# Supplementary Material for "Parallel Rectangle Flip Attack: A Query-based Black-box Attack against Object Detection"

Siyuan Liang[1,2], Baoyuan Wu[3,4,†], Yanbo Fan[5], Xingxing Wei[6], Xiaochun Cao[1,2,†]
[1]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2]University of Chinese Academy of Sciences, Beijing, China
[3]School of Data Science, The Chinese University of Hong Kong, Shenzhen, China
[4]Secure Computing Lab of Big Data, Shenzhen Research Institute of Big Data, Shenzhen, China
[5]Tencent AI Lab, Shenzhen, China
[6]Institute of Artificial Intelligence, Hangzhou Innovation Institute, Beihang University, Beijing, China
{liangsiyuan,caoxiaochun}@iie.ac.cn;wubaoyuan@cuhk.edu.cn;fanyanbo0124@gmail.com;xxwei@buaa.edu.cn

## Organization of the Supplementary Material

In Section A, we show the object detection attack algorithm with parallel rectangle flip attack; in Section B, we add the specific proof in Section 3.3 that using parallel attacks will accelerate the convergence of the algorithm. We can regard random search as a breadth search problem. In Section C, we specifically analyze the rectangular flip attack, and we explain in detail how to use the rectangular flip attack on the depth search. In Section D, we show the one challenge for the black-box attack in object detection. Even if one predicted bounding box is successfully attacked, another sub-optimal bounding box may be detected in similar locations.

## A. Parallel Rectangle Flip Attack Algorithm

Algorithm 1 shows our framework for generating black-box adversarial examples. First, we use the prior-guided detector mentioned in Section 3.2 to compute the high objectness and get the key points to perform the random search. Specifically, the prior-guided detector uses the target detector's output, the mask of the Mask R-CNN mask, or the key points from RepPoints. Next, according to the scheduling algorithm in parameter setting in Section 4.1, we get the side length $a$ of the square perturbations and the number of parallel points $P$. $flip\_indicator$ determines whether to perform rectangle flip. We set $flip\_indicator$ to 0, which means that only a square perturbation is generated in the initial state. After finding the adversarial perturbations $\delta$, we add these to the clean image $\mathbf{x}$ and project it under $\epsilon$ to get the adversarial example $\hat{\mathbf{x}}$.

In Algorithm 2, the pseudo-code takes the rectangular perturbations generated by flipping in the vertical direction as an example, which can easily be extended to the horizontal direction or both. The 'random search' at Line 6 means to randomly search the center coordinate of the adversarial patch with the uniform distribution. According to the number of parallel attack points $P$, we randomly search the center of the adversarial perturbations for $P$ times when generating the square perturbations. The flip state $flip\_indicator$ determines how to flip the adversarial perturbations. When its value is 1, we flip the sign of the perturbations in the upper half of the vertical direction, and when the value is 2, we flip the lower half of the perturbations. The size of the rectangular perturbation is determined by the side length $a$. As the number of iterations $q$ increases, the side length $a$ will decrease.

## B. Proof of Details in Section 3.3

In this section, we provide proof that parallel attacks accelerate convergence. To analyze the convergence rate, we make some common assumptions in the fellowing.

**Assumption 1** We assume all detector's functions $F_m(*)$ are differential and the objective function $H$ in Eq.2 has a Lipschitz

† indicates corresponding authors. Corresponds to wubaoyuan@cuhk.edu.cn and caoxiaochun@iie.ac.cn

**Algorithm 1** Black-Box Adversarial Example Generation with Parallel Rectangle Flip Attack

**Input:** target detector $f_1$, prior-guided detector $f_2$, clean image $\mathbf{x} \in \mathbb{R}^{w \times h \times c}$, ground-truth boxes $G$, classification labels $Y$, maximum number of iterations $Q$, determine whether to perform flip $flip\_indicator$

**Output:** adversarial image $\hat{\mathbf{x}} \in \mathbb{R}^{w \times h \times c}$ with $||\hat{\mathbf{x}} - \mathbf{x}||_\infty \leq \epsilon$

$\hat{\mathbf{x}} \leftarrow init(\mathbf{x})$, $h_{best} \leftarrow H(f(\mathbf{x}), B, Y)$, $q \leftarrow 1$, $flip\_indicator \leftarrow 0$
$key\_points \leftarrow f_2(\mathbf{x})$(obtain key points according to the prior information in Section 3.2)
**while** $q < Q$ **and** $\hat{x}$ is not adversarial **do**
    $a^{(q)} \leftarrow$ the side length of the square(according to schedules in Section 4.1)
    $P^{(q)} \leftarrow$ the number of parallel attack points(according to schedules in Section 4.1)
    $\delta$, $flip\_indicator \leftarrow$ PRFA($flip\_indicator, key\_points, \epsilon, w, h, c, a^{(q)}, P^{(q)}$)(see Alg. 2 for details)
    $\hat{\mathbf{x}}_{new} \leftarrow \Pi_{Bound(x,\epsilon)}(\mathbf{x} + \delta)$
    $h_{new} \leftarrow H(f(\hat{\mathbf{x}}_{new}), B, Y)$
    **if** $h_{new} < h_{best}$ **then** $\hat{\mathbf{x}} \leftarrow \hat{\mathbf{x}}_{new}, h_{best} \leftarrow h_{new}$
    **end if**
    $q \leftarrow q + 1$
**end while**
return adversarial image $\hat{\mathbf{x}}$

**Algorithm 2** Parallel Rectangle Flip Aattack

**Input:** determine whether to perform flip $flip\_indicator$, set of points in an important area $key\_points$, adversarial perturbation limits $\epsilon$, image width $w$, image height $h$, numbers of channels $c$, square size $a$, parallel numbers $P$

**Output:** adversarial perturbations $\delta$

$d \leftarrow w \times h \times c, \mathcal{H} \equiv \{-1, +1\}^d$
$\delta \leftarrow$ An array of shape $w \times h \times c$ with all 0s
**for** $i = 1 \rightarrow P$ **do**
    // Take flipping the rectangle vertically as an example
    **if** not $flip\_indicator$ **then**
        random search
        $(u^{(i)}, v^{(i)}) \in key\_points \subset N^2$
        **for** $j = 1 \rightarrow c$ **do**
            $s \sim \mathcal{U}(\mathcal{H})$ //e.g.,[+1,...,+1]
            $\delta_{u^{(i)}+1:u^{(i)}+a, v^{(i)}+1:v^{(i)}+a, i} \leftarrow s \cdot \mathbb{1}_{a*a}$
        **end for**
        **if** $i == P$ **then** $flip\_indicator \leftarrow 1$
        **end if**
    **else**
        **if** $flip\_indicator == 1$ **then**
            $\delta_{u^{(i)}+1:u^{(i)}+a, v^{(i)}+1:v^{(i)}+a/2, j} * = -1$
            **if** $i == P$ **then** $flip\_indicator \leftarrow 2$
            **end if**
        **else if** $flip\_indicator == 2$ **then**
            $\delta_{u^{(i)}+1:u^{(i)}+a, v^{(i)}+a/2:v^{(i)}+a, j} * = -1$
            **if** $i == P$ **then** $flip\_indicator \leftarrow 0$
            **end if**
        **end if**
    **end if**
**end for**
return $\delta, flip\_indicator$

gradient, that is, there is a constant $L$ for $\forall \mathbf{x}, \forall \mathbf{y}, ||f'(\mathbf{x}) - f'(\mathbf{y}) \leq L||\mathbf{x} - \mathbf{y}||$.

The Lipschitz gradient assumption essentially assumes that the curvature of the objective function $H$ is bounded by $L$.

According to the Assumption 1, we can deduce that $f(\mathbf{y}) - f(\mathbf{x}) \leq \langle f'(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \frac{L}{2}||\mathbf{y} - \mathbf{x}||^2$. The step size is $\gamma$. We apply the Lipschitz gradient assumption in the iterative process $Q$ and obtain the fellowing golden inequality:

$$\begin{aligned} f(\mathbf{x}_{q+1}) - f(\mathbf{x}_q) &\leq \langle f'(\mathbf{x}_q), \mathbf{x}_{q+1} - \mathbf{x}_q \rangle + \frac{L}{2}||\mathbf{x}_{q+1}, \mathbf{x}_q||^2 \\ &= -\gamma \langle f^{'}(\mathbf{x}_q), \delta_q \rangle + \frac{L\gamma^2}{2}||\delta_q||^2 \end{aligned} \tag{1}$$

**Assumption 2** We make a typical assumption the stochastic gradient is unbiased, e.g., $\forall \mathbf{x}, \mathbb{E}_m[F'_m(\mathbf{x})] = f'(\mathbf{x})$ and is with upper bound variance $\sigma$, that is, $\forall \mathbf{x}, \mathbb{E}_m[||F'_m(\mathbf{x}) - f'(\mathbf{x})||^2] \leq \sigma^2$.

According to Assumption 2, we have two important properties. The unbiased gradient in Eq. 1 can be rewritten as $\mathbb{E}[\langle f'(\mathbf{x}_q), \delta_q \rangle] = \langle f'(\mathbf{x}_q), \mathbb{E}[\delta_q] = ||f'(\mathbf{x}_q)||^2$. Where there exists the property of variance, that is, any random vector $\varphi$ satisfies $\mathbb{E}[||\varphi||^2] = ||\mathbb{E}[\varphi]||^2 + \mathbb{E}[||\varphi - \mathbb{E}[\varphi]||]^2$. The bounded stochastic variance can be rewritten as $\mathbb{E}_m[||\delta_q||^2] = ||f'(\mathbf{x}_q)||^2 + E[||\delta_q - f'(\mathbf{x}_q)||^2] \leq ||f'(\mathbf{x}_q)|| + \sigma^2$. Apply these two properties in Eq. 1, we have:

$$\begin{aligned} &\mathbb{E}[f(\mathbf{x}_{q+1})] - \mathbb{E}[f(\mathbf{x}_q)] \\ &\quad \leq -\gamma \mathbb{E}[||f'(\mathbf{x}_q)||^2] + \frac{L\gamma^2}{2}(\mathbb{E}[||f'(\mathbf{x}_q)||^2] + \sigma^2) \\ &\quad \leq -\gamma(1 - \frac{\gamma L}{2})\mathbb{E}[||f'(\mathbf{x}_q)||^2] + \frac{\gamma^2}{2}L\sigma^2. \end{aligned} \tag{2}$$

We can add up Eq. 2 from $q = 0, 1, ...Q$ and have:

$$\mathbb{E}[f(\mathbf{x}_{Q+1})] - f(\mathbf{x}_0) \leq -\gamma(1 - \frac{\gamma L}{2}) \sum_{q=0}^{Q} \mathbb{E}[||f'(\mathbf{x}_q)||^2] + \frac{\gamma^2}{2}QL\sigma^2. \tag{3}$$

We set the step size is $\gamma = \frac{1}{Q + \sigma\sqrt{QL}}$ which makes $(1 - \frac{\gamma L}{2}) > 0$. According to Eq. 3, we can deduce that:

$$\begin{aligned} &\frac{1}{Q}\sum_{q=0}^{Q} \mathbb{E}[||f'(\mathbf{x}_q)||^2] \\ &\qquad \lesssim \frac{f(\mathbf{x}_0) - \mathbb{E}[f(\mathbf{x}_{Q+1})]}{Q\gamma} + \gamma L\sigma^2 \\ &\qquad \lesssim \frac{f(\mathbf{x}_0) - f^*}{Q\gamma} + \gamma L\sigma^2 \\ &\qquad \lesssim \frac{(f(\mathbf{x}_0) - f^*)L}{Q} + \frac{(f(\mathbf{x}_0) - f^*)\sqrt{L}\sigma}{\sqrt{Q}} \end{aligned} \tag{4}$$

We do not repeat the center point's selection during random search with parallel attack points $P$, which can be regarded as for sampling without replacement. The stochastic variance involves a bit more complicated derivation.

**Lemma 1** Define a random variable $\overline{\xi}_{[P]} := \frac{1}{P}\sum_{m=1}^{P} \xi_m$, where $\xi_m$ is uniformly randomly sampled from the random search space $D$ and P is the batch size. Then the fellowing equality holds:

$$\mathbf{Var}[\overline{\xi}] = (\frac{D - P}{D - 1})\frac{\mathbf{Var}[\xi_1]}{P} \tag{5}$$

If sampled point are vectors and satisfy $\frac{1}{M}\sum_{m=1}^{M} \xi_m = 0$, the stochastic gradient can easily varify

$$\mathbb{E}[||\overline{\xi}||^2] = P(\frac{D - P}{D - 1})\mathbb{E}[||\overline{\xi_1}||^2] \tag{6}$$

Let $\mathcal{P}$ is a batch of parallel attack points.The we let $\xi_m := F'_m(\mathbf{x}) - f'(\mathbf{x})$ and we have:

$$\begin{aligned}\mathbb{E}[||\delta^{\mathcal{P}}(\mathbf{x}) - f'(\mathbf{x})||^2] &= \mathbb{E}[||\overline{\xi}||^2] \\ &= (\frac{D-P}{D-1})\frac{\mathbb{E}[||\xi_1||^2]}{P} \\ &\leq (\frac{D-P}{D-1})\frac{\sigma^2}{P} \\ &\leq \frac{\sigma^2}{P}\end{aligned} \tag{7}$$

When we put equations Eq. 6 and Eq. 7 into equation Eq. 4, we can get the fellowing:

$$\begin{aligned}&\frac{1}{Q}\sum_{q=0}^{Q}\mathbb{E}[||f'(\mathbf{x}_q)||^2] \\ &\quad\lesssim \frac{(f(\mathbf{x}_0) - f^*)L}{Q} + \frac{(f(\mathbf{x}_0) - f^*)\sqrt{L}\sigma}{\sqrt{QP}} \\ &\quad\lesssim \frac{L}{Q} + \frac{\sqrt{L}\sigma}{\sqrt{QP}}\end{aligned} \tag{8}$$

In Eq. 8, the influence of queries $Q$ is still more significant for the convergence than the number of parallel attack points $P$. In the initial query, we set the large $P$ and decrease when the query increase.

## C. Details Analyzed in Section 3.4

For adversarial attack in object detection, the maximum change component of the correct label $y$ of an object contained in a patch $\hat{\mathbf{x}}_p$ should be smaller than the maximum change component of other classes. We can divide this patch $\hat{\mathbf{x}}_p$ into $|Y|+1$ cliques $\{T_i\}_{i=1}^{|Y|+1}$, e.g., the clique $\{T_y\}$ represents the cluster of points with the correct label $y$. Then we have:

$$\begin{aligned}\min_{\delta} H = \min_{\delta}[&\sum_{(u,v)\in T_y}^{T_y} \max|F(\sum_{i,j=1}^{k}\delta_{u:,v:}\cdot w_{i,j}) + \alpha_y| \\ &- \max\sum_{l\neq y}^{|Y|}[\sum_{(u,v)\in T_l}^{T_l}\max|F(\sum_{i,j=1}^{k}\delta_{u:,v:}\cdot w_{i,j}) + \alpha_l|]]\end{aligned} \tag{9}$$

### C.1. Proof of Proposition 2

If we want to maximum change component of the correct label $y$ in the same patch $\hat{\mathbf{x}}_p$, two points's perturbation in the same patch $\hat{\mathbf{x}}_p$ should be same, that is, $(u_1, v_1) \in T_l$ and $(u_2, v_2) \in T_l$, then $\delta_{u_1:,v_1:} = \delta_{u_2:,v_2:}$.

Let the sign $s \in \mathbb{R}^{a\times a}$ of perturbations $\delta_p$. The $s^{consisty}$ represents the sign in this perturbations is consisty and $s^{random}$ represents the sign is random. The $c$ is a constant. Using the Khintchine and Jensen inequalities similarly, we have:

$$\begin{aligned}\mathbb{E}|\langle\delta_p, s^{random}\rangle| &= \mathbb{E}|\langle\delta_{u:,v:}, s^{random}_{u:,v:}\rangle| \\ &\geq \frac{\sqrt{2}ch^2}{w^2}||s||_2\end{aligned} \tag{10}$$

The fellowing upper bound of Eq. 11 can be calculated as:

$$\mathbb{E}|\langle\delta_p, s^{random}\rangle| \leq \frac{2ch}{w}||s||_2 \tag{11}$$

Thus $\mathbb{E}|\langle\delta_p, s^{random}\rangle| = O(||s||_2)$. If the sign of perturbations is consisty, which means that the $s_{u:,v:}$ has a constant

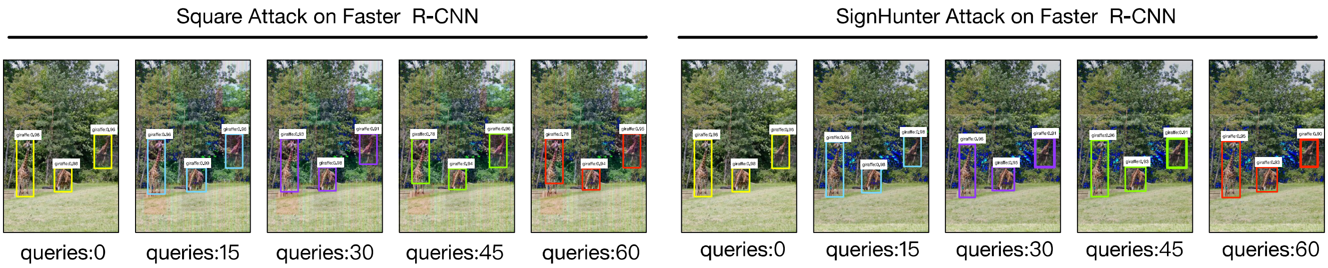


Figure 1. The results of the two black-box attack methods on Faster R-CNN with different iterations. The rectangular boxes represent the detection results of the detector. Even if one predicted bounding box is successfully attacked(the yellow one), another sub-optimal bounding box may be detected in similar locations.

sign. Then we have $\mathbb{E}|\langle\delta_p, s^{consisty}\rangle| = O(||s||_1)$. This implies that for an approximately constant block $\mathbb{E}|\langle\delta_p, s^{consisty}\rangle|$ will be larger than $\mathbb{E}|\langle\delta_p, s^{random}\rangle|$. If we want to minimize the $H$ in Eq. 9, we can minimize the upper bound of the $\sum_{(u,v)\in T_y}^{T_y} \max|F(\sum_{i,j=1}^{k} \delta_{u:,v:} \cdot w_{i,j}) + \alpha_y|$. So we minimize the upper bound $\mathbb{E}|\langle\delta_p, s^{consisty}\rangle| = O(||s||_1)$. Points that belong to the same clique in a patch are highly similar in spatial location (close to value) and semantic feature (close in the property). Hence, if the sign of perturbations can change the semantic feature of one point, it will be effective for other points with high probability. We can generate a rectangular perturbation by flipping the sign, which makes points in the same clique are inconsistent and pushes them into different classified classification boundaries.

## D. Prediction Boxes and Sub-optimal Boxes in Object Detection

Object detection is very effective after using an end-to-end model based on deep learning. The commonly used object detection models, whether they are one-stage(YOLO) or two-stage(RCNN) algorithms, non-maximum suppression are an essential component of them. In the existing object detection models, a massive number of candidate rectangular boxes are generated. Many of these rectangular boxes point to the same object, so there are a large number of redundant candidate rectangular boxes. The purpose of the non-maximum suppression algorithm is here. It can eliminate redundant boxes and find the best object detection position.

**Algorithm 3** Non-maximum Suppression Algorithm

**Input:** $B = \{b_1, b_2, ..., b_N\}$ is the list of initial detection boxes,$S = \{s_1, s_2, ..., s_N\}$ contains corresponding detection scores, $N_t$ is the NMS threshold
**Output:** D, S

```
D ← {}
while B ≠ empty do
    m ← argmax(S)
    M ← b_m
    D ← D ⋃ M
    B ← B − M
    for b_i in B do
        if IoU(M, b_i) ≥ N_t then
            B ← B − b_i
            S ← S − s_i
        end if
    end for
end while
return D, S
```

The idea of non-maximum suppression (Non-Maximum Suppression, hereinafter referred to as NMS algorithm) is to search for local maximums and suppress non-maximum elements. Therefore, the object detection task's output can only be the optimal prediction boxes and corresponding scores after NMS. Consequently, even if one predicted bounding box is successfully attacked (*i.e*., not detected), another sub-optimal bounding box may be detected in similar locations.

In Fig. 1, we show the optimization results of Square Attack and SignHunter on the detector Faster R-CNN. When the query is 0, it means the outputs of the Faster R-CNN on the clean image. Even if we can successfully attack the prediction result on the clean image (the yellow part), the detector still predicts sub-optimal prediction boxes, which can still detect objects successfully(blue, purple, green, and red).