Supplementary Materials of Unified Adversarial Patch for Cross-modal Attacks in the Physical World

Xingxing Wei[1,2]*, Yao Huang[1], Yitong Sun[1], Jie Yu[3]
[1] Institute of Artificial Intelligence, Beihang University, Beijing 100191, China
[2] Hangzhou Innovation Institute, Beihang University, Hangzhou 311228, China
[3] School of Computer Science and Engineering, Beihang University, Beijing 100191, China
{xxwei, y_huang, yt_sun, sy2106137}@buaa.edu.cn

1. Discuss Physical Attacks and Physically Realizable Attacks

Abelfattah *et al*. [1] and Tu *et al*. [5] both propose a cross-modal and physically realizable attack, which could work both on images and point clouds. Among their work, an adversarial 3D object is placed on the top of a car in a 3D scene and then rendered to both point clouds and the corresponding RGB images by differentiable renderers. The shape and texture of the object are trainable parameters that are manipulated adversarially.

However, the above works only stress "physically realizable", which are not truly implemented in the real world. Instead, their proposed methods are tested in a simulated 3D environment. They want to prove their methods' physical realizability by adversarial 3D objects' consistency across modalities. But we can easily find that such adversarial 3D objects are complex to produce and it is unrealistic to keep placing a huge 3D object over the vehicle. Therefore, "physically realizable" usually represents a theoretical feasibility in the real world, not an actual application.

Different from such physically realizable attacks, we want to emphasize that our work is **the first cross-modal physical attack** that actually operates on the target in the physical world and achieves a success rather than just ensuring "physically realizable". In addition, we focus on visible-infrared cross-modal attacks, while Abelfattah *et al*. [1] and Tu *et al*. [5] focus on visible-LiDAR cross-modal attacks. This is another key difference between us and them.

2. Centripetal Catmull-Rom Spline Function

In the part of shape representation, to naturally connect anchor points, we choose a method of centripetal catmull-rom spline Interpolation $CCRS(\cdot)$. Here, we will give a specific description of it.

To generate a curve segment C_i between P_i and P_{i+1}, we use four points $P_{i-1}, P_i, P_{i+1}, P_{i+2}$ and knot sequences $t_{i-1}, t_i, t_{i+1}, t_{i+2}$. Firstly, we define P_i and t_i:

$$P_i = [x_i, y_i]^T \tag{1}$$

$$t_i = [\sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}]^\alpha + t_{i-1} (i \geq 1) \tag{2}$$

where $[x_i, y_i]$ are the coordinates of P_i, α ranges from 0 to 1 for knot parameterization, and $t_0 = 0$. For the value of $\alpha = 0.5$, C_i will be a centripetal Catmull-Rom spline.

The curve segment C_i can then be obtained using the spline equations A_1, A_2, A_3, B_1, B_2, which are generated recursively by $P_{i-1}, P_i, P_{i+1}, P_{i+2}$ and $t_{i-1}, t_i, t_{i+1}, t_{i+2}$:

$$C_i = \{C_i(t) | t_i \leq t \leq t_{i+1}\} \tag{3}$$

$$C_i(t) = \frac{t_{i+1} - t}{t_{i+1} - t_i} B_1 + \frac{t - t_i}{t_{i+1} - t_i} B_2 \tag{4}$$

$$B_1 = \frac{t_{i+1} - t}{t_{i+1} - t_{i-1}} A_1 + \frac{t - t_{i-1}}{t_{i+1} - t_{i-1}} A_2 \tag{5}$$

$$B_2 = \frac{t_{i+2} - t}{t_{i+2} - t_i} A_2 + \frac{t - t_i}{t_{i+2} - t_i} A_3 \tag{6}$$

$$A_1 = \frac{t_i - t}{t_i - t_{i-1}} P_{i-1} + \frac{t - t_{i-1}}{t_i - t_{i-1}} P_i \tag{7}$$

$$A_2 = \frac{t_{i+1} - t}{t_{i+1} - t_i} P_i + \frac{t - t_i}{t_{i+1} - t_i} P_{i+1} \tag{8}$$

$$A_3 = \frac{t_{i+2} - t}{t_{i+2} - t_{i+1}} P_{i+1} + \frac{t - t_{i+1}}{t_{i+2} - t_{i+1}} P_{i+2} \tag{9}$$

when $t = t_i, C_i(t) = P_i$ and $t = t_{i+1}, C_i(t) = P_{i+1}$.

Combining the above formulas, we can summarize the process of generating the curve segment C_i into $C_i = CCRS(P_{i_1}, P_i, P_{i+1}, P_{i+2})$.

*Corresponding author

3. Details about the Differential Evolution Algorithm in Our Work.

3.1. Process and Formula

As mentioned in the main manuscript, the Differential Evolution consists of four main parts: starting from an initial population which is a group of randomly generated solutions in the search space, using the crossover and mutation to generate the offspring in the boundary, making the fittest survive according to the fitness function, and finally finding the appropriate solution in the iterative evolution process.

In our case, a population represents a set of patch shapes. Given the population size P, the k-th generation solutions $S(k)$ is represented as

$$S(k) := \{S_i(k) | \theta_j^L \leq S_{ij}(k) \leq \theta_j^U, 1 \leq i \leq P, 1 \leq j \leq n\} \tag{10}$$

where $S_i(k)$ is the i-th patch's shape, and $S_{ij}(k)$ represents the j-th anchor point of $S_i(k)$ in the k-th generation. θ_j^L and θ_j^U together make up the feasible region B_j, which is the moving range of the j-th anchor point in each patch shape.

Then we use crossover and mutation between random individuals and inbreeding of superior individuals to generate candidate populations $Cr(k)$. What's more, since the essence of an evolutionary algorithm is a process of exploration and utilization, we hope that it can explore more freely in the early stage, and can evolve around the current optimal solution when it evolves to a certain extent. Therefore, we divide the $Cr(k)$'s generation into two stages and the process can be formed as follows:

$$Cr_i(x) = clip(S_\gamma(k) + \beta(S_{\gamma_1}(k) - S_{\gamma_2}(k))) \tag{11}$$

where $Cr_i(k)$ is the i-th individual in the k-th candidate population. γ_1, γ_2 are random numbers picked from $\{1, \cdots, n\}$. and $\gamma_1 \neq \gamma_2$. β is the differential weight and $clip(\cdot)$ is a clipping operation to keep individuals within the range. γ is the index number of the individual in $S(k)$ but will change in two stages. We can formulate γ as follows:

$$\gamma = \begin{cases} \gamma_3, & \mathcal{J}(S(k)) < \epsilon \\ \gamma^*, & \mathcal{J}(S(k)) \geq \epsilon \end{cases} \tag{12}$$

where γ_3 is a random number picked from $\{1, \cdots, n\}$, γ^* denotes the index number of the best individual in $S(k)$ and $\gamma^* \neq \gamma_1 \neq \gamma_2 \neq \gamma_3$. ϵ is a threshold.

In the next step, we choose better individuals from $S(k)$ and $Cr(k)$ based on the score-aware iterative function $\mathcal{J}(\cdot)$to develop the next generation $S(K+1)$.

Finally, the whole algorithm will stop when the attack using the optimal individual in the current population as the coordinates of anchor points is successful or when the maximum number of iterations T is reached.

3.2. Hyperparameters

Considering both the attack performance and the time cost, we set the number of the initial population as 30, the epochs of evolution as 200, and the differential weight β as 0.6. All hyperparameters are verified on the validation set.

4. Additional Experiments

4.1. Performances versus Different Detectors

We discuss the attack performances of one-stage detector: YOLOv3 and two-stage detector: Faster RCNN in Section 4.1.1 of the main manuscript. Here, we will show the effect of our unified adversarial patches on some other classical detection models like YOLOv5, YOLOv7[6], SSD[3], and EfficientDet [4] in Table 1. It has to be noted that the effect of our method on the SSD detector gets decreased for its competitive robustness, but from the analysis of the specific results, we also find that if given more epochs, our method can achieve better results.

Table 1. Attack performances in different detection models.

	YOLOv5	YOLOv7	SSD	EfficientDet
ASR	71.67%	80.00%	37.50%	67.50%
AP drop (Visible)	82.11%	89.66%	26.02%	76.48%
AP drop (Infrared)	87.18%	93.84%	26.02%	84.77%

4.2. Effects of Various Shapes

As mentioned in Sec 4.1.2 of the main manuscript, we provide the ablation study to investigate the outcomes of our optimized shapes. In the main manuscript, we only show the average effect of all these basic shapes, but to further demonstrate the effect of each basic shape, we provide results of all basic shapes in Table 2.

4.3. Effects of Patch Sizes

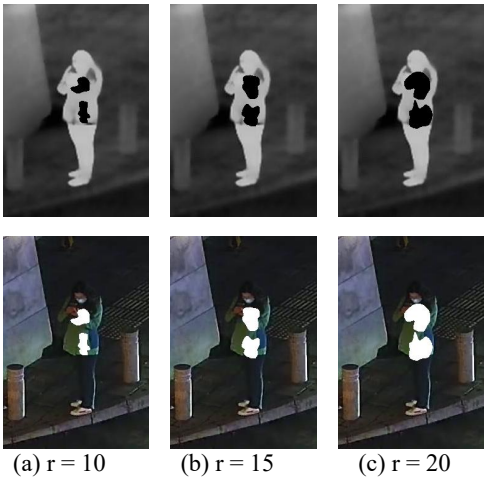
In our method, the radius r of the initial circle affects the size of the patch to some extent. To explore the impact of patch sizes on the attack performance, we set 10, 15 and 20 for r. The corresponding quantitative results are listed in Table 3, where we can see that though the ASR goes up as the patch size increases, our method can still achieve a competitive result with a small patch area, like an ASR of 66.67% with the radius $r = 10$. Additionally, we measure the percentage of patches in the pedestrian area i.e. "Occlusion Rate". The higher the occlusion rate, the easier it is to cover non-effective areas, such as the human head region, as seen in Figure 1 (c). Therefore, to get a trade-off between the attack performance and feasibility, we finally choose a radius of $r = 15$. Some examples of different patch sizes are shown in Figure 1.

Table 2. The ASR(%) and AP drop(%) of all basic shapes.

	Circle	Square	Rectangle(1:2)	Rectangle(2:1)	Triangle
ASR	15.00%	2.50%	0.00%	3.33%	5.83%
AP drop (Visible)	26.83%	22.76%	17.07%	31.71%	26.02%
AP drop (Infrared)	42.28%	4.07%	0.81%	8.94%	13.82%

Table 3. The ASR(%) and AP drop(%) with different patch sizes.

	10	15	20
ASR	66.67%	73.33%	80.83%
Occlusion Rate	5.74%	8.31%	11.09%



(a) r = 10 (b) r = 15 (c) r = 20

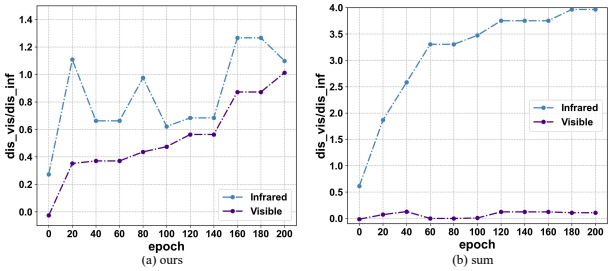
Figure 1. Visual examples of unified adversarial patches with different patch sizes.

4.4. Effects of Score-aware Iterative Evaluation

In Section 4.1.3 of the main manuscript, by comparing the score-aware iterative fitness function with a simple sum, we verify our method's effectiveness in balancing differences between modalities. Here, to further demonstrate the effectiveness of our method, we visualize their specific optimization processes. As shown in Figure 2, with dis_{vis} and dis_{inf} representing the current progress towards the success of attack in the corresponding modality (Eq.(14)-Eq.(15) in the paper, the larger, the closer to success), our method balances the differences between the modalities and achieves simultaneous progress for both modalities, whereas a simple sum tends to focus on a single easy-to-attack modality, such as the infrared modality in Figure 2 (b).

4.5. Comparisons with Other Shape Methods

Wei *et al*. [7] propose a shape optimization of utilizing nine-square-grid shapes to attack infrared detectors (called as hotcold block). Because Wei *et al*. [7] is also a black-box attack, we can easily combine Wei *et al*. [7]'s shape modeling manner with our score-aware iterative function to conduct the comparison. We ensure to use the same patch number and patch size for these two methods. From Table 4, we can see that the hotcold block only has an ASR of 30.83% compared with our unified adversarial patches' 73.33%, which supports our belief that the search space of such nine-square-grid shapes is greatly limited. Moreover, as Figure 3 shown, the hotcold block's optimized positions may not be easy for patches to fix at, causing the instability of physical implementation.



(a) ours (b) sum

Figure 2. Visualization of specific optimization processes in score-aware iterative function and a simple sum function. dis-vis/dis-inf in y-axis denotes the value of dis_{vis} and dis_{inf}, respectively.

(a) Unified Adversarial Patch (b) HoldCold Block

Figure 3. A visual example about unified adversarial patches and hotcold blocks.

Table 4. The Comparison between ours and hotcold block

	Ours	Hotcold block[7]
ASR	**73.33%**	30.83%
AP drop (Visible)	**99.19%**	51.54%
AP drop (Infrared)	**74.31%**	45.53%

We also compare our method with deformable shape [2]. Because deformable shape is optimized using the gradients under a white-box setting, we don't directly compare with the attack performance. Instead, we give a visualization comparison for the optimized shapes between [2] and ours in Figure 4, where we can see that the deformable shape is heteromorphic, and is not easy to implement in the physical world. In contrast, our shape is more natural, and thus is easy-to-implement to attach on the pedestrian to achieve an effective physical attack. Besides, [2] aims to attack image classifiers in the visible domain, while ours aim to attack

object detectors in the visible and infrared domain.

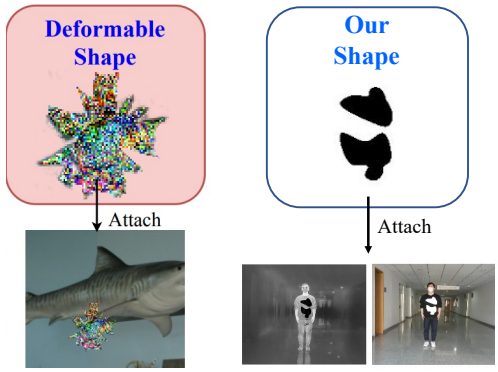


Figure 4. A visualization comparison for the optimized shape between [2] and ours.

References

[1] Mazen Abdelfattah, Kaiwen Yuan, Z Jane Wang, and Rabab Ward. Towards universal physical attacks on cascaded camera-lidar 3d object detection models. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 3592–3596. IEEE, 2021. 1

[2] Zhaoyu Chen, Bo Li, Shuang Wu, Jianghe Xu, Shouhong Ding, and Wenqiang Zhang. Shape matters: deformable patch attack. In *European Conference on Computer Vision*, pages 529–548. Springer, 2022. 3, 4

[3] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 21–37. Springer, 2016. 2

[4] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790, 2020. 2

[5] James Tu, Huichen Li, Xinchen Yan, Mengye Ren, Yun Chen, Ming Liang, Eilyan Bitar, Ersin Yumer, and Raquel Urtasun. Exploring adversarial robustness of multi-sensor perception systems in self driving. *arXiv preprint arXiv:2101.06784*, 2021. 1

[6] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, 2022. 2

[7] Hui Wei, Zhixiang Wang, Xuemei Jia, Yinqiang Zheng, Hao Tang, Shin'ichi Satoh, and Zheng Wang. Hotcold block: Fooling thermal infrared detectors with a novel wearable design. *arXiv preprint arXiv:2212.05709*, 2022. 3