Dynamic Physical Adversarial LED Patterns via Reinforcement Learning

Konstantin Zyryanov
Department of Physics, Lomonosov Moscow State University
119991, Moscow, Russia
zyryanov.konstantin.sergeevich@gmail.com

Nikita Lotts
Faculty of Computer Science, HSE University
109028, Moscow, Russia
nklotts@edu.hse.ru

Abstract

Physical adversarial attacks expose limitations of computer vision systems, but most existing methods rely on static patches or textures. We study closed-loop physical adversarial control, where an agent adapts a programmable emissive actuator using environmental observations and black-box detector feedback during training. We implement this setting using a wearable RGB LED array and Physically Regularized Soft Actor-Critic (PR-SAC) with latent action parameterization. Across YOLOv8, Faster R-CNN, and RetinaNet, dynamic PR-SAC achieves attack success rates of 89.1%, 84.7%, and 86.8%, outperforming optimized static LED patterns by 38.3–42.1 percentage points. Evaluation in unseen locations shows reduced but substantial attack effectiveness. We further analyze input-processing defenses and persistent training instability, including Q-drift and entropy collapse. Our results highlight the importance of evaluating adaptive physical attacks through direct evasion metrics, environmental generalization, and training stability.

1 Introduction

Automated person detection is increasingly used in video surveillance, access control, and scene monitoring. While these systems provide useful capabilities, their deployment also raises questions about privacy, consent, and the reliability of automated visual monitoring. Understanding whether physically realizable, adaptive perturbations can disrupt person detection is therefore relevant both to robustness evaluation and to research on privacy-preserving technologies. Our work investigates this vulnerability under a constrained black-box threat model; it does not propose a ready-to-deploy privacy protection product.

Most physical adversarial attacks rely on static patches, printed textures, or fixed clothing patterns. Such perturbations cannot adapt their appearance when the scene changes. We instead study *closed-loop physical adversarial control*, in which a policy changes a physical actuator in response to observations of the environment while receiving only detector-level feedback during training. This introduces challenges absent from digital attacks: sensor noise, changing illumination, viewpoint variation, physical rendering constraints, and expensive environment interactions.

We investigate this setting using a wearable programmable RGB LED array. The garment is an experimental testbed rather than a proposed consumer device: its non-planar surface, self-occlusion, wearer motion, and limited visible area make adaptive control particularly challenging. Unlike printed textures, the LEDs emit light and can change their patterns in real time. The underlying formulation

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

may also be relevant to other controllable emissive devices, although transfer to such actuators is not established by our experiments.

We formulate LED control as a black-box reinforcement learning problem and study Physically Regularized Soft Actor-Critic (PR-SAC) with latent action parameterization. Beyond confidence and mAP@0.5, we evaluate attack success rate, compare dynamic control against optimized static LED patterns, and examine generalization to previously unseen locations. We also analyze physical training instability and the effectiveness of input-processing defenses.

Our contributions are as follows:

- We introduce closed-loop physical adversarial control as a new reinforcement learning problem setting.
- We introduce and evaluate latent action parameterization with deterministic upsampling as a way to stabilize large-scale LED control.
- We analyze the instability of off-policy RL in physical adversarial control, including Q-drift, entropy collapse, replay mismatch, and degradation after peak performance.
- We evaluate dynamic PR-SAC against optimized static LED patterns and black-box baselines using detection confidence, mAP@0.5, and attack success rate, and assess generalization to unseen locations and robustness under input-processing defenses.

2 Related Work

2.1 Digital adversarial examples

Adversarial examples were first studied in the digital domain, where small perturbations can significantly alter model predictions [8]. Stronger optimization-based attacks such as Carlini & Wagner [3] and robust optimization frameworks [18] further demonstrated the vulnerability of deep networks. Transferability across models [5] and generative approaches [26] have also been explored. In addition, localized perturbations such as LaVAN [12] highlight the effectiveness of structured adversarial noise. However, these methods assume direct access to digital inputs and do not account for real-world constraints.

2.2 Physical adversarial attacks

Physical attacks introduce sensor noise, geometric transformations, illumination changes, and environmental variability. Early work demonstrated physically robust attacks on classifiers and detectors [6, 7], followed by transformation-aware optimization such as ShapeShifter [4]. Subsequent work explored adversarial patches [2, 17, 14, 24, 31, 30], including naturalistic and translucent variants [9]. Recent studies have highlighted real-world limitations of patch-based attacks [22].

Beyond patches, physically realizable attacks include adversarial clothing and textures [27, 11, 10], accessories [23], and camouflage patterns [29]. Other directions include attacks via camera manipulation [15] and invisibility-style perturbations [25]. While these approaches demonstrate physical feasibility, most rely on static perturbations. In contrast, we study closed-loop dynamic LED control, where attack optimization itself occurs through noisy physical feedback.

2.3 Black-box optimization and reinforcement learning

In realistic scenarios, attackers often lack access to gradients, requiring black-box optimization. Prior approaches include query-based and transfer-based attacks [20], which scale poorly in high-dimensional spaces. Reinforcement learning provides an alternative by treating the model as part of the environment and optimizing perturbations via reward signals. However, existing RL-based methods primarily focus on digital settings or low-dimensional control, limiting their applicability to physical systems.



Figure 1: Real-world adversarial setup. A programmable LED garment, driven by an Orange Pi controller, is captured by an external detector camera, while onboard cameras provide observations for the RL agent.

2.4 Robustness, defenses, and detection models

Various defense strategies have been proposed, including input preprocessing such as feature squeezing [28] and robust training [18]. A morphological defense targeting physical perturbations was introduced in [32].

Modern object detectors, including Faster R-CNN [21], RetinaNet [16], and YOLO-based models [1], serve as primary targets and exhibit varying sensitivity to perturbations, motivating cross-model evaluation.

3 Statement of the problem

3.1 Hardware setup

The system implements a closed-loop physical pipeline (Figures 1–2) consisting of LED control, physical rendering, camera capture, object detection, and reward computation. The wearable array contains 39 panels of 16×16 RGB LEDs, totaling $N_{\mathrm{LED}} = 9{,}984$ individually addressable LEDs and $d = 3N_{\mathrm{LED}} = 29{,}952$ RGB control values.

Each LED is controlled by an RGB triplet in $\{0, \ldots, 255\}^3$; color and brightness are jointly determined by these values, without a separate intensity channel. An Orange Pi 5 Pro drives the LED matrix, while an NVIDIA GTX 1050 GPU (3 GB VRAM) supports training and detector inference.

Two onboard cameras, facing forward and backward, capture the surrounding scene for the policy encoder. They are observation sensors, not evaluation cameras. A separate external camera records the illuminated garment and provides frames to the target person detector. This separation allows the policy to condition its actions on environmental observations while measuring attack performance independently.

3.2 Formalization of the problem as reinforcement learning

We formulate LED pattern generation as a continuous reinforcement learning problem in a partially observable and non-stationary physical environment. Unlike digital adversarial optimization, the agent has no access to gradients and interacts with the system solely through detector outputs.

Let I_t^{front} and I_t^{back} denote the images captured by the forward-facing and backward-facing onboard cameras at time t. We define the observation as $o_t = (I_t^{front}, I_t^{back})$ and the state representation as $s_t = f(o_t)$, where $f(\cdot)$ is an image feature extractor. Due to sensor noise and environmental variability, the observed images are noisy measurements of the surrounding scene:

$$o_t = \bar{o}_t(\theta_t) + \varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0, \sigma I), \tag{1}$$

where θ_t denotes physical scene parameters such as distance, viewpoint, pose, and illumination.

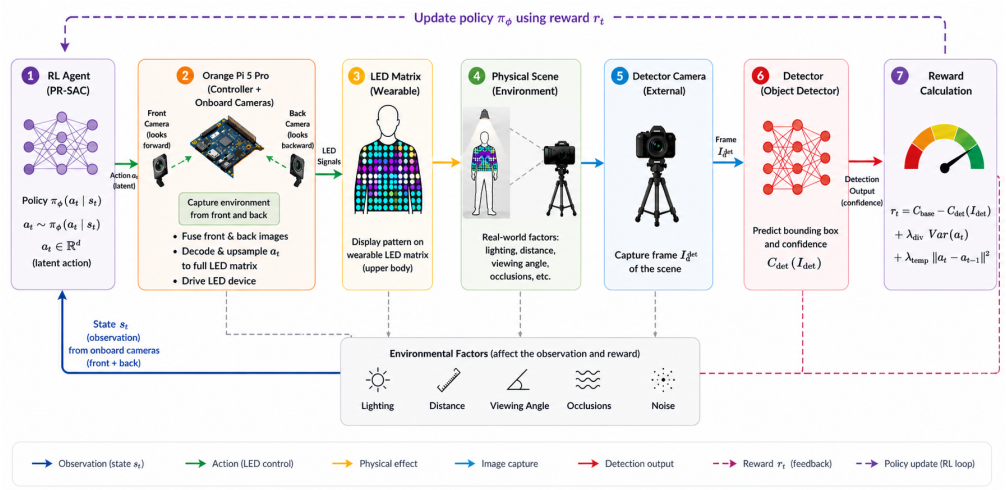


Figure 2: Closed-loop physical adversarial optimization pipeline. A PR-SAC agent generates LED control actions, which are executed by an Orange Pi controller and rendered on the wearable LED matrix. The camera captures the resulting scene, the detector outputs confidence, and the reward is used to update the policy.

The agent outputs a continuous LED action $a_t \in [0, a_{\max}]^d$, where $d = 3N_{\text{LED}}$ corresponds to RGB values for all LEDs. The action is rendered by the physical LED garment and observed by a separate detector camera. Let $I_t^{det} = R(a_t, \theta_t)$ denote the detector-camera image after physical rendering, and let $C_{det}(I_t^{det}) \in [0, 1]$ be the detector confidence for the person class. The reward is

$$r_t = 1 - C_{det}(I_t^{det}) + \lambda_{\text{div}} \operatorname{Var}(a_t) + \lambda_{\text{temp}} \|a_t - a_{t-1}\|^2, \tag{2}$$

where the first term encourages confidence reduction, while the auxiliary terms encourage action diversity and temporal variation.

Overall, the reward can be written as a stochastic physical composition

$$r_t = g(D(R(a_t, \theta_t))) + \xi_t, \tag{3}$$

where R is the physical rendering process, D is the detector, $g(\cdot)$ maps detector output to reward, and ξ_t denotes measurement noise. This creates a non-stationary and noisy reward signal, motivating entropy-regularized off-policy RL while also causing critic instability and replay mismatch.

3.3 Threat model

The adversary controls a wearable RGB LED array but cannot modify detector weights, architecture, cameras, or digital inputs. Training uses black-box detector-confidence feedback and observations from two onboard cameras. At evaluation time, the learned policy generates patterns from onboard observations without detector feedback. The objective is to evade person detection by an external camera; transfer to other target classes or actuators is not evaluated.

4 Method

4.1 General architecture

An encoder combines front and rear camera observations into a state representation. PR-SAC predicts a latent action, decoded and upsampled into an LED pattern. Detector confidence from the external camera provides the training reward.

4.2 Physically Regularized Soft Actor-Critic

Figure 3 illustrates the SAC architecture used in our method. We build on Soft Actor-Critic (SAC), but direct SAC is unstable in our physical loop due to noisy detector feedback, scene drift, and

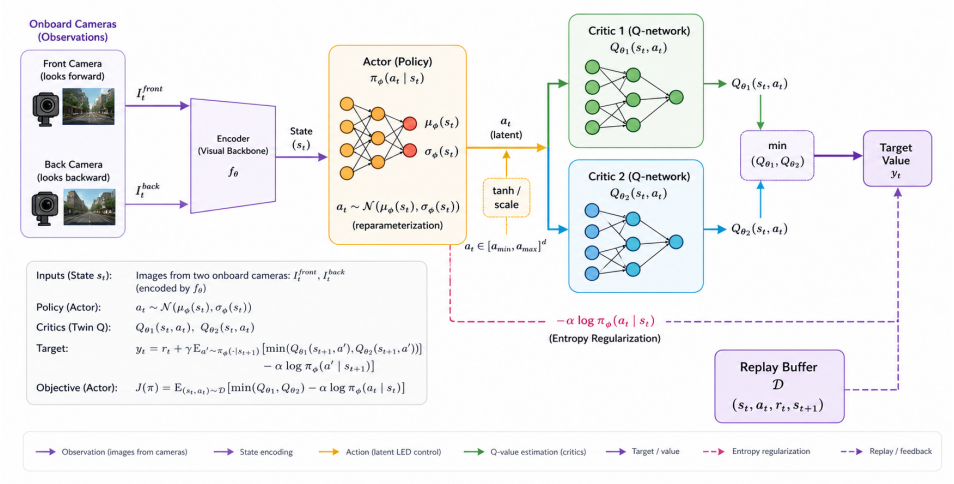


Figure 3: Base SAC architecture underlying PR-SAC. The actor samples bounded continuous actions, while twin critics estimate action values with clipped double Q-learning and entropy regularization.

high-dimensional LED control. PR-SAC biases learning toward physically consistent trajectories using smooth actions, reliable feedback, and recent replay samples, which improves stability in the physical setting, although formal convergence guarantees are not available.

First, we regularize the actor toward spatially smooth and temporally consistent LED patterns:

$$J_\pi = \mathbb{E}\left[\alpha \log \pi_\phi(a_t \mid s_t) - Q_\theta(s_t, a_t) + \lambda_s \|\nabla_{\text{LED}} a_t\|^2 + \lambda_t \|a_t - a_{t-1}\|^2\right], \tag{4}$$

where ∇_{LED} denotes finite differences over neighboring LEDs.

Second, we down-weight noisy physical transitions using

$$\mathcal{L}_Q = \tilde{w}_i \left(Q_\theta(s_i, a_i) - y_i\right)^2, \quad w_i = \frac{1}{\epsilon + \hat{\sigma}_r^2(s_i, a_i)}, \tag{5}$$

where y_i is the SAC target value for replay transition i.

Third, we reduce replay staleness with recency weights

$$\rho_i = \exp(-\beta(T - t_i)), \qquad \tilde{w}_i = \rho_i w_i. \tag{6}$$

4.3 Action-space parameterization

Direct LED-level control has dimension $d = 3N_{\text{LED}}$, which can reach several thousand and leads to unstable critic estimates. We therefore use a latent action that is decoded and upsampled into the full LED pattern. This creates a trade-off between expressiveness and stability: low upsampling factors preserve high-frequency detail but are unstable, while high factors enforce smoothness but reduce attack strength, as shown in Table 1.

Table 1: Effect of action parameterization on stability. Values are mean $\pm$ standard deviation over 15 runs; 95% confidence intervals are in brackets. Lower is better.

Parameterization	Final confidence	Degradation after peak
Direct LED control	0.71 ± 0.08 $[0.67, 0.75]$	0.19 ± 0.07 $[0.15, 0.23]$
Latent action	0.60 ± 0.07 $[0.56, 0.64]$	0.18 ± 0.06 $[0.15, 0.21]$
Latent + structured upsampling	0.43 ± 0.08 $[0.39, 0.47]$	0.13 ± 0.03 $[0.11, 0.15]$

4.4 Off-Policy Instability

The reward in the physical setting is a stochastic and non-stationary composition

$$r_t = g(D(R(a_t, \theta_t))) + \xi_t, \tag{7}$$

where scene parameters θ_t change over time. Hence the Bellman operator is time-dependent,

$$(\mathcal{T}_t Q)(s, a) = r_t(s, a) + \gamma \mathbb{E}_{s', a' \sim \pi} Q(s', a'), \tag{8}$$

which implies that critic targets are computed under a drifting reward process. In addition, replay samples follow $d_{\mathcal{D}}(s, a)$ rather than the current $d_\pi(s, a)$, producing an off-policy bias

$$\mathbb{E}_{d_{\mathcal{D}}}[Q] - \mathbb{E}_{d_\pi}[Q]. \tag{9}$$

In high-dimensional LED control, this mismatch is amplified by noisy exploration and sparse effective regions of the action space, leading to Q-drift, entropy collapse, and degradation after early improvement.

4.5 Experimental protocol

We evaluate YOLOv8, Faster R-CNN, and RetinaNet using detector confidence, mAP@0.5, and attack success rate (ASR). ASR is the percentage of frames without a person detection matching the target ground-truth box at IoU ≥ 0.5 and confidence ≥ 0.6. Confidence provides the dense reward signal during training.

Baselines include random LED patterns, vanilla SAC, and CMA-ES, which optimizes a static pattern in the same latent space with the same reward and query budget. The best-frozen baseline holds a pattern from a trained PR-SAC policy fixed during evaluation. Each configuration uses 15 independent seeds and 300 evaluation frames per run. We report run-level means, standard deviations, and 95% Student-t confidence intervals. Checkpoints are selected by minimum mean confidence on a held-out validation sequence.

An operator varies pose, viewpoint, distance, and illumination under a shared randomization schedule. Distance and angle tests isolate the respective factor; unseen-location tests use new backgrounds, geometry, and illumination without fine-tuning. Training uses 2,000 episodes of approximately 10–12 seconds each (roughly 5–6 hours per run).

5 Results

Unless otherwise stated, all reported uncertainties are computed across 15 independent training runs, not across frames. Confidence intervals are computed over run-level averages.

5.1 Dynamic versus static LED control

To determine whether attack effectiveness results from adaptive control rather than merely displaying an unusual LED pattern, we compare PR-SAC against optimized static patterns and non-adaptive controls. Table 2 reports attack success rate (ASR) across three detectors.

Dynamic PR-SAC exceeds the best-frozen baseline by 38.3, 42.1, and 40.0 percentage points of ASR on YOLOv8, Faster R-CNN, and RetinaNet, respectively. The corresponding detector confidences are 0.40/0.47/0.43 for dynamic PR-SAC and 0.63/0.71/0.67 for the frozen baseline.

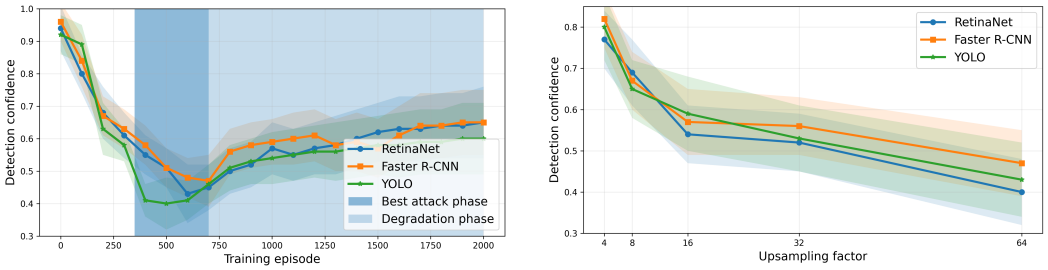
Generalization to unseen locations. We evaluate trained policies without fine-tuning in locations excluded from training, with changes in background, geometry, and illumination. ASR decreases from 89.1/84.7/86.8% to 77.3/69.8/74.5% across the three detectors, while mean detection confidence increases from 0.40/0.47/0.43 to 0.46/0.53/0.49. Despite this degradation, dynamic PR-SAC in unseen locations retains higher ASR than the evaluated static baselines in familiar conditions. These results support partial environmental generalization, without establishing robustness to arbitrary outdoor or sensor conditions.

5.2 Main training dynamics

Figure 4a shows PR-SAC training dynamics over 15 runs, while Figure 4b illustrates the effect of the upsampling factor on final adversarial performance.

Table 2: Attack success rate (ASR, %) for static and dynamic LED control. Results are mean ± standard deviation over 15 independent runs; higher is better. Unseen denotes evaluation in locations excluded from training.

Method	YOLOv8	Faster R-CNN	RetinaNet
Clean	0.4 ± 0.2	1.4 ± 0.9	0.6 ± 0.5
Random LED	8.7 ± 3.8	10.6 ± 4.1	7.9 ± 3.6
CMA-ES (static)	43.5 ± 5.4	45.1 ± 5.1	47.8 ± 5.3
Vanilla SAC	61.9 ± 6.0	55.8 ± 6.8	54.6 ± 6.3
PR-SAC (frozen)	50.8 ± 5.9	42.6 ± 5.5	46.8 ± 5.4
PR-SAC (random snapshot)	22.5 ± 4.9	20.8 ± 4.5	15.7 ± 4.3
PR-SAC (dynamic)	$\mathbf{89.1 \pm 4.3}$	$\mathbf{84.7 \pm 5.0}$	$\mathbf{86.8 \pm 3.8}$
PR-SAC (unseen)	77.3 ± 5.4	69.8 ± 5.6	74.5 ± 5.2



(a) Detection confidence vs. training episodes.

(b) Effect of upsampling factor on final adversarial performance.

Figure 4: Analysis of training dynamics and spatial parameterization. (a) Detection confidence over training episodes, averaged across 15 independent runs; shaded/error variation corresponds to variability across runs. (b) Intermediate upsampling provides the best balance between spatial smoothness and pattern expressiveness.

5.3 Cross-model transferability

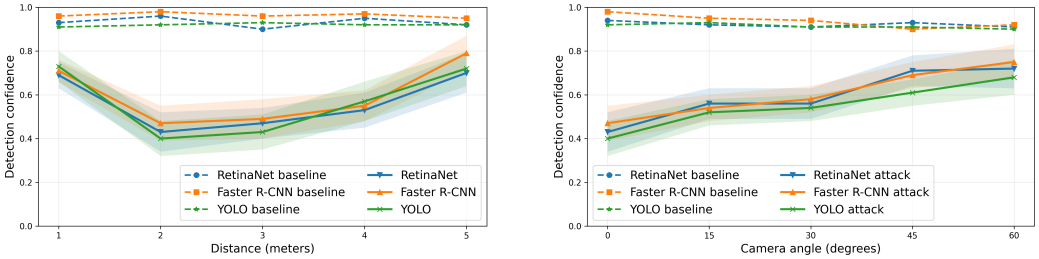
Table 3 evaluates cross-model transfer. The strongest effect is observed on the training detector, but the learned patterns partially transfer to other architectures.

Table 3: Cross-model transferability of the learned attack. Values indicate mean detection confidence ± standard deviation over 15 independent training runs; 95% confidence intervals are shown in brackets. Lower confidence corresponds to a stronger attack.

Train on \ Test on	YOLO	Faster R-CNN	RetinaNet
YOLO	0.40 ± 0.08 $[0.36, 0.44]$	0.48 ± 0.06 $[0.45, 0.51]$	0.51 ± 0.07 $[0.47, 0.55]$
Faster R-CNN	0.52 ± 0.06 $[0.49, 0.55]$	0.47 ± 0.08 $[0.43, 0.51]$	0.56 ± 0.06 $[0.53, 0.59]$
RetinaNet	0.54 ± 0.07 $[0.50, 0.58]$	0.55 ± 0.06 $[0.52, 0.58]$	0.43 ± 0.09 $[0.38, 0.48]$

5.4 Robustness to distance and viewing angle

Figure 5 shows robustness to geometric variation. Attack effectiveness decreases with distance and viewing angle, as the LED pattern occupies less image area and undergoes stronger geometric distortion.



(a) Detection confidence vs. camera distance.

(b) Detection confidence vs. viewing angle.

Figure 5: Robustness of the adversarial LED pattern under real-world variations. The attack effectiveness decreases with distance and viewing angle, indicating sensitivity to scale and geometric distortions. Lower confidence corresponds to a stronger adversarial effect.

5.5 Ablation studies

Table 4 compares reward formulations. Regularized rewards improve performance, but excessive shaping can reduce stability. Unless otherwise stated, the default reward is $1 - C$ with additional regularization terms.

Table 4: Impact of different reward formulations on attack performance across detectors. Values are reported as mean detection confidence $\pm$ standard deviation over 15 independent training runs; 95% confidence intervals are shown in brackets

Reward Function	YOLO	Faster R-CNN	RetinaNet
$-C$	0.62 ± 0.06 $[0.59, 0.65]$	0.64 ± 0.08 $[0.60, 0.68]$	0.67 ± 0.06 $[0.64, 0.70]$
$1 - C$	0.59 ± 0.07 $[0.55, 0.63]$	0.60 ± 0.08 $[0.56, 0.64]$	0.63 ± 0.08 $[0.59, 0.67]$
$-\log C$	0.56 ± 0.05 $[0.53, 0.59]$	0.54 ± 0.06 $[0.51, 0.57]$	0.57 ± 0.06 $[0.54, 0.60]$
$(1 - C)^2$	0.55 ± 0.06 $[0.52, 0.58]$	0.53 ± 0.09 $[0.48, 0.58]$	0.61 ± 0.07 $[0.57, 0.65]$
+ variance	0.43 ± 0.09 $[0.38, 0.48]$	0.51 ± 0.07 $[0.47, 0.55]$	0.50 ± 0.09 $[0.45, 0.55]$
+ temporal diff	0.40 ± 0.08 $[0.36, 0.44]$	0.47 ± 0.08 $[0.43, 0.51]$	0.43 ± 0.09 $[0.38, 0.48]$

Tables 5 and 6 compare PR-SAC with random search, CMA-ES, and vanilla SAC. PR-SAC achieves the strongest average confidence reduction and mAP@0.5 degradation, while vanilla SAC has higher variance.

Table 5: Main attack performance compared to clean baseline, random LED patterns, black-box evolutionary optimization, vanilla SAC, and PR-SAC. Values are reported as mean detection confidence $\pm$ standard deviation over 15 independent runs; 95% confidence intervals are shown in brackets. Lower confidence indicates a stronger attack.

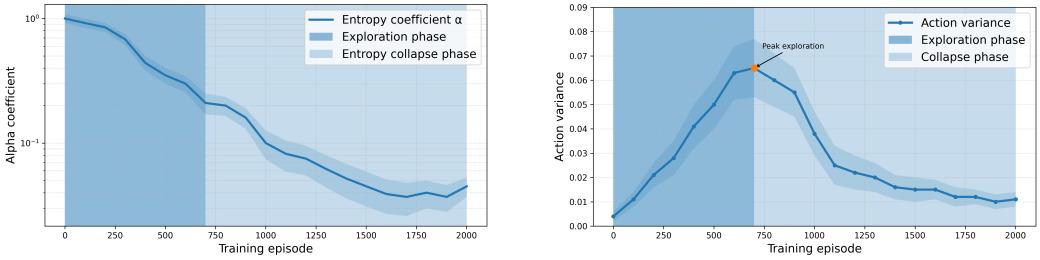
Method	YOLOv8	Faster R-CNN	RetinaNet
Baseline	0.92 ± 0.04 $[0.90, 0.94]$	0.96 ± 0.03 $[0.94, 0.98]$	0.94 ± 0.03 $[0.92, 0.96]$
Random Search	0.81 ± 0.04 $[0.79, 0.83]$	0.79 ± 0.04 $[0.77, 0.81]$	0.82 ± 0.05 $[0.79, 0.85]$
CMA-ES	0.71 ± 0.05 $[0.68, 0.74]$	0.70 ± 0.05 $[0.67, 0.73]$	0.68 ± 0.06 $[0.65, 0.71]$
Vanilla SAC	0.59 ± 0.08 $[0.55, 0.63]$	0.66 ± 0.07 $[0.62, 0.70]$	0.67 ± 0.08 $[0.63, 0.71]$
PR-SAC	0.40 ± 0.08 $[0.36, 0.44]$	0.47 ± 0.08 $[0.43, 0.51]$	0.43 ± 0.09 $[0.38, 0.48]$

Table 6: Detection-level attack performance measured by mAP@0.5 for the person class, including random search, CMA-ES, vanilla SAC, and PR-SAC. Values are reported as mean ± standard deviation over 15 independent training runs; 95% confidence intervals are shown in brackets. Lower mAP indicates stronger degradation of detector performance.

Method	YOLOv8	Faster R-CNN	RetinaNet
Baseline	0.88 ± 0.03 $[0.86, 0.90]$	0.85 ± 0.03 $[0.83, 0.87]$	0.83 ± 0.03 $[0.81, 0.85]$
Random Search	0.74 ± 0.04 $[0.72, 0.76]$	0.72 ± 0.04 $[0.70, 0.74]$	0.73 ± 0.04 $[0.71, 0.75]$
CMA-ES	0.64 ± 0.05 $[0.61, 0.67]$	0.63 ± 0.05 $[0.60, 0.66]$	0.62 ± 0.05 $[0.59, 0.65]$
Vanilla SAC	0.54 ± 0.06 $[0.51, 0.57]$	0.58 ± 0.06 $[0.55, 0.61]$	0.57 ± 0.06 $[0.54, 0.60]$
PR-SAC	0.42 ± 0.06 $[0.39, 0.45]$	0.47 ± 0.05 $[0.44, 0.50]$	0.45 ± 0.05 $[0.42, 0.48]$

5.6 Stability analysis

Figure 6 shows that both the entropy coefficient and action variance decrease during training. PR-SAC delays but does not eliminate action-diversity collapse, supporting the hypothesis that entropy collapse is a contributing factor to performance degradation in physical off-policy adversarial control.



(a) Entropy coefficient α over training episodes.

(b) Action variance over training episodes.

Figure 6: Training dynamics of the PR-SAC agent in the physical adversarial setting. Both metrics decrease over time, indicating reduced exploration and eventual collapse of policy diversity, which correlates with degradation in attack performance.

5.7 Defense analysis

Table 7 evaluates defense interaction. The defense partially restores confidence, but does not eliminate the PR-SAC attack.

Table 7: Attack vs. defense comparison across detectors. Values are reported as mean detection confidence ± standard deviation over 15 independent training runs; 95% confidence intervals are shown in brackets. Lower confidence indicates a stronger attack.

Detector	PR-SAC	AdvTshirt [27]	AdvTexture [10]	AdvCaT [11]	Defense [32]
YOLOv8	0.40 ± 0.08 $[0.36, 0.44]$	0.62 ± 0.07 $[0.58, 0.66]$	0.55 ± 0.06 $[0.52, 0.58]$	0.52 ± 0.06 $[0.49, 0.55]$	0.69 ± 0.07 $[0.65, 0.73]$
Faster R-CNN	0.47 ± 0.08 $[0.43, 0.51]$	0.65 ± 0.08 $[0.61, 0.69]$	0.58 ± 0.07 $[0.54, 0.62]$	0.55 ± 0.06 $[0.52, 0.58]$	0.73 ± 0.05 $[0.70, 0.76]$
RetinaNet	0.43 ± 0.09 $[0.38, 0.48]$	0.60 ± 0.08 $[0.56, 0.64]$	0.53 ± 0.07 $[0.49, 0.57]$	0.50 ± 0.07 $[0.46, 0.54]$	0.66 ± 0.06 $[0.63, 0.69]$

Robustness against input-processing defenses. We further evaluate five defense configurations on the same PR-SAC evaluation frames, including Gaussian blur, JPEG compression, a morphological defense [32], UDFilter [19], and RAPID [13].

Table 8: Effect of input-processing defenses on PR-SAC. Each entry reports ASR (%) / mean detector confidence. Lower ASR and higher confidence indicate stronger defense.

Defense	YOLOv8	Faster R-CNN	RetinaNet
None	89.1 / 0.40	84.7 / 0.47	86.8 / 0.43
Gaussian blur	61.2 / 0.54	56.8 / 0.56	69.4 / 0.51
JPEG compression	63.6 / 0.53	58.9 / 0.59	71.0 / 0.56
Morphological	39.8 / 0.69	30.9 / 0.73	46.5 / 0.66
UDFilter	35.7 / 0.70	27.8 / 0.74	41.2 / 0.68
RAPID	24.6 / 0.74	17.6 / 0.78	29.8 / 0.72

Table 8 reports ASR and detector confidence after applying each defense.

Generic transformations reduce attack effectiveness but leave substantial residual evasion: ASR remains between 56.8% and 71.0% under blur or JPEG compression. Dedicated physical-attack defenses perform better. RAPID yields the lowest observed ASR, ranging from 17.6% to 29.8% across detectors. Adaptive adversarial training and certified defenses remain outside the scope of our evaluation.

6 Discussion

Physical off-policy training remains unstable due to noisy detector feedback, changing scene conditions, and replay mismatch. We observe Q-drift, entropy collapse, and performance degradation after early improvements. PR-SAC mitigates these effects through spatial and temporal regularization, reliability-weighted critic updates, and recency-weighted replay, without guaranteeing convergence.

Latent upsampling improves the stability of high-dimensional LED control. Selecting checkpoints using held-out validation confidence limits the practical impact of late-training degradation, but does not resolve the underlying instability.

7 Limitations

Our experiments cover three person detectors and a single wearable LED platform. Performance depends on lighting, viewpoint, distance, and camera characteristics; unseen-location results do not establish robustness under uncontrolled conditions. Other target classes, actuators, adversarially trained detectors, and certified defenses remain untested. PR-SAC also suffers from training instability and requires approximately 5–6 hours per run.

8 Broader Impact

This work studies vulnerabilities of computer vision systems under physically realizable adaptive attacks. Although such methods could potentially be misused to evade surveillance or safety-critical perception systems, our goal is to improve robustness evaluation under realistic physical conditions. By analyzing transferability, environmental robustness, and defense interaction, we aim to support the development of more reliable vision models. To mitigate misuse risks, we do not release real-time deployment tools or optimized policies for specific targets.

9 Conclusion

We introduced closed-loop physical adversarial LED control using PR-SAC. Dynamic policies achieve 89.1%, 84.7%, and 86.8% ASR on YOLOv8, Faster R-CNN, and RetinaNet, outperforming optimized frozen patterns by 38.3–42.1 percentage points. Experiments also reveal partial environmental generalization and persistent training instability, motivating stronger evaluation of adaptive physical attacks.

References

[1] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. 2020.

[2] Tom B. Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. 2018.

[3] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. 2017.

[4] Shang-Tse Chen, Cory Cornelius, Jason Martin, and Duen Horng Chau. Shapeshifter: Robust physical adversarial attack on faster r-cnn object detector. In *Machine Learning and Knowledge Discovery in Databases*, page 52–68. Springer International Publishing, 2019.

[5] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. 2018.

[6] Kevin Eykholt et al. Robust physical-world attacks on deep learning visual classification. In *CVPR*, 2018.

[7] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Florian Tramer, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Physical adversarial examples for object detectors. 2018.

[8] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. 2015.

[9] Yu-Chih-Tuan Hu, Jun-Cheng Chen, Bo-Han Kung, Kai-Lung Hua, and Daniel Stanley Tan. Naturalistic physical adversarial patch for object detectors. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7828–7837, 2021.

[10] Zhanhao Hu, Wenda Chu, Xiaopei Zhu, Hui Zhang, Bo Zhang, and Xiaolin Hu. Physically realizable natural-looking clothing textures evade person detectors via 3d modeling. 2024.

[11] Zhanhao Hu, Siyuan Huang, Xiaopei Zhu, Fuchun Sun, Bo Zhang, and Xiaolin Hu. Adversarial texture for fooling person detectors in the physical world. 2022.

[12] Danny Karmon, Daniel Zoran, and Yoav Goldberg. Lavan: Localized and visible adversarial noise. 2018.

[13] Heemin Kim, Byeong-Chan Kim, Sumi Lee, Minjung Kang, Hyunjee Nam, Sunghwan Park, Il-Youp Kwak, and Jaewoo Lee. RAPID: Robust multi-patch masker using channel-wise pooled variance with two-stage patch detection. *Journal of King Saud University - Computer and Information Sciences*, 36(8):102188, 2024.

[14] Mark Lee and Zico Kolter. On physical adversarial patches for object detection. 2019.

[15] Juncheng Li, Frank R. Schmidt, and J. Zico Kolter. Adversarial camera stickers: A physical camera-based attack on deep learning systems. 2019.

[16] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. 2018.

[17] Xin Liu, Huanrui Yang, Ziwei Liu, Linghao Song, Hai Li, and Yiran Chen. Dpatch: An adversarial patch attack on object detectors. 2019.

[18] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. 2019.

[19] Zimin Mao, Shuiyan Chen, Zhuang Miao, Heng Li, Beihao Xia, Junzhe Cai, Wei Yuan, and Xinge You. Enhancing robustness of person detection: A universal defense filter against adversarial patch attacks. *Computers & Security*, 146:104066, 2024.

[20] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. 2017.

[21] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. 2016.

[22] Jakob Shack, Katarina Petrovic, and Olga Saukh. Breaking the illusion: Real-world challenges for adversarial patches in object detection. 2024.

[23] Mahmood Sharif et al. Accessorize to a crime. In *CCS*, 2016.

[24] Simen Thys, Wiebe Van Ranst, and Toon Goedemé. Fooling automated surveillance cameras: adversarial patches to attack person detection. 2019.

[25] Zuxuan Wu, Ser-Nam Lim, Larry Davis, and Tom Goldstein. Making an invisibility cloak: Real world adversarial attacks on object detectors. 2020.

[26] Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. Generating adversarial examples with adversarial networks. 2019.

[27] Kaidi Xu, Gaoyuan Zhang, Sijia Liu, Quanfu Fan, Mengshu Sun, Hongge Chen, Pin-Yu Chen, Yanzhi Wang, and Xue Lin. Adversarial t-shirt! evading person detectors in a physical world. 2020.

[28] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. In *Proceedings 2018 Network and Distributed System Security Symposium*. Internet Society, 2018.

[29] Yang Zhang, Hassan Foroosh, Philip David, and Boqing Gong. CAMOU: Learning physical vehicle camouflages to adversarially attack detectors in the wild. In *International Conference on Learning Representations*, 2019.

[30] Wenjun Zhu et al. Tpatch: A trigger-based physical adversarial patch attack. In *USENIX Security*, 2023.

[31] Alon Zolfi, Moshe Kravchik, Yuval Elovici, and Asaf Shabtai. The translucent patch: A physical and universal attack on object detectors. 2020.

[32] Konstantin Zyryanov and Alexey Chulichkov. Morphological real-time defense against physical attacks. In *ICMV*, 2025.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction accurately reflect the contributions, including the formulation of closed-loop physical adversarial control and the experimental findings across multiple detectors.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: A dedicated section discusses limitations including hardware constraints, environmental sensitivity, and training instability .

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not present formal theoretical results requiring proofs.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper describes the hardware setup, RL formulation, reward function, and evaluation protocol in sufficient detail .

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The code and instructions for reproducing the experiments are provided in the supplementary material via .zip file.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies training setup, reward design, evaluation protocol, and experimental conditions.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Results are reported with mean, standard deviation, and 95% confidence intervals across 15 runs.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Experiments were conducted using an Orange Pi 5 Pro for LED control and a local machine with an NVIDIA GTX 1060 GPU (3GB VRAM) for training and evaluation.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research complies with NeurIPS ethical guidelines and does not involve harmful data collection or unethical experimentation.

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper discusses both risks (e.g., misuse for evasion) and positive impact (improving robustness evaluation).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [Yes]

Justification: The paper explicitly states that deployment tools and optimized policies are not released to mitigate misuse risks.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All used models (YOLOv8, Faster R-CNN, RetinaNet) are properly cited.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [No]

    Justification: No new datasets or code assets are released with the paper.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The work does not involve human subjects or crowdsourcing.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The work does not involve human subject research.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not part of the proposed method or experiments.