Gradient-Reweighted Adversarial Camouflage for Physical Object Detection Evasion

Supplementary Material

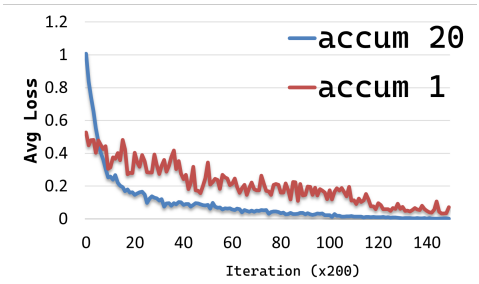

Figure 8. Illustration of training loss w/ and w/o gradient reweighting.

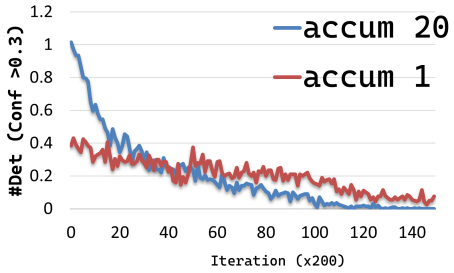

Figure 9. Illustration of number of detection of target w/ and w/o gradient reweighting.

A. Gradient Reweighted Optimization

In this section, we demonstrate how gradient reweighting enhances the adversarial camouflage optimization process. As shown in Figure 8, with gradient reweighting, the loss decreases more smoothly, as indicated by the "accum 20" line. In contrast, greedily updating the texture at each step—where only the adversarial effect from a specific perspective is considered—can result in convergence oscillations. Additionally, gradient reweighting leads to a lower overall loss compared to greedy updating in the end.

However, we also observe that greedy updates can reduce the loss more rapidly in the initial stages. This occurs because our method updates the texture after certain accumulations, resulting in fewer updates; the smaller changes to the texture in the early iterations make the optimization less effective initially. Nevertheless, our method surpasses greedy updating after a sufficient number of iterations. In Figure 9, we illustrate this observation by showing the number of detections for the target perspective, confirming the consistency of our findings.

B. Algorithm for Gradient Reweighted Camouflage Generation

We present the detailed algorithm for our Gradient Reweighted Camouflage Generation method in Algorithm 1.

Algorithm 1 Gradient Reweighted Camouflage Generation

Require: Initial texture $\mathbf{T}$, optimization steps N_s, number of batches N_b, batch size q, camera transforms $\mathbf{C}$, ground truth labels $\mathbf{Y}$

Ensure: Optimized texture $\hat{\mathbf{T}}$

Initialize: $\hat{\mathbf{T}} \leftarrow \mathbf{T}$, $\mathbf{W} \leftarrow \mathbf{1}$, $S_T \leftarrow \emptyset$
for batch $i = 1$ **to** N_b **do** ▷ Iterate over each batch
  ▷ Collect per-view gradients in the current batch
  for sample $j = 1$ **to** q **do**
    $(c, y) \leftarrow (\mathbf{C}_i[j], \mathbf{Y}_i[j])$
    $\mathbf{G}_j \leftarrow \nabla_{\mathbf{T}} \mathcal{L}\left(\mathbb{F}_\theta(\mathbf{I}(\mathbf{T}, c)), y\right)$
      ▷ Compute gradient of loss w.r.t. texture
    $S_T \leftarrow S_T \cup \{\mathbf{G}_j\}$
  end for
  $\mathbb{G}(\mathbf{W}) \leftarrow \sum_{j=1}^{q} \mathbf{W}_j \odot \mathbf{G}_j$ ▷ Weighted gradient
  for step $k = 1$ **to** N_s **do** ▷ Optimize weights
    for sample $j = 1$ **to** q **do**
      $(c, y) \leftarrow (\mathbf{C}_i[j], \mathbf{Y}_i[j])$
      $\mathcal{L}_{\mathbf{W}}^j \leftarrow \mathcal{L}(\mathbb{F}_\theta(\mathbf{I}(\mathbf{T}+\mathbb{G}(\mathbf{W}), c)), y) + \|\mathbf{W}\|_2$
        ▷ Compute loss w.r.t. weights
    end for
    $\mathbf{W} \leftarrow \mathbf{W} - \nabla_{\mathbf{W}} \sum_j \mathcal{L}_{\mathbf{W}}^j$
  end for
  $\hat{\mathbf{T}} \leftarrow \hat{\mathbf{T}} + \mathbb{G}(\mathbf{W})$
    ▷ Update texture with weighted gradient
  Reset: $\mathbf{W} \leftarrow \mathbf{1}$, $S_T \leftarrow \emptyset$, $\mathbf{T} \leftarrow \hat{\mathbf{T}}$
end for
return $\hat{\mathbf{T}}$

C. Examples under different environment conditions

We present additional visualizations of test examples under different weather conditions (in Figure 10), camera distances (in Figure 12), and real-world conditions (in Figure 11).

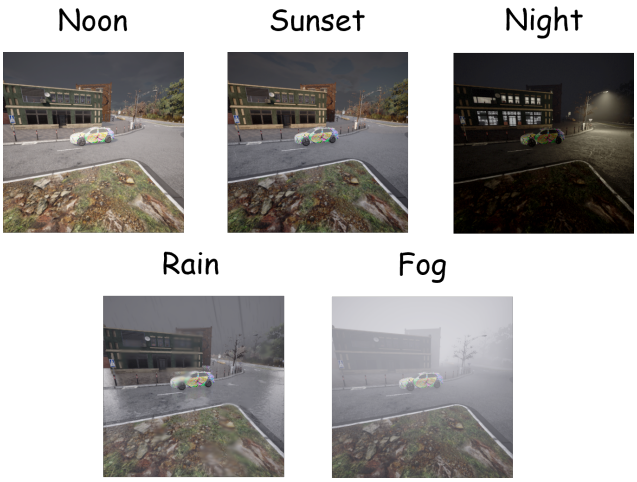

Figure 10. Examples of adversarial camouflage under different weather conditions.

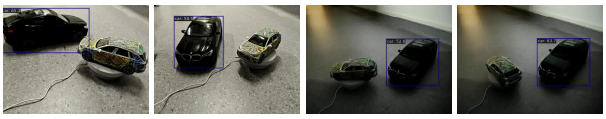

Figure 11. Visualization of detection in real world environment.

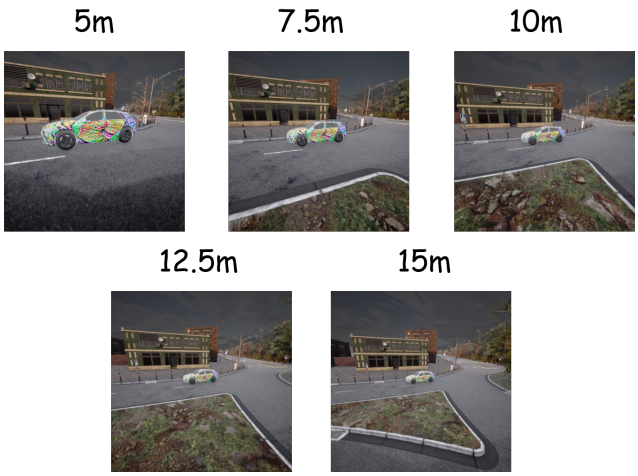

Figure 12. Examples of adversarial camouflage at various distances.

D. Comparison Using Different Environmental Simulation

We evaluate the effectiveness of environmental simulation in degrading detector performance, using RAUCA's NRP module [46] as a baseline. As shown in Table 8, both NRP and our method significantly reduce detection accuracy (AP@0.5), demonstrating their ability to simulate challenging environmental conditions. NRP achieves a stronger attack effect, yielding greater performance drops on both YOLOv3 and Faster R-CNN. However, it incurs higher computational cost, limiting its use in practical settings. In contrast, our method offers a slightly weaker attack performance but a more favorable efficiency-attack trade-off.

Table 8. Comparison of different environmental simulation.

Models	w\o Sim.	w\ NRP	w\ Ours
YOLOv3	6.08	2.76	2.94
FrRCN	20.44	16.57	17.35

E. Evaluation with Complementary Recall Metric

In addition to AP@0.5, we also report Recall@0.5 (R@0.5) scores to further evaluate the effectiveness of our attack under various illumination conditions. Table 9 presents the R@0.5 results on the target car class. As shown, our attack method achieves significantly lower recall compared to RAUCA [46], demonstrating a more severe degradation in detection performance.

Table 9. Evaluation under varying illumination conditions, values are R@0.5 (%) of the target car.

Methods	**Illumination Setting**					***Avg***
	Noon	Sunset	Night	Fog	Rain	
Normal	91.23	92.52	91.23	78.58	93.77	89.47
RAUCA [46]	23.83	12.32	24.24	10.44	22.14	18.59
Ours	**13.50**	**3.56**	**4.10**	**4.82**	**3.84**	**5.96**