

# ADC: Adversarial attacks against object Detection that evade Context consistency checks

Mingjun Yin\* Shasha Li\* Chengyu Song M. Salman Asif Amit K. Roy-Chowdhury  
Srikanth V. Krishnamurthy  
University of California, Riverside

{myin013,sli057}@ucr.edu, csong@cs.ucr.edu, {sasif,amitrc}@ect.ucr.edu, krish@cs.ucr.edu

## Abstract

*Deep Neural Networks (DNNs) have been shown to be vulnerable to adversarial examples, which are slightly perturbed input images which lead DNNs to make wrong predictions. To protect from such examples, various defense strategies have been proposed. A very recent defense strategy for detecting adversarial examples, that has been shown to be robust to current attacks, is to check for intrinsic context consistencies in the input data, where context refers to various relationships (e.g., object-to-object co-occurrence relationships) in images. In this paper, we show that even context consistency checks can be brittle to properly crafted adversarial examples and to the best of our knowledge, we are the first to do so. Specifically, we propose an adaptive framework to generate examples that subvert such defenses, namely, Adversarial attacks against object Detection that evade Context consistency checks (ADC). In ADC, we formulate a joint optimization problem which has two attack goals, viz., (i) fooling the object detector and (ii) evading the context consistency check system, at the same time. Experiments on both PASCAL VOC and MS COCO datasets show that examples generated with ADC fool the object detector with a success rate of over 85% in most cases, and at the same time evade the recently proposed context consistency checks, with a “bypassing” rate of over 80% in most cases. Our results suggest that “how to robustly model context and check its consistency,” is still an open problem.*

## 1. Introduction

Deep Neural Networks (DNNs) have been shown to be highly expressive and have achieved state-of-the-art (SOTA) performance on a wide range of computer vision tasks, such as object detection and classification. However, in 2014, Szegedy *et al.* [22] found that DNNs are vulnerable to carefully crafted and usually, visually inconspicuous

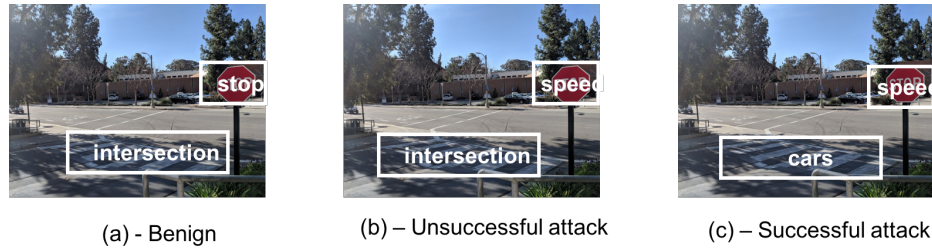
(to humans) perturbed inputs named adversarial examples, which lead DNNs to make incorrect predictions. Since then, an arms race between the generation of adversarial example attacks and defenses to thwart them, has taken off. Researchers have proposed attacks and defenses in image and video classification [20, 11, 12, 24, 28, 8, 25, 10], and object detection [3, 21, 30, 13].

Among existing defense mechanisms, checking intrinsic context consistencies within the input data has recently been showcased to be very effective, in various tasks. For example, spatial consistency has been used to detect adversarial attacks against semantic segmentation [26]; temporal consistency has been used to detect adversarial attacks against video classification [8, 25]; object-object context along with other kinds of context has been used to detect adversarial examples against objection detection [13]; audio-visual correlation has been used to detect adversarial examples against audio-visual speech recognition [15].

While these context-consistency-based defenses are shown to be effective against SOTA attacks, whether they can resist more advanced adaptive attacks remains unexplored. We hypothesize that if the context is (or can be) extracted using a neural network and the consistency checks are also performed using neural networks, then adaptive attacks could be feasible via a more complex optimization formulation. Specifically, we expect that it would be feasible to use existing optimization techniques to compute adversarial examples that can (a) fool the DNNs to make wrong prediction (e.g., misclassify an object) and (2) fool the consistency check modules thus bypassing the defense. This is shown diagrammatically using an example in Fig. 1.

In this work, we conduct the *first* study on composing adaptive attacks against context-consistency-based defenses. Using the very recent context-aware object detector proposed in [13] as an exemplar of defense, we demonstrate the feasibility of attacking context-consistency based models. We analyze two types of attack scenarios that are as follows. In a white-box attack scenario, we assume attackers have full knowledge of the consistency check modules.

\*equal contribution



**Figure 1:** Motivation for our proposed approach. (a) A benign image where a detector detects the stop sign and intersection. (b) Only the stop sign is perturbed and detected as a speed limit; however, a context-aware adversarial defense like [13] is able to easily detect that a speed limit sign does not occur at an intersection, and thus the attack fails. (c) Our goal is to not only perturb the stop sign into a speed limit sign, but also perturb other parts of the image so that it is contextually consistent, e.g., the intersection to objects that co-occur with the speed limit sign like other cars. Such an attack would be difficult to detect using methods like [13]. While we do not explicitly perturb the other objects, the proposed method will add perturbations such that the detections in the overall image become contextually consistent.

In our gray-box attack scenario, we assume attackers are aware of the consistency check mechanism (e.g., the neural network architecture), but have no access to either the parameters of the modules or the training data.

Specifically, we develop a framework that we call ADC for “Adversarial attacks against object Detection that evade Context consistency checks” to develop white box based adversarial examples. To generate these white-box adaptive attacks in ADC, we formulate a joint optimization problem where the loss function is composed of two parts: (a) the loss function of the DNN prediction, and (b) the loss function relating to the consistency check. One particular challenge in solving the joint optimization problem is that in current context aware defenses, the context is extracted from the intermediate layers of the DNN, while the consistency checks are done with separate DNN module(s). For instance, the context-aware object detector proposed in [13] extracts context profiles from internal Gated Recurrent Units (GRUs) and checks for relationship inconsistencies using a bank of auto-encoders. To solve this challenge, we propose a 3-step optimization technique.

With respect to the gray-box adaptive attack, our aim is to expand ADC to investigate the transferability of “context-aware” attacks. Specifically, since we do not have full access to the consistency check modules, we train a surrogate defense module and then solve the joint optimization problem with the surrogate module. Then, we test the generated attacks with the gray-box model.

The main contributions of our work are as follows:

- To the best of our knowledge we are the first to investigate adaptive adversarial attack against consistency check based defenses.
- We develop a framework, ADC, for generating practically viable adaptive adversarial examples against context consistency based defenses. To generate adversarial examples using ADC, we formulate a joint optimization problem, and find the approximate solution efficiently by

solving three sub-optimization problems in a pipeline.

- We conduct extensive experiments on the large-scale PASCAL VOC and MS COCO dataset. Our method yields very high fooling rates against the object detection system while simultaneously, evading the context consistency check with high success rates, for three typical attack goals [3, 21, 30]: mis-categorization, hiding and appearing attacks.

## 2. Related Work

To craft adversarial examples, early methods [22, 6, 9] either take one step to apply perturbations to the input along the sign of gradient of the designed misclassification loss function, or take multiple small steps iteratively while adjusting the direction after each step. Some notable works that considered both attacks and defenses include [17, 2, 7, 27, 4, 1].

One promising defense strategy explored recently, is to leverage violations of intrinsic **consistencies** that are expected to exist in the input data (i.e., context) to detect adversarial examples. Specifically, context relates to relationships that are typical in the training samples; adversarial examples are expected to cause violations in such relationships. Xiao *et al.* [26] show via an empirical study that the surrounding spatial context information in semantic segmentation has different characteristics for benign and adversarial examples, and proposed a spatial consistency-based adversarial detection method. Jia *et al.* [8] define an exception frame as one whose prediction label differs from the labels of adjacent frames and showed that exception frames are more likely in adversarial video inputs than in benign video streams. Xiao *et al.* [25] proposed to use temporal consistency to detect adversarial examples in video. Ma *et al.* [15] found that the correlation between audio and video in adversarial examples is lower than benign examples due to the added perturbations. Yin *et al.* [29] abstracted a pseudo-language to describe the scene and utilizing natu-

ral language processing technique to measure the overall context consistency for a given image. Wang *et al.* [23] proposed a context consistency check mechanism to detect adversarial attacks against ReID systems.

In a very recent work, Li *et al.* [13] construct the context profile, which captures four types of relationships among region proposals (i.e., spatial context, object-object context, object-background context, and object-scene context) in the object detection task to detect adversarial perturbations. Based on the observation that the context profile of each object category is usually unique, they use a bank of auto-encoders to learn the benign context profile distribution for each category and detect adversarial attacks as context consistency violations (i.e., high reconstruction error of auto-encoders) during testing. Experiments show that the detection ROC-AUC is over 0.95 for various adversarial attacks. Although their method has been very successful with most attacks that are currently prevalent, in this paper, we show that it is possible to design context-aware attacks that can defeat such systems. To the best of our knowledge, we are the first to propose adaptive attacks to evade consistency check based defenses in object detection.

### 3. The ADC Framework

In this section, we describe how we design our ADC framework to generate adaptive attacks against object detection systems with a context-inconsistency-based adversarial example detection mechanism. We first give a brief introduction of how a context-inconsistency-based attack detection mechanism like [13] works. Then we describe the threat model in detail. Subsequently, we formulate an optimization problem to jointly fool the detector and evade the context consistency check (referred to as joint optimization) to generate our white-box adaptive attack and describe how we find approximate solutions to this optimization problem via a three-step strategy. Lastly, we describe the strategy we use in ADC to realize a gray-box adaptive attack.

#### 3.1. Context-inconsistency-based attack detection

In this subsection, we illustrate how context-inconsistency-based attack detection mechanisms work. In general, these defense mechanisms extract some context profiles from the input data and check whether the extracted context profiles match some known distributions. If a testing time context profile does not fit into a known distribution, then an anomaly is detected. Such an anomaly could be introduced by previously unknown data or adversarial attacks.

Take the SCEME model proposed by Li *et al.* [13] as an example. SCEME aims to defend the Faster RCNN [19] object detector against adversarial examples. Given an input image, Faster RCNN will output a set of region proposals, each of which is associated with bounding box information,

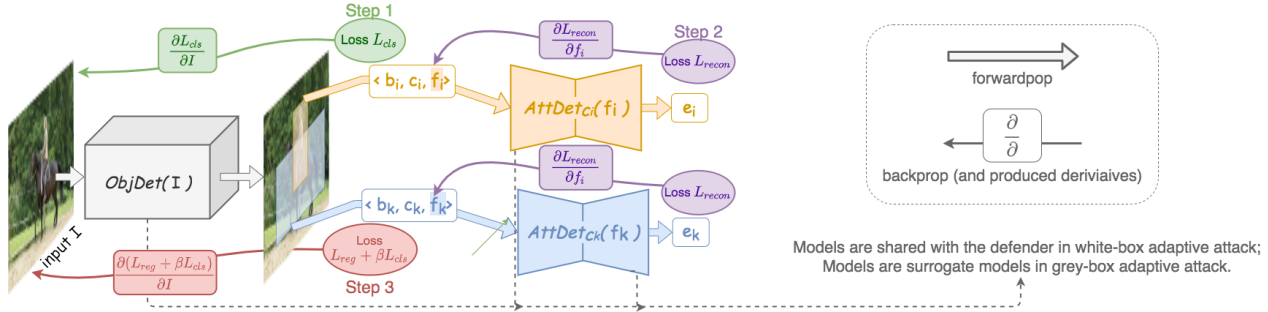
and a category probability vector. To detect adversarial attacks, SCEME extracts a context profile for each proposed region. Each context profile is a vector of intermediate features, computed based on a fully connected context graph, where each node is a region proposal and edge weights are learned from the training data. Because each category of objects is likely to have distinguished object-object relationships, SCEME uses an auto-encoder to learn the benign distribution of context profiles for each category. To detect context inconsistencies, the context profile is input to the auto-encoder associated with the predicted category. A high reconstruction error of the auto-encoder implies an abnormal context profile and thus, the corresponding region would be marked as containing adversarial perturbations.

#### 3.2. Problem definition and threat model

In this subsection, we generalize and formulate the problem of attacking an object detector with context-inconsistency-based attack detection, such as the model proposed by Li *et al.* [13]. We denote the input scene image as  $I$ , the object detection function as  $ObjDet(\cdot)$ , and the detection results as  $R_I = ObjDet(I) = [r_1, r_2, \dots, r_N]$ , where  $N$  is the total number of proposed regions over  $I$ . For simplicity of exposition, we will use  $R$  instead of  $R_I$  hereafter. To detect adversarial attacks, the context-consistency checker extracts a context profile for each proposed region. Therefore, for each region proposal, we have the predicted category label denoted as  $c_i$ , the bounding box coordinates denoted as  $b_i$ , and the context profile denoted as  $f_i$ , i.e.,  $r_i = \langle b_i, c_i, f_i \rangle$ . If we use  $C = [c_1, c_2, \dots, c_N]$  to denote the predicted categories of all region proposals,  $B = [b_1, b_2, \dots, b_N]$  to denote corresponding bounding boxes, and  $F = [f_1, f_2, \dots, f_N]$  to denote all context profiles, then  $R = \langle B, C, F \rangle$ . Because each category of objects is likely to have distinguished object-object relationships, we denote the function of the attack detector (i.e., auto-encoder) associated with each category  $c \in L$  ( $L$  is the complete category set) as  $AttDet_c(\cdot)$ . For each region, we can further calculate the reconstruction error of a context profile as  $e_i = AttDet_{c_i}(f_i)$ .

We aim to perform targeted attacks where the perturbed object is misclassified to an attacker desired category. Compared to untargeted attacks, targeted attacks are harder and potentially provide more insights since subversion to certain target categories could be harder to bypass the context-consistency checks than others. We consider two threat models in designing ADC.

- **White-box adaptive attack** where we assume attackers have full knowledge about the context-consistency check mechanism, i.e.,  $ObjDet(\cdot)$  and  $AttDet_c(\cdot), \forall c \in L$  are known functions to attackers.
- **Gray-box adaptive attack** where we assume the attackers are aware of the context-consistency check mecha-



**Figure 2:** Our proposed ADC framework. There are two attack goals in ADC: a) fooling the object detector; b) bypassing the context-inconsistency checks. The first goal is achieved with the optimization problem defined in step 1 which perturbs the input image to make the object detector output the target classification labels. The second goal is achieved with the two optimization problems defined in step 2 and step 3. In step 2, the context profile features  $f_i$  are perturbed to make the attack detector (auto-encoders) output low anomaly scores. In step 3, the input image is further perturbed to output the target perturbed context profiles and the target classification labels. Note that in gray-box attack, the  $ObjDet(\cdot)$  and  $AttDet_c(\cdot)$  are trained separately by the attacker, and it is not the same model used by the defender.

nism, e.g., the definition of context profile and the use of the auto-encoder bank. However, they have *no* access to either the parameters of the models or the training data. In other words,  $ObjDet(\cdot)$  and  $AttDet_c(\cdot), \forall c \in C$  are unknown functions to attackers, but the attackers could estimate these functions based on their own training samples.

### 3.3. Joint optimization for white-box attack

Our hypothesis is that if the context profile is extracted and checked using a neural network, and the parameters of the neural network are known (i.e., the white-box attack scenario), then we should be able to search for adversarial perturbations that can both fool the object detector and satisfy context-consistency constraints using joint optimization. In this subsection, we describe how to realize our white-box adaptive attack. Specifically, we will use the SCEME model as an example to show how we calculate context-aware perturbations that can both fool its object detector and bypass the inconsistency check applied by its auto-encoder bank.

#### 3.3.1 Formulation

**Fooling the object detector.** To fool the object detector to misclassify the victim object into a target label, perturbations should be calculated such that the labels of the corresponding regions are flipped from  $c_s$  to  $c_t$ , where  $c_s$  denotes the ground truth label of the victim object, and  $c_t$  denotes the target label assigned by the attacker. We use  $C^*$  to denote the target labels for all the region proposals where ground truth labels are used for untargeted objects and label  $c_t$  are used for the targeted object. We use  $\Delta I_{fool}$  to denote the perturbation needed to fool the object detector. To provide imperceptibility of the perturbation, the  $L_p$  norm (e.g.  $L_1, L_2, L_{inf}$  and etc.) of  $\Delta I_{fool}$  is constrained to be lower than a chosen threshold viz.,  $\tau_{fool}$ . Therefore, the optimization problem can be formulated as in Eqn. 1. The

classification loss  $L_{cls}$  is the cross-entropy loss between the predicted probability vector  $P_C$  from the object detector and the target label  $C^*$ . Note that, in the optimization process we do not change the bounding box coordinates of the region proposals.

$$\begin{aligned} & \text{minimize} && L_{cls}(P_C, C^*), P_C \leftarrow ObjDet(I + \Delta I_{fool}) \\ & \text{subject to} && \|\Delta I_{fool}\|_p \leq \tau_{fool} \end{aligned} \quad (1)$$

**Bypassing the context-inconsistency checks.** To bypass context-inconsistency checks, perturbations should be calculated such that the context profiles of all region proposals should have below threshold anomaly scores (i.e., all auto-encoders in SCEME exhibit low reconstruction errors). We use  $\Delta I_{bypass}$  to denote the perturbation needed on the image pixels, to bypass the context-inconsistency check. The optimization problem can be formulated as in Eqn.2. The reconstruction loss  $L_{recon}$  is the smooth  $L_1$  loss as used in [13] between the original context profile and the context profile reconstructed with the auto-encoders and  $\tau_{bypass}$  is the threshold that bounds the perturbation. Note that the auto-encoder used for each context profile  $f_i$  could be dynamically chosen according to the predicted category  $c_i$ .

$$\begin{aligned} & \text{minimize} && \sum_{f_i \in F} L_{recon}(f_i, AttDet_{c_i}(f_i)), \\ & \text{subject to} && F \leftarrow ObjDet(I + \Delta I_{bypass}) \\ & && \|\Delta I_{bypass}\|_p \leq \tau_{bypass} \end{aligned} \quad (2)$$

**Joint optimization formulation.** If we denote the perturbations on the input image  $I$  as  $\Delta I$ , then the overall optimization problem formulation, that would yield a perturbation that fools the object detector, while at the same time evades the context-inconsistency checks is given by Eqn. 3. In this equation,  $\alpha$  is a parameter that is used to achieve a good trade-off between fooling the object detector and evading

the consistency check.  $\Delta I$  is the total adversarial perturbation applied on the image, which combines the fooling and bypassing perturbations.  $\tau$  is the threshold which bounds the total perturbation that can be applied (to ensure imperceptibility).

$$\begin{aligned} & \underset{\Delta I}{\text{minimize}} && L_{cls}(P_C, C^*) + \alpha \sum_{f_i \in F} L_{recon}(f_i, AttDet_{c_i}(f_i)), \\ & && P_C, F \leftarrow ObjDet(I + \Delta I) \\ & \text{subject to} && \|\Delta I\|_p \leq \tau \end{aligned} \quad (3)$$

### 3.3.2 Three-step strategy

Unfortunately, the joint optimization problem in Eqn. 3 cannot be solved directly, for two reasons: (1) the anomaly detector  $AttDet_c(\cdot)$  could be dynamically chosen according to the predicted category of each region proposal and (2) the anomaly detection networks could be disconnected from the object detection network and in such cases (as in [13]), back-propagation over the pair of networks (object detection and anomaly detection) is infeasible. To overcome these key challenges, we propose a three-step optimization to search for an approximate solution, as depicted in Algorithm 1.

In step one (Lines 1-4 in Algorithm 1), we use existing attack techniques like IFSGM [9] to search for perturbations  $\Delta I_{fool}$  that can cause the target object to be misclassified (i.e., solving Eqn. 1).  $t$  is the iteration counter for IFSGM. We denote the perturbed image calculated in step one as  $I' = I + \Delta I_{fool}$ . Similar to [2],  $NormProjection(\cdot)$  is to project the perturbation back to the norm ball so that the  $L_p$  norm of the perturbation is under the pre-defined threshold  $\tau_{fool}$ . For example, if the  $L_{inf}$  norm is used, then  $NormProjection(x)$  is equal to  $\tau_{fool}$  if  $x > \tau_{fool}$ , and is equal to  $-\tau_{fool}$  if  $x < -\tau_{fool}$ <sup>1</sup>, and is equal to  $x$  otherwise.

Next, we extract the context profiles  $F$  (which consists of node features, and relationships between region proposals as edge features [13]), from the perturbed image  $I'$  (Line 6 in Algorithm 1). In step two (Lines 7-11 in Algorithm 1), we search for perturbations  $\Delta F$ , such that the perturbed context profiles yield low reconstruction errors (i.e., solving Eqn.4).

$$\begin{aligned} & \underset{\Delta F}{\text{minimize}} && \sum_{f_i \in F} L_{recon}(f_i + \Delta f_i, AttDet_{c_i}(f_i + \Delta f_i)) \\ & \text{subject to} && \|\Delta f_i\|_p \leq \tau_F, \forall \Delta f_i \in \Delta F \end{aligned} \quad (4)$$

We denote the perturbed context profiles as  $F^* = F + \Delta F$ , and will use these as the target context profiles in step three. Note that Eqn.4 is different from Eqn. 2; in Eqn. 2 the

<sup>1</sup>Note that appropriate thresholds are used when this function used at different places in the algorithm.

---

### Algorithm 1: White-box Adaptive Attack

---

**Input** :  $I, C^*, ObjDet(\cdot), AttDet_c(\cdot), \forall c \in L$   
**Output**: Adversarial example  $I'$

- 1  $I' = I$
- 2 **for**  $t \leftarrow 1$  to  $T_1$  **do**
- 3      $\Delta I_{fool} \leftarrow$  solve *Equ. 1* with  $I'$  and  $C^*$
- 4      $I' = I + NormProjection(I' + \Delta I_{fool} - I)$
- 5 **end**
- 6  $F \leftarrow ObjDet(I')$
- 7  $F^* = F$
- 8 **for**  $t \leftarrow 1$  to  $T_2$  **do**
- 9      $\Delta F \leftarrow$  solve *Equ. 4* with  $F^*$
- 10     $F^* = F + NormProjection_F(F^* + \Delta F - F)$
- 11 **end**
- 12  $I'' = I'$
- 13 **for**  $t \leftarrow 1$  to  $T_3$  **do**
- 14      $\Delta I'_{bypass} \leftarrow$  solve *Equ. 5* with  $I', C^*$  and  $F^*$
- 15      $I'' = I + NormProjection(I'' + \Delta I'_{bypass} - I)$
- 16 **end**
- 17 **return**  $I'$

---

optimization is on the input image instead of extracted the context profiles. Similarly, Eqn. 2 involves both the object detector and the auto-encoder bank instead of only the auto-encoder bank in Eqn.4. In step three (Lines 12-16 in Algorithm 1), with the target context profiles  $F^*$  and the target category labels  $C^*$ , we search for perturbations  $\Delta I'_{bypass}$  so that the perturbed image with this additional perturbation i.e.,  $I'' = I' + \Delta I'_{bypass}$ , will output both the target category labels  $C^*$  and the target context profiles  $F^*$  (i.e., solving Eqn 5).

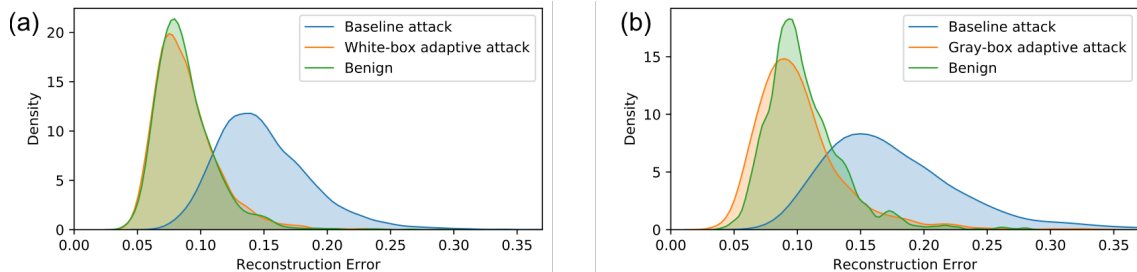
$$\begin{aligned} & \underset{\Delta I'_{bypass}}{\text{minimize}} && \sum_{f_i \in F} L_{reg}(F, F^*) + \beta L_{cls}(P_C, C^*) \\ & && P_C, F \leftarrow ObjDet(I' + \Delta I'_{bypass}) \\ & \text{subject to} && \|\Delta I_{fool} + \Delta I'_{bypass}\|_p \leq \tau \end{aligned} \quad (5)$$

### 3.4. Leveraging transferability for gray-box attack

The proposed framework can be extended to a gray-box setting. As noted in Fig. 2, in this setting, the attacker needs to train its own model for the context-aware object detection. We assume that the attacker has access to data similar to that used by the defender, i.e., data comes from the same distribution and label space; however, the samples used to train the models on the attacker’s side is different from that on the defender’s side. This makes the optimization problem mentioned above more challenging as the object detection and context profiles are expected to have higher uncertainty. This is analyzed experimentally in § 4.3.

## 4. Experimental Analysis

We conduct comprehensive experiments on two large-scale object detection datasets to evaluate attacks generated



**Figure 3:** (a)The reconstruction error distributions of the context profiles from benign samples, adversarial samples generated by the baseline non-adaptive method, and adversarial sample generated by ADC method. The adversarial examples are generated in the white-box setting. We observe that ADC method is able to perturb the context profiles and mimic the benign context profile distributions.(b)The reconstruction error distributions of the context profiles from benign samples, adversarial samples generated by the baseline non-adaptive method in the white box setting, and adversarial sample generated by ADC method in the gray-box setting. We observe that when the attacker has no access to the oracle models, the context profile distribution learnt with surrogate models is very similar to the ground truth benign context profile distribution. The context profiles transfer well across models, so as the perturbation on the context profiles.

**Table 1:** Attack performance for three attack goals on the PASCAL VOC dataset.

Attack Method	Fooling Rate			Bypass Rate		
	Mis-categorization	Hiding	Appearing	Mis-categorization	Hiding	Appearing
Non-adaptive attack (baseline)	97.02%	98.57%	90.51%	19.92%	30.14%	43.62%
Adaptive attack (ours)	88.63%	98.57%	68.20%	90.51%	87.20%	96.39%

by ADC in both white-box and gray-box settings. Inspired by previous works [3, 21, 30], we evaluate our approach with three attack goals:

- *Mis-categorization attack* where the object detector is to mis-categorize the perturbed object as belonging to a different category.
- *Hiding attack* where the object detector is to fail in recognizing the presence of the perturbed object, which happens when the confidence score is low or the object is recognized as background.
- *Appearing attack* where the object detector is to wrongly conclude that the perturbed background region contains an object of a desired category.

For the mis-categorization attack and the appearing attack, we perform targeted attacks and randomly choose the target category labels, in our experiments.

We evaluate the performance of ADC with two metrics. Recall that we have two goals: (a) to fool the object detector to achieve the mis-categorization attack, the hiding attack, or the appearing attack; and (b) to bypass the consistency checks of the attack detectors (i.e., the auto-encoder bank).

- *Fooling rate* is the metric for the former goal, and indicates how many attacks from all the tried ones, succeed in fooling the object detector.
- *Bypass rate* is the metric for the later goal, and quantifies how many attacks from those that fool the object detector, bypass the context consistency checks.

An image is detected as natural/benign if the reconstruction errors of all the context profiles in the image, are lower

than the threshold. It is detected as containing adversarial perturbations, if the reconstruction error of any context profile in the image is higher than the threshold. In other words, the maximum reconstruction error computed from the image (referred to as reconstruction error for simplicity hereon), is used to decide whether the image has violated the context consistency check or not. To report the bypass rate, we need to fix the threshold. The threshold for the reconstruction error is chosen so as to make the false positive rate equal to 0.1; here, the false positive rate is the probability that a benign image is wrongly detected as perturbed.

#### 4.1. Implementation details

**Datasets.** We use both PASCAL VOC [5] and MS COCO [14]. PASCAL VOC contains 20 object categories. Each image, on average, has 1.4 categories and 2.3 instances. *VOC07trainval* and *VOC12trainval* are used as the training set, and the testing is carried out on *VOC07test*. MS COCO contains 80 categories. Each image, on average, has 3.5 categories and 7.7 instances. *coco14train* and *coco14valminusminival* are used for training, and the test evaluations are carried out on *coco14minival*. Note that COCO has few examples for certain categories. Similar to [13], we evaluate with the 11 categories that have the largest numbers of extracted context profiles.

**Attack Implementations.** To launch our adaptive attack, we add perturbations on the whole image, which is typical for digital attacks. We will show the experimental results on adding perturbations to individual object regions in § 4.4. The hyper-parameters  $T_1$ ,  $T_2$ , and  $T_3$  are empirically chosen

**Table 2:** Attack performance for three attack goals on the MS COCO dataset

Attack Method	Fooling Rate			Bypass Rate		
	Mis-categorization	Hiding	Appearing	Mis-categorization	Hiding	Appearing
Non-adaptive attack (baseline)	93.07%	97.27%	78.13%	11.89%	11.19%	19.35%
Adaptive attack (ours)	86.90%	90.82%	66.36%	83.67%	75.52%	86.08%

to be 10, 1, and 1 separately. The step size for updating the perturbation on the input image is 1 and that for updating the perturbation on the context profile is 0.1.  $L_{inf}$  is used and the  $\tau$  is set to be 10, and is the same as in previous works [16, 18, 12].  $\tau_F$  is set to be 0.1.  $\beta$  is set to be 1.

## 4.2. Evaluation of white-box attack performance

**Baseline.** To understand the performance of the proposed context-aware adaptive attack method, we use a vanilla non-adaptive attack method as the baseline where the perturbation is calculated purely according to Eqn. 1 (i.e., simply aiming to fool the object detector without bypassing the context-consistency checks).

**Visualizing the reconstruction error.** We plot the reconstruction error calculated from benign images, perturbed images with the non-adaptive method, and perturbed images with the adaptive method in Fig. 3(a), with mis-categorization as the attack goal. We observe that the distribution of reconstruction errors of benign images differs from the distribution of perturbed images. Specifically, the reconstruction errors of the perturbed images are generally higher because context is violated in these cases. This is why, the context consistency check based detection works in [13]. After we apply the perturbation with the proposed adaptive attack method, the reconstruction error of the perturbed images is low and the distribution is very similar to the that of the benign images. We also plot the distribution figure per target attack category in the supplementary material and we have consistent observations for all the categories. Therefore, we conclude qualitatively that the adaptive attack method is context-aware and offers promise in bypassing the context consistency check.

**Adaptive attack performance.** To quantitatively evaluate the adaptive attack performance, we report the fooling rate and the bypass rate on both PASCAL VOC and MS COCO in Tab. 1 and Tab. 2. We observe that ADC achieves a lower fooling rate compared to the non-adaptive baseline attack. For example the fooling rate of mis-categorization attack on the PASCAL VOC dataset decreases from 97.02% to 88.63%. However, ADC significantly outperforms the baseline method in terms of the bypass rate. For example, only 19.92% of the baseline mis-categorization attacks evade the context-inconsistency based attack detection, while 90.51% of the mis-categorization attacks launched by ADC evade this attack detection. We observe that on the MS COCO dataset, ADC improves the bypass rate from less than 20% (with the baseline) to more than 75% for all the three attack

goals (mis-categorization, hiding and appearing), which corresponds to over a 55% improvement.

## 4.3. Evaluation of gray-box attack performance

In gray-box attack, we use *VOC12trainval* as the training set to train a surrogate system (object detector + auto-encoder bank) and use *VOC07test* to train the target system’s model. We generate perturbations by solving the optimization problem with the surrogate neural networks and the test it on the target system.

**Visualizing the reconstruction error.** We calculate the reconstruction error with the target system and plot the distributions in Fig. 3(b). The attack goal is also mis-categorization. We observe that although the perturbations are calculated with the surrogate system, they seem to transfer well onto the target system and the distribution of the reconstruction error of the perturbed images mimicked that of the benign images quite well.

**Adaptive attack performance.** We report in Tab. 3 the fooling rate and bypass rate of both white-box and gray-box adaptive attacks for all the three attack goals. We observe that compared to the white-box attack, the fooling rates of gray-box attack for all the three attack goals are slightly lower. However, we observe that, the bypass rates in gray-box setting are comparable to the white-box setting. This together with Fig. 3(b), implies that the context profile distribution learnt with surrogate models is very similar to the ground truth benign context profile distribution. The context profiles transfer well across models, and so do the perturbations on the context profiles.

## 4.4. Ablation study

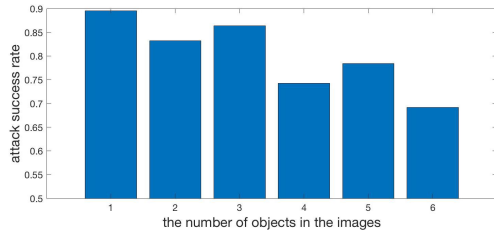
Next, we explore if we can bypass the context consistency check by perturbing only the target object region (defined by the ground-truth bounding box). By constraining the perturbation region, we have less power on perturbing the context profiles. The results are shown in Tab. 4. Compared to whole-image perturbations, although the single-object perturbations are able to retain comparable bypass rates, they lead to much lower fooling rates. The fooling rates of mis-categorization, appearing, and hiding attacks drop from 88.63% to 66.74%, from 68.20% to 34.54%, and from 98.57% to 88.63%, respectively. This result is aligned with the expectation that (a) perturbing a single object is more likely to cause context violations in mis-categorization and appearing attacks and thus the attacks are less likely to succeed, and (b) hiding attacks are more

**Table 3:** The gray-box attack performance for three attack goals on the PASCAL VOC dataset

Threat Model	Fooling Rate			Bypass Rate		
	Mis-categorization	Hiding	Appearing	Mis-categorization	Hiding	Appearing
White-box	88.63%	98.57%	68.20%	90.51%	87.20%	96.39%
Gray-Box	88.05%	85.33%	68.09%	92.34%	86.57%	96.38%

**Table 4:** Attack performance when attacking only the target object region for three attack goals on the PASCAL VOC dataset

Attack Region	Fooling Rate			Bypass Rate		
	Mis-categorization	Hiding	Appearing	Mis-categorization	Hiding	Appearing
Whole Image	88.63%	98.57%	68.20%	90.51%	87.20%	96.39%
Target Object	66.74%	88.63%	34.54%	88.70%	89.33%	95.15%

**Figure 4:** The attack success rates on MS COCO dataset for images with different number of objects. We observe that when more objects present, the attack success rates (fooling rate \* bypass rate) tend to decrease. The reason could be with more objects, it is harder to perturb the target object and all the other objects in a context consistent way.

likely to succeed because they generally do not cause context violations.

**Figure 5:** Two adversarial attack examples: (a) to misclassify the selected car instance into TV monitor and bypass the context-inconsistency check, ADC perturbed a background region into table; (b) to misclassify the selected car instance into bicycle and bypass the context-inconsistency check, ADC perturbed a background region into person.

#### 4.5. Analysis Study

We explore what kind of attacks are harder to succeed compared to others. Specially, we test whether the number of objects in the images affect the attack performance. Here we define a new metric, attack success rate, which de-

scribes how many attacks out of all the tried ones succeed in *both* fooling and bypassing (i.e., captured by fooling rate multiplied by bypass rate). As shown in Fig. 4, we observe that when more objects are present, the attack success rates tend to decrease. We suspect the reason is that with more objects, it is harder to perturb the target object and all the other objects in a context-consistent way.

We then dive deep into one adaptive attack example and see how the context is perturbed. Fig. 5(a) attack goal is to mis-categorize the selected car instance into a TV monitor. After attacking with ADC, we observe that not only the car instance is misclassified to TV monitor, the parking lot is also misclassified as a table instance to make the context more consistent and thus help to bypass the consistency check. Similarly, in Fig. 5(b), when we try to mis-categorize the selected car instance into a bicycle with ADC, we observe that a person instance appears, overlapping the “bicycle” to make the context more consistent.

## 5. Conclusions

In this paper, we show that recent defense strategies that use context consistency checks for detecting adversarial examples, can be subverted by appropriately crafted attacks that jointly consider the objectives of fooling the object detector and bypassing the consistency checks. We develop a framework which we call ADC, to generate both adaptive white-box and gray-box attacks, that are successful in this joint endeavor. To the best of our knowledge, we are the first to show this possibility, and our work highlights the inadequacies in current context models for defending adversarial examples. Our experiments on both the PASCAL VOC and MS COCO datasets show very high rates of both fooling the object detector (typically over 85 %) and evading the context consistency checks (typically over 80 %). We believe that future research on building better context models, possibly tailored to adversarial example defense, may be needed to truly make such approaches robust.

**Acknowledgments.** This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Agreement No. HR00112090096. Approved for public release; distribution is unlimited.



## References

- [1] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International conference on machine learning*, pages 274–283. PMLR, 2018.
- [2] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57. IEEE, 2017.
- [3] 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.
- [4] Guneet S Dhillon, Kamyar Azizzadenesheli, Zachary C Lipton, Jeremy Bernstein, Jean Kossaifi, Aran Khanna, and Anima Anandkumar. Stochastic activation pruning for robust adversarial defense. *arXiv preprint arXiv:1803.01442*, 2018.
- [5] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.
- [6] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.
- [7] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens Van Der Maaten. Countering adversarial images using input transformations. *arXiv preprint arXiv:1711.00117*, 2017.
- [8] Xiaojun Jia, Xingxing Wei, and Xiaochun Cao. Identifying and resisting adversarial videos using temporal consistency. *arXiv preprint arXiv:1909.04837*, 2019.
- [9] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net, 2017.
- [10] Shasha Li, Abhishek Aich, Shitong Zhu, M Salman Asif, Chengyu Song, Amit K Roy-Chowdhury, and Srikanth Krishnamurthy. Adversarial attacks on black box video classifiers: Leveraging the power of geometric transformations. *Conference on Neural Information Processing Systems*, 2021.
- [11] Shasha Li, Karim Khalil, Rameswar Panda, Chengyu Song, Srikanth V Krishnamurthy, Amit K Roy-Chowdhury, and Ananthram Swami. Measurement-driven security analysis of imperceptible impersonation attacks. *arXiv preprint arXiv:2008.11772*, 2020.
- [12] Shasha Li, Ajaya Neupane, Sujoy Paul, Chengyu Song, Srikanth V Krishnamurthy, Amit K Roy Chowdhury, and Ananthram Swami. Adversarial perturbations against real-time video classification systems. *arXiv preprint arXiv:1807.00458*, 2018.
- [13] Shasha Li, Shitong Zhu, Sudipta Paul, Amit Roy-Chowdhury, Chengyu Song, Srikanth Krishnamurthy, Ananthram Swami, and Kevin S Chan. Connecting the dots: Detecting adversarial perturbations using context inconsistency. In *European Conference on Computer Vision*, pages 396–413. Springer, 2020.
- [14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.
- [15] Pingchuan Ma, Stavros Petridis, and Maja Pantic. Detecting adversarial attacks on audio-visual speech recognition. *arXiv preprint arXiv:1912.08639*, 2019.
- [16] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1765–1773, 2017.
- [17] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 582–597. IEEE, 2016.
- [18] Konda Reddy Mopuri, Utkarsh Ojha, Utsav Garg, and R Venkatesh Babu. Nag: Network for adversary generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 742–751, 2018.
- [19] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015.
- [20] Mahmood Sharif, Sruti Bhagavatula, Lujio Bauer, and Michael K Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 1528–1540, 2016.
- [21] Dawn Song, Kevin Eykholt, Ivan Evtimov, Earlene Fernandes, Bo Li, Amir Rahmati, Florian Tramer, Atul Prakash, and Tadayoshi Kohno. Physical adversarial examples for object detectors. In *12th {USENIX} Workshop on Offensive Technologies ({WOOT} 18)*, 2018.
- [22] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.
- [23] Xueping Wang, Shasha Li, Min Liu, Yaonan Wang, and Amit K. Roy-Chowdhury. Multi-expert adversarial attack detection in person re-identification using context inconsistency. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15097–15107, October 2021.
- [24] Xingxing Wei, Jun Zhu, Sha Yuan, and Hang Su. Sparse adversarial perturbations for videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8973–8980, 2019.
- [25] Chaowei Xiao, Ruizhi Deng, Bo Li, Taesung Lee, Benjamin Edwards, Jinfeng Yi, Dawn Song, Mingyan Liu, and Ian Molloy. Advit: Adversarial frames identifier based on temporal consistency in videos. In *Proceedings of the IEEE*

- International Conference on Computer Vision*, pages 3968–3977, 2019.
- [26] Chaowei Xiao, Ruizhi Deng, Bo Li, Fisher Yu, Mingyan Liu, and Dawn Song. Characterizing adversarial examples based on spatial consistency information for semantic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 217–234, 2018.
  - [27] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. *arXiv preprint arXiv:1711.01991*, 2017.
  - [28] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. In *Proceedings of the Network and Distributed System Security Symposium (NDSS'18)*, San Diego, CA, February 2018.
  - [29] Mingjun Yin, Shasha Li, Zikui Cai, Chengyu Song, M. Salman Asif, Amit K. Roy-Chowdhury, and Srikanth V. Krishnamurthy. Exploiting multi-object relationships for detecting adversarial attacks in complex scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7858–7867, October 2021.
  - [30] Yue Zhao, Hong Zhu, Ruigang Liang, Qintao Shen, Shengzhi Zhang, and Kai Chen. Seeing isn't believing: Towards more robust adversarial attack against real world object detectors. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 1989–2004, 2019.