T-SEA: Transfer-based Self-Ensemble Attack on Object Detection

Hao Huang*
Peking University
Beijing, China
huanghao@stu.pku.edu.cn

Ziyan Chen*
Peking University
Beijing, China
chen.ziyan@outlook.com

Huanran Chen*
Peking University
Beijing, China
huanran_chen@outlook.com

Yongtao Wang†
Peking University
Beijing, China
wyt@pku.edu.cn

Kevin Zhang
Peking University
Beijing, China
kevinzyz@pku.edu.cn

Abstract

Compared to query-based black-box attacks, transfer-based black-box attacks do not require any information of the attacked models, which ensures their secrecy. However, most existing transfer-based approaches rely on ensembling multiple models to boost the attack transferability, which is time- and resource-intensive, not to mention the difficulty of obtaining diverse models on the same task. To address this limitation, in this work, we focus on the single-model transfer-based black-box attack on object detection, utilizing only one model to achieve a high-transferability adversarial attack on multiple black-box detectors. Specifically, we first make observations on the patch optimization process of the existing method and propose an enhanced attack framework by slightly adjusting its training strategies. Then, we analogize patch optimization with regular model optimization, proposing a series of self-ensemble approaches on the input data, the attacked model, and the adversarial patch to efficiently make use of the limited information and prevent the patch from overfitting. The experimental results show that the proposed framework can be applied with multiple classical base attack methods (e.g., PGD and MIM) to greatly improve the black-box transferability of the well-optimized patch on multiple mainstream detectors, meanwhile boosting white-box performance. Our code is available at https://github.com/VDIGPKU/T-SEA.

1. Introduction

With the rapid development of computer vision, deep learning-based object detectors are being widely applied to many aspects of our lives, many of which are highly related to our personal safety, including autonomous driving and intelligent security. Unfortunately, recent works [16, 19, 34, 37] have proved that adversarial examples can successfully disrupt the detectors in both digital and physical domains, posing a great threat to detector-based applications. Hence, the mechanism of adversarial examples on detectors should be further explored to help us improve the robustness of detector-based AI applications.

In real scenes, attackers usually cannot obtain the details of the attacked model, so black-box attacks naturally receive more attention from both academia and industry.

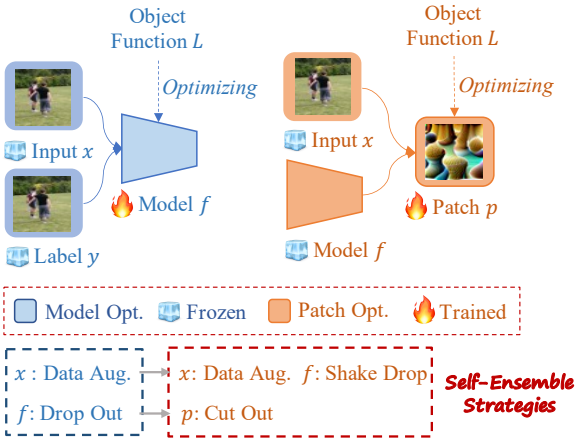

Figure 1. Model optimization usually augments the training data and drop out neurons to increase generalization, motivating us to propose self-ensemble methods for adversarial patch optimization. Specifically, inspired by data augmentation in model optimization, we augment the data x and the model f via constrained data augmentation and model ShakeDrop, respectively. Meanwhile, inspired by drop out in model optimization, we cut out the training patch τ to prevent it overfitting on specific models or images.

*These authors contributed equally to this work.
†Corresponding author.

Generally speaking, black-box adversarial attacks can be classified into 1) query-based and 2) transfer-based. For the former, we usually pre-train an adversarial perturbation on the white-box model and then fine-tune it via the information from the target black-box model, assuming we can access the target model for free. However, frequent queries may expose the attack intent, weakening the covertness of the attack. Contrarily, transfer-based black-box attacks utilize the adversarial examples' model-level transferability to attack the target model without querying and ensure the secrecy of the attack. Thus, how to enhance the model-level transferability is a key problem of transfer-based black-box attacks. Most existing works apply model ensemble strategies to enhance the transferability among black-box models, however, finding proper models for the same task is not easy and training adversarial patch on multiple models is laborious and costly. To address these issues, in this work, we focus on how to enhance the model-level transferability with only one accessible model instead of model ensembling.

Though the investigation of adversarial transferability is still in its early stage, the generalizability of neural networks has been investigated for a long time. Intuitively, the association between model optimization and patch optimization can be established by shifting of formal definitions. Given the input pair of data $x \in \mathcal{X}$, and label $y \in \mathcal{Y}$, the classical model learning is to find a parametric model f_θ such that $f_\theta(x) = y$, while learning an adversarial patch treats the model f_θ, the original optimization target, as a fixed input to find a hypothesis h of parametric patch τ to corrupt the trained model such that $h_\tau(f_\theta, x) \neq y$. Hence, it is straight forward to analogize patch optimization with regular model optimization. Motivated by the classical approaches for increasing model generalization, we propose our **T**ransfer-based **S**elf-**E**nsemble **A**ttack (**T-SEA**), ensembling the input x, the attacked model f_θ, and the adversarial patch τ from **themselves** to boost the adversarial transferability of the attack.

Specifically, we first introduce an enhanced attack baseline based on [32]. Observing from Fig. 4 that the original training strategies have some limitations, we slightly adjust its learning rate scheduler and training patch scale to revise [32] as our enhanced baseline (E-baseline). Then, as shown in Fig. 1, motivated by *input augmentation* in model optimization (*e.g.*, training data augmentation), we introduce the constrained data augmentation (data self-ensemble) and model ShakeDrop (model self-ensemble), virtually expanding the inputs of patch optimization (i.e., the input data x and the attacked model f) to increase the transferability of the patch against different data and models. Meanwhile, motivated by the *dropout* technique in model optimization, which utilizes sub-networks of the optimizing model to overcome overfitting and thus increasing model generalization, we propose patch cutout (patch self-ensemble), randomly performing cutout on the training patch τ to overcome overfitting. Through comprehensive experiments, we prove that the proposed E-baseline and self-ensemble strategies perform very well on widely-used detectors with mainstream base attack methods (*e.g.*, PGD [24], MIM [13]). Our contributions can be summarized as the following:

- We propose a transfer-based black-box attack T-SEA, requiring only one attacked model to achieve a high adversarial transferability attack on object detectors.
- Observing the issues of the existing approach, we slightly adjust the training strategies to craft an enhanced baseline and increase its performance.
- Motivated by approaches increasing generalization of deep learning model, we propose a series of strategies to self-ensemble the input data, attacked model, and adversarial patch, which significantly increases the model-level adversarial transferability without introducing extra information.
- The experimental results demonstrate that the proposed T-SEA can greatly reduce the AP on multiple widely-used detectors on the black-box setting compared to the previous methods, while concurrently performing well with multiple base attack methods.

2. Related Work

2.1. Black-box Adversarial Attack

Since the discovery of adversarial attack by [31], black-box attacks have gradually attracted more attention, owing to the difficulty of obtaining details of the attacked model in real scenes. Black-box attack can be separated into query-based methods [5, 12, 21] and transfer-based methods [2, 20, 26]. The former utilizes the outputs of the target model to optimize the adversarial examples, which may disclose the attack behavior due to the frequent queries. Hence, in our work, we concentrate on the latter, optimizing adversarial examples in a substitute model without relying on any knowledge of the black-box models. Though the transfer-based attacks ensure the covertness of the attack, their performance is generally limited due to the lack of specialized adjustment catered to the target model. In this work, we propose the self-ensemble strategies to address the issue and greatly boost the black-box attack transferability of the existing transfer-based detector attack.

2.2. Object Detectors

Object detection is a fundamental computer vision technology predicting objects' categories and positions and is widely used in many perception tasks. In the past ten years, deep learning-based models have greatly improved the performance of object detectors. The mainstream methods

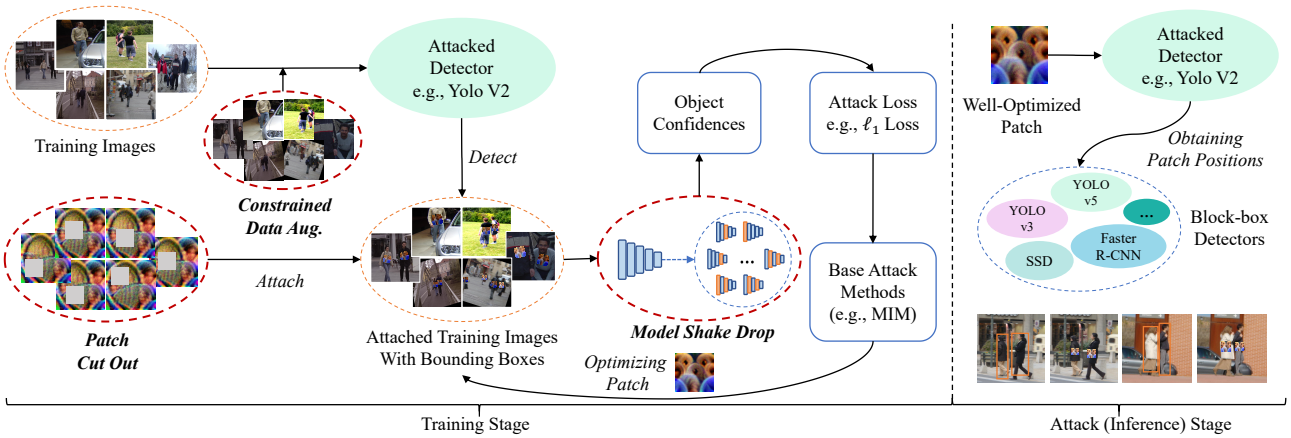

Figure 2. The Overall Pipeline of T-SEA. During the training stage, we utilize the self-ensemble strategies (i.e., the constrained data augmentation, patch cutout and model ShakeDrop) to enhance the transferability of the well-optimized adversarial patch. During the attack (inference) stage, we attach the crafted patch into different images to disrupt the detection process of multiple widely-used detectors on black-box setting.

based on deep learning can be roughly divided into one-staged and two-staged methods. The former directly predicts the positions and classes of the object instances and are thus faster; the latter first use the region proposal network (RPN) to generate proposals, and then predict the labels and positions of the selected proposals. In this paper, we select eight mainstream detectors of both one-stage and two-stage, including YOLO v2 [27], YOLO v3 & YOLO v3tiny [28], YOLO v4 & YOLO v4tiny [4], YOLO v5 [17], Faster R-CNN [29] and SSD [22] to systemically verify the proposed T-SEA framework.

2.3. Attacks on Object Detector

The security of deep learning-based models is receiving increasing attention [8, 9], especially with the existence of adversarial attacks. Recent works have explored adversarial robustness of object detectors. To begin, [37] apply adversarial attack on object detectors, performing iterative gradient-based method to misclassify the proposals, and the similar ideas are also carried out by [19, 34]. Ensuingly, for enhancing the attack capability in the physical world, [7, 32] propose real-world adversarial patches to attack the mainstream detectors, *e.g.*, YOLO and Faster R-CNN. Recently, model ensemble approaches are used to improve the attack transferability among multiple detectors, such as [16,35,41] simultaneously attacking multiple detectors to generate cross-model adversarial examples. Different from the above, we focus on how to make the most of the limited information (i.e., a single detector) to carry out a high-transferability black-box attack on multiple black-box detectors.

3. Method

In this section, we first give the problem formulation in Sec. 3.1 and describe the overall framework in Sec. 3.2. Then, we respectively introduce the details of our enhanced baseline and self-ensemble strategies in Sec. 3.3 and Sec. 3.4.

3.1. Problem Formulation

In this work, we carry out a transfer-based black-box attack with only one white-box detector to decrease the average precision (AP) of both white-box and black-box detectors. Given the target input data distribution $\mathcal{D}(\mathcal{X}, \mathcal{F})$, we regard a single pre-trained detector $f_w \in \mathcal{F}$ as the white-box attacked model, $x_{1,..,N} \in \mathcal{X}$ as the input images, where N is the number of training samples. We are committed to crafting a universal adversarial patch τ from the adversarial distribution $\mathcal{T}$ to disrupt the detection process,

$$\hat{\tau} = \arg\min_{\tau \sim \mathcal{T}} \sum_{i}^{N} L(\widetilde{x}_i, \widetilde{f}_w, \widetilde{\tau}) + \lambda J(\tau), \tag{1}$$

where L is the loss function to measure the corruption of detector, $\widetilde{x}_i, \widetilde{f}_w$ and $\widetilde{\tau}$ denote self-ensembled data, model and patch, respectively, and $J(\cdot)$ is a regularization term which we employ total variation of τ in our work.

3.2. Overall Framework

As depicted in Fig. 2, we divide the entire T-SEA pipeline into training stage and attack stage. During the training stage, the input images are first augmented with the constrained data augmentation. Ensuingly, the white-box detector (i.e., the attacked detector) is used to locate the target objects from the augmented images. Then, we attach adversarial patches to the center of each detected object of the target class, along with patch cutout for mitigating overfitting. After that, the images with adversarial patches go through the shake-dropped models, and we minimize the object confidence and continuously optimize the training

adversarial patches until reaching maximum epoch number. During the attack stage, we apply our well-optimized adversarial patch on the test images to disrupt the detection process of multiple black-box detectors.

3.3. Enhanced Baseline

In this section, we introduce the details of the enhanced baseline, which is described in Algorithm 1. Firstly, we randomly initialize a patch τ_0, prepare the training images and attacked model f_w. During each training epoch, for every input image batch X, we first obtain their detection results via the white-box detector f_w. Ensuingly, we utilize the transformation function T to apply the training patch into each image to generate adversarial image batch X^{adv}. Next, we calculate the object confidence of X^{adv}, and the attack loss of the image batch. Here we use ℓ_1 loss as the attack loss, unless otherwise stated. Finally, we update the τ via base attack method $\hbar$ and adjust the learning rate after each epoch until reaching the maximum number.

Algorithm 1 Enhanced Baseline Based on [32]

Require: $x_{1,..,N}$ (training images), f_w (white-box detector), M (maximum epoch), BS (batch size), τ_0 (the initial patch), T (patch applier), $\hbar$ (base attack method), $scheduler$ (learning rate scheduler).

Ensure: Well-optimized Adversarial Patch τ

$\tau \Leftarrow \tau_0$
for each $i \in [1, M]$ **do**
  for each $j \in [1, \frac{N}{BS}]$ **do**
    $X \Leftarrow x_{(j-1)\cdot BS+1}, \ ..., \ x_{j\cdot BS}$
    $bbox^{clean}, \ conf^{clean} \Leftarrow f_w(X)$
    $X^{adv} \Leftarrow T(X, \ bbox^{clean}, \ \tau)$
    $bbox^{adv}, \ conf^{adv} \Leftarrow f_w(X^{adv})$
    $loss \Leftarrow Avg(conf^{adv})$
    $\tau \Leftarrow \hbar(\tau, \ loss)$
  end for
  update lr via $scheduler$
end for

Compared to AdvPatch [32], we slightly adjust two training strategies. One is the learning rate scheduler. We observe that a saddle point during patch optimization may lead the original plateau-based scheduler to drastically drop the learning rate, causing inadequate optimization. Hence, we adjust the learning rate decline strategy with the following criterion,

$$lr \Leftarrow lr * \mu, if \ (\ell_t - \ell_{t-1}) < \epsilon_1 \ and \ \frac{(\ell_t - \ell_{t-1})}{\ell_t} < \epsilon_2, \quad (2)$$

where μ is the decay factor, ℓ_t denotes the mean loss at epoch t, ϵ_1, ϵ_2 are thresholds to control learning rate updates. In our experiments, we ensure that the learning rate decrease more stably via the above hyper-parameters. The other adjustment is to reduce the patch scale at training stage s_t. Here, the patch scale is the length ratio between the patch and the bounding box of the specific object. As Fig. 4 shows, a proper scale helps the optimized patch to learn more global patterns, which will not be disrupted easily when the patch is scaled, *e.g.*, a local pattern will lose more information when its corresponding patch is scaled to a very small size.

3.4. Self-Ensemble Strategies

3.4.1 Theoretical Analysis

Typical model training usually attempts to find a mapping function f from a finite hypothesis space $\mathcal{F}$ that describes the relationship between the input data x and label y following the underlying joint distribution $\mathcal{D}(\mathcal{X}, \mathcal{Y})$. Generally, since $\mathcal{D}$ is unknown, we use training data set $S = \{(x_i, y_i)|i = 1, ..., N\}$ to train the model, which means the optimization involves the empirical risk minimization(ERM) [33]:

$$\widehat{R}_S(f) = \frac{1}{N}\sum_{i}^{N} L(y_i, f(x_i)), \quad (3)$$

where L is a loss function to measure the difference between the model prediction $f(x)$ and the label y. However, the empirical risk is unable to provide generalization on unseen data [40]. Fortunately, we can derive that the generalization error $R_{\mathcal{D}}(f)$ is bounded by $\widehat{R}_S(f)$ for a given confidence $1 - \sigma \in (0, 1)$ [3, 25],

$$\forall f \in \mathcal{F}, \ R_{\mathcal{D}}(f) \leq \widehat{R}_S(f) + \sqrt{\frac{log \ c + log \ \frac{1}{\sigma}}{N}}, \quad (4)$$

where N denotes the input data size, c is a complexity measure of $\mathcal{F}$, such as VC-dimension [25] or covering numbers [1]. This generally yields bounds of the generalization gap $R_{\mathcal{D}}(f) - \widehat{R}_S(f) = \mathcal{O}(\sqrt{\frac{log \ c}{N}})$, which gives us a lens to analyze model generalization: we can sample more training data or lower model complexity to help close the generalization gap. In practice, limited training data and large-scaled model parameters make the model prone to overfit [39], causing an undesirable generalization gap increase. Fortunately, past studies have widely adopted a series of regularization strategies on both input data and the model, such as data augmentation [11, 40, 42], Stochastic Depth [15], and Dropout [30].

Motivated by the aforementioned model regularization methods, we discuss how to improve attack transferability of the adversarial patch. During patch optimization, the model f becomes an input instead of the optimization objective, which means the original input distribution

$\mathcal{D}(\mathcal{X}, \mathcal{Y})$ is shifted to a joint input distribution $\mathcal{D}(\mathcal{X}, \mathcal{F})$. Naturally, we can improve generalization of the adversarial patch by increasing training data scale $N_{x,f}$, i.e. sampling more training images x and ensembling more white-box models f_w. However, obtaining massive images or models of the same task is generally expensive or impractical for the attacker. To address this issue, we propose to virtually expand the input by data augmentation and model ShakeDrop to help improve patch transferability. Meanwhile, although the large-scale models have achieved superior performance in most tasks, these models still benefit from reducing model complexity at training stage like stochastic depth [15] and Dropout [30]. Thus we propose the patch cutout to reduce its capacity c_τ in training process as a new regularization to alleviate overfitting.

3.4.2 Data: Constrained Data Augmentation

Motivated by data augmentation in increasing model generalization, we can similarly improve patch transferability by virtually expanding the training set S to more closely approximate the underlying distribution $\mathcal{D}$. Therefore, we employ a constrained policy to avoid unnatural augmentation which may not appear in natural scenes. In our work, we 1) mildly resize and crop the input images, 2) slightly alter the brightness, contrast, saturation and hue, and 3) randomly rotate the input images in a small range, to generate natural augmented images.

3.4.3 Model: ShakeDrop

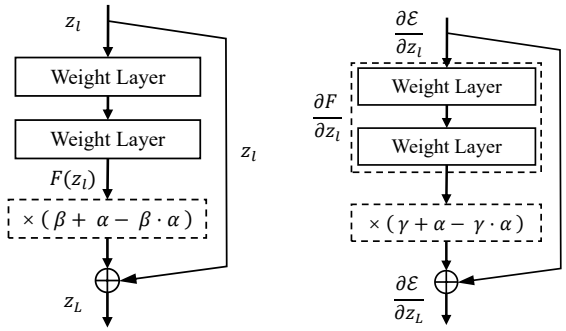

Figure 3. We utilize the model ShakeDrop to linearly combine the outputs of the stacked layers and identity of a residual block.

As discussed above, when training the patch, sampling more white-box models f from the joint input distribution $\mathcal{D}$ can improve the adversarial transferability of the crafted patch. However, it is difficult to obtain multiple models of the same task in reality. Inspired by stochastic depth [15], which randomly drops a subset of layers to virtually ensemble multiple model variants for improving generalization, we utilize ShakeDrop [38] to linearly combine the outputs of stacked layers and identity of a residual block, generating massive variants of the attacked white-box model as the Fig. 3 illustrated.

Specifically, in forward propagation, we combine the identity z_l and the output of stacked layers $F(z_l)$ of a residual block to generate z_L with the following formula:

$$z_L = z_l + (\beta + \alpha - \beta \cdot \alpha) \cdot F(z_l), \tag{5}$$

where β is sampled from a Bernoulli distribution with $P(\beta = 1) = \varphi_s$, α is sampled from a continuous uniform distribution $\alpha \sim U(1-e, 1+e)$ (e is a constant). In backward propagation, denoting the loss function as $\mathcal{E}$, from the chain rule of backpropagation, ShakeDrop can be formulated as

$$\begin{aligned} \psi(\frac{\partial \mathcal{E}}{\partial z_l}) &= \psi(\frac{\partial \mathcal{E}}{\partial z_L} \cdot \frac{\partial z_L}{\partial z_l}) \\ &= \frac{\partial \mathcal{E}}{\partial z_L}(1 + (\gamma + \alpha - \gamma \cdot \alpha) \cdot \frac{\partial F}{\partial z_l}), \end{aligned} \tag{6}$$

where γ is another Bernoulli variable and identically distributed with β.

3.4.4 Patch: Cutout

Dropout [30] is designed to randomly drop neural units along with their connections, effectively preventing overfitting of the model. Inspired by Dropout, we propose a patch cutout strategy, randomly masking a region of the adversarial patch to reduce the patch complexity at training stage and thus to prevent overfitting on specific model and data. From an other angle, model Dropout prevents excessive co-adapting of neural units and spread out features over multi neurons to alleviate overfitting. Similarly, patch cutout also prevents the adversarial nature of patch from relying too much on the patterns of a certain area.

The proposed patch cutout is similar to cutout [11] or random erasing [42] in model training. Specifically, for a normalized input image $I_{x,y} \in [0, 1]$ of size $H \times W$, we carry out the following process with a probability φ_c before attaching patches to the target objects: 1) we firstly conduct random sampling to obtain one point $p = (x_0, y_0)$ within the given patch; 2) then we cover a $\eta H \times \eta W$ square area(η is the ratio) centered at p with a specific value $\eta \in [0, 1]$.

4. Experiment

In this section, we first introduce the implementation details in Sec. 4.1. Then, we present the main results of T-SEA in Sec. 4.2, reporting its performance on different detectors and attack methods. After that, we compare the proposed method with other state-of-the-art detection attack algorithms in Sec. 4.3 and perform the ablation study

Methods	YOLO v2	YOLO v3	YOLO v3tiny	YOLO v4	YOLO v4tiny	YOLO v5	Faster R-CNN	SSD	Black-Box Avg↓
AdvPatch	**5.66**	40.26	18.07	48.49	24.44	43.38	39.27	41.28	36.46
T-SEA(ours)	**1.73**	4.48	2.41	5.68	7.75	6.91	16.38	20.55	9.16 $^{27.30\downarrow}$
AdvPatch	51.85	**13.89**	51.17	57.16	58.43	70.47	51.46	59.79	57.19
T-SEA(ours)	31.02	**5.76**	35.38	42.37	25.3	58.02	37.62	52.26	40.28 $^{16.91\downarrow}$
AdvPatch	56.02	66.12	**1.39**	69.64	51.56	72.16	56.61	60.73	61.83
T-SEA(ours)	38.85	47.13	**0.51**	52.75	31.01	61.18	49.12	55.44	47.93 $^{13.90\downarrow}$
AdvPatch	37.41	37.18	17.58	**19.67**	26.91	46.37	44.67	43.07	36.17
T-SEA(ours)	9.34	6.46	9.49	**4.22**	16.65	11.66	16.24	29.43	14.18 $^{21.99\downarrow}$
AdvPatch	47.41	59.59	37.14	66.48	**14.50**	69.51	55.95	55.22	55.90
T-SEA(ours)	37.69	44.98	15.81	51.14	**4.11**	51.37	46.06	49.39	42.35 $^{13.55\downarrow}$
AdvPatch	46.9	54.93	29.20	62.16	46.15	**13.39**	47.71	50.73	48.25
T-SEA(ours)	11.61	16.77	10.73	32.53	17.55	**1.37**	17.14	30.78	19.59 $^{28.66\downarrow}$
AdvPatch	24.53	23.37	14.58	26.54	25.6	30.46	**8.62**	42.15	26.75
T-SEA(ours)	9.28	4.08	3.99	8.55	13.58	10.45	**3.08**	26.46	10.91 $^{15.84\downarrow}$
AdvPatch	32.43	62.93	52.03	66.22	49.07	51.19	47.42	**15.10**	51.61
T-SEA(ours)	12.06	30.90	9.73	27.33	9.08	17.38	28.54	**5.13**	19.29 $^{32.32\downarrow}$

Table 1. Comparisons between T-SEA and AdvPatch [32]. We attack each detector separately and use the remaining seven detectors as the black-box detectors. The proposed T-SEA performs much better than [32] on both white-box setting and black-box setting, demonstrating the effectiveness of the proposed self-ensemble strategies.

Method		White Box ↓	Black-Box Avg↓
Adam	AdvPatch	13.39	48.25
	T-SEA(ours)	1.37	19.59
SGD	AdvPatch	20.39	55.97
	T-SEA(ours)	1.66	15.85
MIM	AdvPatch	11.91	33.51
	T-SEA(ours)	1.43	15.43
BIM	AdvPatch	8.47	32.93
	T-SEA(ours)	1.62	19.15
PGD	AdvPatch	13.58	38.70
	T-SEA(ours)	1.60	18.06

Table 2. Comparisons of T-SEA and AdvPatch with Different Base Attack Methods on YOLO v5 (white-box). We select five classical base attack methods, including optimization-based methods (Adam and SGD) and iterative methods (MIM, BIM and PGD). The results show that T-SEA can enhance the performance of all these methods and performs much better than AdvPatch.

in Sec. 4.4. Finally, we show the transferability of T-SEA across the datasets and scenes in Sec. 4.5 and Sec. 4.6.

4.1. Implementation Details

Datasets In our experiments, we utilize the INRIA person dataset [10] to train and test our adversarial patch, whose training set and test set consist of 614 and 288 images, respectively. Meanwhile, to verify the transferability of the crafted patch, we select images containing person from COCO validation set (named COCO-person) and CCTV Footage of Humans* (named CCTV-person) as additional test data. The former contains 1684 human images in different scenes (e.g., sports playground, transportation routes, oceans and forests) and the latter contains 559 in-person images from camera footage.

Optimization Details We use INRIA train set as the training set and regard person as the target attack class. The patch size is 300×300, the input image size is 416×416, the batch size $BS = 8$, and the maximum epoch number $M = 1000$. For the E-baseline, we adjust the training patch scale from 0.2 in AdvPatch [32] to 0.15, set $\epsilon_1 = 1e-4, \epsilon_2 = 1e-4$ for the learning rate scheduler, and adopt the Adam [18] as the optimizer. For constrained data augmentation, we carry out constrained data augmentation via horizontal flip, slight color jitter, random resized crop, and random rotation; for model ShakeDrop, the constant e is set to 1 so that $\alpha \sim U(0, 2)$, and we perturb the model with the probability of $\varphi_s = 0.5$; for patch cutout, we set the fill value of the erased area $k = 0.5$, the ratio $\eta = 0.4$, and the probability $\varphi_c = 0.9$.

Evaluation Metric Following [14, 32], we use the Average Precision (AP) to measure the attack capability of the crafted patch (the lower AP, the better attack) and regard the detector's predictions of the clean data as the ground truth (i.e., $AP = 1$).

*https://www.kaggle.com/datasets/constantinwerner/human-detection-dataset

Method	White Box ↓	Black Box ↓							Black-Box Avg↓
	YOLO v2	YOLO v3	YOLO v3tiny	YOLO v4	YOLO v4tiny	YOLO v5	Faster R-CNN	SSD	
Gray	67.75	76.22	80.69	75.22	76.89	81.86	61.75	72.05	-
Random Noise	70.67	75.8	82.44	75.1	78.74	81.79	63.41	72.9	-
White	68.52	74.89	80.2	74.73	76.09	80.09	60.35	69.41	-
NPAP [14]	**38.03**	56.85	58.04	67.74	67.43	66.85	56.6	56.66	**61.45**
AdvCloak [36]	**33.74**	54.77	53.42	67.57	56.12	68.05	55.19	60.82	**59.42**
AdvPatch [32]	**5.66**	40.26	18.07	48.49	24.44	43.38	39.27	41.28	**36.46**
E-baseline(ours)	**3.61**	15.32	5.50	18.58	13.39	9.77	29.73	23.82	**16.59**
T-SEA(ours)	**1.73**	4.48	2.41	5.68	7.75	6.91	16.38	20.55	**9.16**

Table 3. Comparisons with Existing Detection Attack Methods. For clearer controlled observations, we list the results of gray, random noise, and white patch. Compared with existing methods, T-SEA achieves the best performance on both white-box and black-box attack.

4.2. Main Results

4.2.1 Results on Different Attacked Models

We systematically investigate attack performance of the proposed T-SEA on eight widely-used object detectors, and report the quantitative results in Tab. 1. Since T-SEA is improved based on AdvPatch [32], we also report the results of AdvPatch. The proposed T-SEA achieves significant improvements on both white-box and black-box performance of all eight detectors compared to [32], demonstrating the effectiveness of the proposed strategies. Meanwhile, for some white-box detectors (*e.g.*, YOLO v2), the black-box average AP can drop to around 10, exhibiting effective black-box performance of T-SEA.

4.2.2 Results of Different Attack Methods

T-SEA is not designed only for a specific base attack method; it is important for T-SEA to perform well on different base attack methods. In Tab. 2, we give the detailed results of T-SEA with different base attack methods on YOLO v5, which includes the optimization-based methods (e.g., Adam and SGD) and iterative methods (e.g., MIM, BIM and PGD). The results show that the proposed T-SEA can improve both white-box and black-box attack performance on all methods above, demonstrating that the performance gain caused by T-SEA is not limited to a specific method. That is to say, T-SEA may potentially work well with future base attack methods to further increase the transferability of their crafted AdvPatch.

4.3. Comparison with SOTA Methods

In this section, we compare T-SEA with the SOTA detection attack approaches, which all regard the YOLO v2 as the white-box model and evaluate on seven black-box detectors, and we also report the results of gray/random noise/white patches as control group. As reported on Tab. 3, 1) all adversarial patch performs much better than the control group; 2) our E-baseline and T-SEA achieve the highest performance among all adversarial approaches with the same inference setting (i.e., we ensure the perturbed area is same for each method), showing that compared to the existing single model attack method, the training strategies adjustment and self-ensemble strategies can effectively enhance the attack capability on both white-box and black-box of the crafted adversarial patch.

	White Box ↓	Black Box ↓
E-baseline	13.39	48.25
+ Constrained Data Aug.	18.47	42.42
+ Model ShakeDrop	12.54	39.40
+ Patch cutout	6.80	32.54
+ Combined Strategies	1.37	19.59

Table 4. Ablation Study of Self-Ensemble Strategies on YOLO v5 (white-box). Each self-ensemble strategy is added to the E-baseline to verify its individual performance gain. The results show that: 1) though using data augmentation alone degrades the white-box performance, all strategies can obviously improve the black-box performance of E-baseline; 2) combining all achieves non-trivial results on both white-box and black-box setting.

4.4. Ablation Study

4.4.1 Training Strategies Adjustment

Here we explore the performance gain from the training strategies adjustments. As discussed in the Sec. 3.3, we modify the learning rate scheduler and training patch scale of [32]. As illustrated in Fig. 4, the modified training scale can reduce the AP in test set more quickly and effectively, and the crafted patch has clearer adversarial patterns. The adjusted scheduler can also lead the detection loss to decrease more, and can be further combined with the modified training scale to achieve a better result.

4.4.2 Self-Ensemble Strategies

In this work, we are motivated by model training approaches that enhance generalization ability, and we propose self-ensemble strategies on the input data, the attacked

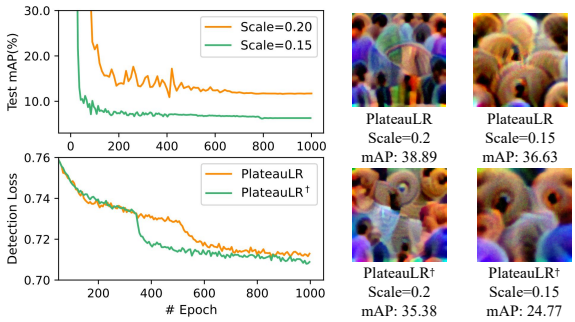

Figure 4. The improvements after adjusting learning rate scheduler (tagged as PlateauLR†) and patch scale at training stage. A small patch scale during training will cause the test AP to drop lower and faster, and the adjusted PlateauLR† can also cause the detection loss to further decrease.

Dataset	Method	White Box ↓	Black Box ↓
COCO-person	AdvPatch [32]	45.83	52.54
	T-SEA	37.28	38.87
CCTV-person	AdvPatch [32]	38.07	34.08
	T-SEA	38.71	19.91

Table 5. The Cross Datasets Transferability of T-SEA on YOLO v5 (white-box). Compared to the [32], the adversarial patch crafted by T-SEA has much stronger attack adaptability on person images from different datasets.

model, and the training patch, greatly augmenting adversarial transferability from themselves. Here we individually inspect the proposed strategies on our E-baseline (YOLO v5 as white-box detector) to explore the impact of each strategy. As reported in Tab. 4, though the constrained data augmentation will slightly decrease the white-box performance (other two strategies both improve the white-box performance), all these self-ensemble strategies increase the black-box results separately. and combining them can achieve the best performance.

4.5. Cross-Dataset Verification

The capability of performing cross-dataset attack is also significant for a well-optimized patch. Here, we apply the INRIA-trained patch on two different datasets, COCO-person and CCTV-person. Compared to INRIA, COCO-person has much more person images on different scenes, while CCTV-person focus on persons from the security camera. As shown in Tab. 5, the patch crafted by T-SEA achieves comparable results on the white-box setting with the AdvPatch, but obtains a much better black-box attack capability on both COCO-person and CCTV-person, indicating its strong black-box cross-data attacking capability.

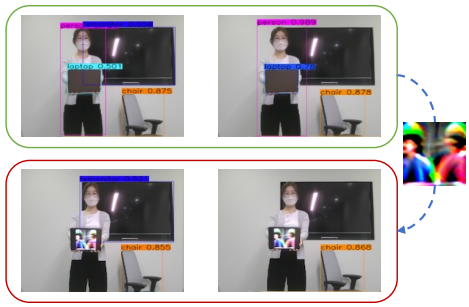

Figure 5. Physical Attack Demo of T-SEA. After showing the optimized patch on iPad, the YOLO v5 and SSD can not detect person.

4.6. Physical Verification

Although applying the patch attack in the physical world is not the main goal of our work, we believe the optimized patch that has high transferability may perform well in physical settings. Here we show a simple case in Fig. 5 that the patch shown on a iPad can successfully attack the human detection of YOLO v5 and SSD, without disrupting the detection process of other objects.

5. Future Work

Since most existing mainstream detectors are CNN-based, we focus on applying the self-ensemble strategies on these detectors in this work. However, the transformer-based detectors [6, 23] are achieving promising results, and the proposed model ShakeDrop can not directly applied on transformers (the others can), we will follow the motivation of designing the model ShakeDrop, to propose new approach and generate variants of transformer-based detector.

6. Conclusion

In this paper, we propose a novel transfer-based self-ensemble black-box attack on object detectors, achieving stable and excellent performance gains with various base attack methods on multiple popular object detectors. Firstly, with only slight training strategy adjustments, we improve the existing method's performance and regard it as our enhanced baseline. Then, based on this baseline, we propose a series of self-ensemble strategies to augment the input data, the attacked model, and the training patch from itself to significantly enhance the adversarial transferability of the optimized patch on black-box detectors. The comprehensive experimental results reveal the potential risk that an attacker with only one model could still succeed in a high transferability black-box attack.

7. Acknowledgements

This work was supported by National Key R&D Program of China (Grant No. 2022ZD0160305) and National Key Laboratory Open Fund of China (Grant No. 6142113210204).

References

[1] Martin Anthony and Peter Bartlett. *Neural network learning: Theoretical foundations*. cambridge university press, 1999. 4

[2] Song Bai, Yingwei Li, Yuyin Zhou, Qizhu Li, and Philip HS Torr. Adversarial metric attack and defense for person re-identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(6):2119–2126, 2020. 2

[3] Mikhail Belkin, Siyuan Ma, and Soumik Mandal. To understand deep learning we need to understand kernel learning. In *International Conference on Machine Learning*, pages 541–549. PMLR, 2018. 4

[4] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020. 3

[5] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *arXiv preprint arXiv:1712.04248*, 2017. 2

[6] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 8

[7] Shang-Tse Chen, Cory Cornelius, Jason Martin, and Duen Horng Polo Chau. Shapeshifter: Robust physical adversarial attack on faster r-cnn object detector. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 52–68. Springer, 2018. 3

[8] Zhaoyu Chen, Bo Li, Shuang Wu, Jianghe Xu, Shouhong Ding, and Wenqiang Zhang. Shape matters: deformable patch attack. In *European Conference on Computer Vision*, pages 529–548. Springer, 2022. 3

[9] Zhaoyu Chen, Bo Li, Jianghe Xu, Shuang Wu, Shouhong Ding, and Wenqiang Zhang. Towards practical certifiable patch defense with vision transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15148–15158, 2022. 3

[10] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. Ieee, 2005. 6

[11] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 4, 5

[12] Yinpeng Dong, Qi-An Fu, Xiao Yang, Tianyu Pang, Hang Su, Zihao Xiao, and Jun Zhu. Benchmarking adversarial robustness on image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 321–331, 2020. 2

[13] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193, 2018. 2

[14] Yu-Chih-Tuan Hu, Bo-Han Kung, Daniel Stanley Tan, Jun-Cheng Chen, Kai-Lung Hua, and Wen-Huang Cheng. Naturalistic physical adversarial patch for object detectors. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 6, 7

[15] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *European conference on computer vision*, pages 646–661. Springer, 2016. 4, 5

[16] Hao Huang, Yongtao Wang, Zhaoyu Chen, Zhi Tang, Wenqiang Zhang, and Kai-Kuang Ma. Rpattack: Refined patch attack on general object detectors. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2021. 1, 3

[17] Glenn Jocher, Alex Stoken, Jirka Borovec, NanoCode012, ChristopherSTAN, Liu Changyu, Laughing, tkianai, Adam Hogan, lorenzomammana, yxNONG, AlexWang1900, Laurentiu Diaconu, Marc, wanghaoyang0106, ml5ah, Doug, Francisco Ingham, Frederik, Guilhen, Hatovix, Jake Poznanski, Jiacong Fang, Lijun Yu, changyu98, Mingyu Wang, Naman Gupta, Osama Akhtar, PetrDvoracek, and Prashant Rai. ultralytics/yolov5, Oct. 2020. 3

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[19] Yuezun Li, Daniel Tian, Ming-Ching Chang, Xiao Bian, and Siwei Lyu. Robust adversarial perturbation on deep proposal-based models. *arXiv preprint arXiv:1809.05962*, 2018. 1, 3

[20] Jiadong Lin, Chuanbiao Song, Kun He, Liwei Wang, and John E Hopcroft. Nesterov accelerated gradient and scale invariance for adversarial attacks. In *International Conference on Learning Representations*. 2

[21] Siao Liu, Zhaoyu Chen, Wei Li, Jiwei Zhu, Jiafeng Wang, Wenqiang Zhang, and Zhongxue Gan. Efficient universal shuffle attack for visual object tracking. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2739–2743. IEEE, 2022. 2

[22] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 3

[23] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 8

[24] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 2

[25] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018. 4

[26] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pages 506–519, 2017. 2

[27] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017. 3

[28] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 3

[29] 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*, 28, 2015. 3

[30] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014. 4, 5

[31] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014*, 2014. 2

[32] Simen Thys, Wiebe Van Ranst, and Toon Goedemé. Fooling automated surveillance cameras: adversarial patches to attack person detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 0–0, 2019. 2, 3, 4, 6, 7, 8

[33] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 1999. 4

[34] Xingxing Wei, Siyuan Liang, Ning Chen, and Xiaochun Cao. Transferable adversarial attacks for image and video object detection. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 954–960, 2019. 1, 3

[35] Shudeng Wu, Tao Dai, and Shu-Tao Xia. Dpattack: Diffused patch attacks against universal object detection. *arXiv preprint arXiv:2010.11679*, 2020. 3

[36] Zuxuan Wu, Ser-Nam Lim, Larry S Davis, and Tom Goldstein. Making an invisibility cloak: Real world adversarial attacks on object detectors. In *European Conference on Computer Vision*, pages 1–17. Springer, 2020. 7

[37] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan Yuille. Adversarial examples for semantic segmentation and object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1369–1378, 2017. 1, 3

[38] Yoshihiro Yamada, Masakazu Iwamura, and Koichi Kise. Shakedrop regularization. 2018. 5

[39] Xue Ying. An overview of overfitting and its solutions. In *Journal of physics: Conference series*, volume 1168, page 022022. IOP Publishing, 2019. 4

[40] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 4

[41] Yusheng Zhao, Huanqian Yan, and Xingxing Wei. Object hider: Adversarial patch attack against object detectors. *arXiv preprint arXiv:2010.14974*, 2020. 3

[42] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13001–13008, 2020. 4, 5