BUDGET-AWARE ADAPTIVE ADVERSARIAL PATCHES FOR BLACK-BOX OBJECT DETECTION

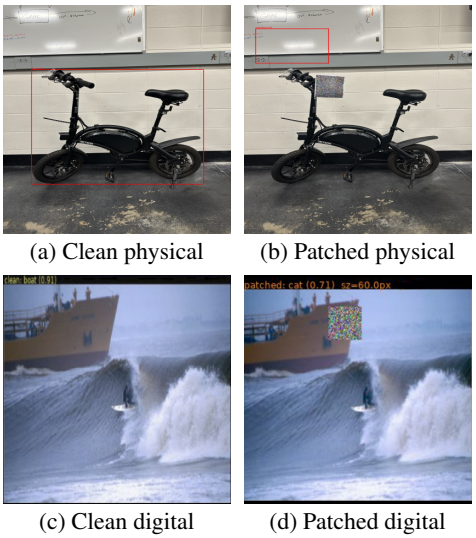
Pedram MohajerAnsari, Amir Salarpour, David Fernandez, Mert D. Pesé

School of Computing, Clemson University, USA
{pmohaje, asalarp, dferna3, mpese}@clemson.edu

ABSTRACT

Adversarial patches pose a practical threat to modern object detectors. Prior work shows vulnerability, but three gaps limit actionable insight: (i) few *score-based black-box* attacks *jointly* optimize patch *location, texture, and size* under tight query budgets; (ii) success is rarely tied to the patch's *visual footprint*; and (iii) evaluations often conflate EOT robustness with plain-view suppression. We present PATCHBANDIT, a query-efficient, budget-adaptive black-box attack that couples a lightweight *Contextual Thompson-Sampling* placer with NES-style pixel updates, growing the patch only when progress stalls. Reporting is anchored by a *strict plain-image* suppression test; EOT is audited but never used as a substitute for success, and optional appearance/printability weights expose strength–visibility trade-offs. Across YOLOv5, Faster R-CNN, and YOLOS, PATCHBANDIT achieves strong suppression on CNN-based detectors and substantial suppression on the transformer-based detector, using compact patches and exposing clear query–footprint trade-offs relative to fixed-size and heuristic baselines. A print–capture pilot further shows transfer across unseen physical objects and viewpoints. Code is available at https://github.com/pedram-mohajer/PatchBandit.

***Index Terms*—** Adversarial patch, Black-box attack, Query-efficient optimization, Object detection



(a) Clean physical (b) Patched physical

(c) Clean digital (d) Patched digital

Fig. 1: A 160 px physical patch flips the top YOLO prediction from *bicycle* (0.95) to *motorbike* (0.36); a 60 px digital patch flips a clean *boat* (0.91) to *cat* (0.92).

1. INTRODUCTION

In critical applications such as autonomous driving, robotics, and surveillance, computer vision systems increasingly depend on object detectors to detect and classify objects within a scene. These detectors span one-stage CNN architectures such as YOLO [1], two-stage CNN detectors such as Faster R-CNN [2], and transformer-based detectors such as YOLOS [3]. Like image classifiers, these detectors are vulnerable to adversarial perturbations—small, crafted input changes that induce incorrect predictions [4]. Among the most practical threats are adversarial patches [5]: localized patterns that can be printed and placed in the environment so cameras repeatedly capture them and downstream models are consistently misled. Because a single patch can be reused, scaled, and repositioned across scenes, patch attacks align naturally with physical-world threat scenarios.

However, existing patch methods leave a gap between what is demonstrated in controlled settings and what is feasible against deployed detectors. Prior work includes universal white-box patches [5, 6], physical demonstrations [7], and query-efficient black-box attacks on classifiers [8, 9], yet many real detectors are score-only black boxes [10], which limits access to gradients and internal signals. This challenge applies across detector architectures, from CNN-based detectors to transformer-based models such as YOLOS.

Under this interface, three issues remain. First, attackers must jointly decide where to place a patch and what texture to optimize under a tight query budget, but many approaches fix location, assume loose budgets, or rely on surrogate gradients [11]. Second, success is rarely reported as a function of patch area or appearance cost, despite the practical importance of visual footprint [12]. Third, Expectation-over-Transformation (EOT) training is common [13], but EOT-

averaged progress can diverge from suppression on the plain camera view encountered in deployment [14], motivating evaluation that separates plain-view success from robustness across transforms.

We present PATCHBANDIT, a score-based black-box adversarial patch attack designed around these constraints. PATCHBANDIT interleaves a lightweight contextual Thompson-sampling placer to efficiently explore placement cells with NES-style zeroth-order texture updates, and grows patch size only when progress stalls, exposing explicit success–versus–area trade-offs. Evaluation is anchored by a strict plain-image suppression criterion; EOT is audited and logged, but never substituted for success. We evaluate PATCHBANDIT across three detector families: YOLOv5 as a one-stage CNN detector, Faster R-CNN as a two-stage CNN detector, and YOLOS as a transformer-based detector. On YOLOv5, PATCHBANDIT achieves 77.5% strict suppression with compact patches (Figure 1); on Faster R-CNN, it reaches 89.7% strict suppression with very small median patch area; and on YOLOS, it achieves 59.1% strict suppression, indicating that the attack remains effective but is less pronounced on the transformer-based detector. A print–capture pilot further indicates that digitally optimized patches can transfer across viewpoints and object instances, repeatedly inducing mis-detections.

This paper makes the following contributions:

- A score-based black-box detector patch attack that jointly optimizes placement, texture, and size under explicit query budgets by combining contextual Thompson sampling, NES updates, and a progress-triggered growth policy.
- A footprint-aware reporting protocol that prioritizes strict plain-image suppression and reports EOT robustness separately, enabling clear success–visibility trade-off analysis (including $J = q_{\text{med}} \times a_{\text{med}}$).
- An empirical evaluation on YOLOv5, Faster R-CNN, and YOLOS showing strong suppression on CNN-based detectors and substantial suppression on a transformer-based detector, plus a print–capture pilot demonstrating transfer across physical viewpoints and objects.

2. RELATED WORK

Adversarial Patches and Physical Attacks on Vision Models. Adversarial patches are localized visual triggers that induce model misbehavior when inserted into a scene. Brown *et al.* introduced *universal, printable* patches that drive classifiers toward a chosen target under varied viewing conditions [5]. Liu *et al.* extended this idea to object detectors with DPATCH, degrading both classification and localization in Faster R-CNN and YOLO [6]. Thys *et al.* demonstrated wearable/held printed patches that reduce person-detection confidence in real-world surveillance settings [7]. Zolfi *et al.* proposed a *translucent* lens-mounted sticker that suppresses stop-sign detections without modifying the sign itself [15].

Black-Box and Query-Efficient Adversarial Example Generation. Early score-based black-box attacks estimated gradients via *zeroth-order* optimization (e.g., ZOO), avoiding substitute models [16]. Ilyas *et al.* improved query efficiency with Natural Evolution Strategies (NES) [17] and later "Bandits & Priors," which leverages structured gradient priors through bandit optimization to further reduce queries [9]. Randomized, region-wise methods such as the Square Attack provided another query-efficient alternative for high-dimensional inputs by using structured, localized updates [18].

Joint Optimization of Patch Content and Placement. Patch effectiveness depends on both *what* is printed and *where* it is placed. Prior work has treated placement as a decision variable, using RL or black-box optimization to jointly adapt position and appearance under query budgets [11]. Other approaches use evolutionary/annealing-style searches in camouflage-style pipelines (CamoPatch) [19], and recent score-based detector attacks optimize patches at locations via zeroth-order updates (BB-Patch) [20]. Earlier detector work such as DPATCH observed some location invariance but did not perform dense, adaptive placement search [6]. Building on this line, our contextual bandit provides a lightweight, query-aware policy for selecting promising placement cells without full RL training.

Stealth, Camouflage, and Printability Constraints. Because effective patches are often visually conspicuous, prior work constrains appearance for camouflage and physical realizability. Camo-style evolutionary methods encourage texture-matching while remaining adversarial [19], and Patch of Invisibility uses generative priors to produce natural-looking, printable black-box detector patches [12]. Wearable person-detector attacks similarly highlight the trade-off between attack strength (often high-contrast) and visual plausibility [7].

Evaluation Protocols, EOT, and Physical Robustness. A key challenge is ensuring patches remain effective under real-world transformations (viewpoint, scale, lighting). Athalye *et al.* introduced *Expectation Over Transformation* (EOT) to optimize perturbations over a transformation distribution [13]. Eykholt *et al.* showed that digital gains can collapse in physical settings without rigorous robustness-oriented evaluation [21]. Detector patch work such as DPATCH and Patch of Invisibility similarly tests under diverse, realistic environmental variation, though heavy augmentation can obscure whether the plain image is actually suppressed [6, 12].

Positioning of PATCHBANDIT. We target *score-based black-box* detectors and combine three elements that earlier studies explored in isolation: a lightweight contextual bandit selects promising cells under a query budget, NES performs gradient-free pixel updates, and a budget-adaptive size ladder enlarges the patch only when progress stalls. Strict success is recorded on the plain image, while EOT behavior is logged separately; the attack loop records queries, area, and confidence trajectories for reproducible ablations.

Table 1 places this design beside prior work. [5], [6], [7] and [15] rely on white-box gradients to train large, universal

Table 1: **Representative adversarial patch attacks.** Published results are reported for prior methods, while PATCHBANDIT is evaluated under our protocol. WB = white-box; BB = score-only black-box; cls = image classification; det = object detection.

Method	Threat	Phys.	Task	Atk	Reported effect / protocol	Notes
Brown17 [5]	WB	●	cls	T	98–99% top-1 targeted success (digital); 93% at 1 m (print)	Single reusable patch trained with EOT against Inception & VGG.
DPATCH18 [6]	WB	○	det	B	VOC07 mAP ↓ 65.70% → 0.00% (U); patch-only detections (T)	40×40 px universal patch; cross-model transfer; digital-only.
Thys19 [7]	WB	●	det	U	YOLOv2 person AP ↓ 100% → 25.5%	40×40 cm printed patch; needs torso placement.
Zolfi21 [15]	WB	●	det	U	Stop-sign miss rate 42.3% (physical); AP ↓ 95.17% → 52.71%	Translucent lens sticker; hides a selected class without scene changes.
Lapid23 [12]	BB	○	det	U	AP 23.2% (↓ 43 pp) within 110 k queries	GAN-style universal patch; query-efficient NES; no physical test yet.
PATCHBANDIT (ours)	BB	●	det	U	77.5% strict suppression under our protocol; 8.3% median patch area	Instance-specific, size-adaptive patch; bandit placer + NES.

● Yes ○ No Atk: T = Targeted, U = Untargeted, B = Both

overlays that require fixed footprints and long optimization. [12] move to the same score-only interface but use a GAN latent search that lowers Tiny-YOLOv3 average precision to 23 % at a cost of $\sim 1.1{\times}10^5$ queries. In the same setting, PATCHBANDIT suppresses the original top class on 77.5 % of images with a median of 49 calls and an 8.3 % patch area—over three orders of magnitude fewer queries and a smaller footprint. White-box universals remain useful when gradients are revealed or a single reusable sticker is desired, yet that threat model is less representative of commercial detectors. Under the more restrictive but realistic score-only interface, our results show that near-complete suppression is attainable with limited queries and an instance-specific, size-adaptive patch. Experiments with the patch placed on a bottle confirm that the attack transfers across varying viewing distances and angles.

3. METHODOLOGY

Algorithm 1 summarizes PATCHBANDIT, a query-bounded black-box patch attack that jointly optimizes (i) where to place a patch, (ii) the patch texture, and (iii) the patch size. Each iteration selects a candidate location via contextual Thompson sampling, updates patch pixels with NES-style zeroth-order optimization, and grows the patch when progress stalls.

Threat model and objective. We consider a score-based black-box detector f queried on an RGB image $I \in [0,1]^{H\times W\times 3}$, returning detections $\{(b_j, c_j, s_j)\}$ with confidences $s_j \in [0,1]$. For any class c, let $c(I,c) = \max_{j:c_j=c} s_j$ denote its maximum confidence (0 if absent). We define the initial top class as $t = \arg\max_c c(I,c)$ with clean score $s_0 = c(I,t)$. The attacker chooses a square patch $\theta \in [0,1]^{3\times m\times m}$ and a paste location $\ell = (x,y)$, obtaining $I' = \text{PASTE}(I,\theta,\ell)$. Success is measured by strict suppression of the original top class, using only score queries to f (no gradients or parameters).

Patch pasting and constraints. Given θ and top-left ℓ, we overwrite the corresponding image crop; candidates that spill outside the image are discarded. In digital mode we clamp $\theta \in [0,1]$; in printable mode we clamp to $[\theta_{\min}, \theta_{\max}]$.

Candidate locations and context features. We discretize the image into a $G \times G$ grid of feasible top-left locations $\mathcal{L}$. Each location ℓ has a lightweight context vector $\phi(\ell) \in \mathbb{R}^4$ capturing proximity/overlap with the nearest ground-truth box, local texture variance, and whether the cell touches the border. These features bias search toward object-bearing, high-contrast regions that tend to yield stronger suppression [6].

Contextual Thompson sampling for placement. We model the per-location reward as a linear function of context, $r = \langle w_\star, \phi(\ell)\rangle + \xi$ with $\xi \sim \mathcal{N}(0,\sigma^2)$ and prior $w \sim \mathcal{N}(0, \lambda^{-1}I)$. At iteration t, we sample $\tilde{w} \sim \mathcal{N}(\mu_t, \Sigma_t)$ from the posterior and select $\ell_t = \arg\max_{\ell\in\mathcal{L}} \langle \tilde{w}, \phi(\ell)\rangle$, then update the posterior with the observed reward. (We use w for bandit weights to avoid overloading the patch symbol θ.)

NES-style patch update. With the chosen location fixed, we update patch pixels via Natural Evolution Strategies (NES) [17]. Each iteration samples M perturbations $\epsilon_i \sim \mathcal{N}(0,I)$, evaluates the reward for candidate patches $\theta^{(i)} = \Pi(\theta + \sigma_{\text{nes}}\epsilon_i)$, and applies a stochastic ascent step to improve θ under the same clamping operator $\Pi(\cdot)$.

Reward and appearance terms. For the core digital setting shown in Alg. 1, the reward is the reduction in the original top-class confidence: $r(\theta,\ell) = s_0 - c(\text{PASTE}(I,\theta,\ell), t)$. When desired, we augment this objective with (i) an EOT-averaged confidence term [13] and (ii) optional appearance/printability penalties to expose strength–visibility trade-offs.

Adaptive size ladder. We use a size ladder $\mathcal{S} = \{m_1 < \cdots < m_L\}$. Optimization starts at m_1 and tracks the best reward so far; if reward fails to improve for τ_{grow} iterations, we upsample θ to the next size and rebuild the placement grid. This grows patch area only when needed.

Query budget and strict success. Each iteration evaluates

Algorithm 1 PATCHBANDIT: Contextual Bandit + NES Patch Attack

Require: Image I, detector f, iterations T, size ladder $\mathcal{S} = \{m_1 < \cdots < m_L\}$, growth patience τ_{grow}, NES $(M, \sigma_{\text{nes}}, \eta)$, success thresholds $(\Delta_{\min}, \tau_{\text{abs}})$

Target class $t \leftarrow \arg\max f(I)$ with score s_0; init $\theta \sim \mathcal{U}[0,1]^{3\times m_1 \times m_1}$, grid $\mathcal{L}$ with features $\phi(\ell) \in \mathbb{R}^4$, bandit posterior (A, b), stall counter

for iter $= 1$ **to** T **do**

  Sample $\tilde{w} \sim \mathcal{N}(A^{-1}b, \sigma^2 A^{-1})$; pick $\ell \leftarrow \arg\max_{\ell' \in \mathcal{L}} \langle \tilde{w}, \phi(\ell') \rangle$

  Draw $\{\epsilon_i\}_{i=1}^{M} \sim \mathcal{N}(0, I)$; evaluate rewards $r_i \leftarrow s_0 - c\big(f(\text{PASTE}(I, \Pi(\theta + \sigma_{\text{nes}}\epsilon_i), \ell)), t\big)$

  Update $\theta \leftarrow \Pi(\theta + \frac{\eta}{M}\sum_i \tilde{r}_i \epsilon_i)$ $\{\tilde{r}_i$ z-scored$\}$

  $r^\star \leftarrow \max_i r_i$; if $(s_0 - c^\star) \geq \Delta_{\min}$ and $c^\star \leq \tau_{\text{abs}}$:

  return (θ, ℓ)

  If no improvement for τ_{grow} iters and $m < m_L$: grow patch via UPSAMPLE and rebuild $\mathcal{L}$

  Update bandit: $A \leftarrow A + \phi(\ell)\phi(\ell)^\top$, $b \leftarrow b + r^\star \phi(\ell)$

end for

return best (θ, ℓ) found

M patch candidates (and optionally additional transformed views); the run halts when the query cap is reached. We declare success only when the learned patch reduces the original top-class confidence by at least $\Delta_{\min}$ and pushes it below an absolute cap τ_{abs}:

$$(s_0 - c_{\text{plain}}(\theta, \ell)) \geq \Delta_{\min} \quad \wedge \quad c_{\text{plain}}(\theta, \ell) \leq \tau_{\text{abs}}. \tag{1}$$

We use $\Delta_{\min} = 0.25$ and $\tau_{\text{abs}} = 0.10$ in all experiments.

4. EVALUATION

4.1. Digital Results

Experimental palette. We evaluate 18 controlled variants of PATCHBANDIT that each toggle a single design knob (growth patience, query schedule, context masking, budget stress tests, stealth bias, EOT robustness, and fixed-size controls). Unless stated otherwise, results use the same protocol as the baseline; selected settings are repeated across seeds to assess variability.

Effect on headline metrics. Table 2 summarizes how each representative *configuration family* performs on digital-domain attack metrics: strict success rate, target-score drop, patch area, and number of detector queries. Clear trade-offs emerge across YOLO, Faster R-CNN, and YOLOS. For instance, `gp5` matches the baseline's success rate while roughly halving the median queries on YOLO (1079→529) due to shortened growth patience, with only a small area increase on Faster R-CNN. `nocx_strict` slightly outperforms the baseline on YOLO despite removing contextual placement, while it is nearly tied on Faster R-CNN. On YOLOS, strict success is lower than on the CNN detectors but remains substantial, with PATCHBANDIT reaching 59.1%. Conversely, `fixed160` succeeds using minimal queries but requires significantly larger patches. Robustness and stealth-oriented settings (`eot10`, `stealth-strong`) maintain high success with query counts that are broadly comparable to the baseline.

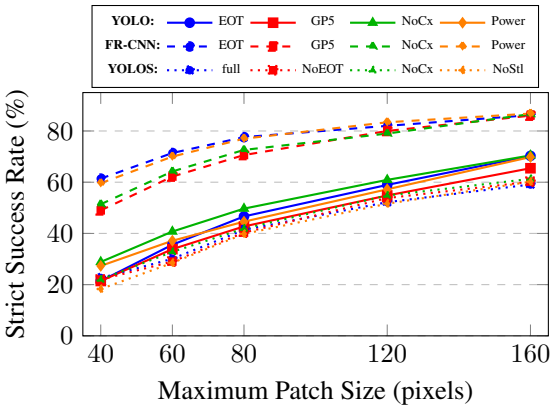


Fig. 2: Cumulative success vs. patch size across three detector architectures. **Solid**: YOLO (one-stage CNN). **Dashed**: Faster R-CNN (two-stage CNN). **Dotted**: YOLOS (ViT-backbone transformer). Adaptive variants reach roughly 43–50% by 80 px and 65–70% at 160 px on YOLO, 70–80% by 80 px and 86–87% at 160 px on Faster R-CNN, and 40–42% by 80 px before plateauing near 60% at 160 px on YOLOS.

Efficiency trade-offs. Given consistent mAP degradation, we now assess the effort per successful attack. Table 2 combines median detector queries and patch area into the joint cost $J_{\text{frac}} = q_{\text{med}} \times (a_{\text{med}}/100)$. On YOLO, the non-adaptive `fixed160` attains the lowest J by accepting conspicuously large patches; among *adaptive* variants, `gp5` substantially reduces J relative to the baseline by halving queries without enlarging patches. On Faster R-CNN, the baseline achieves the lowest J thanks to very small patches; `powerfast` and `eot10` remain competitive. On YOLOS, `gp5` offers the best adaptive efficiency, while `fixed160` achieves the lowest overall J at the cost of a larger footprint. Overall, minimizing queries alone is insufficient if patch size remains large, highlighting the value of adaptive growth when visibility matters.

Growth dynamics. Figure 2 shows how success accumulates as the patch grows. On YOLO (solid), variants reach 20–30% at 40 px, rise to 45–50% by 80 px, and reach 65–70% at full size. Faster R-CNN (dashed) climbs faster: 50–60% at 40 px, $>70\%$ by 60 px, and plateaus near 85–87% by 120 px. YOLOS (dotted) follows a slower but steady trajectory, reaching about 40–42% by 80 px and plateauing near 60% at 160 px. These trends highlight the benefit of adaptive growth: many images succeed at small or moderate patch sizes, with larger patches used only when necessary. GP5's steeper early rise reflects reduced growth patience; fixed-size baselines (omitted) capture only terminal outcomes.

Table 2: Digital attack performance and efficiency. **Performance columns** (blue): Strict success (%), confidence drop, patch area (%), queries. **Efficiency columns** (orange): queries/success, area/success, joint cost $J_{\text{frac}} = q_{\text{med}} \times (a_{\text{med}}/100)$.

Variant	YOLO (one-stage CNN)							Faster R-CNN (two-stage CNN)							YOLOS (ViT transformer)						
	Performance				Efficiency			Performance				Efficiency			Performance				Efficiency		
	Strict	Drop_t	Area	Q.	Q./s	A./s	J_{frac}	Strict	Drop_t	Area	Q.	Q./s	A./s	J_{frac}	Strict	Drop_t	Area	Q.	Q./s	A./s	J_{frac}
PATCHBANDIT (ctx-strict)	77.5	0.452	8.32	1079	63.26	10.74	6.794	89.7	0.075	0.92	177	8.92	1.03	**0.092**	59.1	0.467	8.32	290	149.95	4.58	24.089
GP5 (fast growth)	77.7	0.455	8.32	529	30.88	10.71	3.307	90.0	0.080	2.08	155	7.78	2.31	0.180	60.8	0.558	14.79	47.5	32.88	8.13	7.027
Nocx_Strict ε-greedy	78.9	0.406	8.32	1013	58.32	10.55	6.153	89.6	0.079	1.31	353	17.86	1.46	0.261	61.3	0.502	8.32	283	157.34	4.56	23.548
PowerFast (top-swap)	69.8	0.531	8.32	903	58.74	11.92	7.002	86.8	0.077	0.92	177	9.22	1.06	0.098	53.8	0.430	8.32	245	167.00	5.20	20.384
Fixed (no adapt)	71.1	0.585	14.8	133	8.44	20.82	**1.757**	79.6	0.087	14.8	111	6.28	18.59	1.167	55.4	0.777	14.79	35.0	15.49	14.79	**5.178**
EOT variant	79.3	0.416	8.32	1582	64.33	10.49	6.748	87.6	0.070	0.92	249	9.13	1.05	0.096	60.2	0.492	8.32	288	151.77	4.72	23.964
Stealth variant	78.4	0.425	8.32	1035	59.95	10.62	6.367	89.0	0.075	0.92	188	9.55	1.03	0.098	60.2	0.584	8.32	297	161.04	5.07	24.713

Table 3: Fixed-size patch ablation (digital, contextual placement). Efficiency cost $J_{\text{frac}} = q_{\text{med}} \times (a_{\text{med}}/100)$.

Detector	Config	Strict (%)	Drop_t	Area (%)	Queries	J_{frac}
YOLO	method full	77.5	0.452	8.32	49.0	4.080
	fixed–160	71.1	0.585	14.79	6.0	0.887
	fixed–80	48.2	0.810	3.70	20.0	**0.740**
Faster R-CNN	method full	89.7	0.075	0.92	8.0	**0.074**
	fixed–160	79.6	0.087	14.79	5.0	0.740
	fixed–80	66.2	0.102	3.70	5.0	0.185
YOLOS	method full	59.1	0.467	8.32	290.0	24.089
	fixed–160	55.4	0.777	14.79	35.0	**5.178**
	fixed–80	36.0	0.974	3.70	200.0	7.396

4.2. Ablation Studies

Fixed–size patches. To isolate the value of adaptive growth, we compare PATCHBANDIT to fixed 160 px and 80 px patches. Table 3 shows that adaptive growth generally preserves higher strict success with a smaller footprint than fixed–160, while fixed-size settings reduce J_{frac} only by lowering success substantially (fixed–80) or requiring larger patches (fixed–160).

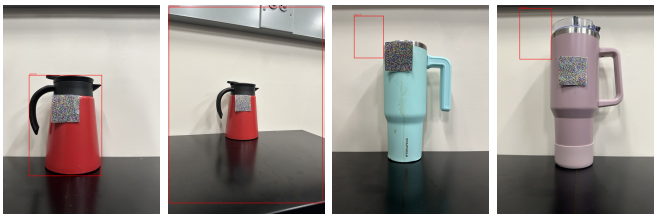
Ablation: growth patience and query schedule. Reducing growth patience (`gp5`) roughly halves YOLO median queries (1079→529) with essentially unchanged strict success (77.5→77.7) and unchanged area (8.32%). By contrast, `powerfast` yields a smaller query reduction (1079→903) but lowers strict success (77.5→69.8), indicating that shortcut schedules can trade effectiveness for efficiency.

4.3. Physical Pilot Evaluation

To assess preliminary physical transfer, we printed a `bottle` patch optimized entirely in simulation and evaluated it under controlled changes in viewpoint, distance, and object instance. Table 4 summarizes the print–capture outcomes, and Figure 3 shows representative captures. In clean shots, YOLOv5 detected `bottle` with confidence > 0.90; with the patch, the tested conditions produced misclassification, low-confidence detection, or incorrect localization.

Table 4: Print–capture pilot outcomes for the physical PATCHBANDIT bottle patch. Patched outcomes report the top YOLOv5 prediction.

Condition	Distance	Azimuth	Object	Prediction	Outcome
Frontal view	40 cm	0°	Original	`chair` (0.59)	Misclassified
Tilted view	50 cm	10°	Original	`tvmonitor` (0.35)	Misclassified
Transfer #1	54 cm	0°	New #1	*bottle* (0.41)	Wrong box / low conf.
Transfer #2	54 cm	0°	New #2	*bottle* (0.81)	Wrong box



(a) Frontal (b) Tilted (c) Transfer #1 (d) Transfer #2

Fig. 3: Representative print–capture results for the physical PATCHBANDIT bottle patch. Quantitative outcomes are reported in Table 4.

5. CONCLUSION

PATCHBANDIT combines contextual Thompson-sampling placement, NES zeroth-order updates, and progress-triggered growth into a score-only attack loop. Using a strict plain-view suppression criterion and reporting EOT separately, it achieves strong suppression across CNN-based detectors and substantial suppression on a transformer-based detector, while exposing query–footprint trade-offs against fixed-size and heuristic baselines. Compact, budget-aware patches remain a realistic black-box threat, with placement playing a key role in attack efficiency.

Acknowledgments

This work was supported in part by a grant from The BMW Group.

6. REFERENCES

[1] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[2] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[3] Yuxin Fang, Bencheng Liao, Xinggang Wang, Jiemin Fang, Jiyang Qi, Rui Wu, Jianwei Niu, and Wenyu Liu, "You only look at one sequence: Rethinking transformer in vision through object detection," *Advances in Neural Information Processing Systems*, vol. 34, pp. 26183–26197, 2021.

[4] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[5] Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer, "Adversarial patch," *arXiv preprint arXiv:1712.09665*, 2017.

[6] Xin Liu, Huanrui Yang, Ziwei Liu, Linghao Song, Hai Li, and Yiran Chen, "Dpatch: An adversarial patch attack on object detectors," *preprint arXiv:1806.02299*, 2018.

[7] Lien Thys, Wiebe Van Ranst, and Toon Goedemé, "Fooling automated surveillance cameras: Adversarial patches to attack person detection," in *CVPR Workshops*, 2019.

[8] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin, "Black-box adversarial attacks with limited queries and information," in *International conference on machine learning*. PMLR, 2018, pp. 2137–2146.

[9] Andrew Ilyas, Logan Engstrom, and Aleksander Madry, "Prior convictions: Black-box adversarial attacks with bandits and priors," *preprint arXiv:1807.07978*, 2018.

[10] Siyuan Liang, Baoyuan Wu, Yanbo Fan, Xingxing Wei, and Xiaochun Cao, "Parallel rectangle flip attack: A query-based black-box attack against object detection," *arXiv preprint arXiv:2201.08970*, 2022.

[11] Xingxing Wei, Ying Guo, Jie Yu, and Bo Zhang, "Simultaneously optimizing perturbations and positions for black-box adversarial patch attacks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, no. 7, pp. 9041–9054, 2022.

[12] Raz Lapid, Eylon Mizrahi, and Moshe Sipper, "Patch of invisibility: Naturalistic physical black-box adversarial attacks on object detectors," *arXiv preprint arXiv:2303.04238*, 2023.

[13] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok, "Synthesizing robust adversarial examples," in *International conference on machine learning*. PMLR, 2018, pp. 284–293.

[14] Chenglin Yang, Adam Kortylewski, Cihang Xie, Yinzhi Cao, and Alan Yuille, "Patchattack: A black-box texture-based attack with reinforcement learning," in *European Conference on Computer Vision*. Springer, 2020.

[15] Alon Zolfi, Moshe Kravchik, Yuval Elovici, and Asaf Shabtai, "The translucent patch: A physical and universal attack on object detectors," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 15232–15241.

[16] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh, "Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models," in *Proceedings of the 10th ACM workshop on artificial intelligence and security*, 2017, pp. 15–26.

[17] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin, "Black-box adversarial attacks with limited queries and information," in *ICML*, 2018.

[18] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein, "Square attack: a query-efficient black-box adversarial attack via random search," in *European conference on computer vision*. Springer, 2020, pp. 484–501.

[19] Lifeng Huang, Chengying Gao, Yuyin Zhou, Cihang Xie, Alan L Yuille, Changqing Zou, and Ning Liu, "Universal physical camouflage attacks on object detectors," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 720–729.

[20] Satyadwyoom Kumar, Saurabh Gupta, and Arun Balaji Buduru, "Bb-patch: Blackbox adversarial patch-attack using zeroth-order optimization," *arXiv preprint arXiv:2405.06049*, 2024.

[21] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song, "Robust physical-world attacks on deep learning visual classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1625–1634.