Supplementary Material: Efficient Decision-based Black-box Patch Attacks on Video Recognition

1. Overview

In this supplementary material, we provide more details of STDE, organized as follows:

In Section 2, we provide further details about the population initialization algorithm.

In Section 3, we further analyze all hyper-parameters in STDE.

In Section 4, we provide more implementation details of TPA [8], Patch-Rs [3], AdvW [5], and BSCA [2].

In Section 5, we show more visualizations of our method compared with other SOTA methods for untargeted attacks and targeted attacks respectively.

In Section 6, we provide more analysis about the complexity of time and space between STDE and BSCA.

2. Population Initialization Algorithm

We provide the pseudo-code for population initialization in Algorithm 1, where $\mathrm{rand}(a, b, c)$ means randomly sampling c integers in $[a, b)$, and c defaults to 1. Population initialization aims to generate N populations that correspond to generated adversarial examples can mislead the video recognition model. For definitions of specific formulas and symbols, please see Section 3.3 of the main body.

Algorithm 1 Population Initialization (Population_init)

Input: video recognition model $F(\cdot)$, clean video x, ground truth y, target video x_{tar}, target video label y_{tar}
Parameter: population size N, initialization rate μ, frame coverage rate cf
Output: V, G, q

$V = \emptyset, G = \infty, q = 0$.
for $i \in [1, N]$ **do**
  $cnt = 0, h_l = H \times \mu, w_l = W \times \mu$.
  while True **do**
    $P = \emptyset, M = 0$.
    $FK = \mathrm{rand}(0, 1, T),\ s.t.\ \mathrm{sum}(FK)/T = cf$.
    for $t \in [0, T)$ **do**
      $p^{(0)} = \mathrm{rand}(0, h_l), p^{(1)} = \mathrm{rand}(0, w_l)$.
      $p^{(2)} = \mathrm{rand}(H - h_l, H)$.
      $p^{(3)} = \mathrm{rand}(W - w_l, W)$.
      $P = P \cup p$.
      $M^{(t)}(p^{(0)} : p^{(2)}, p^{(1)} : p^{(3)}) = FK^{(t)}$.
    end for
    Get x_{adv} using Eq. 3 with x, x_{tar}, M.
    Calculate $g(x_{adv})$ using Eq. 4.
    $cnt = cnt + 1$.
    if $g(x_{adv}) < G_i$ **then**
      $G_i = g(x_{adv}), V = V \cup (P, FK)$.
      break
    end if
    if $cnt > 10$ **then**
      $h_l = w_l = 1$.
    end if
    if $cnt > 11$ **then**
      $cf = 1$.
    end if
  end while
  $q = q + cnt$.
end for
return V, G, q

3. Hyper-parameters Tuning

We randomly select one video from each category on the UCF-101 test set, which does not overlap with the videos in Table 1 of the main body. We select C3D as the threat model.

3.1. Population Size N

Table 1 shows the effect of different population sizes N. As the population size N gets larger, the FR remains at 100% with smaller AOA but larger AQN. To balance performance and efficiency, we choose $N = 15$.

3.2. Initialization Rate μ and Frame Coverage Rate cf

The μ parameter constrains the area of patch initialization on each frame in the spatial domain, and cf constrains the distribution of patches across the video in the temporal domain. According to Algorithm 1, the larger μ is and the smaller cf is, the smaller the area of the initialized generated populations corresponding to the generated adversarial

N	Untargeted Attack				Targeted Attack			
	FR(%)	AOA(%)	AOA*(%)	AQN	FR(%)	AOA(%)	AOA*(%)	AQN
5	**100**	4.86	2.07	**2650**	**100**	22.00	8.99	**2640**
10	**100**	4.47	1.87	2660	**100**	21.10	8.70	2770
15	**100**	4.23	1.73	2740	**100**	20.80	8.51	2700
20	**100**	4.14	1.71	2730	**100**	20.20	8.22	3870
30	**100**	**4.06**	**1.69**	2990	**100**	**19.80**	**8.06**	4850

Table 1. Hyper-parameters tuning on population size N.

example patches, which means that the starting optimization point is closer to the global optimum. However, this also leads a larger query consumption for population initialization. Table 2 and Table 3 show the performance results for different values of μ and cf respectively. Considering the query efficiency and attack performance, for untargeted attacks, we choose $\mu = 0.4, cf = 0.6$, for targeted attacks, we choose $\mu = 0.4, cf = 0.7$.

μ	Untargeted Attack				Targeted Attack			
	FR(%)	AOA(%)	A0A*(%)	AQN	FR(%)	AOA(%)	AOA*(%)	AQN
0.05	**100**	5.33	2.16	2660	**100**	21.90	8.86	2810
0.10	**100**	4.88	2.04	**2560**	**100**	21.40	8.74	2810
0.15	**100**	4.43	1.87	2640	**100**	21.80	8.80	2730
0.20	**100**	4.52	1.84	2790	**100**	21.70	8.80	2840
0.25	**100**	3.91	1.63	2780	**100**	**20.70**	8.46	2780
0.30	**100**	4.51	1.88	2690	**100**	20.80	8.47	2780
0.35	**100**	4.23	1.73	2740	**100**	20.80	8.51	**2700**
0.40	**100**	**3.89**	**1.59**	2750	**100**	**20.70**	**8.42**	2730

Table 2. Hyper-parameters tuning on initialization rate μ

cf	Untargeted Attack				Targeted Attack			
	FR(%)	AOA(%)	AOA(%)*	AQN	FR(%)	AOA(%)	AOA*(%)	AQN
Random	**100**	4.28	1.84	2810	**100**	21.70	8.77	2760
0.4	**100**	4.08	1.67	2730	**100**	21.40	8.75	2750
0.5	**100**	3.93	1.61	2750	**100**	21.20	8.60	2800
0.6	**100**	**3.81**	**1.55**	2740	**100**	21.20	8.60	2710
0.7	**100**	3.89	1.59	2750	**100**	**20.70**	8.42	**2730**
0.8	**100**	4.01	1.68	**2700**	**100**	21.10	8.53	2770
0.9	**100**	4.14	1.68	2760	**100**	**20.70**	**8.40**	2840
1.0	**100**	3.91	1.59	2758	**100**	21.70	8.77	2790

Table 3. Hyper-parameters tuning on frame coverage rate cf. *Random* denotes no constrains for FK.

3.3. Mutation Rate γ and Crossover Rate α

Mutation rate γ and crossover rate α increase the diversity of populations in sparsity and temporal domains respectively. Table 4 shows the effect of different mutation rates γ on attack performance. Table 5 shows the effect of different crossover rates α on attack performance. For the untargeted attack, we both choose the smallest values of γ and α: $\gamma = 1, \alpha = 1$, which means more complex variants can inhibit performance for the untargeted attack. For the targeted attack, we choose $\gamma = 1, \alpha = 2$, which balances attack performance and query budgets.

3.4. I_t module and Lambda λ

The video model makes the final action recognition based on the intra-frame spatial semantic information and

γ	Untargeted Attack				Targeted Attack			
	FR(%)	AOA(%)	AOA*(%)	AQN	FR(%)	AOA(%)	AOA*(%)	AQN
1	**100**	**3.89**	**1.59**	2750	**100**	**20.70**	**8.42**	**2730**
2	**100**	5.01	2.20	**2690**	**100**	24.50	9.920	2380
3	**100**	5.09	2.17	3140	**100**	24.30	10.20	2960
4	**100**	5.83	2.67	2830	**100**	26.20	11.00	**2730**

Table 4. Hyper-parameters tuning on mutation rate γ.

α	Untargeted Attack				Targeted Attack			
	FR(%)	AOA(%)	AOA*(%)	AQN	FR(%)	AOA(%)	AOA*(%)	AQN
1	**100**	**3.58**	**1.47**	**2720**	**100**	22.40	9.03	2790
2	**100**	3.89	1.59	2750	**100**	20.70	**8.42**	**2730**
3	**100**	4.35	1.78	2730	**100**	**20.60**	8.44	2950
4	**100**	4.40	1.77	2740	**100**	21.70	8.86	2750

Table 5. Hyper-parameters tuning on crossover rate α.

frame-to-frame temporal semantic information. Based on the analysis of the targeted patch attacks in Section 4.6 of the main body, it is clear that adversarial patches achieve targeted attacks by transferring the model's attention to patches. From the attacker's point of view, if the patches have more temporal-domain semantic information for the same patch area, then the attack performance can be further improved. Therefore, we introduce the I_t module in the fitness function to add more temporal-domain semantic information of patches. Specifically, we suggest that patches with larger temporal-domain intersections have more temporal-domain semantic information. We first define a temporal-domain weighted intersection matrix W as follows:

$$W = \begin{cases} \sum_{t=0}^{T-1} M^{(t)}(i,j), & \text{if } \sum_{t=0}^{T-1} M^{(t)}(i,j) > 1, \\ 0, & \text{otherwise}, \end{cases} \tag{1}$$

where $\sum_{t=0}^{T-1} M^{(t)}(i,j)$ denotes the cumulative number of pixels that patch at position (i,j) in all frames. $W(i,j) > 1$ means the position (i,j) belongs to temporal-domain intersection, and the number of intersections is used as the weight. The remaining positions are set to 0. Then, I_t is the cumulative sum of W.

Table 6 shows the effect of different λ on attack performance. The role of λ is to adjust the weight of different parts in the fitness function $g(\cdot)$. We find that for the untargeted attack, the contribution of I_t is not obvious, but for the targeted attack, as we expected, I_t can significantly improve the attack performance. Due to the trade-off of the targeted attack and untargeted attack, we choose $\lambda = 1.0$.

4. Implementation Details

We only set the hyper-parameters related to patch area and maximum allowed query number in each method, and the other parameters are the default values set. Follow-

λ	Untargeted Attack				Targeted Attack			
	FR(%)	AOA(%)	AOA*(%)	AQN	FR(%)	AOA(%)	AOA*(%)	AQN
0	**100**	**3.58**	**1.47**	2780	**100**	21.50	8.70	2780
0.5	**100**	3.68	1.58	2780	**100**	20.80	8.47	2770
1.0	**100**	**3.58**	**1.47**	**2720**	**100**	**20.70**	**8.42**	2730
1.5	**100**	4.16	1.74	2750	**100**	21.40	8.65	**2700**
2.0	**100**	4.21	1.74	2740	**100**	22.10	8.94	2730

Table 6. Hyper-parameters tuning on lambda λ.

ing [8], we set maximum allowed query number to 10,000 for untargeted attacks and 50,000 for targeted attacks. For patch area, we ensure that the patch area of other methods is not less than the corresponding patch area of our STDE.

4.1. TPA

We set n_occlu and TPA_N_agents to 1. For untargeted attacks, we set $rl\_batch = 400, steps = 25$. For targeted attacks, we set $rl\_batch = 1000, steps = 50$. Table 7 shows the values of $area\_occlu$ on each dataset and model, where $area\_occlu$ denotes the percentage (%) occluded by patches in total video area.

Model	UCF-101		Kinetics-400	
	Untargeted	Targeted	Untargeted	Targeted
C3D	5.00	25.00	4.00	37.00
NL	2.00	14.00	4.00	19.00
TPN	4.00	14.00	7.00	22.00

Table 7. $area\_occlu$ in TPA on each dataset and model.

4.2. Patch-Rs

For untargeted attacks, we set $n\_queries = 10,000$. For targeted attacks, we set $n\_queries = 50,000$. Table 8 shows k values on each dataset and model, where k denotes the number of pixels in the patch area on every frame.

Model	UCF-101		Kinetics-400	
	Untargeted	Targeted	Untargeted	Targeted
C3D	752	3136	502	4641
NL	1003	8530	2007	9031
TPN	2509	6523	3512	11540

Table 8. k in Patch-Rs on each dataset and model.

4.3. AdvW

We set np=50, F=0, CR=0.9, generation=7, len_x = 3 to ensure that the maximum allowed query number is 10,000. For the area of watermark, sl in AdvW denotes the scale between the watermark and the frame. On UCF-101 [6], values of sl for C3D [4], NL [7], and TPN [9] are 4.48, 7.22, 5.00 respectively, while on Kinetics-400 [1], values of sl are 5.77, 5.77, 3.77 respectively.

4.4. BSCA

For controlling the area of bullet screens, we set the font type is $DejaVuSerif$, the font of height is 9. On UCF-101, the numbers of BSC for C3D, NL, and TPN are 3, 6, 6 respectively, while on Kinetics-400, the numbers of BSC are 3, 6, 12 respectively. For controlling queries, we set $rl\_batch = 20, rl\_step = 500$ on C3D model and $rl\_batch = 100, rl\_step = 100$ on NL and TPN models.

5. More Visualizations

In this section, we provide more visualizations of our STDE with other state-of-the-art methods for untargeted attacks in Figure 1 and targeted attacks in Figure 2.

6. Analysis of Complexity

Space Complexity. BSCA adopts the reinforcement learning framework of TPA and generates rl_step parallel replicas for each clean video, which are then inputted into the video model. In contrast, our method only requires the input of a single newly generated adversarial video to compute fitness in the video model. As a result, the space complexity of BSCA and STDE is $O(n)$ and $O(1)$ respectively.

Time Complexity. The time complexity is mainly related to the number of queries and the time required to generate adversarial videos. Although BSCA adopts a space-time trade-off strategy by parallelizing the input queries, adding subtitles to the video requires traversing every video in rl_step, every frame of the video, and every bullet screens on each frame. Therefore, the time complexity of BSCA is $O(m \times T \times Q)$, where m denotes the number of bullet screens. In contrast, our method quickly generates adversarial videos using Eq. 3 in the main body, resulting in a time complexity of $O(T \times Q)$.

References

[1] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 3

[2] Kai Chen, Zhipeng Wei, Jingjing Chen, Zuxuan Wu, and Yu-Gang Jiang. Attacking video recognition models with bullet-screen comments. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 312–320, 2022. 1

[3] Francesco Croce, Maksym Andriushchenko, Naman D Singh, Nicolas Flammarion, and Matthias Hein. Sparse-rs: a versatile framework for query-efficient sparse black-box adversarial attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 6437–6445, 2022. 1

[4] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6546–6555, 2018. 3

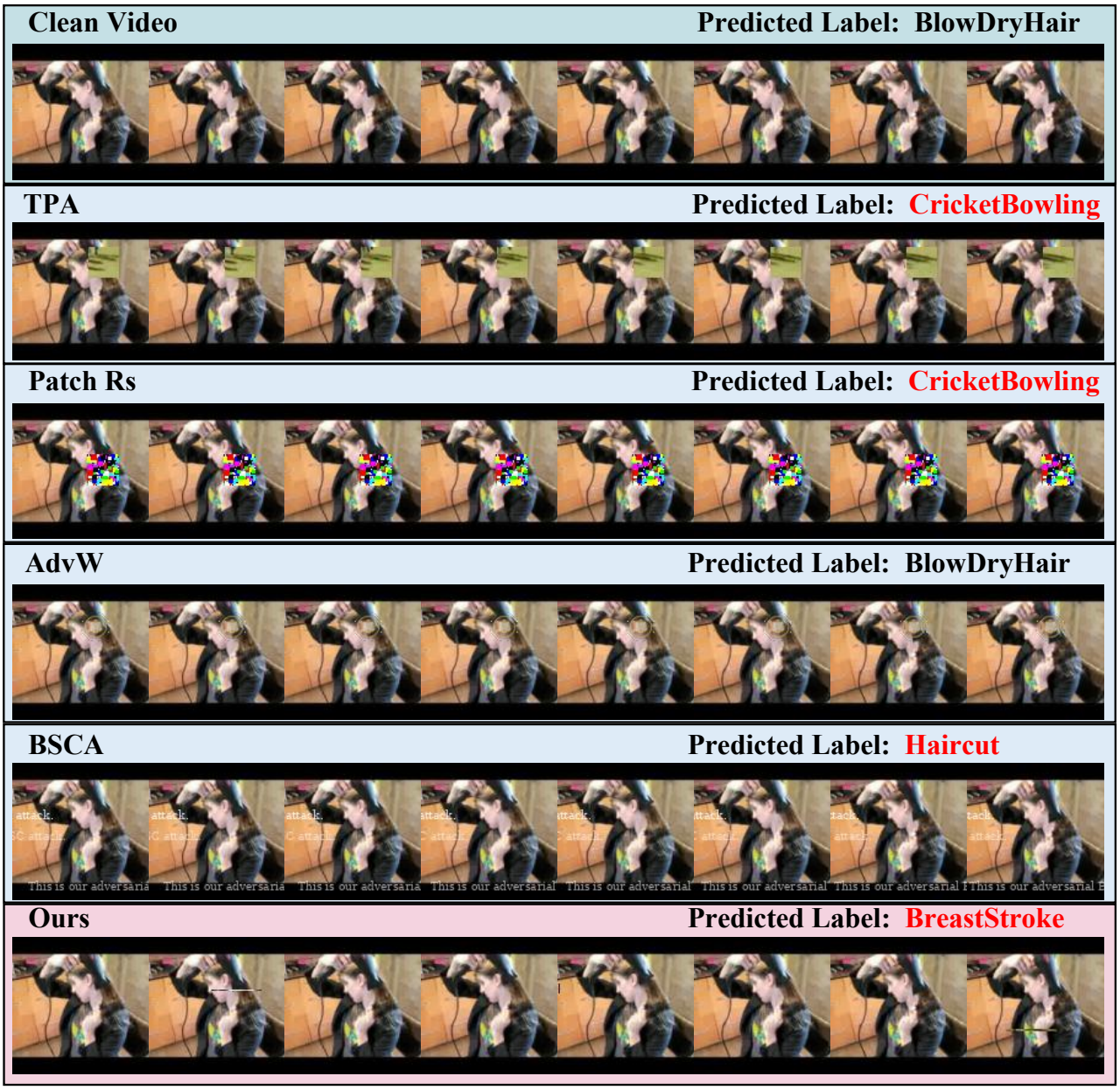

Figure 1. We visualize our method compared with TPA, Patch-Rs, AdvW, and BSCA on UCF-101 dataset for untargeted attacks against C3D model. Among them, all the other attack methods except AdvW achieve successful untargeted attacks.

[5] Xiaojun Jia, Xingxing Wei, Xiaochun Cao, and Xiaoguang Han. Adv-watermark: A novel watermark perturbation for adversarial examples. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1579–1587, 2020. 1

[6] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 3

[7] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 3

[8] Chenglin Yang, Adam Kortylewski, Cihang Xie, Yinzhi Cao, and Alan Yuille. Patchattack: A black-box texture-based attack with reinforcement learning. In *European Conference on Computer Vision*, pages 681–698. Springer, 2020. 1, 3

[9] Ceyuan Yang, Yinghao Xu, Jianping Shi, Bo Dai, and Bolei Zhou. Temporal pyramid network for action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 591–600, 2020. 3

Figure 2. We visualize our method compared with TPA, Patch-Rs on UCF-101 dataset for targeted attacks against C3D model. Among them, only TPA and Ours achieve successful targeted attacks. Compared with other methods, our method has smaller patch area.