Supplementary materials: 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

Our supplementary material includes more details on requests, baselines as well as our evaluation metrics in Sec. 1 and Sec. 2. Qualitative and more quantitative results of all methods can be found in Sec. 3 and Sec. 4 Finally, we report the limitations of our proposed GLOW in Sec. 5

1. More details about generic request

R2 Compared to R1, R2 takes one step further in terms of relaxing the attack request. Specifically, R2 comes in a more vague manner where users only specifies the target label c_p. In terms of selecting c_p, R2 shares the same intuition with R1. In short, we first find out the out-of-context $c_p \notin \mathcal{S}_n^d$. Then we compute the averaged distance of each c_p to existing s_n^d according to Eq.1 in our main paper, followed by ranking each c_p w.r.t. $v_d(c_p)$. Finally, the ones with top 95%, 50% and 5% distance are selected as the target label c_p R2-5, R2-50 and R2-95, respectively.

R3 As described in our main paper, R3 adds additional restrictions on the object amount, e.g. show me two cars. Without losing generalization, we design a special type of R3 where the request is always giving me *two* c_p. For instance, give me two cars if car is our target label. Similarly, out-of-context is explored to figure out c_p in R3-5, R3-50 and R3-95. In particular, the target labels of R3-5, R3-50 and R3-95 are the same as R2-5, R2-50 and R2-95, respectively.

2. Baselines and evaluation metrics

Baselines In this section, we include one more baseline, or TOG [3]+SAME Similar to TOG+RAND., attack plan generated by TOG+SAME assign target labels to all objects. The only difference lies in the mapping function $g(s_n^d)$. Here we enforce $g(s_n^d) = c_p$, meaning all objects share the same target label c_p.

We visualize the attack plan of different methods under R2-5 in Fig. 2. As can be found in this figure, TOG cares only about the randomly selected victim object, which may lead to context inconsistency in predictions. Though TOG+RAND. is aware of this limitation and proposes to assign labels to all objects, including the ones that are not victim objects. It potentially suffers from the oc-occurrence inconsistency as well since the target labels of these objects are randomly selected. Both Zikui [2] and TOG+SAME are able to address such inconsistency problems. Specifically, TOG+SAME provides an ad-hoc way by enforcing all objects belonging to the same target label c_p. While Zikui [2] turns to a co-occurrence matrix when looking for target labels of objects that are not originally our victim.

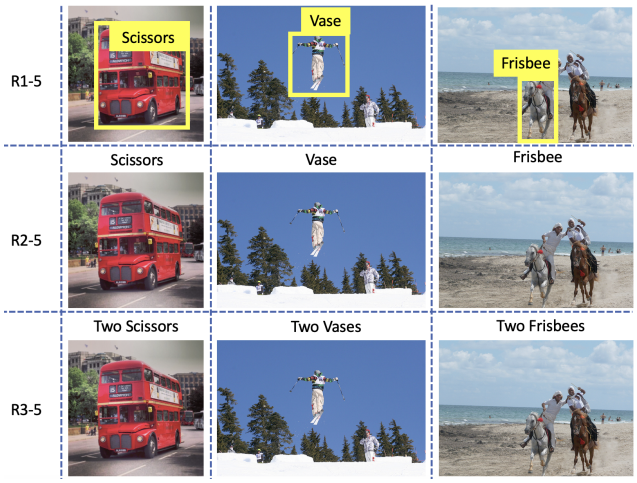

Figure 1. We visualize the R1-5, R2-5 and R3-5 of the same victim image and they share the same target label c_p in our design. While R1 assumes that the victim object (highlighted with a yellow bounding box in each figure) is given, R2 relaxes the constraint by only providing a target label. In contrast, R3 further restricts the amount of target objects.

Evaluation metrics We describe all criteria we have used and then introduce our evaluation metrics based on them. Note all the following evaluations are performed on predictions of perturbed victim image.

*Equal contribution.
†Corresponding author.

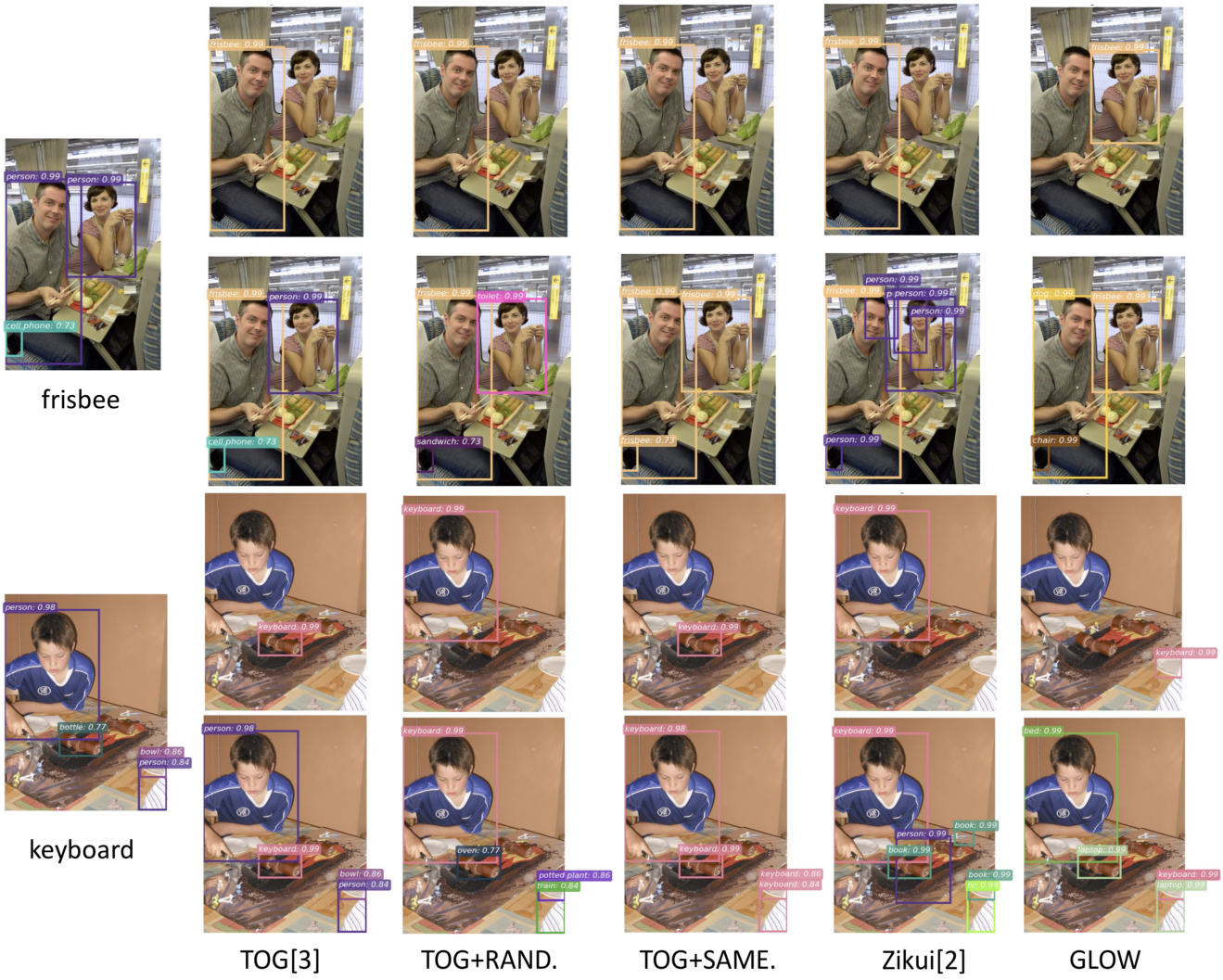

Figure 2. We visualize two examples of R2-5. From left to right, we have victim image I_d image with the $\mathcal{O}_d$ as well as the target class c_p, victim object (top) and attack plan (bottom) with TOG [3], victim object (top) and attack plan (bottom) with TOG [3]+RAND., victim object (top) and attack plan (bottom) with TOG [3]+SAME, victim object (top) and attack plan (bottom) with Zikui [2], victim object (top) and attack plan (bottom) with GLOW.

- a) Victim object is perturbed as target label while IOU scores greater than 0.3 compared to GT.
- b) Predictions pass the co-occurrence check. For instance, if at least the labels of two predicted objects never co-occur in co-occurrence matrix, this image is context-inconsistent.
- c) Averaged weighted w_p on victim object. If combined with other criteria, an attack is successful iff the averaged w_p on victim objects is above 0.02.
- d) Overall layout recall reflects the percentage of images whose maximum recall rate is above 0.5. Specifically, for each prediction on the perturbed image, we compare it with all annotated images $I_t \in \mathcal{A}$. Then we find the best match that has the maximum recall rate. For example, when comparing predictions on perturbed I_d and $\mathcal{O}_t$, an object is regarded as recalled as long as its label is agreed with the matched object in $\mathcal{O}_t$ and the IoU score between these two objects is above 0.5. After obtaining the recall rate on each I_t, we find the one with the highest recall rate. If the recall rate is above 0.5, we believe this image is layout-consistent.
- e) If we have the target label c_p exists in predictions, this victim image is successfully attacked.
- f) One attack is successful if both c_p exist and their amount satisfies the request.

As for fooling rate (**F**) [1], criteria a) and b) are combined. **T** to measure the consistency on victim objects and we utilize c) as our criterion when combining with others. To measure the overall layout consistency, we introduce **R** that is technically obtained with d). We further design two metrics, **E** and **C**, on R2 and R3 to report successful rates.

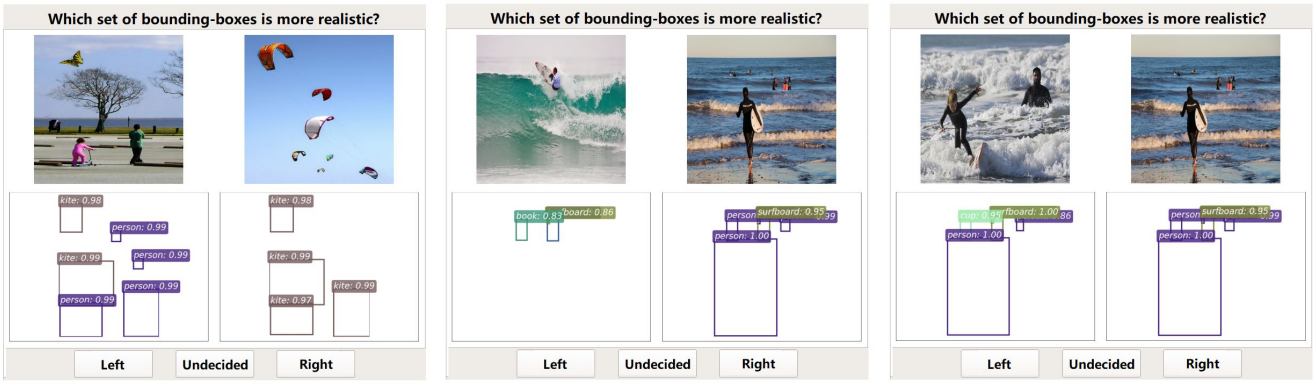

Figure 3. Three examples of our user interface. Specifically, we provide both the attacked results, which include the bounding boxes, as well as their closest pair in terms of layout similarity in COCO. Users will decide which attack result is more realistic based on their common senses as well as the reference image. They are also allowed, though not encouraged, to choose "undecided" if the results are equally bad or good. Please note that which methods these two results are from are anonymous to users.

And they are based on e) and f) respectively.

Human analysis As described in our main paper, we perform human analysis on the attacked results of all methods, which provides a more reliable evaluation of how consistent-wise realistic each attacked method is.

Specifically, we ask humans to perform a pair-wise comparison between two attacked results. Humans will decide whether one result is more realistic according to context, or they are equally well. We design a user interface and give an example in Fig. 3. As a reference, the users believe that results are equally good on the left-most example. While the results on the right are better than those on the left for the middle and right-most examples. As can be found in this figure, we provide both the attacked results, which only include the bounding boxes, as well as their closest pair in terms of layout similarity in COCO. After iterating out all baselines, we report the percentage of cases where one method is believed to be superior to the other when they co-occur. Our human analysis includes the annotation results from three persons, each of them with a different background. Results on Pascal can be found in Tab. 1 and Tab. 2. We also conduct the same analysis on COCO and demonstrate our results in Tab. 3, 4 and 5.

Experiment details In experiment, we set Q to 5. For each victim image, we set the iteration number to 50 regardless of the perturbation budgets. And the $clip()$ is used to truncate the accumulated per-pixel perturbation if it is greater than the pre-defined perturbation budget. The weight λ is set to 0.5 empirically.

3. Qualitatively results

We visualize the victim object, the attack plan, and the final predictions of all methods on one example victim image in Fig. 4. As can be found in this figure, victim object selection or localization is very important, where different selections would lead to various results. TOG, TOG+RAND., and Zikui fail the task as NO refrigerator occurs in the final prediction. Though TOG+SAME can turn multiple objects into refrigerator in the final prediction, it fails the victim object it selects. Our GLOW, in contrast, is able to both fool the model's prediction on localized victim objects, as well as generate layout-consistent final predictions.

Methods	TOG [3]	[3]+RAND	[3]+SAME	Cai [2]	GLOW
TOG [3]	-	.57	.52	.31	.22
[3]+RAND	.39	-	.21	.15	.06
[3]+SAME	.52	.75	-	.37	.11
Cai [2]	.64	.81	.63	-	.37
GLOW	.76	.92	.78	.57	-

Table 1. Human analysis on Pascal under R1-5. We colored results under white and black settings in red and blue respectively. The bottom-left 0.76 means that 76% of GLOW results are voted to be better than TOG [3] by humans.

Methods	TOG [3]	[3]+RAND	[3]+SAME	Cai [2]	GLOW
TOG [3]	-	.63	.60	.31	.32
[3]+RAND	.39	-	.41	.14	.18
[3]+SAME	.56	.72	-	.15	.24
Cai [2]	.61	.74	.56	-	.50
GLOW	.76	.86	.68	.69	-

Table 2. Human analysis on Pascal under R3-5. We colored results under white and black settings in red and blue respectively.

4. More quantitative results

4.1. coco17val

We report the attack results on coco17val with perturbation budget set to 10 in Tab. 6, 7 and 8. We highlight the best

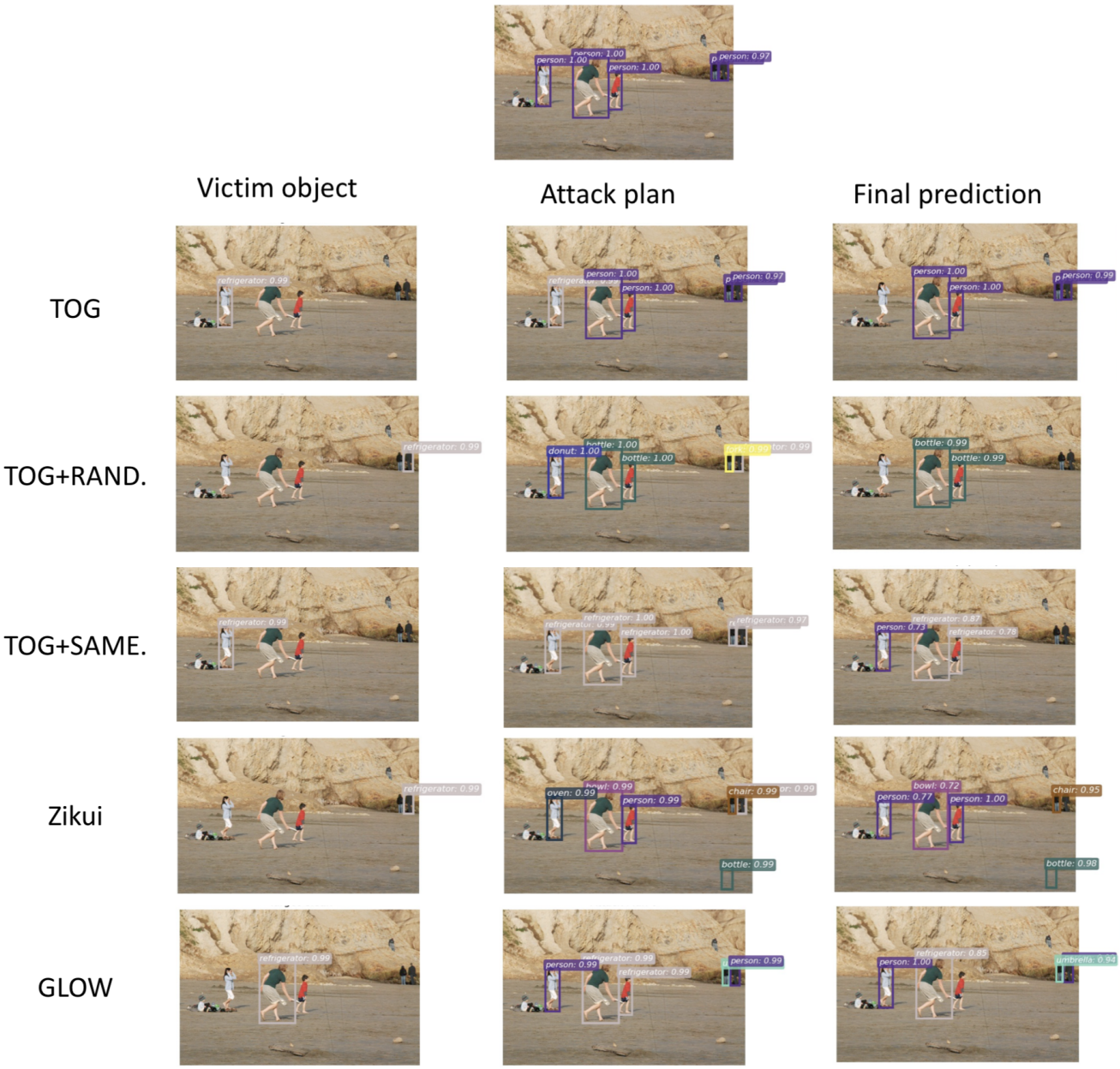

Figure 4. We visualize the victim object, the attack plan, and final predictions of all methods on one example victim image. We are asked to fool the model so that we have a refrigerator in our final prediction. Unlike other methods that select the victim by random, our GLOW localizes the victim object w.r.t. the target label, as well as accounts for the center and size. Furthermore, GLOW is further able to generate layout-consistent attack plans, leading to better attacks in general.

and second-best performances in bold and underline, respectively.

Compared to perturbation budget 30, one most noticeable observation we have is that the overall performance is much worse, which is clearly reflected in **F** where the victim object cannot be easily fooled with only a limited budget in white-box setting. Failing to fool victim object in the white-box setting leads to even worse performance under the black-box setting.

Similarly, we observe the same trend as in perturbation 30 where simpler task, e.g. R2-95, gives better performance compared to harder task such as R2-5. This, again, supports our experimental design of different target label c_p under the same request.

Interestingly, we notice that GLOW can always give either the best or second-best results compared to all baselines in

Methods	TOG [3]	[3]+RAND	[3]+SAME	Cai [2]	GLOW
TOG [3]	-	.71	.70	.58	.52
[3]+RAND	.32	-	.40	.30	.27
[3]+SAME	.68	.97	-	.42	.28
Cai [2]	.81	.98	.73	-	.37
GLOW	.91	.98	.80	.63	-

Table 3. Human analysis on COCO under R1-5. We colored results under white and black settings in red and blue respectively.

Methods	TOG [3]	[3]+RAND	[3]+SAME	Cai [2]	GLOW
TOG [3]	-	.86	.79	.68	.53
[3]+RAND	.18	-	.33	.23	.11
[3]+SAME	.66	.97	-	.51	.17
Cai [2]	.83	.98	.73	-	.29
GLOW	.91	.98	.83	.76	-

Table 4. Human analysis on COCO under R2-5. We colored results under white and black settings in red and blue respectively.

Methods	TOG [3]	[3]+RAND	[3]+SAME	Cai [2]	GLOW
TOG [3]	-	.79	.76	.72	.58
[3]+RAND	.23	-	.30	.35	.22
[3]+SAME	.79	.97	-	.44	.29
Cai [2]	.83	.94	.71	-	.32
GLOW	.93	.99	.81	.78	-

Table 5. Human analysis on COCO under R3-5. We colored results under white and black settings in red and blue respectively.

all evaluation metrics. TOG+SAME is a strong counterpart under a small budget. This is expected as a small budget cannot fulfill attack plans generated with all methods and TOG+SAME hacks the goal with simple ad-hocs. We would like to argue that TOG+SAME is not generic as requests cannot be always of the same target label c_p. And it would be de-generate to TOG+RAND. or Zikui [2] if multiple various c_ps are given. Therefore, our claim that GLOW is more generic and a better choice under both white-box and black-box settings is still valid.

4.2. Pascal

We also report more results on Pascal in Tab. 9, 10 and 11. Please note that in our main paper, the white-box models are Faster-RCNN+YOLO and we turn to RetinaNet as our black-box victim model. While in our supplementary, we exploit both Faster-RCNN as our white-box victim models. And DETR is our victim model when working on black-box settings.

4.3. Dataset distribution

We also showcase the data distribution on two victim datasets in Fig. 5. As can be found in this figure, the data distribution is more balanced in coco17val while Pascal has about 38% of victim images with only two objects. We argue that this can be one reason that our GLOW does not demonstrate

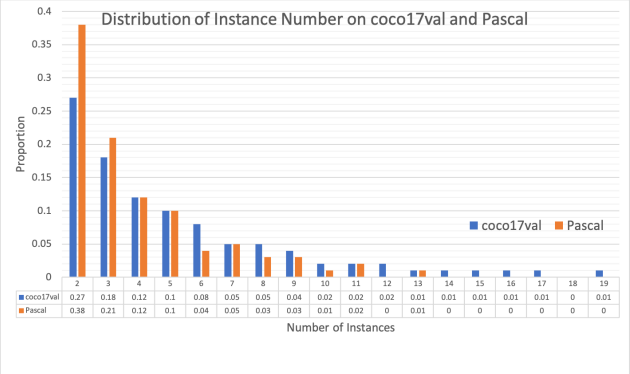

Figure 5. We visualize the data distribution of coco17val and Pascal in this figure. As can be found in this figure, about 38% of Pascal victim images have 2 objects while the number of instances distribution is more balanced on coco17val in comparison.

strong performance improvement over existing methods on Pascal compared to coco17val under R3. More specifically, R3 requests to attack two object instances in victim images where 38% of Pascal images share the same attack plan for all methods, leading to a more challenging scenario where the superiority of GLOW can be demonstrated with the remaining images on Pascal. Nevertheless, we still observe that GLOW is the safest choice under R3 on Pascal such that it outperforms existing methods with majority evaluation metrics.

5. Limitations

More complex requests, such as turning the furthest brown chair into lamb, require a high-level understanding of both the scene and language. They are beyond the scope of GLOW and will be discussed in future work.

Meanwhile, as discussed in our main paper, there is room for improvement in the design of R3. Theoretically, GLOW works with scenarios where objects are of the same or different categories. In experiments, we simulate the target category selection process with automatic generation, where word vector is utilized to measure similarities between semantic labels. When it comes to instances with multiple categories, additional heuristics are needed to avoid semantic inconsistency, e.g. toilet and elephant, since none of these criteria, including visual, contextual, or word similarity, would guarantee contextual consistent combinations. Moreover, delicate designs are requested for evaluation metrics and similarity measurement with the increasing number of categories. Currently, our R3 provides the first trivial towards multiple-object attacks. And we leave the principle design and evaluation on more challenging requests with various while consistent categories for future work.

Methods	White-box (Faster-RCNN)						Zero query black-box (Faster-RCNN → DETR)					
	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 [3]	.43	.07	.54	.11	.68	.17	.04	.00	.07	.01	.12	.01
TOG+RAND	.37	.10	.47	.14	.55	.15	.04	.00	.09	.01	.15	**.02**
TOG+SAME	**.58**	**.18**	**.66**	.19	**.72**	.19	**.09**	.00	**.15**	.01	**.20**	**.02**
Zikui [2]	.55	.13	.63	.14	.68	.13	.07	.00	.13	.01	.17	.01
GLOW	.54	.17	.61	**.21**	.67	**.21**	.07	.00	.12	.01	.17	.01

Table 6. Performance of R1 under perturbation budget 10 on coco17val

Methods	White-box (Faster-RCNN)								
	R2-5			R2-50			R2-95		
	T	F+T	E+R	T	F+T	E+R	T	F+T	E+R
TOG [3]	.17	.19	.09	.19	.27	.17	.22	.35	.19
TOG+RAND	.17	.16	.07	.17	.21	.09	.21	.27	.11
TOG+SAME	.18	.26	.24	.19	.33	.26	.22	.40	.25
Zikui [2]	.21	.25	.17	.22	.31	.19	.24	.37	.18
GLOW	**.32**	**.35**	**.26**	**.36**	**.41**	**.30**	**.41**	**.47**	**.30**
	Zero query black-box (Faster-RCNN → DETR)								
TOG [3]	.22	.02	.00	.28	.04	.01	.30	.06	.01
TOG+RAND	.19	.02	**.01**	.22	.04	.01	.30	.07	.02
TOG+SAME	.23	**.04**	**.01**	.23	**.08**	**.02**	.28	**.13**	**.03**
Zikui [2]	.28	**.04**	.00	.29	.06	.01	.31	.11	.02
GLOW	**.32**	**.04**	**.01**	**.37**	**.08**	.01	**.44**	**.13**	.02

Table 7. Performance of R2 under perturbation budget 10 on coco17val

Methods	White-box (Faster-RCNN)								
	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 [3]	.17	.16	.05	.19	.23	.08	.22	**.28**	**.09**
TOG+RAND	.17	.14	.05	.18	.20	.08	.22	.24	.10
TOG+SAME	.18	.08	**.08**	.19	.10	**.10**	.22	.12	.10
Zikui [2]	.21	.08	.03	.21	.13	.03	.24	.14	.04
GLOW	**.29**	**.21**	.07	**.31**	**.26**	**.10**	**.33**	**.28**	**.09**
	Zero query black-box (Faster-RCNN → DETR)								
TOG [3]	.24	.00	.00	.22	**.01**	.00	.29	**.02**	**.01**
TOG+RAND	.22	.00	.00	.23	**.01**	.00	.28	**.02**	**.01**
TOG+SAME	.23	.00	.00	.24	**.01**	.00	.29	.01	**.01**
Zikui [2]	**.28**	.00	.00	.26	.00	.00	.31	.01	.00
GLOW	**.28**	.00	.00	**.32**	**.01**	.00	**.36**	.01	.00

Table 8. Performance of R3 under perturbation budget 10 on coco17val

Methods	White-box (Faster-RCNN)						Zero query black-box (Faster-RCNN → DETR)					
	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 [3]	.93	.32	.95	.39	.96	.48	.29	**.02**	.34	.03	.42	.07
TOG+RAND	.71	.14	.75	.16	.76	.13	.26	.01	.35	.03	.39	.06
TOG+SAME	**.95**	.21	**.98**	.35	.97	.35	**.35**	.01	**.47**	.03	**.49**	.05
Zikui [2]	.92	.17	.97	.16	**.98**	.15	.34	.01	.43	.02	.47	.04
GLOW	.92	**.46**	.96	**.54**	.96	**.56**	.34	**.02**	.39	**.04**	.45	**.08**

Table 9. Performance of R1 with victim model F-RCNN on Pascal

Methods	White-box (Faster-RCNN)								
	R2-5			R2-50			R2-95		
	T	F+T	E+R	T	F+T	E+R	T	F+T	E+R
TOG [3]	.16	.54	.30	.19	.53	.37	.18	.45	.46
TOG+RAND	.19	.36	.06	.20	.42	.12	.17	.34	.12
TOG+SAME	.20	.56	.22	.20	.57	.35	.17	.45	.36
Zikui [2]	.17	.55	.17	.19	.53	.19	.16	.43	.18
GLOW	**.36**	**.71**	**.49**	**.35**	**.70**	**.53**	**.33**	**.56**	**.50**
	Zero query black-box (Faster-RCNN → DETR)								
TOG [3]	.24	.13	.02	.22	.19	.05	.20	.13	.06
TOG+RAND	.22	.17	.02	.18	.17	.03	.25	.19	.05
TOG+SAME	.23	.26	**.04**	.24	**.30**	**.06**	.22	.24	**.09**
Zikui [2]	.25	.22	.02	.22	.23	.02	.22	.22	.03
GLOW	**.40**	**.31**	.03	**.36**	.26	.05	**.39**	**.30**	**.09**

Table 10. Performance of R2 with victim model F-RCNN on Pascal

Methods	White-box (Faster-RCNN)								
	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 [3]	.20	**.56**	.11	.19	.54	**.33**	.17	.43	**.33**
TOG+RAND	.19	.46	.08	.19	.45	.27	.17	.35	.26
TOG+SAME	.19	.23	.08	.19	.25	.25	.17	.17	.25
Zikui [2]	.19	.31	.02	.21	.23	.06	.18	.15	.05
GLOW	**.31**	.51	**.17**	**.27**	**.55**	.28	**.27**	**.49**	.25
	Zero query black-box (Faster-RCNN → DETR)								
TOG [3]	.20	.03	**.01**	.22	**.06**	.02	.20	**.06**	**.05**
TOG+RAND	.26	.03	**.01**	.22	.05	.02	.23	**.06**	.04
TOG+SAME	.24	.02	**.01**	.23	.03	**.03**	.22	.05	**.05**
Zikui [2]	.24	.00	.00	.25	.02	.01	.21	.03	.02
GLOW	**.33**	**.05**	**.01**	**.31**	.04	.02	**.32**	.05	.02

Table 11. Performance of R3 with victim model F-RCNN on Pascal

References

[1] 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. 2

[2] 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, 5, 6

[3] 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. 1, 2, 3, 5, 6