GLOW: Global Layout Aware Attacks on Object Detection

Jun Bao*[1], Buyu Liu*[2], Kui Ren[2], and Jun Yu†[3,4]

[1]The State Key Laboratory of Blockchain and Data Security [2]Zhejiang University
[3]Hangzhou Dianzi University [4]Harbin Institute of Technology (Shenzhen)

{baojun, buyu.liu, kuiren}@zju.edu.cn, yujun@hdu.edu.cn

Abstract

Adversarial attacks aim to perturb images such that a predictor outputs incorrect results. Due to the limited research in structured attacks, imposing consistency checks on natural multi-object scenes is a practical defense against conventional adversarial attacks. More desired attacks should be able to fool defenses with such consistency checks. Therefore, we present the first approach GLOW that copes with various attack requests by generating global layout-aware adversarial attacks, in which both categorical and geometric layout constraints are explicitly established. Specifically, we focus on object detection tasks and given a victim image, GLOW first localizes victim objects according to target labels. And then it generates multiple attack plans, together with their context-consistency scores. GLOW, on the one hand, is capable of handling various types of requests, including single or multiple victim objects, with or without specified victim objects. On the other hand, it produces a consistency score for each attack plan, reflecting the overall contextual consistency that both semantic category and global scene layout are considered. We conduct our experiments on MS COCO and Pascal. Extensive experimental results demonstrate that we can achieve about 30% average relative improvement compared to state-of-the-art methods in conventional single object attack request; Moreover, such superiority is also valid across more generic attack requests, under both white-box and zero-query black-box settings. Finally, we conduct comprehensive human analysis, which not only validates our claim further but also provides strong evidence that our evaluation metrics reflect human reviews well.

1. Introduction

Object detection aims to localize and recognise multiple objects in given images with their 2D bounding boxes and corresponding semantic categories [14, 19]. Due to the physical commonsense and viewpoint preferences [16], detected bounding boxes in natural images are not only semantically labeled but also placed relative to each other within a coherent scene geometry, reflecting the underlying 3D scene structure. Such bounding box representation allows us to derive a notion of both semantic and geometric constraints. For example, co-occurrence matrix is a commonly exploited semantic constraint where certain object categories are more likely to co-occur, e.g., bed and pillow [20]. Geometric constraints, on the other hand, leverage the inductive bias of scene layout [11], such as when oc-occurring in a scene, traffic light is more likely to be appeared on the upper region with a smaller bounding box compared to car.

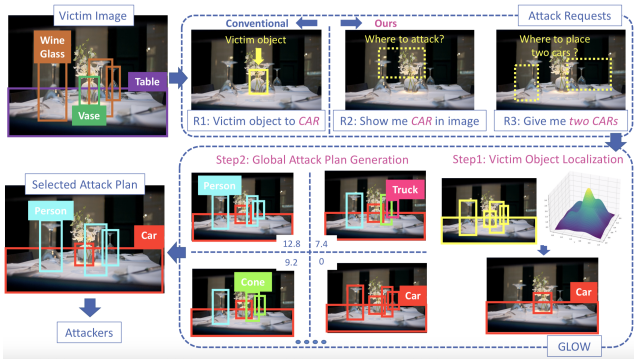

Figure 1. We propose a novel attack generation algorithm GLOW that manages both conventional single targeted object (R1) and our generic attack requests (R2,R3). Specifically, GLOW consists of two steps. The first step localizes victim objects, if not provided. The second step generates various attack plans with their consistency scores. Then the one with the highest score is our final attack plan and parsed to attackers. Best viewed in color.

Adversarial attacks on object detectors mainly focus on targeted victim setting [7, 45] where the goal is to perturb a specific victim object to target class. In this case, the location, ground truth and target class of the victim object are assumed to be known to attackers. Naturally, contextual cues are leveraged in attack and defense mechanisms [6, 7, 48] on detectors to enhance or detect holistic context (in)consistency [6]. Though being well-motivated and demonstrating good performances in conventional setting, the state-of-the-art meth-

*Equal contribution.
†Corresponding author.

ods [6, 7] suffer the following problems in practice. Firstly, the assumption of known location and ground truth label of victim object might be too strong due to annotation cost [2]. Therefore, more vague attack requests where victim objects are not specified, e.g. show me an apple and a chair, should be considered in practice, which are beyond those existing methods. Secondly, global geometric layout is commonly neglected as existing methods either model semantic co-occurrence [6] or consider relative sizes and distance w.r.t. given victim object [7].

In this work, we introduce a novel yet generic attack plan generation algorithm GLOW on both conventional and generic attack requests to effectively leverage both categorical and global layout cues, leading to superior white-box attack performance and better transferability in black-box setting. As for generic requests, we firstly loose the assumption of known specific victim object by requesting only the existence of certain target label, e.g. show me category X in image. Compared to conventional setting, our request demands the modelling of the locations and sizes of target label X. Our second request further constrains label amount, e.g. give me N objects of category X and M objects of category Y, which necessitates the global layout of victim image. To fulfill these requests, we propose a novel attack plan generation method GLOW that accounts for both categorical and geometrical relations. Specifically, GLOW aims to figure out the most context-consistent attack plan for each victim image according to its underlying layout while considering the hard constraints, e.g. existence or amount of some target labels under generic requests or a specific victim object under conventional request. The first step in GLOW localizes victim objects with given target label or amount on victim image by modeling the joint distribution of bounding box sizes and centers. And it enables generic attack requests. Given these victim objects, the second step further leverages the layouts of victim image to generate globally context-consistent attack plans with consistency scores. This is achieved by reformulating the plan generation task as a layout similarity measurement problem. Therefore, those consistency scores are similarity scores. Finally, the plan with the highest score would be our selected attack plan. We then implement the selected plan with existing attack generation methods, or attackers. Details of our proposed requests and GLOW can be found in Fig. 1.

We validate our ideas on coco2017val [32] as well as Pascal [18] with both white-box and zero-query black-box settings. And we design new evaluation metrics as well as introduce human analysis to measure layout consistency thus mimicking consistency defenses. We demonstrate that in white-box setting, our proposed method achieves superior performance with both conventional and proposed generic attack setting compared to SOTAs. More importantly, GLOW provides significantly better transfer success rates on zero-query black-box setting compared to existing methods.

Our contributions can be summarized as follows:

- A novel method GLOW that is capable of generating context consistent attack plans while accounts for both semantic and geometric coherency.
- Two generic attack requests and consistency evaluation metrics to mimic realistic scenarios and delicate defenses.
- State-of-the-art performances on coco2017val and Pascal images under both white-box and zero-query black-box settings. Code, model and requests will be available.

2. Related Work

Object detection The goal of object detection is to predict a set of bounding boxes and category labels for each object of interest. Starting from [14, 19], object detection explored extensive cues, including semantic [27], geometric [46] and other contextual cues [47], to improve its performance as well as interpretability. Recently, deep neural networks (DNNs) [26] have significantly improved many computer vision [21, 26] and natural language processing tasks [15, 23]. Modern detectors follow the neural networks design, such as two-stage models where proposals are firstly generated and then regression and classification are performed [5, 21] and one-stage models [33, 43, 52] that simultaneously predict multiple bounding boxes and class probabilities with the help of pre-defined anchors or object centers. More recently, transformer-based models [8, 53] are proposed to further simplify the detection process by formulating the object detection as a set prediction problem where unique predictions can be achieved by bi-partite matching, rather than non-maximum suppression [4, 24]. Similarly, contextual cues are also explored in modern detectors [1, 3, 12, 35, 51] with various forms. In this paper, we focus on adversarial attacks on DNNs-based detectors. And our GLOW generates contextually coherent attack plans with various requests, which are also transferable to detectors of different architectures.

Adversarial attacks and defenses in object detection Despite impressive performance boosts, DNNs are vulnerable to adversarial attacks, e.g. adding visually imperceptible perturbations to images leads to significant performance drop [9, 22, 42]. Adversarial attacks can be categorized into white-box [22, 36] and black-box [17, 31], depending on whether parameters of victim models are accessible or not. Attacks such as DAG [45], RAP [30] and CAP [50] are architecture-specific white-box attacks on detectors where two-stage architecture is required since they work on proposals generated by the first stage. More generic attacks, such as UAE [44] and TOG [13], are capable of attacking all different kinds of models regardless of their architectures. Compared to the aforementioned methods that perturb the image globally [45], patch-based attacks [34] also showcase their ability in terms of fooling the detectors without touching the victim objects [25]. In contrast, black-box at-

tacks [10, 28, 29] are more practical yet challenging where either a few queries or known surrogate models are exploited to fool an unknown victim model. Observing the impacts of adversarial attacks on detectors, various defense methods are proposed to detect such attacks, wherein contextual cues are explored [48]. However, contextual cues are almost always represented in the form of semantic co-occurrence matrix where global layouts are largely neglected [6, 7]. In contrast, we propose a generic attack plan generation algorithm that leverages both semantic and geometric coherency, e.g. scene layout. Consequently, it manages both conventional single targeted victim setting and generic attack requests where locations are unknown or object amount is further restricted, translating to SOTA performance under white-box and black-box settings.

3. Method

We introduce the attack requests, our proposed GLOW and attackers in Sec. 3.1, Sec. 3.2 and Sec. 3.3 respectively.

3.1. Attack requests

To attack a victim image, user may or may not specify victim objects, e.g. providing their locations or labels. Therefore, besides considering conventional attack request where a specific object and its targeted label are given, more generic requests should also be addressed, such as give me 2 cats or mis-classify the rightmost boat to car. Let's denote $\mathcal{D}$ as the set of victim images. $\mathcal{C} = \{c_p\}_{p=1}^{C}$ is the label space with C semantic categories. Given a known object detector f, which can be the victim model in white-box attack or the surrogate model under black-box setting, we can obtain a set of predicted objects $\mathcal{O}$ on victim image $I \in \mathcal{D}$, or $\mathcal{O} = f(I) = \{l_n, s_n\}_{n=1}^{N}$, consisting the locations and semantic categories of N objects. l_n defines the location of the n-th object, including its bounding box center coordinates, height and width. And $s_n \in \mathcal{C}$ is its semantic label.

R1: mis-classify the object s_n to c_p. This is the conventional attack request where the n-th object is our specific victim object and c_p is the targeted label.

Though one can always choose random object as victim and random category as c_p, we observe that the choices of victim object and target label play an important role in attack performances (see Sec. 4). To this end, we set different selection criteria for victim object and targeted label to evaluate attack methods in various aspects. As for victim object, it is unpractical to assume that ground-truth locations can be provided by the users, e.g. bounding box annotations can be time-consuming [2]. Therefore, we turn to the predictions as reliable sources to help us to determine where the attack should take place. In practice, choosing the one that has the largest bounding box among all predictions with confidence score above 0.85 provides good estimation for GT.

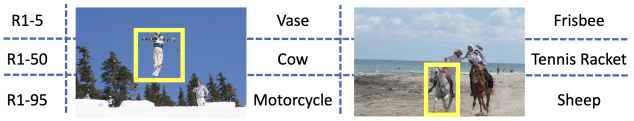

Figure 2. Examples of R1. Victim objects s_n are highlighted with yellow bounding boxes and target labels c_p generated with R1-5, R1-50 and R1-95 are on the right side of each victim image.

As for target label c_p, we mainly follow [6, 7] where the out-of-context attack is considered. Specifically, to eliminate the chance of miscounting the existing objects as success, c_p is selected if and only if c_p is not present in the I. Rather than randomly selecting c_p among all unpresented categories [6, 7], our decision is made according to distance in word vector space [49] as it captures the semantic and syntactic qualities of words. Mathematically, for each unpresented c_p, we define its average distance as:

$$v_d(c_p) = \frac{1}{N}\sum_{n} v(c_p, s_n); \forall c_p \notin \mathcal{S}_n \tag{1}$$

where $\mathcal{S}_n = \{s_n\}_{n=1}^{N}$ and $v(c_p, s_n)$ denotes the cosine distance between category c_p and s_n in word vector space.

To evaluate the impact of target label c_p, we collect three c_ps according to $v_d(c_p)$ and visualize them in Fig. 2. Specifically, we firstly rank all $c_p \notin \mathcal{S}_n$ based on $v_d(c_p)$. Then we choose the top 5%, 50% and 95% ones as our target class c_ps, referring as R1-95, R1-50 and R1-5, respectively.

Our ultimate goal of R1 is not only to mis-classify the victim object, but also to fail the potential defenses w.r.t. consistency checks. Therefore, the challenge of R1 mainly lies in figuring out the attack plan that is contextually consistent and beneficial for the mis-classification in practice.

R2: show me the category c_p. Rather than assuming that a specific victim object is known to attackers as in R1, R2 takes one step further in terms of relaxing the attack request. Specifically, R2 comes in a much vague manner where user only specifies the target label c_p.

Though it seems that asking for the existence of c_p is an easier task compared to R1 as one can always flip a random object to c_p, we argue that this conclusion is valid if only coarse semantic consistency check/defense, e.g. co-occurrence matrix [6], is available, which unfortunately neglects geometric context. A more desired consistency check should be capable of capturing both geometric and semantic context. For example, traffic light is less likely to appear on the image bottom while poles usually have slim bounding boxes. And our goal is to fool the victim model and these delicate defenses simultaneously.

Therefore, we claim that R2 is more challenging than R1 as it requests additional understanding of the location-wise distribution of target label c_p. We kindly note our readers that such challenge is beyond [6, 7] (see Fig. 3). We omit the details of c_p in R2-5, R2-50 and R2-95 as they are selected based on the same criteria as that of R1 (see supplementary).

Figure 3. We visualize one victim image with their R2-5, R2-50 and R2-90 labels. And we highlight the victim object s_n, which is localized by GLOW, with yellow bounding box.

Figure 4. Challenge of R3. We have victim image on the left and four example proposals based on request R3-5 on the right. Among these four proposals, the left most one is more plausible than one in middle considering the layout relations. And the right two are totally wrong as they violate the amount restriction.

R3: give me multiple c_ps. R3 reflects another realistic attack scenario, e.g. have a monitor and a mouse in victim image I. Besides not specifying the victim object by providing only target label information, R3 enforces additional constraint on object amount, making it more challenging. Specifically, multi-object relationship should be considered together with hard restrictions on the amount of objects (see Fig. 4). For example, besides modelling locations of mouse and monitor individually, estimating their layout, e.g. monitor is more likely to be above the mouse, is also essential to achieve context consistent yet fooled predictions.

Theoretically, R3 can be multiple victim objects of the same or different categories, which do not affect our following GLOW method. In practice, when it comes to objects with various categories, additional heuristics are needed to avoid semantic inconsistency as v_d does not guarantee contextual consistent combinations. Moreover, such problem becomes more severe with increasing number of objects, together with the emerge of new challenge of underlying constraints on object amount in natural image. For instance, ten *apples* in I can be natural but not for ten *stop signs*. Therefore, we leave objects of different categories as our future work and focus on two objects of the same c_p. Details of R3-5, R3-50 and R3-95 can be found in supplementary.

3.2. GLOW: Global LayOut aWare attacks

Contextually consistent attack has been discussed in many previous work [6, 7]. The main motivation is that perturbing only the victim object may lead to inconsistency in context thus global attack plan should be considered. Specifically, an attack plan assigns target labels to all objects in victim image, including ones that are not victims originally, to both avoid inconsistency and benefit the attack request. Though well-motivated, existing methods largely rely on semantic context [6], neglecting geometric context such as scene layout. In addition, the ability of modelling prior knowledge, such as having more than ten beds in an image is unlikely to happen while obtaining ten books can be feasible, is lacking.

To this end, we propose a novel attack plan generation method GLOW that accounts for both semantic and geometric context, such as object locations and overall scene layout. GLOW consists of two steps. The first localization step aims to locate victim objects based on target labels and their amounts under generic attack requests R2 and R3. Then the second generation step further produces multiple context-consistent attack plans as well as their scores with given victim objects. Afterwards, the plan with the highest score is selected as our final attack plan and then parsed to existing attackers. See Fig. 5 for more details.

Victim object localization We aim to localize victim objects under R2 and R3, where constraints on target labels and/or their amount are available.

Let's first assume there exist some images from annotated detection dataset, which, in the simplest case, can be the set that our victim/surrogate model is trained on. We denote this dataset as $\mathcal{T}$, including T images and their bounding box annotations $\mathcal{A} = \{\mathcal{A}_t\}_{t=1}^{T}$, where $\mathcal{A}_t = \{l_m^t, s_m^t\}_{m=1}^{M}$ is the set of bounding box annotations on the t-th image I_t. As denoted in $\mathcal{A}_t$, I_t consists of M annotated objects. The m-th object instance is further represented by its location l_m^t and corresponding semantic category $s_m^t \in \mathcal{C}$.

Determining the location of victim object under R2 and R3 is equivalent to estimating the center, height and width of bounding boxes of target label c_p. And we formulate the localization as a probability maximization problem. This is achieved by modelling the joint probability of bounding box center, height and width per category. Specifically, for each $c_p \in \mathcal{C}$, we have $\mathcal{L}_{c_p} = \{l_m^t | s_m^t = c_p\}_{t,m}$, where l_m^t is normalized by image height and width. Then we apply GMM [37] to fit $q = \{1, \dots, Q\}$ Gaussians $\mathcal{N}_q^p(\mu_q^p, \delta_q^p)$ on $\mathcal{L}_{c_p}$, where μ_q^p and δ_q^p are mean and co-variance of q-th Gaussian at class c_p. pdf_q^p and π_q^p are the probability density function and the weight of $\mathcal{N}_q^p$ respectively. Q is set to 5 based on experiment on $\mathcal{T}$. Given any $x \in \mathbb{R}^4$, our GMM is able to provide a weighted probability density $w(x)$ by:

$$w_p(x) = \frac{1}{Q} \sum_q \pi_q^p \times pdf_q^p(x) \tag{2}$$

Simply going through all x and choosing ones with highest $w_p(x)$ ignore overall scene layout, which might result in significant layout changes, e.g. large bounding box on objectless area or heavy occlusions, leading to less plausible overall layouts. Alternatively, we narrow down our search space to existing bounding boxes and find the optimal location among all l_n. As for R2, the victim object can be found by $n^* = \arg\max_n w_p(l_n)$. As for R3, we rank and select top ones depending on detailed request, e.g. choose the top 2 if R3 is to have two objects of same target label c_p.

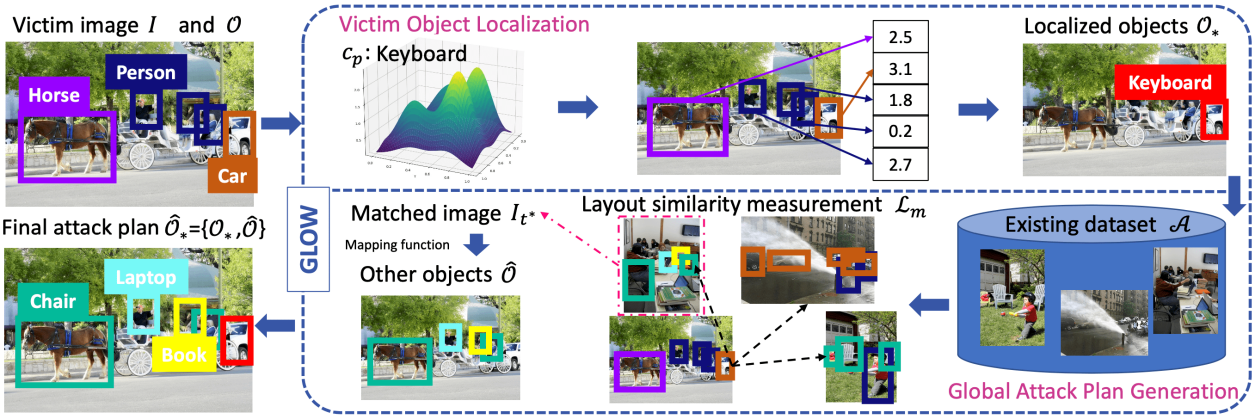

Figure 5. Overview of GLOW. The first step of GLOW aims to locate the victim object $\mathcal{O}_*$ under generic attack requests according to dataset distribution. Afterwards, the GLOW produces various context-consistent attack plans, together with their consistency scores. The plan with highest score is selected as our final attack plan $\hat{\mathcal{O}}_*$.

We then denote the victim objects in I as $\mathcal{O}_* = \{l_p^*, c_p^*\}_*$, where c_p^* equals to c_p in R1 and R2. And $\{c_p^*\}_*$ is the set of requested target labels in R3. Similarly, l_p^* is l_n in R1 and are from estimation in R2 (l_{n^*}) and R3. We further denote the number of target objects as X and $X = 1$ under R1 and R2. Example victim objects $\mathcal{O}_*$ can be found in Fig. 6.

Global attack plan generation Given victim object $\mathcal{O}_*$, our next step is to generate target labels on objects that are not victim. Specifically, it aims to find an mapping function $g(s_n) \in \mathcal{C}$ that perturbs the label of these objects, resulting in $\hat{\mathcal{O}} = \{l_n, g(s_n)\}_n$. The overall generated attack plan on I would be $\hat{\mathcal{O}}_* = \{\mathcal{O}_*, \hat{\mathcal{O}}\}$.

Theoretically, there exist $(N - X)^C$ possible configurations in $\hat{\mathcal{O}}$. Instead of permuting all possible solutions, we restrict ourselves with only feasible ones that occur in existing dataset $\mathcal{T}$ as scene layouts are naturally context-consistent therein. To this end, we formulate our global attack plan generation as a layout similarity measurement problem, with hard constraint on victim objects $\mathcal{O}_*$. Our goal is therefore to map the bounding box labels according to the best match based on layout similarity in $\mathcal{T}$. Intuitively, the more similar these layouts are, the more confident we are in terms of performing mapping. Therefore the layout similarity score reflects context consistency to some extent. Our insights lie in the following design choices of obtaining mapping function g and score s:

- Generate $\mathcal{T}^* = \{\mathcal{T}_{c_p^*}\}_{c_p^*}$ where $\mathcal{T}_{c_p^*}$ consists of images from $\mathcal{T}$ where target label c_p^* is present.
- Compute the Intersection over Union (IoU) score between victim objects $\mathcal{O}_*$ and objects that share the same target labels in $I_t \in \mathcal{T}_{c_p^*}$, Mathematically,:

$$s_1(I_t) = \frac{1}{X} \sum_{l_p^* \in \mathcal{O}_*} s(l_p^*) \tag{3}$$

where $s(l_p^*) = \max_m \mathbb{1}_{\{c_p^* = s_m^t\}} IoU(l_p^*, l_m^t)$. The IoU score between victim object location l_p^* and m-th bounding box in I_t is obtained by $IoU(l_p^*, l_m^t)$.

- Perform Hungarian matching [8, 41] between objects in I_t and those in victim image I. Specifically, we find a bipartite matching between these two sets by searching for a permutation of M elements $\mathbb{S}_M$ with the lowest cost:

$$\begin{aligned} \delta_t^* &= \arg\max_{\delta \in \mathbb{S}_M} \sum_{l_n \notin \mathcal{O}_*} \mathcal{L}_m(l_n, l_{\delta(n)}^t) \\ &= \arg\max_{\delta \in \mathbb{S}_M} \sum_{l_n \notin \mathcal{O}_*} L1(l_n, l_{\delta(n)}^t) + GIoU(l_n, l_{\delta(n)}^t) \end{aligned} \tag{4}$$

where $L1()$ and $GIoU()$ denote the L1 and GIoU [40] scores between bounding boxes. $\delta_t^*(n)$ is the index of the best match of n-th object which is not victim originally in victim image I. And the match loss of $\delta_t^*(n)$ can be obtained with $s2(I_t) = \frac{1}{N-X} \sum_{l_n \notin \mathcal{O}_*} \mathcal{L}_m(l_n, l_{\delta_t^*(n)}^t)$. The temporary mapping function based on the t-th image I_t is then defined as $g_t(s_n) = s_{\delta_t^*(n)}^t$.

The overall similarity score between $I_t \in \mathcal{T}^*$ and I is obtained by $s(I_t) = s_1(I_t) - \lambda s_2(I_t)$, where λ is a hyper-parameter chosen by experiment. We would like to note that score s accounts for not only the victim objects reflecting by s_1, but also the overall layout similarity incorporated in s_2.

Afterwards, we define the I_{t^*} as it 1) gives the highest similarity score and 2) matches more than 95% of objects in I. Consequently, the mapping function $g(s_n)$ then equals to the temporary mapping function of the t^*-th image, or $g_{t^*}(s_n)$. We refer the readers to Fig. 6 for more details.

3.3. Implementation of attack plan

To generate $\hat{\mathcal{O}}_*$, evasion attacks can be implemented using our victim model itself under white-box setting or a single or multiple surrogate model(s) under zero-query black-box setting. In white-box scenario, our implementation of attack plan is based on TOG [13] so that we can perform fair comparisons with existing methods [7](see Sec. 4). Specifically, we fix the weight of victim model f and learns a perturbation

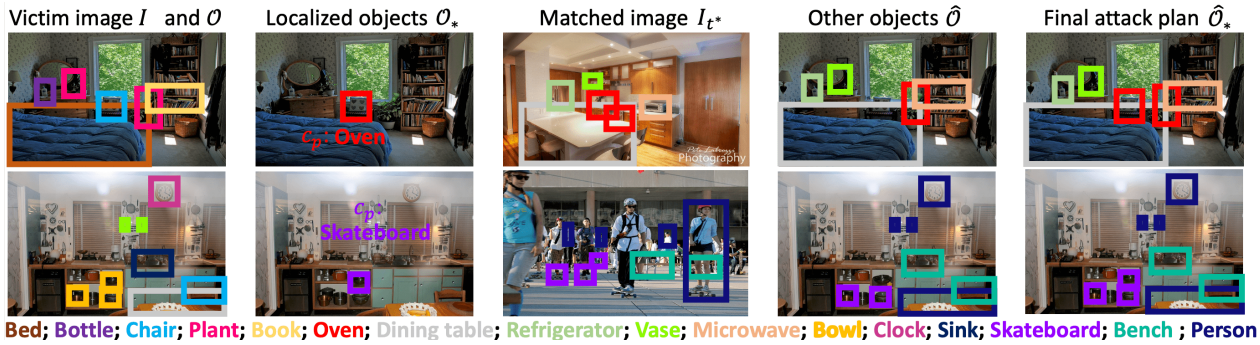

Figure 6. Step-wise illustration of GLOW. From left to right, we have victim image I with their initial prediction results $\mathcal{O}$, target label c_p and the localized victim object $\mathcal{O}_*$, best matching image I_{t^*}, the plan for other objects $\hat{\mathcal{O}}$ and our final attack plan $\hat{\mathcal{O}}_*$.

Victim set $\mathcal{D}$		coco17val (3792/80)	Pascal (500/20)
Victim f	Whi.	$\mathbb{F}$	$\mathbb{F}$+$\mathbb{Y}$
	Blk.	$\mathbb{F} \rightarrow \mathbb{D}$	$\mathbb{F}$+$\mathbb{Y} \rightarrow \mathbb{T}$
Victim model f is trained on and $\mathcal{T}$			coco17train

Table 1. Experimental setup. Our victim model is trained on coco17train only and the victim images are from coco17val and Pascal. The former consists of 3792 images with 80 categories while the latter has 500 random images of 20 categories.

image δ for I by minimizing $\mathbf{L}(clip(I+\delta); \hat{\mathcal{O}}_*)$ at every iteration [22]. $clip()$ is enforced to ensure bounded perturbation. Thereafter, the perturbed image $clip(I+\delta)$ is parsed to another unknown victim model, mimicking the zero-query black-box setting.

4. Experiment

To evaluate GLOW under various requests, we perform extensive experiments on coco2017val [32] and Pascal [18], with both white-box and black-box settings. As can be found in Tab. 1, our victim model f can be Faster-RCNN-R50-FPN-1X-COCO($\mathbb{F}$) [39] and F-RCNN+YOLO($\mathbb{F}$+$\mathbb{Y}$) [38] under white-box setting. These aforementioned victim models are later utilized as the surrogate model in our black-box attacks where DETR($\mathbb{D}$) [8] and RetinaNet($\mathbb{T}$) [33] are our victim models f. Our black-box attack is zero-query based, meaning no feedback from victim model is available. Our GLOW is generally applicable to different victim detectors and we choose the aforementioned models mainly for efficiency and re-productivity purpose [7]. We report our performance under both perturbation budget 10 and 30. Due to the space limitation, we refer the readers to supplementary materials for results with the former and visualized examples. And our claims are valid with different perturbation budgets.

Baselines We compare GLOW with four baselines. To perform fair comparison, attack plan implementations are all obtained with TOG [13] thus we describe only the attack plan generation process in the following:

- TOG [13] The attack plan generated by the TOG is context-agnostic, or $g(s_n) = s_n$. Victim object is given in R1 and will be randomly selected under R2 and R3.
- TOG+RAND. TOG+RAND. focuses on both victim objects and other objects. Victim object is provided in R1 and randomly selected under R2 and R3. Mapping function $g(s_n)$ is a random permutation function.
- TOG+SAME. Attack plan generated by TOG+SAME. includes all objects. And we enforce $g(s_n) = c_p$, meaning all objects share the same target label c_p.
- Cai [7] can be directly apply to R1. As for R2 and R3, Cai [7] firstly selects random objects as victims and then generates the attack plan.

Evaluation Metrics We follow the basic metric from [6] and also introduce others for generic attack requests. Fooling rate (**F**) [6] is used to evaluate the attack performance on victim objects. Specifically, one attack succeeds if (1) victim object is perturbed as target label while IOU is score greater than 0.3 compared to GT and (2) it pass the co-occurrence check. And we define the fooling rate as the percentage of the number of test cases for which the above two conditions are satisfied. Besides, we further introduce **T** to measure the consistency on victim objects. **T** itself reveals the averaged $w_p(l_p^*)$. When combined with other metrics, **T** is satisfied as long as the averaged $w_p(l_p^*)$ is above 0.02 (see Sec. 3.2). To measure the overall layout consistency, we introduce **R** that reflects the percentage of images whose maximum recall rate compared to $\mathcal{A}$ is above 0.5. We further design two metrics, **E** and **C**, on R2 and R3 to report successful rate. **E** checks whether target label c_p exists in predictions. While **C** further verify the amount of c_p. One attack is successful if both target labels and their amount satisfy the request in R3. We refer the readers to supplementary for more details of all metrics and give some visual examples in Fig. 7.

4.1. Main results

Attack performance on R1 We report our main results on conventional attack request R1 in Tab. 2 where perturbation budgets is set to 30. In general, we observe that under white-box setting, our **F** is comparable to existing methods on coco, which is reasonable as this metric accounts for only oc-occurrence matrix and both TOG+SAME and Cai [7] consider such semantic consistency. When considering global

Methods	White-box (coco17val/Pascal)						Zero query black-box (coco17val/Pascal)					
	R1-5		R1-50		R1-95		R1-5		R1-50		R1-95	
	F	F+R	F	F+R	F	F+R	F	F+R	F	F+R	F	F+R
TOG [13]	.64/**.67**	.11/.16	.75/**.77**	.15/.22	.87/**.82**	.20/.27	.08/**.13**	.01/.04	.16/.19	.02/.08	.23/.27	.03/.13
TOG+RAND	.45/.48	.06/.08	.54/.61	.08/.14	.58/.66	.07/.14	.12/.10	.01/.02	.21/.17	**.03**/.04	.27/.26	**.04**/.06
TOG+SAME	**.89**/.52	.18/.13	**.90**/.68	.18/.22	**.91**/.75	.18/.22	**.21**/.10	.01/.04	**.34/.22**	**.03**/.11	**.38**/.31	.03/.11
Cai [7]	.86/.46	.09/.08	.87/.63	.09/.09	.90/.74	.07/.11	.18/.08	.01/.03	.29/.19	.02/.08	.34/.30	.02/.11
GLOW	.85/.61	**.20/.20**	.87/.76	**.22/.28**	.89/.79	**.21/.29**	**.21**/.11	**.02/.05**	.30/**.22**	**.03/.12**	.35/**.33**	**.04/.18**

Table 2. Overall performance of R1. As described in Sec. 3.1, we have three different target labels, R1-5, R1-50 and R1-95, for each victim object and we report the results on all of them. We highlight the best scores in bold.

Figure 7. We visualize failure and successful cases of different evaluation metrics and their key factors under R3, where our goal is to have two elephants in given victim image.

layout **R**, we observe 30% performance improvement over existing methods under all scenarios (R1-5, R1-50, R1-95), reflecting that GLOW is able to fool the victim object with more contextually consistent layout. Noticeably, our observation of 30% averaged improvement is also valid under challenging zero-query black-box setting, which further demonstrates the transferability of our proposed attack plan generation. Please note that results on Pascal are obtained with victim models that trained on coco17train, which further showcase the generality and superiority of GLOW.

There are also other interesting observations in Tab. 2. Firstly, there exists a trend of performance improvement over all methods when comparing R1-5, R1-50 and R1-95, indicating the choice of target label plays an important role in terms of performance. This trend validates our hypothesis that far-away labels, e.g. R1-5, are harder to attack compared to close-by ones, which in return proves the necessity of systematic design on target label rather than random generation. Secondly, though TOG+SAME simply assigns all labels of existing objects to be target label c_p, it gives good performance under **F**. This observation further supports our design of more delicate consistency check metrics, e.g. **R**, as co-occurrence matrix is vulnerable to such simple hacks.

Attack performance on R2 The advantages of GLOW are more noticeable in R2 where victim object is requested to be localized by algorithm itself rather than being provided. There are two main observations based on Tab. 3. Firstly, GLOW almost always beats the SOTAs in terms of all evaluation metrics under R2-5, R2-50 and R2-95 in white-box setting, e.g. about 35% relative improvement compared to the second best in terms of **F+T** and **E+R** under R2-5. This observation is also valid when victim models are trained on coco17train and tested on Pascal. Interestingly, unlike R1 where victim object is fixed among all methods, results of **T** in R2 showcase that the victim object selection matters under generic request. Though neither TOG+SAME nor Cai [7] considers the overall layout consistency, the former gives better score than the latter as it naively enforces all objects to share the same target label and **E** in **E+R** measures only the existence of target label. Please note that **E** and **F** are different. For instance, assuming layout consistency is already satisfied, if the attack on the victim object fails but turns another object into target label, it will be regarded as a success in **E** but a failure in **F**. GLOW, again, produces superior results by leveraging layout explicitly. Our second observation from Tab. 3 is that GLOW has better transfer rates, such as 24% improvement compared the context-aware baseline [7] and various types of random assignment under black-box setting, which further demonstrates the benefits of utilizing global layout in attack plan generation and the potential limitations of exploiting only semantic context. We observe the same trend that the overall performance improves when the target label is closer to presented labels in word space, supporting our design of various target labels.

Attack performance on R3 Results of the most challenging request R3 are provided in Tab. 4. We kindly remind our readers that **C+R** and **F+T+C** reflect different aspects of an algorithm as the former does not care about specific objects but checks both target labels and their the amount. Assuming R3 is to have two apples in victim image and our attacks are contextually consistent, **F+T+C** will be successful if only these two victim objects are perturbed to apple. In contrast, **C+R** reflects the amount of apples in perturbed images and mismatch in numbers would lead to failure. Again, our GLOW is a much safer choice in terms of R3 as it almost always, or about 13 out of all 18 entries, gives the best performance with both white-box and black-box setting.

Human analysis We report the human analysis on Pascal in Tab. 5 and kindly ask readers to check more details in supplementary. In short, humans are asked to perform pairwise comparisons on the attacked results of the same image from

Methods	White-box (coco17val / Pascal)								
	R2-5			R2-50			R2-95		
	T	F+T	E+R	T	F+T	E+R	T	F+T	E+R
TOG [13]	.18 / .18	.31 / .38	.17 / .14	.20 / .20	.41 / .44	.24 / .21	.22 / .20	.49 / .34	.25 / .15
TOG+RAND	.19 / .22	.22 / .29	.04 / .09	.18 / .18	.27 / .35	.06 / .23	.22 / .20	.32 / .34	.06 / .15
TOG+SAME	.18 / .23	.45 / .35	.21 / .22	.20 / .22	.51 / .43	.20 / .27	.23 / .19	.55 / .38	.20 / .26
Cai [7]	.21 / .24	.44 / .30	.11 / .10	.20 / .22	.48 / .38	.11 / .15	.24 / .19	.53 / .37	.09 / .16
GLOW	**.38 / .35**	**.64 / .50**	**.32 / .24**	**.40 / .35**	**.67 / .55**	**.35 / .29**	**.44 / .33**	**.69 / .48**	**.32 / .29**
	Zero query black-box (coco17val / Pascal)								
TOG [13]	.25 / .26	.04 / .08	.01 / .04	.24 / .25	.08 / .15	.02 / .12	.28 / .21	.12 / .12	.03 / .15
TOG+RAND	.20 / .28	.05 / .08	.01 / .03	.20 / .20	.08 / .12	.01 / .05	.29 / .25	.14 / .15	.03 / .07
TOG+SAME	.23 / .30	.12 / .10	.02 / .06	.22 / .25	.20 / **.19**	.03 / **.18**	.27 / .26	.25 / .17	**.05** / .16
Cai [7]	.26 / .37	.10 / .09	.02 / .07	.25 / .26	.15 / .12	.02 / .13	.30 / .21	.20 / .14	.02 / .18
GLOW	**.37 / .38**	**.17 / .14**	**.03 / .10**	**.38 / .35**	**.22** / .18	**.04** / .15	**.44 / .38**	**.29 / .23**	**.05 / .19**

Table 3. Overall performance of R2. Similar to R1, we have three different target labels for victim image. Since the victim object location is not provided in R2, **T**, **F+T** and **E+R** reflects different aspects of layout consistency.

Methods	White-box (coco17val / Pascal)								
	R3-5			R3-50			R3-95		
	T	F+T+C	C+R	T	F+T+C	C+R	T	F+T+C	C+R
TOG [13]	.18 / .21	.35 / **.28**	.10 / .05	.19 / .15	.43 / **.37**	.13 / **.16**	.22 / .19	.50 / .35	**.14 / .17**
TOG+RAND	.18 / **.30**	.27 / .24	.08 / .05	.19 / .20	.33 / .32	.11 / **.16**	.22 / .19	.38 / .31	.12 / .16
TOG+SAME	.18 / .23	.14 / .16	.11 / .06	.20 / .21	.15 / .18	.12 / .14	.22 / .19	.17 / .16	.12 / **.17**
Cai [7]	.20 / .25	.17 / .10	.02 / .02	.21 / .24	.22 / .16	.02 / .06	.23 / .22	.21 / .13	.02 / .04
GLOW	**.32** / .28	**.48** / .27	**.13 / .07**	**.34 / .28**	**.52** / .35	**.15** / .12	**.35 / .27**	**.53 / .37**	**.14** / .11
	Zero query black-box (coco17val / Pascal)								
TOG [13]	.21 / .32	.01 / .01	.00 / .01	.21 / .23	.03 / **.03**	.01 / .03	.28 / .26	**.04 / .04**	**.02 / .05**
TOG+RAND	.22 / .31	.01 / .01	.00 / .01	.21 / .21	.02 / **.03**	.01 / .03	.28 / .25	**.04 / .04**	.01 / .04
TOG+SAME	.21 / .32	.01 / .01	.00 / .01	.22 / .23	.02 / .01	.01 / **.04**	.27 / .25	.03 / .03	**.02 / .05**
Cai [7]	.25 / .35	.01 / .01	.00 / .01	.25 / .31	.01 / .00	.01 / .01	.30 / .27	.02 / .02	.00 / .03
GLOW	**.31 / .38**	**.02 / .02**	**.01 / .02**	**.34 / .38**	**.04** / .02	.01 / .02	**.37 / .36**	**.04 / .04**	.01 / .03

Table 4. Overall performance of R3. Compared to R2, our **C+R** accounts for both layout consistency and amount restriction.

Methods	TOG [13]	[13]+RAND	[13]+SAME	Cai [7]	GLOW
TOG [13]	-	.71	.59	.30	.24
[13]+RAND	.43	-	.25	.18	.14
[13]+SAME	.52	.77	-	.35	.18
Cai [7]	.69	.79	.61	-	.42
GLOW	.78	.89	.80	.70	-

Table 5. Human analysis on Pascal under R2-5. We colored results under white and black settings in red and blue respectively. For instance, the bottom-left 0.78 means that 78% of GLOW results are voted to be better than TOG [13] by humans.

two methods. We observe that our method outperforms all methods significantly in Tab. 5, proving that it gives superior layout-consistent attacks. Such observation is also consistent with our evaluation metrics, indicating that these metrics provide reasonable estimations of layout consistency.

5. Conclusion

In this paper, we propose a novel attack generation algorithm GLOW for adversarial attacks on detectors. Compared to existing work, it explicitly takes both semantic context and geometric layout into consideration. By validating on two datasets, we demonstrate that GLOW produces superior performances under both conventional attack request and more generic ones where victim objects are obtained by estimation. In addition, we would like to highlight that GLOW demonstrates better transfer rates under challenging zero-query black-box setting.

6. Acknowledgement

This work was supported in part by the National Natural Science Foundation of China (No. 62125201, 62020106007).

References

[1] Ehud Barnea and Ohad Ben-Shahar. Exploring the bounds of the utility of context for object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 7412–7420, 2019. 2

[2] Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. What's the point: Semantic segmentation with point supervision. In European conference on computer vision, pages 549–565. Springer, 2016. 2, 3

[3] Sean Bell, C Lawrence Zitnick, Kavita Bala, and Ross Girshick. Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2874–2883, 2016. 2

[4] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-nms–improving object detection with one line of code. In Proceedings of the IEEE international conference on computer vision, pages 5561–5569, 2017. 2

[5] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: high quality object detection and instance segmentation. IEEE transactions on pattern analysis and machine intelligence, 43 (5):1483–1498, 2019. 2

[6] Zikui Cai, Shantanu Rane, Alejandro E Brito, Chengyu Song, Srikanth V Krishnamurthy, Amit K Roy-Chowdhury, and M Salman Asif. Zero-query transfer attacks on context-aware object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 15024–15034, 2022. 1, 2, 3, 4, 6

[7] Zikui Cai, Xinxin Xie, Shasha Li, Mingjun Yin, Chengyu Song, Srikanth V Krishnamurthy, Amit K Roy-Chowdhury, and M Salman Asif. Context-aware transfer attacks for object detection. In Proceedings of the AAAI Conference on Artificial Intelligence, pages 149–157, 2022. 1, 2, 3, 4, 5, 6, 7, 8

[8] 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. 2, 5, 6

[9] Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In Proceedings of the 10th ACM workshop on artificial intelligence and security, pages 3–14, 2017. 2

[10] Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. Adversarial attacks and defences: A survey. arXiv preprint arXiv:1810.00069, 2018. 3

[11] Xinlei Chen and Abhinav Gupta. Spatial memory for context reasoning in object detection. In Proceedings of the IEEE international conference on computer vision, pages 4086–4096, 2017. 1

[12] Zhe Chen, Shaoli Huang, and Dacheng Tao. Context refinement for object detection. In Proceedings of the European conference on computer vision (ECCV), pages 71–86, 2018. 2

[13] Ka-Ho Chow, Ling Liu, Margaret Loper, Juhyun Bae, Mehmet Emre Gursoy, Stacey Truex, Wenqi Wei, and Yanzhao Wu. Adversarial objectness gradient attacks in real-time object detection systems. In 2020 Second IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA), pages 263–272. IEEE, 2020. 2, 5, 6, 7, 8

[14] 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), pages 886–893. Ieee, 2005. 1, 2

[15] Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. Fast and robust neural network joint models for statistical machine translation. In proceedings of the 52nd annual meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1370–1380, 2014. 2

[16] Santosh K Divvala, Derek Hoiem, James H Hays, Alexei A Efros, and Martial Hebert. An empirical study of context in object detection. In 2009 IEEE Conference on computer vision and Pattern Recognition, pages 1271–1278. IEEE, 2009. 1

[17] 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

[18] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. International Journal of Computer Vision, 111(1):98–136, 2015. 2, 6

[19] Pedro Felzenszwalb, David McAllester, and Deva Ramanan. A discriminatively trained, multiscale, deformable part model. In 2008 IEEE conference on computer vision and pattern recognition, pages 1–8. Ieee, 2008. 1, 2

[20] Carolina Galleguillos, Andrew Rabinovich, and Serge Belongie. Object categorization using co-occurrence, location and appearance. In 2008 IEEE Conference on Computer Vision and Pattern Recognition, pages 1–8. IEEE, 2008. 1

[21] Ross Girshick. Fast r-cnn. In Proceedings of the IEEE international conference on computer vision, pages 1440–1448, 2015. 2

[22] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572, 2014. 2, 6

[23] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal processing magazine, 29(6):82–97, 2012. 2

[24] Jan Hosang, Rodrigo Benenson, and Bernt Schiele. Learning non-maximum suppression. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 4507–4515, 2017. 2

[25] Shengnan Hu, Yang Zhang, Sumit Laha, Ankit Sharma, and Hassan Foroosh. Cca: Exploring the possibility of contextual camouflage attack on object detection. In 2020 25th

International Conference on Pattern Recognition (ICPR), pages 7647–7654. IEEE, 2021. 2

[26] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. Communications of the ACM, 60(6):84–90, 2017. 2

[27] L'ubor Ladickỳ, Paul Sturgess, Karteek Alahari, Chris Russell, and Philip HS Torr. What, where and how many? combining object detectors and crfs. In European conference on computer vision, pages 424–437. Springer, 2010. 2

[28] Maosen Li, Cheng Deng, Tengjiao Li, Junchi Yan, Xinbo Gao, and Heng Huang. Towards transferable targeted attack. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 641–649, 2020. 3

[29] Qizhang Li, Yiwen Guo, and Hao Chen. Practical no-box adversarial attacks against dnns. Advances in Neural Information Processing Systems, 33:12849–12860, 2020. 3

[30] 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. 2

[31] Jiadong Lin, Chuanbiao Song, Kun He, Liwei Wang, and John E Hopcroft. Nesterov accelerated gradient and scale invariance for adversarial attacks. arXiv preprint arXiv:1908.06281, 2019. 2

[32] 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. 2, 6

[33] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In Proceedings of the IEEE international conference on computer vision, pages 2980–2988, 2017. 2, 6

[34] Xin Liu, Huanrui Yang, Ziwei Liu, Linghao Song, Hai Li, and Yiran Chen. Dpatch: An adversarial patch attack on object detectors. arXiv preprint arXiv:1806.02299, 2018. 2

[35] Yong Liu, Ruiping Wang, Shiguang Shan, and Xilin Chen. Structure inference net: Object detection using scene-level context and instance-level relationships. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 6985–6994, 2018. 2

[36] 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

[37] Carl Rasmussen. The infinite gaussian mixture model. Advances in neural information processing systems, 12, 1999. 4

[38] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 779–788, 2016. 6

[39] 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. 6

[40] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 658–666, 2019. 5

[41] Russell Stewart, Mykhaylo Andriluka, and Andrew Y Ng. End-to-end people detection in crowded scenes. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2325–2333, 2016. 5

[42] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199, 2013. 2

[43] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In Proceedings of the IEEE/CVF international conference on computer vision, pages 9627–9636, 2019. 2

[44] Xingxing Wei, Siyuan Liang, Ning Chen, and Xiaochun Cao. Transferable adversarial attacks for image and video object detection. arXiv preprint arXiv:1811.12641, 2018. 2

[45] 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, 2

[46] Yi Yang, Sam Hallman, Deva Ramanan, and Charless Fowlkes. Layered object detection for multi-class segmentation. In 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pages 3113–3120. IEEE, 2010. 2

[47] Jian Yao, Sanja Fidler, and Raquel Urtasun. Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In 2012 IEEE conference on computer vision and pattern recognition, pages 702–709. IEEE, 2012. 2

[48] 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, pages 7858–7867, 2021. 1, 3

[49] Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. Deep modular co-attention networks for visual question answering. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 6281–6290, 2019. 3

[50] Hantao Zhang, Wengang Zhou, and Houqiang Li. Contextual adversarial attacks for object detection. In 2020 IEEE International Conference on Multimedia and Expo (ICME), pages 1–6. IEEE, 2020. 2

[51] Ji Zhang, Mohamed Elhoseiny, Scott Cohen, Walter Chang, and Ahmed Elgammal. Relationship proposal networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 5678–5686, 2017. 2

[52] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. arXiv preprint arXiv:1904.07850, 2019. 2

[53] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159, 2020. 2