LogoStyleFool: Vitiating Video Recognition Systems via Logo Style Transfer

Yuxin Cao[1*], Ziyu Zhao[2*], Xi Xiao[1†], Derui Wang[3], Minhui Xue[3], Jin Lu[4]

[1] Shenzhen International Graduate School, Tsinghua University, China
[2] Fan Gongxiu Honors College, Beijing University of Technology, China
[3] CSIRO's Data61, Australia
[4] Ping An Technology (Shenzhen) Co., Ltd., China

Abstract

Video recognition systems are vulnerable to adversarial examples. Recent studies show that style transfer-based and patch-based unrestricted perturbations can effectively improve attack efficiency. These attacks, however, face two main challenges: 1) Adding large stylized perturbations to all pixels reduces the naturalness of the video and such perturbations can be easily detected. 2) Patch-based video attacks are not extensible to targeted attacks due to the limited search space of reinforcement learning that has been widely used in video attacks recently. In this paper, we focus on the video black-box setting and propose a novel attack framework named *LogoStyleFool* by adding a stylized logo to the clean video. We separate the attack into three stages: style reference selection, reinforcement-learning-based logo style transfer, and perturbation optimization. We solve the first challenge by scaling down the perturbation range to a regional logo, while the second challenge is addressed by complementing an optimization stage after reinforcement learning. Experimental results substantiate the overall superiority of LogoStyleFool over three state-of-the-art patch-based attacks in terms of attack performance and semantic preservation. Meanwhile, LogoStyleFool still maintains its performance against two existing patch-based defense methods. We believe that our research is beneficial in increasing the attention of the security community to such subregional style transfer attacks.

Introduction

Short videos have become omnipresent in the current era. With the tentacles of Deep Neural Networks (DNNs) extending from images to videos, the quality in services such as video recognition (Ji et al. 2012; Carreira and Zisserman 2017; Hu et al. 2023), video segmentation (Zhou et al. 2022; Gao et al. 2023) and video compression (Chen et al. 2017; Ma et al. 2019) has been greatly improved. However, they are also encountered with severe security threats – DNNs are vulnerable to adversarial examples which are generated by surreptitiously introducing minuscule perturbations fooling the model (Szegedy et al. 2014; Goodfellow, Shlens, and Szegedy 2015). For example, attackers can perturb malicious / toxic videos to evade DNN-based detectors, which may result in severe social-economic consequences. Therefore, the outpouring of adversarial videos has raised security-critical concerns for machine-learning-assisted systems such as video verification systems (Cao et al. 2023). To date, most attacks against videos focus on crafting imperceptible perturbations restricted by ℓ_p norms. This branch of attacks considers restricted perturbations and consumes a large number of queries under the black-box setting. A recent work utilizes style transfer to introduce pixel-wise unrestricted perturbations that do not affect the semantic information of the video (Cao et al. 2023). However, seldom work utilizes sub-region perturbations, such as adversarial patches (Yang et al. 2020; Chen et al. 2022), in the attack. Nevertheless, the sub-region perturbations are considered as more generalizable and practical in the physical world.

In this paper, we investigate the risk brought by unrestricted sub-region perturbations towards video recognition systems. Such attacks face two main challenges. 1) Style transfer-based attacks add perturbations to all pixels (Cao et al. 2023), which may cause artifacts as a result of naturalness reduction. Such perturbations can also be detected with ease (Xiao et al. 2019; Jia et al. 2019). 2) Due to the limited search space of reinforcement learning (RL), patch-based video attacks (Chen et al. 2022) only support untargeted attacks (the performance of the targeted version plummets), while targeted attacks are in higher demand and of greater harm. From another perspective, despite many existing patch-based attacks in the image domain (Brown et al. 2017; Karmon, Zoran, and Goldberg 2018; Chindaudom et al. 2020; Li, Schmidt, and Kolter 2019; Yang et al. 2020; Gong et al. 2023), directly extending patch-based attacks from images to videos is hard due to the increase in data dimensionality and the lack of temporal consistency.

To address the above challenges, we follow the paths of both style transfer-based attacks and patch-based attacks in the video domain and propose **LogoStyleFool**, a black-box adversarial attack against video recognition systems via logo style transfer. Dissimilar to the existing style transfer-based video attack which imposes perturbations on all pixels, we first plumb the possibility of vitiating video recognition systems by transferring the style of the logo and forming local perturbations to superimpose on the video. Our attack first finds style images with random color initialization and builds

*These authors contributed equally.
†Corresponding author.

a style set based on the intuition that the style image, which can be classified as the target class, carries more information about the target class, thus facilitating the attack efficiency. Then the best logo, style image and position parameters are searched by RL, and the adversarial video is initiated by the original video regionally superimposed with the stylized logo. In contrast to irregular patches, the stylized logos that we integrate into the content encompass valuable semantic information, thereby preserving the inherent naturalness of the video. Additionally, our approach facilitates a meticulous logo placement, striving to position the logo in proximity to the video's corners while suppressing its size. This optimization is achieved through RL, aiming at maximizing the overall visual naturalness. Finally, the adversarial video is updated through iterative optimizations, which alleviates the limited search space of RL in existing methods. We prove the upper bound of both ℓ_∞ and ℓ_2 partial perturbations for videos in the perturbation optimization process. Through experiments compared with three existing patch-based attacks, LogoStyleFool can launch both targeted and untargeted attacks and achieve better attack performance and semantic preservation. Moreover, we demonstrate the performance of LogoStyleFool against two existing patch-based defense methods. In summary, our contributions are listed as follows.

- We propose a brand new attack framework, termed LogoStyleFool, which superimposes a stylized logo on the input video, against video recognition systems. LogoStyleFool sets up a holistic approach to patch-based attacks.
- We provide a better action space in style reference selection and initialize the video by RL-based logo style transfer, which can move the video with a stylized logo close to the decision boundary and improve the attack efficiency. We also design a novel reward function that considers the distance between logos and the corners of the video to ensure their naturalness.
- We also complement a perturbation optimization stage after RL to solve the problem of limited search space widely present in the existing patch/RL-based attacks, making patch/RL-based attacks extensible to targeted attacks. The upper bounds of both the ℓ_∞ and ℓ_2 partial perturbations assure the video's naturalness and temporal consistency.
- We show that LogoStyleFool can achieve superior attack performance and preserve semantic information in both targeted and untargeted attacks while maintaining the performance against patch-based defense methods.

Related Work

Deep-Learning-Based Video Adversarial Attacks. In the early stage, attacks are launched under the white-box setting, where the attacker can access the model architecture and parameters (Wei et al. 2019; Li et al. 2019; Inkawhich et al. 2018; Pony, Naeh, and Mannor 2021; Chen et al. 2021). For commercial systems, it is hard to secure the inner information of the model, thus, black-box attacks are more practical. Black-box video adversarial attacks assume that only the top-1 score and its label are available to attackers. Jiang et al. (Jiang et al. 2019) first proposed V-BAD to dupe video recognition systems by introducing tentative perturbations. Wei et al. (Wei et al. 2020) proposed a heuristic attack to add sparse perturbations to the input sample both temporally and spatially. Cao et al. (Cao et al. 2023) first used style transfer to add unrestricted perturbations to video samples and reduce queries by a large margin. However, perturbations in all pixels may bring about local artifacts (*e.g.,* green skin, blue leaves) that affect the naturalness.

Reinforcement-Learning-Based Adversarial Attacks. Deep RL was originally designed to learn and simulate human decision-making processes. RL has received a lot of attention, including those engaged in the issue of adversarial attacks. Yang et al. (Yang et al. 2020) proposed a patch-based black-box image attack method, PatchAttack, which utilizes RL to optimize the location and texture parameters of each patch to generate adversarial samples. Wei et al. (Wei, Yan, and Li 2022) first applied RL to black-box video attacks by designing an agent based on attack interaction and intrinsic attributes of the video to select the keyframes of the video. Similarly, RL has also been used in key frame/region selection (Wang, Sha, and Yang 2021; Wei, Wang, and Yan 2023) and perturbation optimization (Yan and Wei 2021). Reinforcement-learning-based attacks are encountered with the problem of limited search space, making most of them unadaptable to targeted attacks.

Patch-Based Attacks. In the image domain, Jia et al. (Jia et al. 2020) proposed Adv-watermark, which fooled classifiers by adding meaningful watermarks such as school badges and trademarks. However, the transparency of watermarks weakens the attack capability when the perturbed area is limited and is not effective when attacking higher dimensional data such as videos. Croce et al. (Croce et al. 2022) proposed a versatile framework, Sparse-RS. It incorporates a random color patch method, but tends to yield results that exhibit less naturalness. To the best of our knowledge, BSC (Chen et al. 2022) is the first patch-based attack in the video domain by adding bullet-screen comments. Although BSC maintains the naturalness of the video to some extent and slightly reduces queries, it can only launch untargeted attacks due to the problem of limited search space. A concurrent work (Jiang et al. 2023) proposes an efficient decision-based patch attack for videos using a spatial-temporal differential evolution framework (STDE). However, patches in targeted attacks are large enough (as discussed later) to affect the semantic information of clean videos. To conclude, there is a trade-off between attack efficiency and perturbation stealthiness.

Methodology

For the target model f, it takes a video $x \in \mathbb{R}^{T\times H\times W\times C}$ as input, and outputs its predicted label y and its score $p\left(y|x\right)$, where T, H, W and C respectively represent the frame number, height, width and channel number of the video. The attacking goal is to find a meticulously fabricated logo in a certain style and superimpose it on a certain position of the video, so as to fool the target model, *i.e.,* $f\left(x_{adv}\right) \neq y_0$ for untargeted attacks and $f\left(x_{adv}\right) = y_t$ for targeted attacks, where x_{adv} presents the adversarial video, y_0 and y_t denote the original label and the target label respectively. We consider the black-box setting where the attacker can only access the top-1 score and its label. We also assume that the target model has

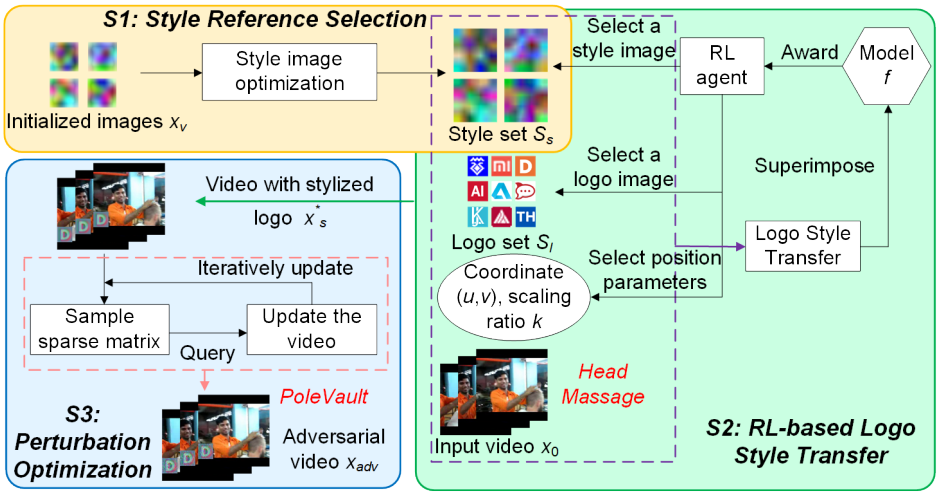

Figure 1: Overview of our proposed LogoStyleFool.

a query limit. The framework of LogoStyleFool is depicted in Figure 1. We separate LogoStyleFool into three stages: style reference selection, reinforcement-learning-based logo style transfer, and perturbation optimization.

Style Reference Selection

In order to make the stylized logo carry more information about the target label for targeted attacks (the other labels for untargeted attacks), and to reduce computational complexity, we use block perturbations to search for style images. Concretely, for a small image $x_v \in \mathbb{R}^{H_b \times W_b \times C}$, where H_b and W_b represent the block height and width respectively, the goal is to find the optimal block perturbations δ_v that satisfy

$$\begin{cases} \min\limits_{\delta_v} L\left(\phi\left(x_v+\delta_v\right), y_t\right), & \text{targeted attacks,} \\ \min\limits_{\delta_v} -L\left(\phi\left(x_v+\delta_v\right), y_0\right), & \text{untargeted attacks,} \end{cases} \tag{1}$$

where L represents the cross-entropy loss, ϕ represents the resize operation that expands the small perturbed image to the same dimension as the input video x. We use a simple black-box attack, SimBA (Guo et al. 2019), to optimize the perturbations and do not restrict the ℓ_p norm of perturbations δ_v here since the naturalness of the style image is not of significance. We denote the style image as $x_v^* = x_v + \delta_v^*$ after obtaining the optimal block perturbations δ_v^*.

As the initial values of x_v affect the style image, we randomly initialize x_v multiple times to obtain N_s style images, which also provides a larger search space for subsequent reinforcement learning. The set consisting of different style images is called the style set S_s for an input video.

Reinforcement-Learning-Based Logo Style Transfer

Logo Style Transfer. We perform style transfer for the logo region to generate more natural but unseen logos. Such a process can help conceal adversarial perturbations and meanwhile increase the difficulty for adversarial detection, since the adversarial samples are diversified when using different style images. With the aid of traditional image style transfer method (Gatys, Ecker, and Bethge 2015; Justin, Alexandre, and Li 2016), we consider content loss $L_{content}$, style loss L_{style}, and total variance loss L_{tv} for logo image l and style image x_v^*. The stylized logo l_s^* can be expressed as

$$l_s^* = \arg\min_{l_s} \lambda_c L_{content}\left(l, l_s\right) + \lambda_s L_{style}\left(l_s, x_v^*\right) + \lambda_{tv} L_{tv}\left(l_s\right), \tag{2}$$

where λ_c, λ_s and λ_{tv} represent weight coefficients. For more details about style transfer losses, please refer to the work done by Justin et al. (Justin, Alexandre, and Li 2016).

Logo Set. LLD (Sage et al. 2018) is a publicly available large logo dataset that includes over 600,000 logos worldwide. Some logos have a large number of transparent pixels, which may, after style transfer, cause abnormal perturbations (irregular colors on the white background) and weaken the concealment of stylized logos. Therefore, these logos are not suitable for style transfer. We automatically filter the logos in the dataset, remove logos with transparent pixels or a large number of white pixels, and then randomly select N_l logos to construct the desired logo set S_l.

Reinforcement Learning. Choosing the appropriate location, logo, and style for the video is a key step for attacks. Specifically, define the search space S as a set of $(u, v, k, l_{ind}, s_{ind})$ with dimension W, where (u, v) is the pixel coordinate of the upper left corner of the logo region, k is the scaling ratio of the logo, l_{ind} is the index of the logo selected from the logo set and s_{ind} is the index of the style selected from the style set. Given the height h and width w of the logo, then $u \in [0, H-kh] \cap \mathbb{Z}$, $v \in [0, W-kw] \cap \mathbb{Z}$. The training agent takes 5 actions in sequence to generate an action sequence $a \in S$, *i.e.,* selecting the logo position, scaling size, logo and style. Then the stylized logo is resized and superimposed on the video position.

We raise three requirements for stylized logos. Firstly, videos with stylized logos can be moved as close as possible to the decision boundary of the target label for targeted attacks (the original label for untargeted attacks). Secondly, the logo area should be small, making it less prominent in the video. Thirdly, the logo should be located in the corners of the video, which is not enough to affect the core semantic

part of the video. Thus, the reward function is defined as

$$R = \begin{cases} \log p\left(y_t|x_s\right) - \mu_a k^2 hw - \mu_d d, & \text{targeted}, \\ \log\left(1 - p\left(y_0|x_s\right)\right) - \mu_a k^2 hw - \mu_d d, & \text{untargeted}, \end{cases} \tag{3}$$

where x_s denotes the video superimposed with a stylized logo, $d = \min\{d_1, d_2, d_3, d_4\}$ denotes the shortest distance from the four corners of the logo to the nearest corner in the video, $d_1 = \|(u, v) - (0, 0)\|_2$, $d_2 = \|(u, v + kw) - (0, W)\|_2$, $d_3 = \|(u + kh, v) - (H, 0)\|_2$, $d_4 = \|(u + kh, v + kw) - (H, W)\|_2$, μ_a and μ_d are the area and distance coefficients balancing the penalties on the logo size and the distance from the logo to the corner.

Following prior work (Yang et al. 2020; Chen et al. 2022), the policy network, which is composed of an LSTM and a fully-connected layer, outputs the probability distribution of actions $p\left(a_t|\left(a_1, \cdots, a_{t-1}\right)\right)$ in step t, the action of which a_t is later sampled from the Categorical function. After traversing all actions, the agent will generate a sampling probability π_{θ_p} and a reward function R for an action sequence. Since the purpose of the policy network is to minimize the distance between the sampling distribution and the reward, the loss function is defined as $L\left(\theta_p\right) = -\mathbb{E}_{\tau\sim\pi_{\theta_p}}\left[R\left(\tau\right)\right]$. The policy network parameter θ_p can be updated by calculating the approximated gradient after multiple trajectory sampling (Williams 1992).

$$\nabla_{\theta_p} L\left(\theta_p\right) \approx -\frac{1}{\Omega}\sum_{\tau=1}^{\Omega}\sum_{t=1}^{W}\nabla_{\theta_p}\log\pi_{\theta_p}\left(a_t|h_t\right)R\left(\tau\right), \tag{4}$$

where Ω denotes the number of sampling trajectories, h_t denotes the hidden state of LSTM. By continuously optimizing action sequences through agents, a set of optimal action sequences can be obtained, which can move videos added with a stylized logo towards the decision boundary.

Perturbation Optimization

SimBA-DCT (Guo et al. 2019), the DCT (Discrete Cosine Transform) version of SimBA, is a query-efficient attack in the image domain. However, squarely extending SimBA-DCT to videos faces challenges. The video dimension is much larger than that of images so the attack cannot succeed if we just change every channel by step size only once. On the other hand, the attack difficulty surges if we perturb only a small region in the video. To solve these obstacles, we improve SimBA-DCT to optimize the perturbations for the logo region after reinforcement learning and name it LogoS-DCT. Optimization becomes the following.

$$\min_{\delta} L\left(x_s^* + M \odot \delta, y_t\right), \text{s.t.} \|M \odot \delta\|_p \leq \varepsilon, \tag{5}$$

where x_s^* represents the video superimposed with the optimized logo after reinforcement learning, $M \in \mathbb{R}^{T\times H\times W\times C}$ denotes the logo mask where the value in the logo region is 1, otherwise 0, δ represents the perturbation, ε stands for the perturbation threshold.

Following SimBA-DCT (Guo et al. 2019), we iteratively optimize the perturbation by randomly sampling from the orthogonal frequency set Q_{DCT} extracted by DCT. According to previous experimental attempts, most videos cannot be attacked in one round for targeted attacks, which is due to the fact that the dimension of videos is much larger than images. To address this challenge, we modify the perturbation update strategy. Attackers can perform multiple rounds of modification on all pixels in the input sample until the attack succeeds, but each pixel in each round can only be changed by one of $\{-\eta, 0, \eta\}$ compared to the initial pixel. η denotes the step size.

Proposition 1 *The perturbation after K steps can be expressed as*

$$M \odot \delta_K = M \odot \sum_{m=1}^{\min\{K,d\}} \Psi\left(A^T\left(\gamma_m \eta \mathrm{p}_m\right) A\right), \tag{6}$$

where $\Psi\left(x\right) = \mathrm{clip}_{-\varepsilon}^{+\varepsilon}\left(x\right)$ for ℓ_∞ restriction, x for ℓ_2 restriction, clip *stands for the clip operation to restrict the perturbation within the ℓ_∞ ball. A represents the DCT transformation matrix, which satisfies $A_{ij} = c\left(i\right)\cos\left[\frac{(j+0.5)\pi}{d}i\right]$, where $c\left(i\right)$ equals to $\sqrt{\frac{1}{d}}$ if $i = 0$, $\sqrt{\frac{2}{d}}$ otherwise. $\gamma_m \in \{-1, 0, 1\}$ denotes the product of signs in each round in the m-th pixel.* $\mathrm{p_m}$ *is a sparse matrix where only one element is 1, which indicates that only one channel in a certain pixel is considered at a time to optimize the direction.*

Given Proposition 1, we can derive the upper bound of perturbation for videos as follows, which means that the perturbation optimized by LogoS-DCT will not significantly break the naturalness of the video with a stylized logo.

Theorem 1 *The ℓ_∞ norm of the video adversarial perturbations in the logo area is upper bounded by $\varepsilon\rho\sqrt{\min\{K, d\}}$, where $\rho = k\sqrt{\frac{hw}{HW}}$ signifies the square root of the ratio of the logo area to a frame image.*

The upper bound of the perturbation can be extended to ℓ_2 restrictions due to the peculiarity of DCT transformation matrix A and the orthogonality of sparse matrix $\mathrm{p_m}$.

Lemma 1 *The ℓ_2 norm of any row in the DCT transformation matrix A is 1.*

Theorem 2 *The ℓ_2 norm of the video adversarial perturbations in the logo area is upper bounded by $\eta\rho\sqrt{\min\{K, d\}}$.*

Therefore, the upper bound of the perturbation is guaranteed. Proofs are given in the supplement. It is obvious that the increase in either the step size (for ℓ_2 norm), the logo size or the step number will result in a decrease in the other two. Consequently, it is necessary to find a dynamic balance among the three parameters mentioned above, *i.e.,* reasonably controlling the step size and logo size, to reduce the query number (reflected by step number) without affecting visual perception. We provide results for attacks restricted by ℓ_2 and ℓ_∞ norms, respectively, in the experimental section.

LogoStyleFool Recap

To sum up, LogoStyleFool can be separated into three stages. Firstly, the style set is built by finding multiple style images that can be misclassified. Then the optimal combination of style index, logo index, position, and size is selected through RL to obtain the video superimposed with the best stylized logo which is close enough to the decision boundary. Finally, the perturbation in the logo area

Algorithm 1: LogoStyleFool.

Input: Black-box classifier f, input video x_0, original label y_0, target label y_t, style image number N_s, logo set S_l, search space S, orthogonal frequency set Q_{DCT}, step size η, perturbation threshold ε.

Output: Adversarial video x_{adv}.

$S_s \leftarrow \texttt{find\_style}(f, N_s, y_t)$; // replace y_t with y_0 for untargeted

while *not meeting the termination condition* **do**

  $a \leftarrow$ an action sequence $(u, v, k, l_{ind}, s_{ind})$ sampled from S;

  $x_v^* \leftarrow S_s(s_{ind})$;

  $l_s^* \leftarrow \texttt{logo\_transfer}(S_l(l_{ind}), x_v^*)$;

  $M \leftarrow \texttt{cal\_mask}(u, v, k, x_0)$;

  $x_s \leftarrow x_0 + M \odot \texttt{pad}(l_s^*)$;

  Calculate reward R for targeted/untargeted attacks;

  Calculate RL loss gradient $\nabla_{\theta_p} L(\theta_p)$;

  Update policy network and the best video x_s^*;

$x_{adv} \leftarrow \texttt{LogoS\_DCT}(f, x_s^*, y_t, M, Q_{DCT}, \eta, \varepsilon)$. // replace y_t with y_0 for untargeted

is ulteriorly optimized through LogoS-DCT to obtain the adversarial video. The overall process of LogoStyleFool is shown in Algorithm 1. `find_style` outputs the style set where the images are adversarial and obtained through random initialization and unrestricted SimBA optimization. `logo_transfer` denotes the style transfer for the logo image. `cal_mask` outputs a mask matrix according to the logo position and size. The concrete process of LogoS-DCT is provided in the supplement. The source code is available at https://github.com/ziyuzhao-zzy/LogoStyleFool.

Experiments

Experimental Setup

Datasets and Models. We choose UCF-101 (Soomro, Zamir, and Shah 2012) and HMDB-51 (Kuehne et al. 2011), two datasets that are popularly used in video adversarial attacks, to verify the attack performance. We select two frequently used video recognition models, C3D (Tran et al. 2015) and I3D (Carreira and Zisserman 2017), as our target models. We beforehand trained the two models on two datasets. The video recognition accuracy for C3D and I3D on UCF-101 is 83.54% and 61.70%, while that on HMDB-51 is 66.77% and 47.92%. Please refer to the supplement for more introduction.

Benchmarks. We choose PatchAttack (Yang et al. 2020), BSC (Chen et al. 2022) and Adv-watermark (Jia et al. 2020) as benchmarks. We extend PatchAttack to videos and consider the rectangular patch with RGB perturbations for a fair comparison. Since BSC only provides results of untargeted attacks, we slightly modify it to adapt to targeted attacks. Both benchmarks optimize the patch iteratively by RL, and the attack is early stopped once the reward converges. However, it is not guaranteed that the attack has succeeded, especially for targeted attacks. Since we consider the query limit in our attack, we enlarge the batch size and the iteration step for both benchmarks to achieve comparative fairness. Owing to the inherent resemblance between watermarks and logos, we extend the application of Adv-watermark (Jia et al. 2020) to video attacks and compare it as a benchmark to our method. Following two existing video attacks V-BAD (Jiang et al. 2019) and StyleFool (Cao et al. 2023), we set the query limit as 3×10^5. The other parameters of benchmarks are set as their default values. As the positions of the bullet-screen comments in the video vary across frames in the BSC attack, the application of our Stage 3 (*i.e.,* perturbation optimization) becomes less feasible. To ensure a fair comparison, we provide outcomes obtained through our method without Stage 3.

Metrics. We use the following metrics to evaluate the attack performance. 1) Fooling rate (FR) and first two-stage fooling Rate (^{2}FR). 2) Average query (AQ), first two-stage average query (^{2}AQ), average query in each stage (AQ_1, AQ_2, and AQ_3). 3) Average Occluded Area (AOA). 4) Temporal Inconsistency (TI) (Lei, Xing, and Chen 2020). We leave the definition of metrics, parameter settings and germane analyses in the supplement.

Experimental Results

Attack Performance. We randomly select 100 videos respectively from UCF-101 and HMDB-51 to attack C3D and I3D. These videos are all correctly classified as their ground-truth labels. Table 1 and Table **??** in the supplement report the attack performance among 4 attack frameworks (we provide both ℓ_∞ and ℓ_2 versions for LogoStyleFool). Results show that although LogoStyleFool does not exhibit a dramatic (yet still pretty good) edge over PatchAttack and BSC in terms of AQ for targeted attacks, the FR of LogoStyleFool increases a lot. Due to the dimension gap between images and videos and the different attack capabilities between watermark and patch caused by transparency, Adv-watermark performs the worst, followed by PatchAttack. While the query count in Adv-watermark remains relatively modest, the corresponding success rate is notably low, with the majority of samples converging with a high loss before reaching the upper query limit. As for BSC, we find that merely increasing batch size and iteration step can increase the FR somewhat, but the attack performance is still limited due to limited search space, the problem of which has not been fundamentally resolved. We discover that the score of the target class is very low (usually below the power of 10e-3) when the reward converges. We deduce that this issue is not obvious in untargeted attacks attributed to the lower difficulty of untargeted attacks. The ^{2}FR and ^{2}AQ of LogoStyleFool also support this conjecture, since LogoStyleFool achieves comparable ^{2}FR if the attack only has the first two stages (RL in Stage 2). From another respect, increasing search space may intuitively increase the fooling rate of RL-based methods, but we find that increasing batch size cannot significantly improve FR, but instead increases AQ. This indicates that RL may not find an adversarial example that can be misclassified to a certain target class even if the search space is large enough. We address the above issues by adding perturbation optimization after RL in LogoStyleFool, resulting in better attack

Model	Attack	UCF-101-Targeted				UCF-101-Untargeted			
		FR(^{2}FR)↑	AQ(^{2}AQ)↓	AOA↓	TI↓	FR(^{2}FR)↑	AQ(^{2}AQ)↓	AOA↓	TI↓
C3D	Adv-watermark	2%	824.3	**4.38%**	5.26	46%	182.1	**4.36%**	**4.26**
	PatchAttack	6%	37,562.5	6.32%	65.35	71%	7,004.7	6.81%	73.53
	BSC	16%	32,886.3	6.29%	5.07	83%	4,611.8	7.50%	4.96
	LogoStyleFool-ℓ_∞	49%(9%)	26,382.4(2,710.5)	5.15%	4.24	97%(**81%**)	**3,308.9**(996.7)	6.02%	4.32
	LogoStyleFool-ℓ_2	**58%**(8%)	**26,003.4**(1,893.1)	5.16%	**4.16**	**98%**(79%)	3,463.0(**933.9**)	5.85%	4.48
I3D	Adv-watermark	1%	876.0	5.02%	5.72	48%	571.8	**4.66%**	4.05
	PatchAttack	2%	31,805.3	6.23%	34.67	66%	2,515.7	5.74%	25.86
	BSC	14%	33,517.2	7.01%	**3.19**	82%	**2,018.0**	6.60%	4.22
	LogoStyleFool-ℓ_∞	**42%**(6%)	**22,856.3**(2,378.6)	**4.89%**	3.51	**97%**(**85%**)	2,279.1(**680.3**)	5.84%	**3.60**
	LogoStyleFool-ℓ_2	31%(5%)	33,013.0(1,441.2)	5.06%	3.67	92%(80%)	3,742.9(741.4)	5.88%	3.65

Table 1: Attack performance comparison on UCF-101. Metric details are provided in the experimental setup.

performance. Though the LogoS-DCT process is the most query-consuming process in three stages, we reduce queries in the first two stages by initializing better style images and stylized logos. Thus, LogoStyleFool can greatly improve the fooling rate with comparable or even less AQ compared with PatchAttack and BSC in targeted attacks. Of course, there are some instances where, despite the stylized logo carrying the target class's information, the original video is too far from the target class's decision boundary, leading to a failed attack due to the query limit. We will try to solve this problem in the future. The difference between LogoStyleFool-ℓ_∞ and LogoStyleFool-ℓ_2 is reflected in Stage 3. The average ℓ_2 distance before and after Stage 3 is only 10.89 (1.17) for targeted (untargeted) attacks, which verifies that using ℓ_2 in video attacks will not cause significant changes to the video.

TI and AOA. Drawing insights from TI results in Table 1 and Table **??** in the supplement, it becomes evident that the PatchAttack demonstrates a significant deficiency in temporal consistency, which can be attributed to the incorporation of solid color patches. In contrast, several alternative methods showcase superior temporal consistency, with LogoStyleFool performing the best across the majority of cases. Attributed to its transparency, the AOA associated with the Adv-watermark is minimal. Furthermore, LogoStyleFool boasts the smallest AOA, resulting in the least amount of video occlusion when compared to the other two methods. The AOA of STDE (Jiang et al. 2023) (18.70% in targeted attacks) is rather high that abundant semantic information is redacted.

Analyses for Query Number. Under the same black-box attack setting, the current video attacks (Jiang et al. 2019; Cao et al. 2023) still require over 10^5 of queries under global perturbations. In comparison, our attack considers regional perturbations based on logos whose semantics are also well-maintained. We argue that achieving an Average Query (AQ) of approximately 20,000 (targeted) and 2,000 (untargeted) is not only a commendable outcome but is also deemed acceptable, particularly in offline attack scenarios. In particular, we achieve an FR of over 80% and an AQ of around 900 in untargeted attacks even when Stage 3 is not considered. It can be seen as a variant of our method, which can be used when the query budget is rather low.

Grad-CAM Visualizations. To further evaluate the attack performance of LogoStyleFool, we use Grad-CAM (Selvaraju et al. 2017) to visualize the local regions where the model focuses. The examples in Figure 2 show that our optimized logos have a strong ability to mislead video recognition models. As a result, adding a stylized logo can move important regions from the semantic regions that are consistent with the original label to the logo area, or even other areas which do not overlap with the logo, since our attack generates adversarial examples once successfully misleading the video classifier and does not require high scores of the target class.

Ablation Study

We verify the effectiveness of our proposed LogoStyleFool by considering different scenarios in each attacking stage. In Stage 1, we consider 1) randomly selecting style images (skipping Stage 1); 2) using solid color to initialize the style image instead of random initialization. In Stage 2, we consider 3) $\mu_a = 0$, *i.e.,* not penalizing the logo area; 4) $\mu_d = 0$, *i.e.,* not penalizing the distance from the logo to the corners. In Stage 3, we consider 5) only optimizing the perturbation one round (similar to SimBA (Guo et al. 2019)); 6) optimizing one point per frame (T points together) in a step. Table **??** reports the results under different scenarios when attacking C3D on UCF-101. In scenarios 1) and 2), the style images are not prone to carry the information of the target class, leading to attack inefficiency in targeted attacks. We observe that due to the high likelihood that random style images are already adversarial, the average query drops a lot in untargeted attacks, which can be regarded as an improvement for LogoStyleFool. We also notice that although ^{2}AQ is slightly lower than LogoStyleFool when the style image is initialized with solid color, Stage 3 consumes many queries. One possible reason is that the monotonous color in the optimized style image increases the difficulty of Stage 3. In scenarios 3) and 4), the FR improves a little, and the AQ reduces significantly in untargeted attacks. However, the larger logo area or its position near the center of the image results in the core action of the video being obstructed, thus affecting human visual perception. We denote average logo area as $\bar{a} = \sum_i k_i^2 hw$ and average minimum distance to the corners as $\bar{d} = \sum_i d_i$ for all adversarial videos. Compared with $\bar{a}$ of 646.7(733.3) and $\bar{d}$ of 13.6(11.7) for targeted (untargeted) at-

Figure 2: Grad-CAM visualizations of LogoStyleFool. Top row: targeted, bottom row: untargeted.

Model	Attack	UCF-101		HMDB-51	
		LGS	PC	LGS	PC
C3D	Adv-watermark	**41%**	38%	44%	39%
	PatchAttack	39%	29%	42%	36%
	BSC	**41%**	**42%**	42%	44%
	LogoStyleFool-ℓ_∞	40%	**42%**	**45%**	47%
	LogoStyleFool-ℓ_2	40%	39%	**45%**	**48%**
I3D	Adv-watermark	38%	42%	39%	41%
	PatchAttack	41%	36%	35%	29%
	BSC	40%	43%	34%	36%
	LogoStyleFool-ℓ_∞	46%	**55%**	44%	**46%**
	LogoStyleFool-ℓ_2	**47%**	52%	**46%**	41%

Table 2: Fooling rate (↑) of defense performance.

tacks in LogoStyleFool, $\bar{a}$ becomes 976.8(894.0) for targeted (untargeted) attacks in scenario 3), and $\bar{d}$ becomes 37.1(35.2) for targeted (untargeted) attacks in scenario 4). Considering the naturalness reduction, we do not adopt these two scenarios. We also consider different optimization strategies in scenarios 5) and 6). Untargeted attacks still exhibit better query performance than LogoStyleFool, as the perturbation required is minimal. For targeted attacks, a single round of optimization is not enough to move the video across the decision boundary, resulting in low FR. Since highly refined modification of each pixel value in the video is needed to achieve challenging targeted attacks, we attribute the loss of attack performance to the dimensionality increase from images to videos. Optimizing one pixel value per frame in each step leads to a higher query in Stage 3 since the simultaneous adjustment of multiple pixels ignores the mutual influence between these pixels. Taking into account both fooling rates and query efficiency, scenarios 1) and 5) can serve as variants of untargeted attacks, while the original LogostyleFool setting achieves higher attack efficiency for targeted attacks.

Defense Performance

We also evaluate the defense performance of our proposed LogoStyleFool against two state-of-the-art patch-based defense methods, Local Gradients Smoothing (LGS) (Naseer, Khan, and Porikli 2019) and PatchCleanser (PC) (Xiang, Mahloujifar, and Mittal 2022). LGS regularizes gradients in the estimated noisy region and inhibits the values of high activation regions caused by adversarial noise. PC uses two rounds of the pixel masking algorithm on the input image to remove the effects of adversarial patches and recover correct predictions. We extend LGS and PC to videos. We compare the performance of our approach with Adv-watermark, PatchAttack and BSC in terms of fooling rate against both defense methods, as shown in Table 2. The fooling rate is averaged on both targeted and untargeted attacks. Although watermarks are transparent and concealed, these two patch-based defenses still have a certain defensive effect on Adv-watermark. Owing to the poor performance of targeted attacks (<14%), the FRs of Adv-watermark, PatchAttack and BSC reduce a lot. The performance of LogoStyleFool is on par with those of other competitors when encountering defenses. We argue that the performance is good enough since we additionally consider perturbation optimization which adds irregular perturbations to the logo, while the perturbations of PatchAttack and BSC are smoother since only RL is considered. If we add perturbation optimization after RL for PatchAttack and BSC, the perturbations added to the bullet-screen comments or solid RGB colors are obvious enough to trigger the alarm of detection methods with humans in the loop. We provide more analyses when Stage 3 is excluded for LogoStyleFool in the supplement. We notice that there is still a gap between current defenses and a plausible defensive mechanism.

Conclusion

In this paper, we study the vulnerability of video recognition systems and propose a novel attack framework LogoStyleFool. To address the naturalness reduction of style transfer-based in all pixels, we consider regional perturbations by adding a stylized logo to the corner of the video. We design a SimBA-based strategy to select style references and use reinforcement learning to search for the optimal logo, style image and location parameters. Next, a perturbation optimizer named LogoS-DCT iteratively optimizes the adversarial video, which mitigates the problem of limited search space of RL-based methods in targeted attacks. In addition, we prove the upper bounds on the perturbations, in both ℓ_∞ and ℓ_2 norms. Comprehensive experiments show that LogoStyleFool can significantly improve the attack performance while preserving semantic information. Furthermore, LogoStyleFool parades its powerfulness and robustness against two existing defenses against patch attacks. Future work will focus on exploring potential defenses towards such subregional perturbations based on style transfer.

Acknowledgments

The authors thank anonymous reviewers for their feedback that helped improve the paper. This work was supported in part by the National Natural Science Foundation of China (61972219), the Overseas Research Cooperation Fund of Tsinghua Shenzhen International Graduate School (HW2021013). This work was also supported in part by facilities of Ping An Technology (Shenzhen) Co., Ltd., China, and CSIRO's Data61, Australia.

References

Brown, T. B.; Mané, D.; Roy, A.; Abadi, M.; and Gilmer, J. 2017. Adversarial patch. *Advances in Neural Information Processing Systems*.

Cao, Y.; Xiao, X.; Sun, R.; Wang, D.; Xue, M.; and Wen, S. 2023. Stylefool: Fooling video classification systems via style transfer. In *2023 IEEE Symposium on Security and Privacy (SP)*, 1631–1648. IEEE.

Carreira, J.; and Zisserman, A. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6299–6308.

Chen, K.; Wei, Z.; Chen, J.; Wu, Z.; and Jiang, Y.-G. 2022. Attacking video recognition models with bullet-screen comments. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 312–320.

Chen, T.; Liu, H.; Shen, Q.; Yue, T.; Cao, X.; and Ma, Z. 2017. Deepcoder: A deep neural network based video compression. In *2017 IEEE Visual Communications and Image Processing (VCIP)*, 1–4. IEEE.

Chen, Z.; Xie, L.; Pang, S.; He, Y.; and Tian, Q. 2021. Appending adversarial frames for universal video attack. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 3199–3208.

Chindaudom, A.; Siritanawan, P.; Sumongkayothin, K.; and Kotani, K. 2020. AdversarialQR: An adversarial patch in QR code format. In *2020 Joint 9th International Conference on Informatics, Electronics & Vision (ICIEV) and 2020 4th International Conference on Imaging, Vision & Pattern Recognition (icIVPR)*, 1–6. IEEE.

Croce, F.; Andriushchenko, M.; Singh, N. D.; Flammarion, N.; and Hein, M. 2022. Sparse-rs: a versatile framework for query-efficient sparse black-box adversarial attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 6437–6445.

Gao, M.; Zheng, F.; Yu, J. J.; Shan, C.; Ding, G.; and Han, J. 2023. Deep learning for video object segmentation: a review. *Artificial Intelligence Review*, 56(1): 457–531.

Gatys, L. A.; Ecker, A. S.; and Bethge, M. 2015. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*.

Gong, H.; Dong, M.; Ma, S.; Camtepe, S.; Nepal, S.; and Xu, C. 2023. Stealthy Physical Masked Face Recognition Attack via Adversarial Style Optimization. *IEEE Transactions on Multimedia*.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and harnessing adversarial examples. In *Proceedings of the International Conference on Learning Representations*.

Guo, C.; Gardner, J.; You, Y.; Wilson, A. G.; and Weinberger, K. 2019. Simple black-box adversarial attacks. In *International Conference on Machine Learning*, 2484–2493. PMLR.

Hu, K.; Jin, J.; Zheng, F.; Weng, L.; and Ding, Y. 2023. Overview of behavior recognition based on deep learning. *Artificial Intelligence Review*, 56(3): 1833–1865.

Inkawhich, N.; Inkawhich, M.; Chen, Y.; and Li, H. 2018. Adversarial attacks for optical flow-based action recognition classifiers. *arXiv preprint arXiv:1811.11875*.

Ji, S.; Xu, W.; Yang, M.; and Yu, K. 2012. 3D convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1): 221–231.

Jia, X.; Wei, X.; Cao, X.; and Foroosh, H. 2019. Comdefend: An efficient image compression model to defend adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6084–6092.

Jia, X.; Wei, X.; Cao, X.; and Han, X. 2020. Adv-watermark: A novel watermark perturbation for adversarial examples. In *Proceedings of the 28th ACM International Conference on Multimedia*, 1579–1587.

Jiang, K.; Chen, Z.; Huang, H.; Wang, J.; Yang, D.; Li, B.; Wang, Y.; and Zhang, W. 2023. Efficient decision-based black-box patch attacks on video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4379–4389.

Jiang, L.; Ma, X.; Chen, S.; Bailey, J.; and Jiang, Y.-G. 2019. Black-box adversarial attacks on video recognition models. In *Proceedings of the 27th ACM International Conference on Multimedia*, 864–872.

Justin, J.; Alexandre, A.; and Li, F. 2016. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Karmon, D.; Zoran, D.; and Goldberg, Y. 2018. Lavan: Localized and visible adversarial noise. In *International Conference on Machine Learning*, 2507–2515. PMLR.

Kuehne, H.; Jhuang, H.; Garrote, E.; Poggio, T.; and Serre, T. 2011. HMDB: a large video database for human motion recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2556–2563. IEEE.

Lei, C.; Xing, Y.; and Chen, Q. 2020. Blind video temporal consistency via deep video prior. *Advances in Neural Information Processing Systems*, 33: 1083–1093.

Li, J.; Schmidt, F.; and Kolter, Z. 2019. Adversarial camera stickers: A physical camera-based attack on deep learning systems. In *International Conference on Machine Learning*, 3896–3904. PMLR.

Li, S.; Neupane, A.; Paul, S.; Song, C.; Krishnamurthy, S. V.; Roy-Chowdhury, A. K.; and Swami, A. 2019. Stealthy Adversarial Perturbations Against Real-Time Video Classification Systems. In *Proceedings of the Symposium on Network and Distributed Systems Security (NDSS)*.

Ma, S.; Zhang, X.; Jia, C.; Zhao, Z.; Wang, S.; and Wang, S. 2019. Image and video compression with neural networks: A review. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(6): 1683–1698.

Naseer, M.; Khan, S.; and Porikli, F. 2019. Local gradients smoothing: Defense against localized adversarial attacks. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 1300–1307. IEEE.

Pony, R.; Naeh, I.; and Mannor, S. 2021. Over-the-Air Adversarial Flickering Attacks Against Video Recognition Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Sage, A.; Agustsson, E.; Timofte, R.; and Van Gool, L. 2018. Logo synthesis and manipulation with clustered generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5879–5888.

Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, 618–626.

Soomro, K.; Zamir, A. R.; and Shah, M. 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2014. Intriguing properties of neural networks. In *Proceedings of the International Conference on Learning Representations*.

Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; and Paluri, M. 2015. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 4489–4497.

Wang, Z.; Sha, C.; and Yang, S. 2021. Reinforcement learning based sparse black-box adversarial attack on video recognition models. In *Proceedings of International Joint Conference on Artificial Intelligence*.

Wei, X.; Wang, S.; and Yan, H. 2023. Efficient Robustness Assessment Via Adversarial Spatial-Temporal Focus on Videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Wei, X.; Yan, H.; and Li, B. 2022. Sparse black-box video attack with reinforcement learning. *International Journal of Computer Vision*, 130(6): 1459–1473.

Wei, X.; Zhu, J.; Yuan, S.; and Su, H. 2019. Sparse adversarial perturbations for videos. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 33, 8973–8980.

Wei, Z.; Chen, J.; Wei, X.; Jiang, L.; Chua, T.-S.; Zhou, F.; and Jiang, Y.-G. 2020. Heuristic black-box adversarial attacks on video recognition models. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 34, 12338–12345.

Williams, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Reinforcement learning*, 5–32.

Xiang, C.; Mahloujifar, S.; and Mittal, P. 2022. {PatchCleanser}: Certifiably Robust Defense against Adversarial Patches for Any Image Classifier. In *31st USENIX Security Symposium (USENIX Security 22)*, 2065–2082.

Xiao, C.; Deng, R.; Li, B.; Lee, T.; Edwards, B.; Yi, J.; Song, D.; Liu, M.; and Molloy, I. 2019. Advit: Adversarial frames identifier based on temporal consistency in videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 3968–3977.

Yan, H.; and Wei, X. 2021. Efficient sparse attacks on videos using reinforcement learning. In *Proceedings of the 29th ACM International Conference on Multimedia*, 2326–2334.

Yang, C.; Kortylewski, A.; Xie, C.; Cao, Y.; and Yuille, A. 2020. Patchattack: A black-box texture-based attack with reinforcement learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Zhou, T.; Porikli, F.; Crandall, D.; Van Gool, L.; and Wang, W. 2022. A survey on deep learning technique for video segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 1–20.