Geometry-Aware Patch Attacks: ShapeAware and PatchCross

Ahmed Abdelkader Ahmed
Student ID: g2025224490
King Fahd University of Petroleum and Minerals
Dhahran, Saudi Arabia

Supervised by: Dr. Muzammil Behzad
muzammil.behzad@kfupm.edu.sa
King Fahd University of Petroleum and Minerals
Dhahran, Saudi Arabia

***Abstract*—Adversarial *patch* attacks on deep vision models are typically black-box and fix geometry (e.g., squares), optimizing only texture and placement. We treat geometry as a first-class decision and introduce two PatchAttack variants: *ShapeAware*, which adds a shape head with an area-aware reward, and *PatchCross*, which specializes to a 1-pixel X for strong camouflage. Under matched query/placement budgets and early stopping, we evaluate on ImageNet with ResNet-50 and ViT-B/16, reporting ASR, ANQ, and APA. ShapeAware matches or improves ASR while consistently reducing queries and union area relative to PatchAttack; PatchCross attains near-perfect ASR with minimal area. Overall, explicit geometry—via compositional shapes or a specialized cross—yields more efficient, compact black-box patch attacks.**

***Index Terms*—Adversarial machine learning, Black-box attacks, Patch attacks, Reinforcement learning, ImageNet**

I. Introduction

A. Background and Significance

Deep learning models have demonstrated remarkable efficiency and capability across a wide range of applications and domains [1]–[3]. However, these models are also vulnerable to carefully crafted adversarial examples that induce erroneous predictions [4]–[6].

In computer vision (CV), adversarial attacks typically modify a small region of pixels to mislead the model. This creates significant security and dependability gaps, particularly as many CV systems now operate without human supervision. Such attacks have been demonstrated across key tasks, including object detection [7], face detection [8], autonomous driving [9], and other safety-critical ML systems [10].

Based on whether a model's internals are accessible, adversarial attacks are commonly categorized as *white-box* (the attacker has access to the model's parameters/architecture) or *black-box* (the attacker can only query the model). White-box attacks such as Fast Gradient Sign Method (FGSM) [5], DeepFool [11], Projected Gradient Descent (PGD) [12], and logit-space projected gradient ascent (LS-PGA) for discrete inputs [13] have achieved strong results. However, unrestricted gradient access is often unrealistic; consequently, black-box settings with limited query budgets constitute a more practical threat model.

Along another axis, attacks can be grouped by how perturbations are distributed: *perturbation attacks* apply small, typically L_∞-bounded changes spread across the image (e.g., FGSM/PGD), whereas *patch attacks* localize changes within a bounded region [6]. Patch attacks are particularly realistic because patches can be printed as physical stickers and remain effective under changes in viewpoint and illumination [14]. Thus, black-box adversarial patch attacks constitute a highly realistic threat model for computer-vision systems [6], [14]. Moreover, by objective, attacks are categorized as *untargeted*—which aim to pull the prediction away from the ground-truth label—and *targeted*—which aim to push a specific label [5].

Taken together, these considerations motivate a focus on *black-box adversarial patch attacks*: they align with realistic attacker capabilities (no gradient access), capture physical-world feasibility (printable artifacts that withstand viewpoint and illumination changes), and directly stress unattended CV pipelines deployed behind APIs with strict query limits. Progress in this setting—especially under area and query budgets—provides a more policy-relevant and safety-critical measure of robustness than white-box-only evaluations [6], [10], [14].

B. Challenges in Current Techniques

In black-box patch attacks, prior work has largely optimized the patch's content (noise/texture) and placement, with comparatively little emphasis on *shape* or on the *camouflage ability* of patches. This gap matters in practice: realistic attacks must navigate area constraints and visual plausibility while maintaining query efficiency, suggesting that explicit shape optimization could improve outcomes under fixed query budgets and tighter localization.

PatchAttack [15] exemplifies this design space: an RL agent selects a patch texture from a predefined, class-conditioned dictionary and chooses where to place it on the image. Crucially, *area and shape are excluded from the action space*—the maximum area is a hyperparameter, and the patch shape is fixed (square). While the approach demonstrates strong success rates, it leaves open whether incorporating geometric decisions (shape choice/composition) could further reduce queries or area for comparable success, especially in targeted settings.

Another relevant line is the cross-shaped patch attack (CSPA) [16], which probes how uncommon shapes influence attack success and the number of queries. Their results highlight that a cross-shaped geometry can be particularly effective and query-efficient among tested alternatives in black-box scenarios. Nevertheless, *targeted* attacks remain more challenging than untargeted ones in the black-box patch regime, often demanding either more queries or greater effective area—underscoring the need for action spaces and rewards that *jointly* reason about shape, overlap/composition, and area usage.

Motivation. These observations motivate expanding black-box patch action spaces beyond texture and location to include shape and to pair this with reward designs that explicitly trade off confidence reduction against used area. Such changes aim to reduce the Average Patch Area (APA), while maintaining Average Number of Queries (ANQ) and Average Success Rate (ASR).

C. Problem Statement

Black-box evasion can be framed as a Markov decision process (MDP) in which an agent iteratively selects perturbation actions from a predefined (typically discrete) action space and observes feedback from the victim model via queries [17], [18]. After each action, the agent receives a reward shaped from the model's outputs (e.g., reduced confidence in the true class) and updates its policy accordingly. A central design tension is *action-space calibration*: too small an action space restricts exploration and yields suboptimal attacks, while overly large spaces, when coupled with weak or noisy rewards, can destabilize learning and impair sample efficiency [19].

In the specific case of black-box *patch* attacks, most prior work optimizes patch *content* (noise/texture) and *location*, with comparatively little emphasis on *shape* or the *camouflage* and compactness of the patch under area constraints.

Research gaps. Current black-box patch methods typically omit s (shape) from the action space and treat Area as a hard cap rather than a learned trade-off, limiting realism and efficiency under tight budgets. Moreover, specialized geometries (e.g., cross-shaped) have been explored in isolation rather than within a unified RL formulation that jointly reasons about texture, placement, and geometry.

To concretely address these gaps, we instantiate two alternatives: (1) *ShapeAware*, which expands the action space with shape primitives and allows multi-primitive compositions, paired with an area-penalized reward to encourage compact, overlapping configurations; and (2) *PatchCross*, which specializes the action space to an X-shaped geometry (parameterized by two anchor points), both under identical query/area budgets and early stopping. These instantiations provide controlled testbeds for evaluating the contribution of explicit geometry and area-aware rewards in black-box patch attacks.

Threat model and access. Black-box, score-based image classification: the attacker can query the model and read scores/logits; no gradients or internals are available. Both untargeted and targeted objectives are considered.

Objectives. We aim to quantify the value of geometry by measuring the effect of making shape a first-class decision variable on ASR and ANQ, and to compare two geometry paradigms—compositional shape-aware actions (circles/triangles/squares) versus a specialized X-shape—under matched settings.

II. Literature Review

A. Overview of Existing Techniques

Adversarial evasion methods can be organized along three axes that matter for our study: (i) *threat model* (white-box vs. black-box), (ii) *locality* of the perturbation (L_p sparse noise vs. localized patches), and (iii) *core search strategy* (gradient-based, randomized, or reinforcement learning).

White-box perturbations. With full access to model gradients, classical methods such as *DeepFool* (minimal L_2 via iterative linearization) and *PGD* (projected steps within an L_∞ ball) remain standard baselines for benchmarking robustness [11], [12].

Black-box perturbations. Square Attack achieves strong untargeted success with low average queries using randomized, localized square updates [20]. Reinforcement-learning approaches cast evasion as an MDP: *RLAB* trains an agent to add/remove localized tiles efficiently [21], while *Adversarial Agents* shows that learning across episodes reduces queries and increases success [18]. In the video setting, *Sparse Black-box Video Attack* selects key frames/regions via RL to cut queries while maintaining high success [22].

White-box patch attacks. Patch methods constrain changes to a compact mask. *Adversarial Patch* demonstrated universal, printable targeted stickers trained by gradient descent [6]. For detection, *DPATCH* optimizes patch content and placement to collapse mAP in object detectors [23].

Black-box patch attacks. Absent gradients, search structure is crucial. Early randomized approaches such as the *Hastings Patch Attack* (HPA) explore monochrome rectangular patches but are often query intensive [24]. *PatchAttack* reframes patch placement and texture choice as RL decisions under area constraints, achieving high success with learned, class-conditioned textures [15]. More recently, geometry itself has been probed: the *Cross-Shaped Patch Attack* explores non-rectangular masks and reports competitive success under budgets, highlighting the potential of shape as an additional degree of freedom [16]. Another common black-box patch attack is *DevoPatch* [25]. It operates in the decision-based (label-only) setting, formulates patch variables in the integer domain, and uses differential evolution over a key-point parameterization—yielding strong ASR with low ANQ, at the cost of high APA.

Takeaway for this work. Across these families, black-box *patch* attacks typically optimize *content* and *location* and treat *shape* and area usage as secondary (often fixed-shape with a hard area cap). This motivates action spaces and rewards that explicitly reason about geometry (including shape composition) and compactness.

B. Related Work

Hastings Patch Attack (HPA) HPA [24] proposes monochrome rectangular patches and applies a Metropolis–Hastings update over a low-dimensional parameterization.

Each proposal is rendered onto the image and evaluated by querying the victim model; HPA is a simple technique serving as a baseline for black-box patch attacks. However, proposals explore a narrow texture/geometry space (monochrome rectangles), tend to require many queries to escape local optima (ANQ high), and, when successful, often cover relatively large regions (APA high), which undermines camouflage.

PatchAttack (score-based RL). PatchAttack [15] formulates black-box patch search as reinforcement learning over a discrete action space. Using a recurrent Long Short-Term Memory (LSTM) network [26], the agent iteratively (i) places a patch, (ii) queries the victim deep convolutional neural network (DCNN), (iii) receives a confidence-based reward, and (iv) updates its parameters online. Training proceeds on-the-fly during attack process. The process repeats until success or a maximum number of patches is reached. PatchAttack introduces two variants: MPA (Monochrome Patch Attack) and TPA (Texture-based Patch Attack). Since TPA outperforms MPA, we adopt TPA throughout; unless otherwise stated, "PatchAttack" refers to the TPA variant. TPA's action space consists of (i) a class-conditioned texture dictionary (prebuilt from training data) and (ii) 2D placement for a square mask. At each step, the agent selects a texture and placement, renders the patch, queries the model for scores/logits, and receives a reward derived from confidence changes (e.g., lowering p_{y^*} for untargeted or raising p_t for targeted), with early stopping upon success. Area is enforced as a hyperparameter, and geometry is fixed to a square. PatchAttack demonstrates strong ASR with moderate budgets—the dictionary stabilizes learning and captures class-specific textures, and score access enables shaped rewards and effective early stopping. A key limitation is that shape is not a decision variable, and compactness is not learned (hard cap rather than a trade-off), which can inflate APA or ANQ under tight budgets.

Cross-Shaped Patch Attack (CSPA) CSPA [16] probes geometry as a lever by constraining the mask to a cross (two orthogonal bars). Its parameterization typically includes center, orientation, bar widths, and lengths; search is gradient-free under a query budget. The rationale is that non-rectangular masks can align with salient structures and disrupt features more efficiently than squares. CSPA demonstrates that geometry influences black-box patch efficacy, with competitive success and query usage under budgets, but reported targeted ASR is typically lower than untargeted and lower than many alternative attacks.

DevoPatch (decision-based, differential evolution) DevoPatch [25] addresses the label-only setting by casting patch search in the integer domain and optimizing via differential evolution over a key-point parameterization (e.g., paired anchors controlling patch extent and placement). The algorithm initializes candidates—often seeded from target-class images—then iteratively applies mutation and crossover, querying only the top-1 label to compute a fitness signal and select improved candidates. It achieves high ASR with low ANQ; unfortunately, the reported APA is high—around 25% of the image on average— which is a bit unrealistic in real physical settings.

Collectively, these black-box patch attacks highlight complementary strengths: RL with scores (PatchAttack) offers stable learning and early stopping; geometry specialization (CSPA) suggests shape matters; and decision-based evolution (DevoPatch) attains low ANQ without scores. Their common gaps—fixed or specialized geometry, hard area caps (rather than learned compactness), and limited camouflage control—motivate shape-aware action spaces and area-aware reward shaping evaluated with ASR, ANQ, and APA.

C. Gaps in Existing Approaches

G0. Patch geometry is not treated as a first-class decision.Most black-box patch attacks fix the mask to a square and optimize only texture and placement [15], and many white-box works train universal square/rectangular patches [6]. Recent non-square designs indicate that geometry matters [16], yet there is no systematic, shape-aware search in the black-box setting that learns when and how to select or compose shapes.

G1. RL action-space and reward design lack a principled study. Existing formulations hand-pick discrete actions and use simple rewards; the trade-off between too small and too large action spaces is rarely quantified, leading to variability in convergence and stability [19].

G2. Reporting and metrics are inconsistent. Papers mix protocols and metrics (ASR, ANQ, APA), which weakens cross-paper comparison. For example, DevoPatch achieves low ANQ at the cost of high APA, reflecting a clear efficiency–compactness trade-off [25].

Together, these gaps motivate making geometry a learned decision and coupling RL with area-aware reward shaping to encourage compactness, aiming to preserve ASR while reducing ANQ and APA under matched query and placement budgets with early stopping.

III. Proposed Methodology

We retain the PatchAttack backbone—score-based RL with a fixed texture dictionary, early stopping, and uniform query/area budgets—and restrict our study to the targeted setting. On this backbone, we introduce two variations that elevate geometry to a learned decision and encourage compactness via an area-aware reward: (i) *ShapeAware* and (ii) *PatchCross*. Figure 1 shows some examples of targeted attack on ResNet-50 using PatchAttack, ShapeAware, and PatchCross.

ShapeAware:

Action-space expansion. Add a shape head to the policy with classes {circle, triangle, square}, enlarging the original PatchAttack (TPA) action space. Multiple agents (or sequential steps) add patches until success or a maximum limit is reached; each agent chooses the shape and pose of its primitive. The rendered mask is the union of all shapes added by all agents.

Area-aware reward. Augment the confidence-based reward with an area penalty controlled by λ. The penalty uses the

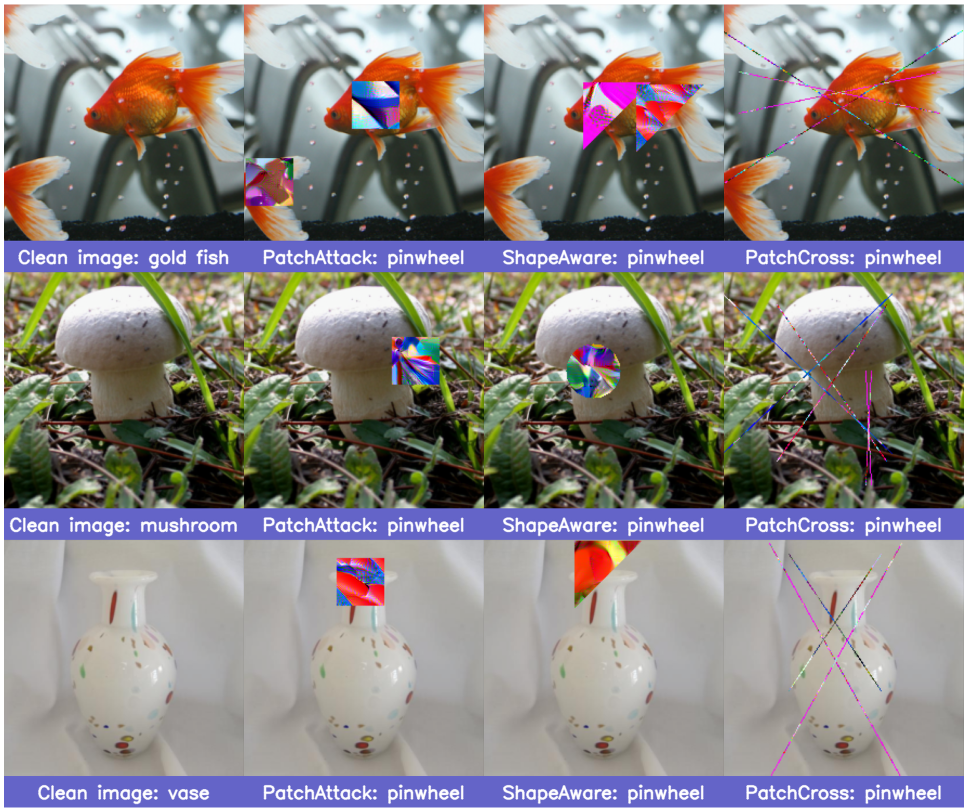

Fig. 1. Targeted examples across multiple classes (Clean, PatchAttack, ShapeAware, PatchCross). Predicted labels are annotated beneath each panel. For PatchAttack and ShapeAware, the occlusion area is set to 4% of the image.

union area contributed by all agents up to the current step. Concretely, with $P^{(i)} = \bigcup_{j \le i} P_j$,

$$r_i \;=\; \underbrace{\log\big(f(x \oplus P^{(i)})\big)}_{r_{\text{logits}}} - \lambda\,\mathrm{Area}\big(P^{(i)}\big),$$

where P_j is the *binary mask added by agent* j, and $\mathrm{Area}\big(P^{(i)}\big)$ counts the 1-valued pixels in $P^{(i)}$ as a fraction of the image area. Implementation overview for ShapeAware is shown in Algorithm 1.

PatchCross:

Cross-shape adaptation. Specialize geometry to an X-shaped mask (two intersecting bars). The policy outputs two anchor coordinates $\mathbf{a}^{(1)} = (x_{\mathrm{UL}}, y_{\mathrm{UL}})$ and $\mathbf{a}^{(2)} = (x_{\mathrm{LR}}, y_{\mathrm{LR}})$ defining the upper-left and lower-right corners of a bounding box. The X is drawn as the union of its two diagonals, $\overline{\mathrm{UL}\to\mathrm{LR}}$ and $\overline{\mathrm{UR}\to\mathrm{LL}}$, with width 1 pixel. The texture aligns with these anchors (no separate texture-crop pose).

Reward. With 1-pixel-wide bars, patch area is negligible, so we keep the original PatchAttack targeted reward based on target-class confidence,

$$r_i \;=\; \log\big(f(x \oplus P^{(i)})\big),$$

Implementation overview for PatchCross is shown in Algorithm 2.

IV. Experimental Design and Evaluation

A. Dataset and Preprocessing

Test dataset and victim models. We evaluate on the ImageNet ILSVRC 2012 validation set [27] using two pretrained classifiers: ResNet-50 [1] and ViT-B/16 [28]. For ResNet-50 we uniformly sample 150 images; for ViT-B/16 we uniformly sample 50 images. We first report clean top-1 accuracy on each model's subset, then ASR, ANQ, and APA for PatchAttack, ShapeAware, PatchCross.

Inputs. Each image is resized to 224×224, converted to $[0, 1]$, and normalized using the victim model's ImageNet mean/std.

Textures. All patch appearances are sampled from the PatchAttack texture dictionary released by the authors, used as-is. Briefly (see the original paper for full details), the dictionary is built on a VGG-19 backbone by extracting style descriptors via Gram matrices from selected convolutional layers, applying Grad-CAM [29] to focus on salient regions, flattening the descriptors, and clustering them per class using K-means (typically $k{=}30$) to obtain a texture embedding; one representative texture is then synthesized per centroid to form the final dictionary [15].

B. Experimental Setup

Budgets. Each episode allows up to $N{=}10$ placement decisions (agents). In the case of PatchAttack and ShapeAware,

Algorithm 1 ShapeAware (targeted)

```
Input: image $x \in [0,1]^{H\times W\times C}$, target $t$, query budget $Q$,
    max placements $N$, per-placement occlusion $area\_occlu$,
    texture dictionary $\mathcal{T}$, victim model $f$, penalty weight $\lambda$,
    stability $\varepsilon$, early-stop bound $\epsilon$
Output: patched image $x^\star$, union patch $P^\star$, success flag $s$,
    queries used $q$
 1: $i \leftarrow 0$;  $q \leftarrow 0$;  $P^{(0)} \leftarrow \mathbf{0}$;  $x^{(0)} \leftarrow x$;  $s \leftarrow$false
 2: while $q < Q$ and $i < N$ and $s =$false do
 3:     $\mathbf{a}_i \leftarrow \pi_\theta(\text{state}(x^{(i)}, P^{(i)}, q, t))$    ▷
        $(c, \mathbf{p}^{\text{shape}}, \mathbf{p}^{\text{text}}, k)$; $c \in \{\circ, \triangle, \square\}$, $k \in \{1, \dots, |\mathcal{T}|\}$,
        $\mathbf{p}^{\text{shape}}, \mathbf{p}^{\text{text}} \in [0, 224)^2$
 4:     $(\text{shape}_{i+1}, \text{pose}^{\text{shape}}_{i+1}, ID, \text{pose}^{\text{text}}_{i+1}) \leftarrow \textsc{Decode}(\mathbf{a}_i)$
 5:     $P_{i+1} \leftarrow \textsc{RenderPatch}(\text{shape}_{i+1}, \text{pose}^{\text{shape}}_{i+1}, ID,$
        $\text{pose}^{\text{text}}_{i+1}, area\_occlu, \mathcal{T})$
 6:     $P^{(i+1)} \leftarrow \textsc{Composite}(P^{(i)}, P_{i+1})$
 7:     $x^{(i+1)} \leftarrow x^{(i)} \oplus P_{i+1}$    ▷ paste the new patch
 8:     $z \leftarrow f(x^{(i+1)})$;  $q \leftarrow q + 1$
 9:     $p_t \leftarrow \text{softmax}(z)_t$;  $\bar{r}_i \leftarrow \log(p_t + \varepsilon)$
10:     $r_i \leftarrow \bar{r}_i - \lambda \cdot \text{Area}(P^{(i+1)})$
11:     update $\pi_\theta$ with $r_i$
12:     if $\arg\max z = t$ then
13:         $s \leftarrow$true;  $x^\star \leftarrow x^{(i+1)}$;  $P^\star \leftarrow P^{(i+1)}$;  break
14:     if $i \geq 2$ and $|\bar{r}_i - 2\bar{r}_{i-1} + \bar{r}_{i-2}| < \epsilon$ then
15:         break
16:     $i \leftarrow i + 1$
17: if $s =$false then
18:     $x^\star \leftarrow x^{(i)}$;  $P^\star \leftarrow P^{(i)}$
19: return $(x^\star, P^\star, s, q)$
```

Algorithm 2 PatchCross (targeted)

```
Input: image $x \in [0,1]^{H\times W\times C}$, target $t$, query budget $Q$,
    max placements $N$, texture dictionary $\mathcal{T}$, victim model $f$,
    stability $\varepsilon$, early-stop bound $\epsilon$
Output: patched image $x^\star$, union mask $P^\star$, success flag $s$,
    queries used $q$
 1: $i \leftarrow 0$;  $q \leftarrow 0$;  $P^{(0)} \leftarrow \mathbf{0}$;  $s \leftarrow$false
 2: while $q < Q$ and $i < N$ and $s =$false do
 3:     $\mathbf{a}_i \leftarrow \pi_\theta(\text{state}(x, P^{(i)}, q, t))$    ▷ policy outputs two
        anchors and a texture ID
 4:     $(ID, \mathbf{a}^{(1)}, \mathbf{a}^{(2)}) \leftarrow \textsc{DecodeX}(\mathbf{a}_i)$    ▷
        $\mathbf{a}^{(1)} = (x_{\text{UL}}, y_{\text{UL}})$, $\mathbf{a}^{(2)} = (x_{\text{LR}}, y_{\text{LR}})$
 5:     $P_{i+1} \leftarrow \textsc{DrawX}(\mathbf{a}^{(1)}, \mathbf{a}^{(2)}, \text{width} = 1)$    ▷ two
        diagonals; 1-pixel bars
 6:     $\theta_{i+1} \leftarrow \textsc{LookupTexture}(ID, \mathcal{T})$
 7:     $P^{(i+1)} \leftarrow P^{(i)} \cup P_{i+1}$
 8:     $x^{(i+1)} \leftarrow x \oplus (\theta_{i+1}, P^{(i+1)})$
 9:     $z \leftarrow f(x^{(i+1)})$;  $q \leftarrow q + 1$
10:     $p_t \leftarrow \text{softmax}(z)_t$;  $\bar{r}_i \leftarrow \log(p_t + \varepsilon)$
11:     $r_i \leftarrow \bar{r}_i$
12:     update $\pi_\theta$ with $r_i$
13:     if $\arg\max z = t$ then
14:         $s \leftarrow$true;  $x^\star \leftarrow x^{(i+1)}$;  $P^\star \leftarrow P^{(i+1)}$;  break
15:     if $i \geq 2$ and $|\bar{r}_i - 2\bar{r}_{i-1} + \bar{r}_{i-2}| < \epsilon$ then
16:         break
17:     $i \leftarrow i + 1$
18: if $s =$false then
19:     $x^\star \leftarrow x^{(i)}$;  $P^\star \leftarrow P^{(i)}$
20: return $(x^\star, P^\star, s, q)$
```

each agent uses an area occlusion of 2% of the image. The query budget is Q=12250 per agent. An attack is deemed a failure if either budget is exhausted before the model's prediction is flipped to the target class.

Targets. All attacks are targeted to a fixed class $t = 723$, as the type of the targeted class no effect in comparing alternatives.

Early stopping — in addition, the policy stops when the mean log-reward plateaus. Let $\bar{r}_s$ be the batch-mean reward at step s, We trigger early stopping once $s \geq 2$ and

$$\left| \bar{r}_s - 2\,\bar{r}_{s-1} + \bar{r}_{s-2} \right| < 10^{-4},$$

i.e., the second finite difference of the mean log-reward falls below the bound (configured as $es\_bnd = 10^{-4}$).

Metrics. For each model and method on the test set, we report three metrics under the stated budgets (Q, N, $area\_oclu$): *ASR* — fraction of images for which the attack reaches the target class within the budgets; *ANQ* — average number of queries made by all agents to the victim model; *APA* — average patch area (as a percentage of the image) computed on the union mask of placements made by all agents. ANQ and APA are reported for successful attacks and, separately, for the entire evaluation set, which includes unsuccessful attempts.

C. Results Comparative Analysis

The comparative results for the three attack variants are reported in Table I.

Results interpretation. On ResNet-50, ASR is effectively identical across methods (the 99.33% for PatchCross is negligible given stochasticity). Both ShapeAware and PatchCross cut query counts substantially; PatchCross uses roughly half the queries of PatchAttack while keeping the patch area nearly negligible. On ViT-B/16, ShapeAware attains perfect ASR with a moderate area, whereas PatchAttack degrades sharply in ASR, APA, and ANQ. PatchCross matches ShapeAware's query efficiency with slightly lower ASR, yet achieving that ASR at APA $\approx$ 2% on ViT-B/16 is a strong outcome.

Area–query trade-off. On ResNet-50, we reran the 150-image evaluation while varying the per-step occlusion $area\_occlu$ (denoted O in the figure) as $O \in \{0.02, 0.04, 0.10, 0.25\}$ for PatchAttack and ShapeAware with N=5.

Figure 2 plots APA$_{\text{all}}$ versus ANQ$_{\text{all}}$, with each point annotated by its ASR and O. Increasing O lowers query counts for both methods, and ASR saturates near 100% by $O \approx$ 0.04–0.10. Across all occlusion levels, PatchAttack points lie consistently *above* ShapeAware points, indicating higher query counts, and are also shifted to the *right* (larger area).

TABLE I
COMPARATIVE RESULTS IN THE TARGETED SETTING ON IMAGENET ILSVRC 2012. BUDGETS: N=6 WITH $area\_occlu$=2% FOR PATCHATTACK AND SHAPEAWARE; N=10 FOR PATCHCROSS; QUERY LIMITS Q=12,250 PER AGENT.

Model	**Method**	**ASR(%)**	$\mathrm{ANQ}_{\mathrm{succ}}$	$\mathrm{ANQ}_{\mathrm{all}}$	$\mathrm{APA}_{\mathrm{succ}}$	$\mathrm{APA}_{\mathrm{all}}$
ResNet-50	PatchAttack	100	22K	22K	5%	5%
	ShapeAware	100	15K	15K	4%	4%
	PatchCross	99.33	12K	12K	2%	3%
ViT-B/16	PatchAttack	82	39K	44K	7%	8%
	ShapeAware	98	23K	23K	5%	5%
	PatchCross	96	22K	22K	2%	2%

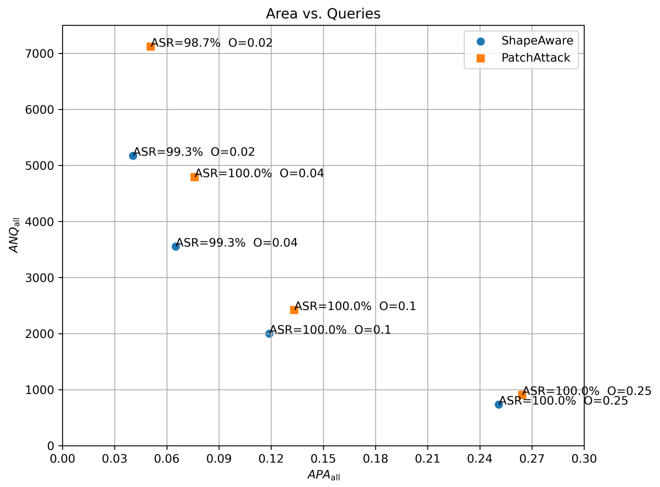

Fig. 2. Area–queries trade-off on ResNet-50 under varying $area\_oclu$ O. Squares: PatchAttack; circles: ShapeAware. X-axis is $\mathrm{APA}_{\mathrm{all}}$, Y-axis is $\mathrm{ANQ}_{\mathrm{all}}$.

This pattern shows that, under matched budgets, ShapeAware dominates PatchAttack in the APA–ANQ trade-off.

Takeaways. As illustrated in Fig. 1, PatchCross, because of its very thin strokes, is likely less noticeable to human observers. ShapeAware tends to guarantee success and adapt to scene content; since image classifiers often lock onto the most salient object, ShapeAware's flexible shapes can cover these regions more effectively than fixed squares.

Overall, shape matters: promoting geometry to a decision variable consistently lowers ANQ under the same per-step occlusion. The ViT-B/16 results amplify this gap, suggesting richer geometry helps on transformer backbones. Finally, for ResNet-50 and for the geometry-aware methods on ViT-B/16, $\mathrm{ANQ}_{\mathrm{all}} = \mathrm{ANQ}_{\mathrm{succ}}$ and $\mathrm{APA}_{\mathrm{all}} = \mathrm{APA}_{\mathrm{succ}}$, indicating near-universal success with early stopping; PatchAttack on ViT-B/16 shows $\mathrm{ANQ}_{\mathrm{all}} > \mathrm{ANQ}_{\mathrm{succ}}$ and $\mathrm{APA}_{\mathrm{all}} > \mathrm{APA}_{\mathrm{succ}}$, meaning failed cases tend to run to the budget limit.

D. Ablation Study

We analyze the contribution of shape choice and the area-aware penalty in ShapeAware. We consider two variants: (1) a square-only version with no shape head and an active penalty ($\lambda = 10$), and (2) a mixed-shape version with a shape head over {circle, triangle, square} and no penalty ($\lambda = 0$). Results are shown in Table II.

TABLE II
SHAPEAWARE ABLATION IN THE TARGETED SETTING. ANQ AND APA WERE THE SAME FOR SUCCESSES AND OVERALL.

Variant	ASR (%)	ANQ	APA
Square-only, λ=10	99.33	23k	4%
Mixed shapes, λ=0	100.00	19k	5%

As shown in Table II, in the square-only case the penalty reduces area, but the number of queries does not improve over PatchAttack and the ASR is nearly unchanged. In the mixed-shape case, removing the penalty increases both ANQ and APA relative to ShapeAware.

Takeaways. Combining shape options with an area penalty works best. A plausible explanation for the weak effect of the penalty with squares only is that the agent cannot build compact, overlapping patterns using just squares. Comparing the mixed-shape ablation to ShapeAware suggests the penalty not only reduces area but also lowers queries, likely because it regularizes the policy and improves the exploration–exploitation balance beyond the weak signal from logits alone.

V. CONCLUSION AND FUTURE WORK

This work elevates patch geometry to a first-class decision in targeted, score-based black-box attacks. Building on PatchAttack, we introduced two variants: *ShapeAware*, which augments the policy with a shape head and an area-aware reward, and *PatchCross*, which specializes to an X-shaped mask. Across ResNet-50 and ViT-B/16, the main finding is consistent: geometry matters. ShapeAware preserves or improves ASR while substantially reducing queries and union area under the same per-step occlusion; PatchCross achieves near-perfect ASR with very low area and strong query efficiency. In addition, PatchCross exhibits *camouflage ability*: the 1-pixel strokes produce visually subtle patches that are easy to miss in qualitative examples

Beyond raw numbers, the study provides a clean evaluation recipe. Keeping the original attack loop and budgets while reporting ASR together with $\mathrm{ANQ}_{\mathrm{succ}}$, $\mathrm{ANQ}_{\mathrm{all}}$, $\mathrm{APA}_{\mathrm{succ}}$, and APA_{all} enables clearer comparisons across models and methods.

Future work. A natural extension is to enrich geometry by creating more complex masks—curved or spline-based primitives—and compositional rules that adapt to object layout. Moving to decision-based (label-only) settings and to physical-world tests (printable patches, varied viewpoints and lighting) will further stress realism. On the model side, transferring to detectors and segmenters, and probing transformer internals with token-aware masks, can broaden impact. In addition, widening the evaluation across architectures and budgets—varying N, $area\_occlu$, and Q—will enrich the empirical picture of the ASR–ANQ–APA trade-offs. Finally, testing ShapeAware and PatchCross against models equipped with adversarial ML defenses (e.g., adversarial training, detection, purification) will help assess resilience and guide defense design.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 770–778.

[2] A. Esteva, B. Kuprel, R. A. Novoa, J. M. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, no. 7639, pp. 115–118, 2017.

[3] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. P. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis, "Mastering the game of Go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[4] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *International Conference on Learning Representations (ICLR)*, 2014.

[5] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *International Conference on Learning Representations (ICLR)*, 2015.

[6] T. B. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer, "Adversarial patch," in *NeurIPS Workshop on Machine Learning and Computer Security (MLSec)*, 2017.

[7] L. Huang, C. Gao, Y. Zhou, C. Xie, A. L. Yuille, C. Zou, and N. Liu, "Universal physical camouflage attacks on object detectors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 717–726.

[8] A. J. Bose and P. Aarabi, "Adversarial attacks on face detectors using neural net based constrained optimization," *arXiv preprint arXiv:1805.12302*, 2018.

[9] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu, "A survey of deep learning techniques for autonomous driving," *Journal of Field Robotics*, vol. 37, no. 3, pp. 362–386, 2020.

[10] N. Papernot, P. McDaniel, A. Sinha, and M. Wellman, "Towards the science of security and privacy in machine learning," *arXiv preprint arXiv:1611.03814*, 2016.

[11] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: A simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2574–2582.

[12] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *International Conference on Learning Representations (ICLR)*, 2018.

[13] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018, pp. 274–283.

[14] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *International Conference on Learning Representations (ICLR) Workshop*, 2017, arXiv:1607.02533.

[15] C. Yang, A. Kortylewski, C. Xie, Y. Cao, and A. L. Yuille, "Patchattack: A black-box texture-based attack with reinforcement learning," in *Computer Vision – ECCV 2020*, ser. Lecture Notes in Computer Science, vol. 12371. Springer, 2020, pp. 681–698.

[16] Y. Ran, W. Wang, M. Li, L.-C. Li, Y.-G. Wang, and J. Li, "Cross-shaped adversarial patch attack," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 34, no. 4, pp. 2289–2303, 2024.

[17] I. Tsingenopoulos, D. Preuveneers, and W. Joosen, "Autoattacker: A reinforcement learning approach for black-box adversarial attacks," in *2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, 2019, pp. 229–237.

[18] K. Domico, J.-C. N. Ferrand, R. Sheatsley, E. Pauley, J. Hanna, and P. McDaniel, "Adversarial agents: Black-box evasion attacks with reinforcement learning," *arXiv preprint arXiv:2503.01734*, 2025.

[19] Y. Tong, H. Liang, H. Ma, S. Zhang, and X. Yang, "A survey on reinforcement learning-driven adversarial sample generation for PE malware," *Electronics*, vol. 14, no. 12, 2025.

[20] M. Andriushchenko, F. Croce, N. Flammarion, and M. Hein, "Square attack: A query-efficient black-box adversarial attack via random search," in *European Conference on Computer Vision (ECCV)*, ser. Lecture Notes in Computer Science, vol. 12368. Springer, 2020, pp. 484–501. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-58592-1_29

[21] S. Sarkar, A. R. Babu, S. Mousavi, S. Ghorbanpour, V. Gundecha, A. Guillen, R. Luna, and A. Naug, "Robustness with query-efficient adversarial attack using reinforcement learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2023, pp. 2330–2337. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2023W/AML/papers/Sarkar_Robustness_With_Query-Efficient_Adversarial_Attack_Using_Reinforcement_Learning_CVPRW_2023_paper.pdf

[22] X. Wei, H. Yan, and B. Li, "Sparse black-box video attack with reinforcement learning," *International Journal of Computer Vision*, vol. 130, no. 6, pp. 1459–1473, 2022. [Online]. Available: https://link.springer.com/article/10.1007/s11263-022-01604-w

[23] X. Liu, H. Yang, Z. Liu, L. Song, H. Li, and Y. Chen, "Dpatch: An adversarial patch attack on object detectors," *arXiv preprint arXiv:1806.02299*, 2018. [Online]. Available: https://arxiv.org/abs/1806.02299

[24] A. Fawzi and P. Frossard, "Measuring the effect of nuisance variables on classifiers," in *Proceedings of the British Machine Vision Conference (BMVC)*. BMVA Press, Sep. 2016, pp. 137.1–137.12.

[25] Z. Chen, B. Li, S. Wu, S. Ding, and W. Zhang, "Query-efficient decision-based black-box patch attack," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 5522–5536, 2023.

[26] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[27] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 248–255.

[28] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations (ICLR)*, 2021, openReview: https://openreview.net/forum?id=YicbFdNTTy.

[29] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 618–626.