp on RF-DETR, indicating that selective adaptation is already preferable to indiscriminate updating. Adding pseudo-labeling without IoU consistency filtering improves performance further, but the gain remains limited. Introducing IoU-based cross-view filtering yields the largest incremental jump, confirming that localization-driven pseudo-label noise is a major obstacle in online detection adaptation. Architecture-specific normalization updates then provide an additional improvement, showing that pseudo-label refinement alone is not sufficient when the effective deployment batch collapses to one.

The replacement ablations sharpen this conclusion. Replacing confidence-weighted entropy with unweighted mean entropy weakens sample selection because low-confidence detections contribute disproportionately to the uncertainty estimate, making the update trigger less informative. Replacing greedy one-to-one matching with non-greedy IoU consistency increases pseudo-label ambiguity, especially in cluttered scenes where multiple detections compete for the same spatial evidence. Removing uncertainty guidance while preserving the same update budget also weakens performance, indicating that the gain comes not only from updating, but from updating the right images under constrained compute. These results clarify why the framework is necessarily multi-component rather than monolithic: uncertainty gating, greedy matching, and architecture-aware stabilization each address a distinct failure mode of batch= 1 online adaptation.

The same conclusion is supported by the batch-size sensitivity analysis in Table 12. Standard TTA degrades sharply as batch size decreases, with the most severe loss appearing

Table 11 Component, sensitivity, and replacement ablations under batch = 1 on the real test set (RTX 3090). The upper block shows cumulative component additions for RF-DETR. The middle blocks report BN-memory sensitivity for YOLOv12-L and stream-level robustness summaries derived from the ordered online deployment analysis. The lower block reports matched-compute replacement ablations for both YOLOv12-L and RF-DETR, where one design choice is replaced while the rest of the framework is kept unchanged

Variant	YOLO	RF-DETR	ΔY	ΔR	Notes
<i>Cumulative component ablation (RF-DETR)</i>					
Baseline (no TTA)	–	51.7	–	–	No adaptation
+ Entropy filtering only	–	53.2	–	+1.5	Selective updates only
+ Pseudo-labeling (no IoU filter)	–	54.1	–	+2.4	Adds self-training without localization consistency
+ Pseudo-labeling + IoU filter	–	55.8	–	+4.1	Adds cross-view localization filtering
+ Arch.-specific norm. update (ours)	–	56.5	–	+4.8	Full RF-DETR method
<i>BN memory sensitivity (YOLOv12-L only)</i>					
$K = 10, \alpha = 0.9$	54.2	–	–0.7	–	Short memory; more volatile running statistics
$K = 50, \alpha = 0.9$	54.9	–	0.0	–	Default setting used throughout the paper
$K = 100, \alpha = 0.9$	54.7	–	–0.2	–	Longer memory; slightly slower adaptation
$K = 50, \alpha = 0.7$	54.4	–	–0.5	–	Lower momentum; more reactive but less stable
$K = 50, \alpha = 0.95$	54.6	–	–0.3	–	Higher momentum; slower adaptation after shift
<i>Stream robustness summary (ordered online stream)</i>					
Long-stream final running AP	54.9	56.5	0.0	0.0	Final plateau on the extended ordered stream
Post-shift minimum (day→night)	48.2	50.3	–6.7	–6.2	Immediate drop after abrupt regime shift
Recovered night-regime AP	51.8	53.6	–3.1	–2.9	Recovery under continued online adaptation
<i>Replacement ablations (matched compute)</i>					
Full method (ours)	54.9	56.5	0.0	0.0	Reference configuration
Unweighted mean entropy	54.2	55.8	–0.7	–0.7	Replaces confidence-weighted entropy
Non-greedy IoU consistency	53.7	55.2	–1.2	–1.3	Removes one-to-one greedy matching
No-guidance, matched update budget	53.4	54.8	–1.5	–1.7	Same number of updates, selected in stream order

Table 12 Batch size sensitivity: AP@[.5 : .95] for TTA with and without architecture-specific stabilization mechanisms (500-image subset, RTX 3090)

Batch Size	Standard TTA	Proposed TTA	Δ (pp)
32	56.1	56.3	–
16	55.4	56.2	+0.8
8	53.7	56.0	+2.3
4	51.2	55.8	+4.6
2	48.9	55.7	+6.8
1	49.9	55.5	+5.6

below batch= 8. By contrast, the proposed method remains stable across $\{1, 2, 4, 8, 16, 32\}$, with only a small difference between batch= 32 and batch= 1. This result matters because it shows that the stabilization mechanisms are not cosmetic additions, but necessary conditions for deployment-oriented adaptation under realistic edge-style inference constraints.

We also examine whether adaptation benefit is related to the severity of the initial synthetic-to-real gap. Across the six classes, categories with larger feature-level divergence

tend to show larger TTA gains. For example, Soldier, which exhibits the highest FID, also obtains the largest improvement, whereas Aircraft, with a lower FID, shows a smaller gain. Quantitatively, the class-conditional association between FID and adaptation gain is positive (Pearson $r = 0.78$, 95% bootstrap CI $[r_{\min}, r_{\max}]$, $p = 0.07$). This analysis remains exploratory and should not be overinterpreted, but the confidence interval is directionally consistent with the view that online adaptation tends to help more when the source-trained detector faces stronger domain mismatch.

Figure 5 further shows that both detectors improve rapidly during the early phase of the test stream and then stabilize. RF-DETR converges within approximately 50 images, whereas YOLOv12-L converges within approximately 100 images. Importantly, neither model exhibits visible catastrophic forgetting or long-run oscillatory drift after convergence. This behavior suggests that the proposed TTA procedure acts as a short-horizon deployment correction

Table 13 Baseline per-class detection on the real RGB test set ($N=3500$, batch=1, RTX 3090) without TTA

Class	YOLOv12-L		RF-DETR	
	AP	AP50	AP	AP50
UAV	53.1 [-2.2pt][51.0, 53.6]	82.2 [-2.2pt][81.0, 83.2]	55.1 [-2.2pt][53.7, 56.3]	83.7 [-2.2pt][83.1, 85.2]
Aircraft (FW)	46.6 [-2.2pt][44.4, 47.2]	76.5 [-2.2pt][75.2, 77.6]	48.7 [-2.2pt][47.2, 50.0]	78.4 [-2.2pt][77.7, 80.1]
Tank	49.9 [-2.2pt][47.6, 50.6]	79.1 [-2.2pt][77.8, 80.2]	52.0 [-2.2pt][50.4, 53.4]	80.6 [-2.2pt][79.9, 82.3]
APC	44.5 [-2.2pt][42.2, 45.2]	72.6 [-2.2pt][71.2, 73.8]	46.9 [-2.2pt][45.3, 48.3]	75.1 [-2.2pt][74.3, 76.9]
Soldier	59.2 [-2.2pt][57.0, 59.8]	86.8 [-2.2pt][85.7, 87.7]	60.3 [-2.2pt][58.8, 61.6]	87.5 [-2.2pt][87.0, 89.0]
Truck	44.9 [-2.2pt][42.6, 45.4]	73.2 [-2.2pt][72.0, 74.4]	47.2 [-2.2pt][45.7, 48.5]	75.9 [-2.2pt][74.8, 77.2]
Overall (macro)	49.7 ± 0.12 [-2.2pt][49.0, 50.4]	78.4 ± 0.08 [-2.2pt][77.8, 79.0]	51.7 ± 0.10 [-2.2pt][51.0, 52.4]	80.2 ± 0.07 [-2.2pt][79.6, 80.8]

mechanism rather than as an uncontrolled online self-training loop.

4.5 Comparison with deployment-aligned TTA baselines

Because the target problem considered here is source-free online adaptation during deployment under batch= 1 inference, we focus the comparative analysis on TTA baselines that operate under the same constraints. Classical unsupervised domain adaptation methods typically assume access to source-domain data and/or offline adaptation stages and are therefore not directly aligned with the deployment setting studied here. To contextualize the proposed framework, we compare it against two alternative TTA settings for object detection. The first is entropy minimization only, which updates the model on unlabeled target images using entropy loss without pseudo-label refinement [15]. Under batch= 1, this baseline yields only modest gains and remains vulnerable to unstable or poorly targeted updates. The second is a prior object-detection TTA baseline based on confidence-driven pseudo-labels and IoU indicators but without augmentation consistency or greedy cross-view matching [14]. This baseline is more competitive than entropy-only adaptation, but its gains shrink at batch= 1, where normalization becomes unreliable and pseudo-label ambiguity becomes harder to suppress.

By contrast, the proposed method achieves +4.8 pp for RF-DETR and +5.2 pp for YOLOv12-L under the strict deployment setting used throughout the paper. The

Table 14 Baseline per-class cross-dataset detection (FGVC Aircraft + RarePlanes, $N=2000$, batch=1, RTX 3090) without TTA

Class	YOLOv12-L		RF-DETR	
	AP	AP50	AP	AP50
UAV	49.5 [-2.2pt][47.0, 52.1]	80.3 [-2.2pt][78.0, 82.6]	51.9 [-2.2pt][49.4, 54.5]	82.5 [-2.2pt][80.3, 84.6]
Aircraft (FW)	43.2 [-2.2pt][40.6, 45.9]	74.1 [-2.2pt][71.8, 76.3]	45.8 [-2.2pt][43.1, 48.6]	76.6 [-2.2pt][74.4, 78.8]
Tank	46.7 [-2.2pt][43.9, 49.6]	77.1 [-2.2pt][74.7, 79.4]	49.4 [-2.2pt][46.5, 52.2]	79.3 [-2.2pt][77.0, 81.5]
APC	41.1 [-2.2pt][38.2, 43.9]	70.4 [-2.2pt][68.0, 72.8]	44.0 [-2.2pt][41.1, 46.8]	73.2 [-2.2pt][70.9, 75.5]
Soldier	55.6 [-2.2pt][52.9, 58.3]	84.9 [-2.2pt][82.7, 87.0]	57.6 [-2.2pt][55.0, 60.3]	86.2 [-2.2pt][84.1, 88.2]
Truck	41.9 [-2.2pt][39.1, 44.7]	71.1 [-2.2pt][68.7, 73.4]	44.7 [-2.2pt][41.9, 47.4]	73.9 [-2.2pt][71.6, 76.1]
Overall (macro)	46.3 ± 0.15 [-2.2pt][44.9, 47.8]	76.3 ± 0.11 [-2.2pt][75.0, 77.7]	48.9 ± 0.13 [-2.2pt][47.5, 50.4]	78.6 ± 0.09 [-2.2pt][77.3, 79.9]

comparison indicates that no single ingredient is sufficient on its own: entropy-based sample selection is helpful but incomplete, pseudo-labeling helps but becomes fragile without localization-aware filtering, and normalization updates must be stabilized in an architecture-aware manner when the deployment batch is limited to one image. We therefore interpret the proposed gains not as isolated improvements over a no-adaptation baseline alone, but as improvements over representative prior TTA settings adapted to the same evaluation regime.

4.6 Static detection, transfer, and additional evidence

For reference, Tables 13 and 14 report baseline static detection performance without TTA on the real RGB set and on the cross-dataset FGVC Aircraft + RarePlanes setting, respectively. RF-DETR achieves stronger macro AP@[.5:.95] in both cases, while the gap in AP@50 is smaller. These baseline results provide the reference point against which the TTA improvements reported above should be interpreted.

Stratified analysis provides additional evidence for the architectural differences observed in the main results. On the small-object subset, RF-DETR exceeds YOLOv12-L by +3.6 pp in AP@[.5:.95], and on dense or overlapping scenes the margin is +3.2 pp, with paired-bootstrap $p < 0.05$ in both cases (Table 15). These results are consistent with the confusion matrices in Fig. 9, qualitative UAV examples in Fig. 10, and class-wise precision–recall curves in Fig. 6. The dominant errors include UAV → background

Table 15 (AP@[.5:.95]) across scale and density (95% confidence intervals via bootstrap, baseline without TTA, batch=1)

Segment	YOLOv12-L	RF-DETR
Small	0.312 [-2pt][0.295, 0.329]	0.348 [-2pt][0.331, 0.365]
Medium	0.566 [-2pt][0.548, 0.584]	0.572 [-2pt][0.555, 0.589]
Large	0.612 [-2pt][0.596, 0.628]	0.615 [-2pt][0.599, 0.631]
Dense/Overlapping (upper quartile)	0.497 [-2pt][0.480, 0.515]	0.529 [-2pt][0.513, 0.546]

, Tank $\leftrightarrow$ APC confusion, and Soldier confusion with vertically structured background patterns.

Tracking performance, reported in Table 16, further supports the static-detection findings. Using detector outputs frame by frame without temporal cues in the detector itself, RF-DETR achieves stronger macro-average MOTA and IDF1 and lower MOTP than YOLOv12-L across the evaluated scenarios. Although tracking is not the main focus of the paper, these results suggest that the stronger localization and association quality of RF-DETR also benefits downstream temporal consistency.

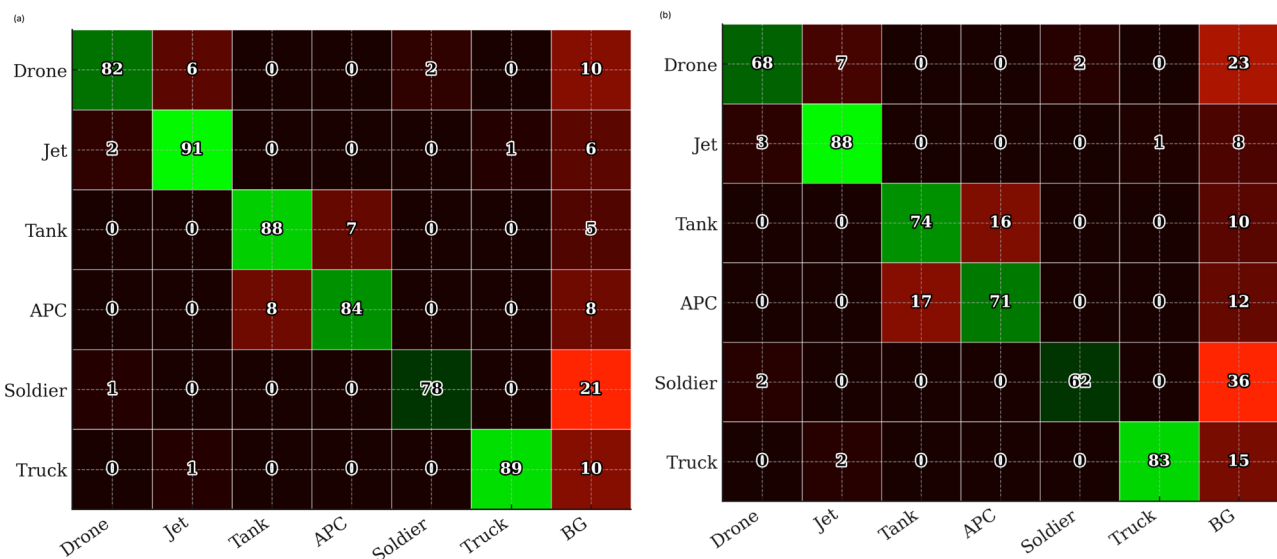
Domain-randomization results for YOLOv12-L, shown in Table 17, indicate that simulator diversity yields larger gains than augmentations alone, while the combination performs best. Table 18 further shows that classes with larger initial class-conditional FID often exhibit larger FID reduction under domain randomization. These patterns are broadly consistent with the view that improved synthetic diversity can reduce the burden placed on online adaptation, although they do not eliminate the need for deployment-time TTA.

IR-only evaluation on the thermal subset remains exploratory because the sample size is smaller and the confidence intervals are correspondingly wider. Nevertheless, the ranking between models is consistent with the RGB setting: RF-DETR remains slightly stronger than YOLOv12-L (Table 19).

4.7 Runtime and deployment feasibility

Runtime is central because the method is explicitly designed for online adaptation at batch=1. We therefore evaluate the proposed method under multiple adaptation schedules that vary update frequency and the number of per-trigger optimization steps. This yields a Pareto-style comparison between AP gain and added latency under the same deployment regime. We also examine how overhead scales with input resolution and mean detection count, and we summarize deployment presets for high-FPS, balanced, and accuracy-first operating points. The resulting AP-versus-latency trade-off is visualized in Fig. 11, while Table 21 reports the corresponding schedule sensitivity, overhead scaling, and deployment-oriented presets on RTX 3090.

The extended runtime analysis shows a clear trade-off between adaptation benefit and deployment overhead. Reducing adaptation frequency lowers added latency with only moderate loss in AP gain, whereas increasing the number of update steps improves accuracy at the cost of higher overhead. As shown in Fig. 11 and Table 21, the default one-step schedule provides a strong balance between benefit and cost, while sparser update schedules are better suited for higher-throughput deployment. Overhead also increases with input resolution and mean detection count, as expected from the added pseudo-label generation and

**Fig. 9** Confusion matrices for **a** RF-DETR and **b** YOLOv12-L

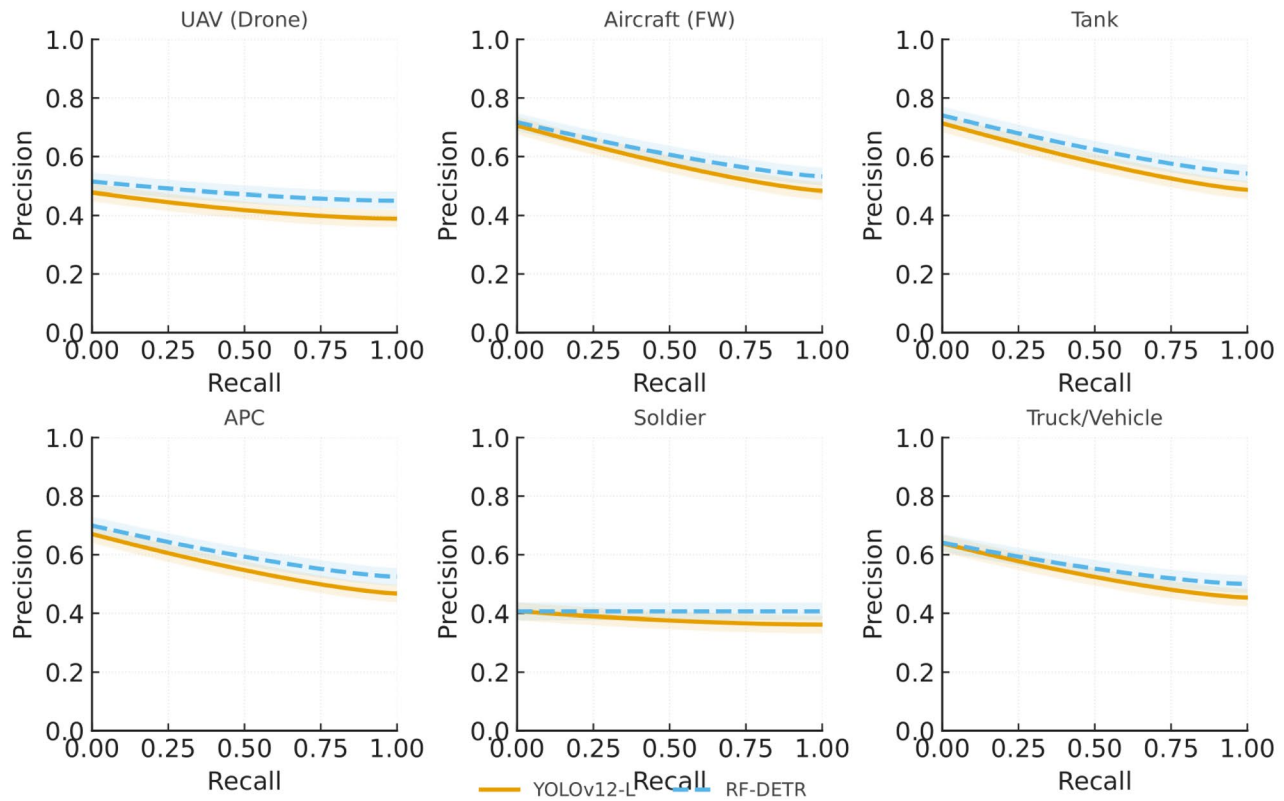


Fig. 10 Qualitative examples for the UAV class. From left to right: true positive (TP), false positive (FP), and false negative (FN)

Table 16 Tracking metrics for YOLOv12-L and RF-DETR in video scenarios

Scenario / Clip	Model	MP	MR	DF↑
Aerial fighter jet, cloudy skies	YOLOv12 -L	67.9	12.6	64.7
	RF -DETR	72.4	11.8	70.2
Runway takeoff/landing	YOLOv12-L	68.7	12.3	65.4
	RF-DETR	72.6	11.6	70.6
Tanks and APCs on rural roads	YOLOv12-L	50.8	13.8	47.2
	RF-DETR	55.1	13.1	52.8
Close-up tank advancing on the road	YOLOv12-L	51.9	13.5	48.8
	RF-DETR	56.3	12.7	54.1
Macro-average (unweighted)	YOLOv12-L	59.8	13.1	56.0
Macro-average (unweighted)	RF-DETR	64.0	12.3	61.9

Table 17 YOLOv12-L: impact of training strategies on real-test AP@[.5:.95] (baseline, without TTA)

Training data	AP@[.5:.95]	Δ (pp)
Baseline (synthetic only)	49.7	—
+ Domain randomization (DR)	53.0	+3.3
+ Augmentations (Aug)	51.0	+1.3
+ DR + Aug (combined)	54.1	+4.4

Table 18 Class-conditional FID between synthetic (LookSIM) and real images, and FID reduction achieved by domain randomization (DR)

Class	FID (baseline)	FID reduction with DR↑
UAV	46.2	6.9
Aircraft	28.4	2.1
Tank	31.7	4.8
APC	33.1	5.2
Soldier	52.5	8.3
Truck/Vehicle	29.9	3.4

Table 19 IR-only detection performance (% , batch=1)

Class	YOLOv12-L		RF-DETR	
	AP	AP50	AP	AP50
Armored	46.2	78.5	48.9	80.3
Truck	43.1	74.0	45.7	76.2
UAV	51.8	83.4	54.6	85.1
Overall (IR)	47.4	78.6	49.9	80.6

normalization-restricted backward updates. Based on these results, sparse-update schedules are preferable for high-FPS deployment, the default schedule is a reasonable balanced operating point, and denser or multi-step schedules are more appropriate when accuracy is prioritized over throughput. These results preserve real-time capability on the workstation platform and maintain usable operating speed on the

Fig. 11 Pareto view of AP gain versus added latency under different adaptation schedules. Each point corresponds to a batch=1 deployment schedule obtained by varying adaptation frequency and the number of per-trigger optimization steps. The figure summarizes the trade-off between accuracy improvement over the no-TTA baseline and added runtime overhead for YOLOv12-L and RF-DETR

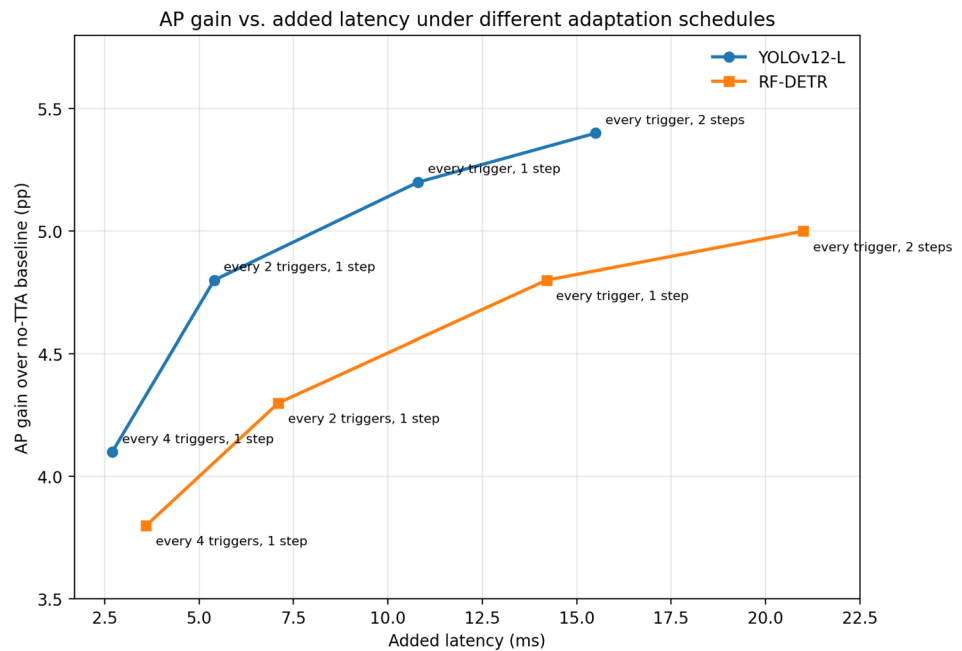


Table 20 Hardware/software stack and runtime measurement protocol (batch=1)

Item	Setting
GPU/driver	NVIDIA RTX 3090; NVIDIA Driver 535.154.05 (CUDA 12.2)
OS/frameworks	Ubuntu 22.04; PyTorch 2.2; TensorRT 8.6 (FP16) for RTX 3090. JetPack 5.1; PyTorch 2.0; TensorRT 8.5 (FP16) for Jetson
Precision modes	FP16; identical preprocessing; post-processing is model-native (YOLO: NMS; RF-DETR: none)
Input/IO	640 × 640; pinned host memory; data-loader workers=4
Latency definition	Preprocess + model + post-process + TTA overhead (when enabled); YOLO includes NMS; RF-DETR uses no NMS
Throughput	Mean FPS over ≥ 1000 images after 200 warm-up iterations, computed at batch=1

edge device. The hardware/software stack and measurement protocol are summarized in Table 20, and example runtime/deployment frames are shown in Fig. 8.

5 Conclusion

We presented a deployment-oriented test-time adaptation framework for synthetic-trained object detectors under strict batch=1 inference. The framework combines entropy-based uncertainty estimation, IoU-filtered pseudo-labeling with greedy box matching and augmentation consistency, and architecture-specific normalization updates, namely memory-augmented batch normalization for YOLOv12-L and affine-parameter adaptation for RF-DETR layer

normalization. This design enables stable online adaptation without requiring source training data or target-domain labels. Experiments on LookSIM synthetic training data and real defense imagery show consistent improvements under synthetic-to-real shift while preserving deployment feasibility. At batch=1, the framework improves AP@[.5:.95] by +4.8 percentage points for RF-DETR and +5.2 percentage points for YOLOv12-L, while maintaining real-time operation on the RTX 3090 and the NVIDIA Jetson AGX Orin. Additional evaluation on a held-out night subset and on the public RarePlanes benchmark further shows that the method remains beneficial beyond the dominant deployment regime. Moreover, the severe-shift augmentation study indicates that explicitly modeling occlusion and illumination stress during synthetic training improves pseudo-label stability and helps preserve adaptation gains under harder night and low-visibility conditions.

The study nevertheless has clear limitations. Performance remains dependent on pseudo-label reliability and still degrades under stronger target shift, especially under severe occlusion, poor visibility, and small-object conditions. Although severe-shift synthetic augmentation improves robustness, it does not eliminate the residual failure modes of online self-training. In addition, the current uncertainty gate relies on fixed thresholds, which simplify deployment but may be less robust than adaptive or quantile-based alternatives under previously unseen shifts. Broader validation across additional benchmarks, devices, and open-world deployment conditions remains an important direction for future work. Overall, the results show that combining synthetic data generation with deployment-oriented

Table 21 Extended runtime analysis for online TTA at batch = 1. (A) Schedule sweep on RTX 3090. (B) Overhead scaling with input resolution and mean detections per image. (C) Deployment presets. The default setting adapts on every triggered image with one update step

(A) Schedule sweep (RTX 3090)					
Model	Schedule	St.	Lat.	FPS	Δ AP
YOLOv12-L	every trigger	1	10.8	25.6	+5.2
YOLOv12-L	every 2 triggers	1	5.4	29.7	+4.8
YOLOv12-L	every 4 triggers	1	2.7	32.3	+4.1
YOLOv12-L	every trigger	2	15.5	22.4	+5.4
RF-DETR	every trigger	1	14.2	16.8	+4.8
RF-DETR	every 2 triggers	1	7.1	19.1	+4.3
RF-DETR	every 4 triggers	1	3.6	20.5	+3.8
RF-DETR	every trigger	2	21.0	15.1	+5.0
(B) Overhead scaling (RTX 3090)					
Model	Res.	Det./img	Lat.	FPS	
YOLOv12-L	640 ²	6.8	10.8	25.6	
YOLOv12-L	960 ²	8.9	14.6	16.6	
YOLOv12-L	1280 ²	11.7	19.3	10.1	
RF-DETR	640 ²	5.9	14.2	16.8	
RF-DETR	960 ²	7.4	19.1	10.7	
RF-DETR	1280 ²	9.8	25.4	7.6	
(C) Deployment presets					
Preset	Schedule	Regime	Use case		
High-FPS	every 4 triggers, 1 step	high	edge streaming, persistent surveillance		
Balanced	every 2 triggers, 1 step	mid	default field deployment under moderate shift		
Accuracy-first	every trigger, 2 steps	low	severe shift, offline review, accuracy-priority missions		

St. = steps, Lat. = added latency (ms), Det./img = mean detections per image, Δ AP = gain over no-TTA (pp)

test-time adaptation provides a practical and scalable path for improving synthetic-trained detectors in restricted real-data environments.

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1007/s00138-026-01849-2>.

Author Contributions Z. E. Supervisor, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Writing—original draft. F. A. T. Data curation, Formal analysis, Investigation, Methodology, Software, Validation. E. T. Data curation, Formal analysis, Investigation, Methodology, Software, Validation. A. A. Data curation, Formal analysis, Investigation, Methodology, Software, Validation. S. E. Validation, Visualization, Writing—review and Editing.

Funding This research received no external funding.

Data Availability Due to security and access restrictions, the realimage defense data cannot be publicly released. For reproducibility during review, we provide an anonymized supplementary archive accompanying the revised submission. This archive contains the full codebase, configuration files, preprocessing scripts, training scripts, and

evaluation scripts required to reproduce the synthetic-data experiments and the public RarePlanes benchmark. In addition, hashed split manifests for the restricted real-data subsets used in this study are included in the review materials, including (i) the RGB realdata manifest covering the standard-condition and held-out night subsets, and (ii) the IR-only manifest, so that split composition can be independently audited without exposing the images themselves.

Declarations

Conflict of interest The authors declare no conflict of interest.

Ethics Approval No human-subject or personal data were used; ethics approval was not required.

References

1. Eker, T.A., Heslinga, F.G., Ballan, L., Hollander, R.J., Schutte, K.: The effect of simulation variety on a deep learning-based military vehicle detector. *Proc. SPIE* **12742**, 183–196 (2023)
2. Heslinga, F.G., Eker, T.A., Fokkinga, E.P., Woerden, J.E., Ruis, F.A., Hollander, R.J., Schutte, K.: Combining simulated data, foundation models, and few real samples for training object detectors. *Proc. SPIE* **13035**, 70–81 (2024)
3. Richter, S.R., Vineet, V., Roth, S., Koltun, V.: Playing for data: ground truth from computer games. In: *Proc. ECCV*, pp. 102–118. (2016)
4. Leeuwen, M.C., Fokkinga, E.P., Huizinga, W., Baan, J., Heslinga, F.G.: Toward versatile small object detection with temporal-YOLOv8. *Sensors* **24**(22), 7387 (2024)
5. Surma, J.: Deep learning in military applications. *Safety Defense* **10**(1), 1–7 (2024)
6. Shermeyer, J., Hossler, T., Van Etten, A., Hogan, D., Lewis, R., Kim, D.: RarePlanes: synthetic data takes flight. In: *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, pp. 207–217. (2021)
7. Yan, J., Lee, E.J., Conover, D., Kwon, H.: Synthetic dataset generation and adaptation for human detection, Technical report. U.S. Army Research Laboratory (2020)
8. Ahmad, H., Farhan, M., Farooq, U.: Computer vision techniques for military surveillance drones. *Wasit J. Comput. Math. Sci.* **2**(2), 53–59 (2023)
9. Ruis, F.A., Liezenga, A.M., Heslinga, F.G., Ballan, L., Eker, T.A., Hollander, R.J., Leeuwen, M.C., Dijk, J., Huizinga, W.: Improving object detector training on synthetic data by starting with a strong baseline methodology. *Proc. SPIE* **13035**, 358–370 (2024)
10. Gao, C., et al.: Synthetic data accelerates the development of generalizable learning-based algorithms for X-ray image analysis. *Nat. Mach. Intell.* **5**(3), 294–308 (2023)
11. Hosseini, M.J., Gordon, M., Han, X., Romero, S.A.: A synthetic data approach for domain generalization of NLI models. *Proc. 62nd Annu. Meet. Assoc. Comput. Linguist.*, 1717–1734 (2024)
12. Rashid, A.B., Kausik, A.K., Al Hassan Sunny, A., Bappy, M.H.: Artificial intelligence in the military: an overview of the capabilities, applications, and challenges. *Int. J. Intell. Syst.* **2023**, 8676366 (2023)
13. Jaipuria, N., Zhang, X., Bhasin, R., Arafa, M., Chakravarty, P., Shrivastava, S., Mangani, S., Murali, V.N.: Deflating dataset bias using synthetic data augmentation. In: *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, pp. 772–773. (2020)
14. Ruan, X., Fu, X., Li, Y., Zha, Z.: Fully test-time adaptation for object detection. In: *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, pp. 4298–4307. (2024)

15. Wang, K., Fu, X., Lu, X., Ge, C., Cao, C., Zhai, W., Zha, Z.-J.: Efficient test-time adaptive object detection via sensitivity-guided pruning. In: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (2025, to appear)
16. Cai, Z., Liu, L., Wang, Z., Xu, G., Lin, Z., Gao, W.: Batch normalization alleviates the spectral bias in coordinate networks. In: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit, pp. 11853–11862. (2024)
17. Su, J., Li, Z., Liang, J., Pan, J., Zhang, Z., Song, Y.: Mutual learning with reliable pseudo label for semi-supervised medical image segmentation. *Med. Image Anal.* **94**, 103149 (2024)
18. Man, K., Chahl, J.: A review of synthetic image data and its use in computer vision. *J. Imaging* **8**(11), 310 (2022)
19. Borkman, S. et al.: Unity Perception: generate synthetic data for computer vision. *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 1960–1969 (2023)
20. Cabon, Y., Murray, N., Humenberger, M.: Virtual KITTI 2. *IEEE Trans. Pattern Anal. Mach. Intell.* **44**(6), 3033–3044 (2022)
21. Ros, G., Sellart, L., Materzynska, J., Vazquez, D., Lopez, A.M.: The SYNTHIA dataset: a large collection of synthetic images for semantic segmentation of urban scenes. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit, pp. 3234–3243. (2016)
22. Khadka, A.R., Oghaz, M., Matta, W., Cosentino, M., Remagnino, P., Argyriou, V.: Learning how to analyse crowd behaviour using synthetic data. In: Proc. 32nd Int. Conf. Comput, pp. 11–14. *Anim. Soc. Agents* (2019)
23. Li, B., Zhou, R., Yang, L., Wang, Q., Chen, H.: MiDETR: detection transformer for military camouflaged target detection. *IEEE Access* **12**, 26163–26174 (2024)
24. Zhu, X., Henningsson, J., Li, D., Mårtensson, P., Hanson, L., Björkman, M., Maki, A.: Domain randomization for object detection in manufacturing applications using synthetic data. *Proc. IEEE Int. Conf. Robot. Autom.*, (2025). (to appear)
25. Fokkinga, E.P., Eker, T.A., Woerden, J.E., Witon, J.-M., Stallinga, S.O., Visser, A., Schutte, K., Heslinga, F.G.: Generative AI methods for synthesis of image data to train AI for automated scene understanding in a military context. *Proc. SPIE* **13459**, 9–31 (2025)
26. Azizi, S., Kornblith, S., Saharia, C., Norouzi, M., Fleet, D.J.: Synthetic data from diffusion models improves ImageNet classification. *IEEE Trans. Pattern Anal. Mach. Intell.* **46**(8), 5469–5481 (2024)
27. Turkcan, M.K., Li, Y., Zang, C., Ghaderi, J., Zussman, G., Kostic, Z.: Boundless: generating photorealistic synthetic data for object detection in urban streetscapes. *IEEE Trans. Intell. Transp. Syst.*, (2025). (early access)
28. Vanherle, B., Moonen, S., Van Reeth, F., Michiels, N.: Analysis of training object detection models with synthetic data. *Mach. Vis. Appl.* **35**(2), 43 (2024)
29. Chan, D.A., Rao, D., Salimans, F., Karras, T., Aila, T.: Rethinking FID: towards a better evaluation metric for image generation. In: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit, pp. 15398–15408. (2024)
30. Westerski, A., Sánchez-Cortes, M., Iglesias, C.: Synthetic data for object detection with neural networks: a comprehensive survey, *ACM Comput. Surv.* **57**(3), Article 67, 39 pages (2024)
31. Xu, X., Zhang, X., Zhang, T., Zhang, W., Shi, J., Wei, S., Shao, Z., Xu, Y., Zeng, T.: Distribution-based anchor assignment and comprehensive score voting with distance penalty IoU loss for SAR remote sensing ship detection. *IEEE Trans. Instrum. Meas.* **73**, 1–18 (2024)
32. Xu, X., Zhang, X., Zhang, T., Zhang, W., Shi, J., Wei, S., Shao, Z., Xu, Y., Zeng, T.: DiffSARShipInst: Diffusion model for ship instance segmentation from synthetic aperture radar imagery. *ISPRS J. Photogramm. Remote. Sens.* **223**, 440–455 (2025)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Zahra Elmi is an assistant professor in the Department of Software Engineering at Beykoz University, Istanbul, Turkey. Her research focuses on computer vision, object detection, synthetic-to-real domain adaptation, test-time adaptation, multimodal visual perception, and open-set recognition. Her recent work addresses the deployment of deep object detectors under distribution shift, particularly in synthetically trained and real-world aerial imaging scenarios.

Furkan Ali Turan received his B.Sc. degree from the Department of Computer Engineering at Istanbul Sabahattin Zaim University, Istanbul, Turkey. His research interests include deep learning, computer vision, object detection, model training, and performance evaluation for visual recognition systems.

Emir Tuna received his B.Sc. degree from the Department of Computer Engineering at Istanbul Sabahattin Zaim University, Istanbul, Turkey. His research interests include machine learning, computer vision, synthetic-to-real adaptation, and experimental analysis of deep detection models.

Abdullatif Akkaya received his B.Sc. degree from the Department of Computer Engineering at Istanbul Sabahattin Zaim University, Istanbul, Turkey. His research interests include artificial intelligence, image processing, applied machine learning, and computer-vision-based detection systems.

Soheila Elmi is a Ph.D. student in the Department of Electrical and Electronics Engineering at Koç University, Istanbul, Turkey. Her research interests include artificial intelligence, machine learning, signal processing, data-driven modeling, and visual perception systems.