

# Attacking Video Recognition Models with Bullet-Screen Comments

Kai Chen, Zhipeng Wei, Jingjing Chen<sup>†</sup>Zuxuan Wu, Yu-Gang Jiang<sup>†</sup>

Shanghai Key Lab of Intelligent Information Processing, School of Computer Science, Fudan University  
Shanghai Collaborative Innovation Center on Intelligent Visual Computing  
{kaichen20, chenjingjing, zxwu, ygj}@fudan.edu.cn, zpwei21@m.fudan.edu.cn

## Abstract

Recent research has demonstrated that Deep Neural Networks (DNNs) are vulnerable to adversarial patches which introduce perceptible but localized changes to the input. Nevertheless, existing approaches have focused on generating adversarial patches on images, their counterparts in videos have been less explored. Compared with images, attacking videos is much more challenging as it needs to consider not only spatial cues but also temporal cues. To close this gap, we introduce a novel adversarial attack in this paper, the bullet-screen comment (BSC) attack, which attacks video recognition models with BSCs. Specifically, adversarial BSCs are generated with a Reinforcement Learning (RL) framework, where the environment is set as the target model and the agent plays the role of selecting the position and transparency of each BSC. By continuously querying the target models and receiving feedback, the agent gradually adjusts its selection strategies in order to achieve a high fooling rate with non-overlapping BSCs. As BSCs can be regarded as a kind of meaningful patch, adding it to a clean video will not affect people’s understanding of the video content, nor will arouse people’s suspicion. We conduct extensive experiments to verify the effectiveness of the proposed method. On both UCF-101 and HMDB-51 datasets, our BSC attack method can achieve about 90% fooling rate when attacking three mainstream video recognition models, while only occluding  $<8\%$  areas in the video. Our code is available at <https://github.com/kay-ck/BSC-attack>.

## Introduction

Deep Neural Networks (DNNs) have demonstrated superior performance in various video-related tasks (Song et al. 2021; Su et al. 2020; Han et al. 2021; Wang, Chen, and Jiang 2021), like video recognition (Karpathy et al. 2014; Carreira and Zisserman 2017; Wu et al. 2016; Zhang et al. 2021), video caption (Yang, Han, and Wang 2017; Liu, Ren, and Yuan 2020) and video segmentation (Nilsson and Sminchisescu 2018; Wang et al. 2019), etc. However, recent works have shown that DNNs are extremely vulnerable to video adversarial examples which are generated by applying negligible perturbations to clean input samples (Wei et al. 2019). The existence of video adversarial examples leads to security concerns of Deep Learning-based video models in real-

<sup>†</sup>Correspondence to: Jingjing Chen, Yu-Gang Jiang.  
Copyright © 2022, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

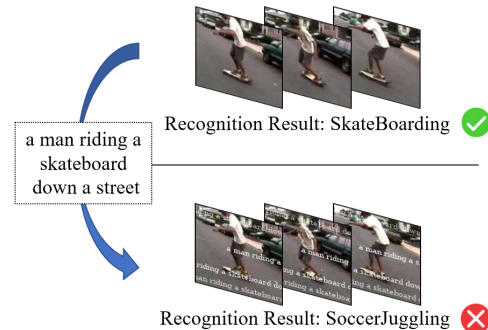


Figure 1: An illustration of adversarial BSC attacks. Given a video, it can successfully fool the video recognition model by adding BSCs.

world applications. Therefore, it has attracted increasing research interest in recent years (Wei et al. 2021b,c, 2020).

Nevertheless, most of the existing works focus on perturbation-based attacks, which introduce imperceptible changes to the clean input samples. The perturbations are constrained to have a small  $L_p$  norm and applied to the whole input. While perturbation-based attacks have been demonstrated to be effective in attacking the video recognition models, they are typically difficult to apply in the physical world. In contrast, patch-based attacks generate adversarial patches by modifying the pixels within a restricted region without any limitations on the range of changes. Therefore, patch-based attacks are stronger and more effective in the physical world. Nevertheless, existing works on patch-based attacks are mostly focused on images, patch-based attacks on videos have seldom been explored.

This paper investigates patch-based attacks on videos in the black-box setting, where the adversary can only access the output of the target model. The challenges of this task mainly come from two aspects. First, a video is a sequence of images on which the adjacent frames are closely correlated. If selecting several frames in the video as in the case of perturbation-based video attacks (Wei et al. 2020) to add adversarial patches, it will increase the perceptibility of the attack. Second, compared to images, the dimension of videos is much higher. If attaching adversarial patches to each frame of the video, it will significantly increase the

computation cost. Hence how to efficiently generate inconspicuous adversarial patches for video models in the black-box setting is the main challenge.

To address the aforementioned challenges, we propose a novel adversarial bullet-screen comment (BSC) attack method against video recognition models. As BSCs are quite popular when viewers watch videos online, people will be less sensitive to such meaningful patches than the rectangular patches (Yang et al. 2020) used in patch-based image attacks. To make the BSCs attached to each video different from each other, we introduce an image captioning model to automatically generate BSCs. Then the position and transparency of adversarial BSCs are selected based on two objectives. First, it should achieve a high fooling rate by placing the BSCs with selected transparencies on the selected positions. Second, BSCs should not overlap with each other in order to avoid obscuring the details of the video significantly. To this end, motivated by PatchAttack (Yang et al. 2020), we formulate the search over the position and transparency of BSCs as a Reinforcement Learning (RL) problem to find the optimal positions and transparencies efficiently, resulting in a query efficient attack. Specifically, in RL, we define the environment as the target model and the agent as the role of position and transparency selection. By continuously querying the target model and receiving the feedback, the agent gradually adjusts its selection strategies in order to achieve a high fooling rate and zero Intersection over Union (IoU) between different BSCs. Figure 1 shows an example of our adversarial BSC attack. As can be seen, the few BSCs do not affect our understanding of the video but fool the video recognition model successfully.

Figure 2 overviews our proposed attack framework. Given a clean video sample, the content of BSCs is generated by an image captioning model. Then the position and transparency of BSCs are optimized through RL, where the agent adjusts the positions and transparencies according to two rewards (fooling rate and IoU between different BSCs) received from the environment (target model). By continuously querying the target model, the optimal positions and transparencies are selected to generate the video adversarial example. For the agent, we use a combination of a Long-Short Term Memory network (LSTM) and a fully connected (FC) layer. In summary, our major contributions are as follows:

- We propose a novel BSC attack method against video recognition models. By formulating the attacking process with RL, our attack method achieves an efficient query.
- We design a novel reward function that considers the IoU between BSCs to ensure that the added few BSCs do not affect the understanding of videos.
- Extensive experiments on three widely used video recognition models and two benchmark video datasets (UCF-101 and HMDB-51) show that our proposed adversarial BSC framework can achieve high fooling rates.

## Related Work

In this section, we provide a short review of perturbation-based attacks on video models and patch-based attacks.

## Perturbation-based Attacks on Video Models

Perturbation-based attacks introduce imperceptible changes to the input that are restricted to have a small  $L_p$  norm and are typically applied to the whole. Perturbation-based attacks on image models are firstly explored by Szegedy et al. (Szegedy et al. 2013), where they add some imperceptible noises on clean images and mislead well-trained image classification models successfully. Sparked by this work, perturbation-based attacks on image models have been extensively studied (Goodfellow, Shlens, and Szegedy 2014; Madry et al. 2017; Carlini and Wagner 2017; Chen et al. 2017; Ilyas et al. 2018; Wei et al. 2021a; Shi and Han 2021). In the past years, perturbation-based attacks have been extended to video models. In terms of white-box attacks, where the adversary has complete access to the target model such as model parameters, model structure, etc, (Wei et al. 2019) first proposes an  $L_{2,1}$  norm regularization-based optimization algorithm to compute sparse adversarial perturbations for videos. (Li et al. 2019) leverages Generative Adversarial Network (GAN) to generate universal perturbations offline against real-time video classification systems, and the perturbations work on unseen inputs. (Chen et al. 2021) proposes to append a few dummy frames to a video clip and then add adversarial perturbations only on these new frames. For black-box attacks, (Jiang et al. 2019) first utilizes tentative perturbations transferred from the image classification model and partition-based rectifications estimated by the Natural Evolutionary Strategies to obtain good adversarial gradient estimates with fewer queries to the target model. To boost the attack efficiency and reduce the query numbers, (Wei et al. 2020) proposes to heuristically search a subset of frames and adversarial perturbations are only generated on the salient regions of selected frames. More recently, (Zhang et al. 2020) proposes a motion-excited sampler to generate sparked prior and obtain significantly better attack performance. However, black-box perturbation-based attacks often require lots of queries and are difficult to apply in the physical world.

## Patch-based Attacks

Patch-based attacks superimpose adversarial patches onto a small region of the input to create the adversarial example, making the attack more effective and applicable in the physical world by breaking the  $L_p$  norm limitations in perturbation-based attacks. At present, patch-based attacks are mainly focused on image models. Adversarial patches are first proposed by (Brown et al. 2017), which fools image classification models to ignore other scenery semantics and make wrong predictions by superimposing a relatively small patch onto the image. (Fawzi and Frossard 2016) introduces the first black-box attack, which searches the position and shape of rectangular patches using Metropolis-Hastings sampling. (Ranjan et al. 2019) further extends adversarial patches to optical flow networks and shows that such attacks can compromise their performance. Although these existing adversarial patches have powerful attack ability, they are highly conspicuous. To make adversarial patches be more inconspicuous, (Liu et al. 2019) introduces GAN to generate

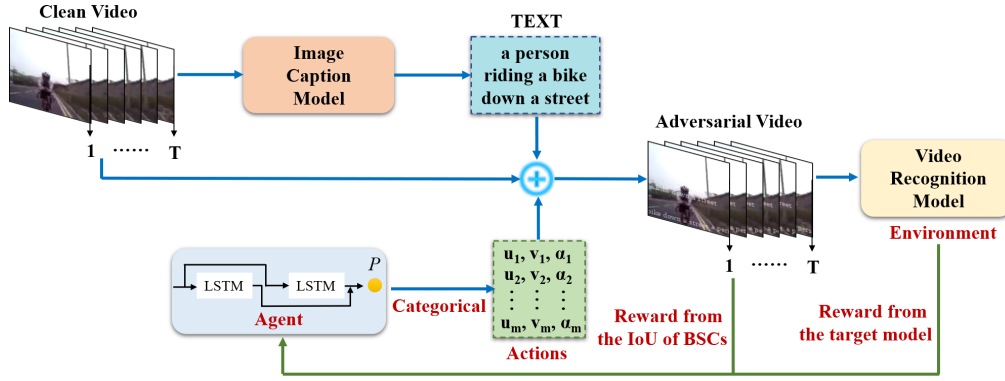


Figure 2: Overview of our black-box adversarial BSC attack method. We formulate the position, transparency selection and attacking the step into an end-to-end RL framework.

visually more natural patches. (Jia et al. 2020) further proposes to camouflage malicious information as watermarks to achieve adversarial stealthiness. This approach assumes that people’s understanding of the image content is not affected by such meaningful perturbations and hence will not arouse people’s suspicion, which is the similar assumption our approach is based on. In contrast, we disguise the adversarial patches as BSCs to attack video recognition models. As BSCs are meaningful and quite common, people will be less sensitive to such type of adversarial patch.

## Methodology

### Problem Formulation

We denote the video recognition model as a function  $F(\cdot)$ , where  $\theta_F$  denotes the model parameters. Given a clean video sample  $x \in X \subset \mathbb{R}^{T \times W \times H \times C}$ , where  $X$  is the video space,  $T, W, H, C$  denote the number of frames, frame width, frame height, and the number of channels respectively. For  $x$ , the associated ground-truth label  $y \in Y = \{1, 2, \dots, K\}$ , where  $Y$  is the label space,  $K$  denotes the number of classes. We use  $F(x) : X \rightarrow Y$  to denote the prediction of the video recognition model  $F(\cdot)$  for an input video  $x$ . The goal of adversarial attacks on video models is to generate an adversarial video  $x_{adv}$  that can fool the video recognition model. There are two types of adversarial attacks: untargeted attacks and targeted attacks. Untargeted attacks make  $F(x_{adv}) \neq y$ , while targeted attacks make  $F(x_{adv}) = y_{adv}$ , where  $y_{adv} \neq y$ . In the case of untargeted attacks, we optimize the following objective function:

$$\arg \min_{x_{adv}} -l(\mathbf{1}_y, F(x_{adv})). \quad (1)$$

where  $\mathbf{1}_y$  is the one-hot encoding of the ground truth label,  $l(\cdot)$  is the loss between the prediction and the ground truth label. In perturbation-based attacks,  $x_{adv}$  is generated by modifying each pixel of the clean video, and the modification is constrained to have a small  $L_p$  norm. In contrast, the only constraint for patch-based attacks is that the modification must be confined to a small region.

In our work, we disguise adversarial patches as meaningful BSCs to achieve stealthiness. Specifically, the BSCs

are confined to a sequence of regions within the video frames  $\epsilon = \{\epsilon_1, \dots, \epsilon_t, \dots, \epsilon_T\}$ , where  $\epsilon_t$  denotes the region of BSCs (i.e., the set of pixels belonging to the region of BSCs) in the  $t$ -th frame.  $\epsilon_t$  can be determined by giving the horizontal coordinate  $u$  and vertical coordinate  $v$  of the BSC’s position in the first frame, the font size  $h$ , and the font type  $\mathbb{T}$ . Hence, the process of determining the  $i$ -th BSC’s region in the first frame can be formalized as  $\epsilon_1^i = R(TEXT, u_i, v_i, h, \mathbb{T}), i \in \{1, \dots, m\}$ , where  $R(\cdot)$  is the function that determines the region of BSCs in the video frames,  $TEXT$  is the content of BSCs generated by the image captioning model,  $m$  is the number of BSCs. To implement the BSCs floating from right to left across the video, we translate  $\epsilon_t$  along the horizontal axis to get the region of BSCs in the  $t + 1$ -th frame. Thus, we have  $\epsilon_{t+1}^i = R(TEXT, u_i - t, v_i, h, \mathbb{T}), i \in \{1, \dots, m\}$ .

To further mitigate the effect of BSCs on the video content, we use alpha blending in (Shen, Sethi, and Bhaskaran 1998) to generate BSCs. When  $(i, j) \in \epsilon_t$ , the generation for  $x_{adv}$  is formulated as:

$$x_{adv}(t, i, j) = (p * \alpha + x(t, i, j) * (255 - \alpha)) / 255. \quad (2)$$

On the contrary, when  $(i, j) \notin \epsilon_t$ ,  $x_{adv}$  is formulated as:

$$x_{adv}(t, i, j) = x(t, i, j), \quad (3)$$

where  $(t, i, j)$  represents the position of the pixel in the video,  $p$  represents the padding of the BSCs’ region which is the color of the BSCs,  $\alpha$  represents the value of the BSC’s alpha channel which refers to the transparency of BSC’s region w.r.t. the video background.

Note that in our paper, we only focus on optimizing the position and transparency of the BSC, instead of the color and rotation, etc.

### Position and Transparency Selection

We use BSCs as adversarial patches, and the generation of video adversarial examples is only related to the position and transparency of BSCs. Searching over the position and transparency of BSCs can be formulated as an RL problem, since RL is demonstrated to be much more effective and efficient than random search strategies in (Yang et al. 2020).

In the RL framework, the agent learns to select the position and transparency of adversarial BSCs by interacting with an environment that provides the rewards and updating its actions to maximize the total expected reward. In our work, the environment consists of  $x$  and  $F(\cdot)$ , and an agent  $\mathbb{A}$  is trained to sequentially search the position and transparency of BSCs. The searching space of BSCs' potential position and transparency is defined as:

$$S = \{u_1, v_1, \alpha_1, \dots, u_i, v_i, \alpha_i, \dots, u_m, v_m, \alpha_m\},$$

$$u_i \in [-w, W], v_i \in [0, H - h], \alpha_i \in [127, 255]. \quad (4)$$

where  $w$  is the width of the BSC, which depends on the content of the BSC. From Equation 4, it can be observed that  $S$  has  $3m$  dimensions, we set the agent  $\mathbb{A}$  to take  $3m$  actions in sequence to generate  $a \in S$  and  $a = \{a_1, \dots, a_{3m}\}$ . Similar to (Yang et al. 2020), we define the agent  $\mathbb{A}$  to be a LSTM topped with a FC layer, its parameters are denoted by  $\theta_{\mathbb{A}}$ . The generation of actions is formulated as:

$$a_0 = 0, \quad (5)$$

$$P = 1, \quad (6)$$

$$\mathbf{h}_t = LSTM(\mathbf{h}_{t-1}, Embedding(a_{t-1})), t = \{1, \dots, 3m\}. \quad (7)$$

$$p(a_t | (a_1, \dots, a_{t-1})) = softmax(\theta_W \times \mathbf{h}_t). \quad (8)$$

$$a_t = Categorical(p(a_t | (a_1, \dots, a_{t-1}))). \quad (9)$$

$$P = P \cdot p(a_t | (a_1, \dots, a_{t-1})). \quad (10)$$

where the initial input  $a_0$  is set as 0, the hidden state  $\mathbf{h}_t \in \mathbb{R}^{30}$  of LSTM evolves over step  $t$ ,  $\theta_W$  represents the weight of the FC layer. The FC layer that ends with the sigmoid function predicts the probability distribution  $p(a_t | (a_1, \dots, a_{t-1}))$  over the possible actions for step  $t$ , and then one action  $a_t$  are sampled via a *Categorical* function and records the probability of the sampled action with  $P$ . The generated  $a_t$  is fed back into LSTM in the next step, which drives the LSTM state transition from  $\mathbf{h}_t$  to  $\mathbf{h}_{t+1}$ . This process is repeated until we have drawn a complete action of  $3m$  steps.

To generate adversarial and non-overlapping BSCs, we define a reward that contains two components: the reward from the feedback of the target model  $r_{attack}$  and the reward from the IoU between different BSCs  $r_{IoU}$ . The reward  $r_{attack}$  and  $r_{IoU}$  complement each other and work jointly to guide the learning of the agent:

$$r = r_{attack} + \lambda \cdot r_{IoU}. \quad (11)$$

The hyperparameter  $\lambda$  is set according to the parameter tuning which will be discussed in Section 17. The former reward  $r_{attack}$  makes the agent generate actions with a higher loss of the target model and is defined as:

$$r_{attack} = \log(1 - \mathbf{1}_y \cdot F(x_{adv})). \quad (12)$$

The reward  $r_{IoU}$  avoids significantly obscuring the details of the video due to the overlap of BSCs, which is defined as:

$$r_{IoU} = -IoU(\epsilon). \quad (13)$$

$IoU(\cdot)$  calculates the intersection area over the union area between different BSCs. In this way,  $r_{IoU}$  not only con-

---

### Algorithm 1: Adversarial BSC attack

---

**Input** : video recognition model  $F(\cdot)$ , clean video  $x$ , ground-truth label  $y$ .  
**Output** : adversarial video  $x_{adv}$ .  
**Parameter**: the number of BSCs  $m$ , the font size  $h$ , the balancing factor  $\lambda$ , the font type  $\mathbb{T}$ .

```

1 for  $i = 1$  to  $epochs$  do
2    $TEXT = I(x[0])$ ;
3    $a, P = \mathbb{A}(0)$ ;
4   for  $t = 0$  to  $T - 1$  do
5      $\epsilon_{t+1}^m = R(TEXT, u_i - t, v_i, h, \mathbb{T}), i \in \{1, \dots, m\}$ ;
6     if  $(i, j) \in \epsilon_{t+1}$  then
7        $x_{adv}(t + 1, i, j) = (p * \alpha + x(t + 1, i, j) * (255 - \alpha)) / 255$ 
8     else
9        $x_{adv}(t + 1, i, j) = x(t + 1, i, j)$ 
10    end
11  end
12   $r_{attack} = \log(1 - \mathbf{1}_y \cdot F(x_{adv}))$ ;
13   $r_{IoU} = -IoU(\epsilon)$ ;
14   $r = r_{attack} + \lambda r_{IoU}$ ;
15  Update the agent  $\mathbb{A}$ .
16 end
17 return  $x_{adv}$ 

```

---

strains the overlap between BSCs but also implicitly constrains the number of BSCs by regarding adversarial examples with overlapping BSCs as failures. Based on this reward, we expect the agent  $\mathbb{A}$  to generate non-overlapping BSCs while successfully attack video recognition models.

Then, we employ the REINFORCE algorithm (Williams 1992) to optimize the parameters  $\theta_{\mathbb{A}}$  of the agent  $\mathbb{A}$  by maximizing the expected reward  $J(\theta_{\mathbb{A}}) = E_P[r]$ :

$$\nabla_{\theta_{\mathbb{A}}} J(\theta_{\mathbb{A}}) = \frac{1}{B} \sum_{n=1}^B \nabla_{\theta_{\mathbb{A}}} r_n \log P_n, \quad (14)$$

where  $B$  is the batch size and is set as 500. We optimize the parameters via Adam with a learning rate of 0.03.

### Overall Algorithm

The overall process of our adversarial BSC attack is summarized in Algorithm 1. To enable automatically generate different BSCs for each video, a pre-trained image captioning model  $I(\cdot)$  takes the first frame of clean video  $x[0]$  as input and outputs the description that used as the BSC. Then, the agent generates an action sequence including position coordinates and transparency of  $m$  BSCs, based on which the BSCs can be attached to the video and the rewards are calculated to optimize the agent finally. The attack process is repeated until we find the adversarial BSC with  $r_{IoU} = 0$ , or the attack fails because the maximum query number is exceeded. Note that if there is more than one adversarial example with  $r_{IoU} = 0$  in the batch, we will select the one with the least salient region occluded by the BSCs. Intuitively, salient regions, for example, the foreground of the frames,

| $m$ | $FR(\%)$ | $AOA(\%)$ | $AOA^*(\%)$ | $AQN$ |
|-----|----------|-----------|-------------|-------|
| 2   | 68.3     | 4.1       | 1.5         | 9084  |
| 3   | 72.3     | 5.5       | 1.8         | 8089  |
| 4   | 79.2     | 7.3       | 2.4         | 7005  |
| 5   | 79.2     | 8.9       | 3.0         | 7292  |
| 6   | 73.3     | 10.0      | 3.5         | 8233  |

Table 1: Effects of the number of BSCs  $m$ .

| $h$ | $FR(\%)$ | $AOA(\%)$ | $AOA^*(\%)$ | $AQN$ |
|-----|----------|-----------|-------------|-------|
| 7   | 78.2     | 7.1       | 2.3         | 7193  |
| 8   | 79.2     | 7.3       | 2.4         | 7005  |
| 9   | 80.2     | 7.6       | 2.5         | 6718  |
| 10  | 81.2     | 8.4       | 2.8         | 6544  |
| 11  | 82.1     | 9.2       | 3.0         | 6263  |
| 12  | 81.2     | 10.6      | 3.7         | 6322  |
| 13  | 76.2     | 10.6      | 3.8         | 7441  |

Table 2: Effects of the font size  $h$ .

have a high probability to be the human’s focus area. Generating adversarial BSCs on the salient regions will be more likely to affect people’s understanding of the video content. Our approach is implemented on a workstation with four GPUs of NVIDIA GeForce RTX 2080 Ti.

## Experiments

### Experimental Setting

**Datasets.** We consider two popular benchmark datasets for video recognition: UCF-101 (Su et al. 2009) and HMDB-51 (Kuehne et al. 2011). UCF-101 is an action recognition dataset collected from YouTube, which contains 13,320 videos with 101 action categories. HMDB-51 is a dataset for human motion recognition and contains a total of 7000 clips distributed in 51 action classes. Both datasets split 70% of the videos as training set and the remaining 30% as test set. We randomly sample 2 videos from each category of the test dataset. During the test, 16-frame snippets are uniformly sampled from each video as input of target models. Note that, the sampled video snippet can all be classified correctly by target models.

**Target Models.** Three video recognition models, Long-term Recurrent Convolutional Network (LRCN) (Donahue et al. 2015), C3D (Hara, Kataoka, and Satoh 2018) and I3D-Slow (Feichtenhofer et al. 2019) are used as our target models. LRCN exploits the temporal information contained in successive frames, with Recurrent Neural Networks (RNNs) capturing long-term dependencies on the features generated by Convolutional Neural Networks (CNNs). In our implementation, Inception V3 (Szegedy et al. 2016) pre-trained on ImageNet is utilized to extract features from video frames and LSTM is utilized for video recognition; C3D applies 3D convolution to learn spatio-temporal features from videos with spatio-temporal filters for video recognition; I3D-Slow preserves the slow pathway, which operates at the low frame

| $\lambda$ | $FR(\%)$ | $AOA(\%)$ | $AOA^*(\%)$ | $AQN$ |
|-----------|----------|-----------|-------------|-------|
| $1e^{-5}$ | 79.2     | 7.7       | 2.6         | 7253  |
| $1e^{-4}$ | 80.2     | 7.8       | 2.5         | 6970  |
| $1e^{-3}$ | 80.2     | 7.6       | 2.5         | 6718  |
| $1e^{-2}$ | 78.2     | 7.5       | 2.6         | 7169  |
| $1e^{-1}$ | 76.2     | 7.8       | 2.6         | 7579  |

Table 3: Effect of the balancing factor  $\lambda$ .

| $\mathbb{T}(DejaVu)$  | $FR(\%)$ | $AOA(\%)$ | $AOA^*(\%)$ | $AQN$ |
|-----------------------|----------|-----------|-------------|-------|
| <i>Sans</i>           | 78.2     | 7.5       | 2.4         | 7426  |
| <i>Serif</i>          | 80.2     | 7.6       | 2.5         | 6718  |
| <i>SansMono</i>       | 76.2     | 7.3       | 2.3         | 7753  |
| <i>SansCondensed</i>  | 69.3     | 9.0       | 3.1         | 8797  |
| <i>SerifCondensed</i> | 67.3     | 8.8       | 3.0         | 9534  |

Table 4: Effect of the font type  $\mathbb{T}$ .

rate and captures spatial semantics in the SlowFast (Feichtenhofer et al. 2019) framework. These three models are the mainstream methods for video recognition. On UCF-101, the recognition accuracies for C3D, LRCN and I3D-Slow are 85.88%, 64.92% and 63.39% respectively, while on HMDB-51, the recognition accuracies are 59.95%, 37.42% and 34.9% respectively.

**Image Captioning Model.** For simplicity and efficiency, we adopt an attention-based image captioning model (Xu et al. 2015) that is pre-trained on Microsoft Common Objects in Context (MS COCO) (Lin et al. 2014) to automatically generate the description for the first frame of videos.

**Metrics.** Three metrics are used to evaluate the performance of our method on various sides. 1) Fooling rate ( $FR$ ): the ratio of adversarial videos that are successfully misclassified. 2) Average occluded area ( $AOA$ ): the average area percentage occluded by BSCs in the entire video. We use  $AOA^*$  to denote the average area percentage occluded by BSCs in the salient region. 3) Average query number ( $AQN$ ): the average number of querying the target models to finish the attacks.

### Effects of Hyperparameters

We conduct a large number of experiments to determine four hyperparameters in Algorithm 1, including the number of BSCs  $m$ , the font size  $h$ , the balancing factor  $\lambda$  in the reward, and the font type  $\mathbb{T}$ . We evaluate the attack performance of our algorithm on the C3D model with different hyperparameters. For the evaluation, we randomly sample 1 video per category from the test set of UCF-101. The sampled videos can be correctly classified by the C3D model. Then, we do a grid search to find the most appropriate values for these four hyperparameters.

Table 1 and Table 2 show the attack performance with different number of BSCs and different font sizes, respectively. The results show that when the number of BSCs  $m$  increases, the  $AOA$  will increase while the  $FR$  will firstly increase and then decrease. When the font size  $h$  increases,

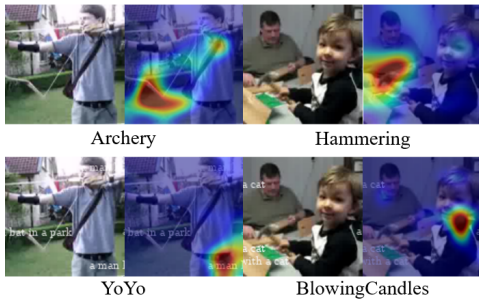


Figure 3: The top row is the clean frames and their corresponding heatmaps. The bottom row is the adversarial frames and their corresponding heatmaps.

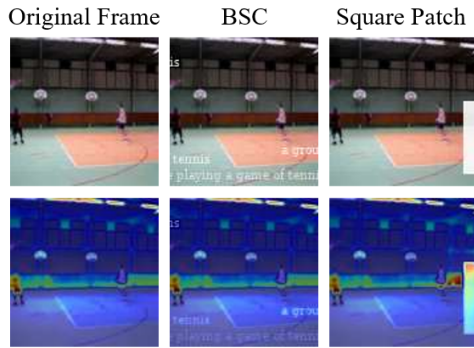


Figure 4: Examples of saliency detection for adversarial patches. We can see that our BSC does not trigger the saliency detection significantly.

$AOA$  and  $FR$  show a similar trend. That is, as the number of BSCs or the font size increases, more areas in the video are occluded, hence achieves a higher fooling rate. However, since we regard the adversarial examples with overlapping BSCs as failures, BSCs are more likely to overlap when the number of BSCs or the font size increases. To strike a balance between  $FR$ ,  $AOA$  and  $AQN$ , we set  $m = 4$  and  $h = 9$  to conduct subsequent experiments. Table 3 shows the attack performance with different balancing factors in the reward. As can be seen from the table, when  $\lambda$  increases,  $FR$  decreases slightly while  $AOA$  remains relatively stable. That is, when the reward  $r_{IoU}$  has a larger weight, the model tends to make the generated BSCs non-overlap rather than optimize the attack success rate, hence results in a lower fooling rate. Therefore, we set  $\lambda = 1e^{-3}$  so that adversarial BSC attack can achieve the highest  $FR$  (%) and the least  $AQN$ . Table 4 shows the attack performance with different DejaVu font types. According to the results, we set  $\mathbb{T} = \text{DejaVuSerif}$  to achieve the best attack performance for the adversarial BSC attack.

### Performance Comparison

We compare our method with PatchAttack(Yang et al. 2020), which is originally proposed to attack image classification models in the black-box setting. Since BSCs are usually in white and untextured, for a fair comparison, we only con-

sider the white square patch in the comparison. Different from the original setting of PatchAttack, we extend PatchAttack to attack video models by selecting the position and transparency of a white square patch with the same area as  $m$  BSCs via RL. Besides, we also compare two variants of our method. One variant uses Basin hopping (BH) (Wales and Doye 1997) instead of RL to search over the position and transparency of BSCs. BH is a stochastic optimization algorithm that can be used to find the global minimum of a multi-variate function. During each iteration, BH generates several new variables with random perturbation, then finds the local minimization, and finally accepts or rejects the new variables according to the minimized function value. The other variant randomly selects the position and transparency of the BSCs. For a fair comparison, we set the number of random trials equal to the query numbers of our method based on RL.

Table 5 lists the performance comparison against different target models on UCF-101 dataset and HMDB-51 dataset. From the results, we have the following observations. First, compared to PatchAttack, our method that uses BSCs as adversarial patches significantly reduces the occluded area. For all models, the occluded area has been reduced by more than 52% on both datasets. It is not surprising that BSCs have much smaller occluded areas since compared to a square patch, BSCs are more scattered. Second, compared to BH, RL is more effective in reducing the number of queries. For C3D and LRCN models, the number of queries has been reduced by more than 22% on both datasets. Besides, RL achieves better performance than random selection under the same query numbers. Third, in most cases, BSCs occlude wider range contents of video than a square patch with the same area and hence increases the fooling rate. Similar results are obtained by conducting experiments on Kinetics-400 (Kay et al. 2017) dataset. In summary, using BSCs as adversarial patches decreases the occluded areas and RL helps to achieve a more effective and efficient attack.

Figure 3 shows two examples of adversarial frames generated by our proposed BSC attack method on UCF-101 dataset. In addition, we further visualize the discriminative regions in the video frames for the C3D model with Gradient-weighted Class Activation Mapping (Grad-CAM) (Selvaraju et al. 2017). From the generated heatmaps, it is clear why the C3D model predicts the input frames as the corresponding correct classes. And embedding the adversarial BSCs into the frame can modify the distribution of the maximum points on the generated heatmap.

To qualitatively evaluate the risks of adversarial patches prone to spot, we use a visual saliency map to show the human-simulated focus area when they take a glance at the image. We compare the BSCs with the square patch, including the original frame as the baseline. Note that both patches occluded the same area of frame for fairness. An example and its saliency map are shown in Figure 4. We can see that the square patch can be easily highlighted in the saliency map. This means adversarial patches have a high probability to be spotted at people’s first glance. In contrast, the BSCs are relatively inconspicuous under human observation at first glance. Besides, even if they are detected, BSCs are less likely to arouse people’s suspicion than square patches.

| Dataset | Target Model | Attack Method                  | Metrics      |            |             |             |
|---------|--------------|--------------------------------|--------------|------------|-------------|-------------|
|         |              |                                | $FR(\%)$     | $AOA(\%)$  | $AOA^*(\%)$ | $AQN$       |
| UCF-101 | C3D          | PatchAttack (Yang et al. 2020) | 73.3         | 16.9       | 5.7         | 7299        |
|         |              | Our method (BH)                | 65.8         | 8.8        | 2.9         | 10473       |
|         |              | Our method (RL)                | <b>90.1</b>  | <b>7.5</b> | <b>2.5</b>  | <b>4273</b> |
|         |              | Our method (Random)            | 68.8         | 9.0        | 3.5         | -           |
|         | LRCN         | PatchAttack (Yang et al. 2020) | 97.4         | 14.0       | 2.6         | <b>1166</b> |
|         |              | Our method (BH)                | 97.4         | 8.5        | 2.8         | 1335        |
|         |              | Our method (RL)                | <b>99.5</b>  | <b>5.5</b> | <b>1.0</b>  | 1673        |
|         |              | Our method (Random)            | 97.4         | 8.6        | 2.8         | -           |
|         | I3D-Slow     | PatchAttack (Yang et al. 2020) | 92.1         | 14.6       | 4.6         | 2480        |
|         |              | Our method (BH)                | 90.1         | 8.2        | 2.7         | 3468        |
|         |              | Our method (RL)                | <b>96.5</b>  | <b>5.8</b> | <b>1.9</b>  | <b>1673</b> |
|         |              | Our method (Random)            | 89.6         | 8.2        | 2.8         | -           |
| HMDB-51 | C3D          | PatchAttack (Yang et al. 2020) | <b>92.2</b>  | 13.5       | 3.5         | <b>2500</b> |
|         |              | Our method (BH)                | 81.4         | 8.2        | 2.7         | 6358        |
|         |              | Our method (RL)                | 91.2         | <b>6.4</b> | <b>1.5</b>  | 3122        |
|         |              | Our method (Random)            | 83.3         | 8.8        | 3.2         | -           |
|         | LRCN         | PatchAttack (Yang et al. 2020) | 96.9         | 12.1       | 1.6         | 1250        |
|         |              | Our method (BH)                | 94.9         | 8.2        | 2.6         | 1617        |
|         |              | Our method (RL)                | <b>99.0</b>  | <b>4.8</b> | <b>0.7</b>  | <b>980</b>  |
|         |              | Our method (Random)            | 93.9         | 8.0        | 2.5         | -           |
|         | I3D-Slow     | PatchAttack (Yang et al. 2020) | <b>100.0</b> | 11.5       | 3.5         | <b>760</b>  |
|         |              | Our method (BH)                | 91.1         | 8.5        | 2.8         | 3453        |
|         |              | Our method (RL)                | 99.0         | <b>4.8</b> | <b>1.6</b>  | 949         |
|         |              | Our method (Random)            | 98.0         | 7.8        | 2.6         | -           |

Table 5: Attack performance on UCF-101/HMDB-51 datasets against C3D/LRCN/I3D-Slow models.

| Dataset | Target Model | Type of Patch      | $FR(\%)$    |
|---------|--------------|--------------------|-------------|
| UCF-101 | C3D          | BSC                | <b>67.9</b> |
|         |              | White Square Patch | 54.2        |
|         | LRCN         | BSC                | <b>81.7</b> |
|         |              | White Square Patch | 75.5        |
|         | I3D-Slow     | BSC                | <b>84.7</b> |
|         |              | White Square Patch | 65.0        |
| HMDB-51 | C3D          | BSC                | <b>70.7</b> |
|         |              | White Square Patch | 59.8        |
|         | LRCN         | BSC                | <b>88.3</b> |
|         |              | White Square Patch | 75.5        |
|         | I3D-Slow     | BSC                | <b>93.9</b> |
|         |              | White Square Patch | 67.3        |

Table 6: Attack performance against the LGS defense.

We also evaluate the performance of our attack method against the patch-based defense method - Local Gradient Smoothing (LGS)(Naseer, Khan, and Porikli 2019). LGS has shown the best adversarial accuracy on the ImageNet dataset against patch-based attacks among the studied patch defenses to date (Chiang et al. 2020). In order to evaluate the robustness of adversarial patches with different types,

we compare the performance of BSCs and a square patch against LGS defense in terms of the fooling rate. Since the approach is designed for images, we apply the LGS defense operation for each frame in the video. From Table 6, it is clear that the BSCs are more robust than the square patch against the LGS defense method. Since adversarial training is difficult to apply on videos, an intuitively effective defense method against our BSC attack is to use strong text removal techniques to detect and remove BSCs.

## Conclusion

In this paper, we proposed the BSC attack, a novel black-box adversarial attack against video recognition models. As the meaningful adversarial patch, few BSCs can not only attack the video model easily but also don't arouse people's suspicion. We formulate the attacking process as an RL problem, where the agent is trained to superimpose BSCs onto the videos in order to induce misclassification. Compared to BH and random selection, RL is much more query efficient and effective. We demonstrated by experiments that compared with the previous PatchAttack, the BSC attack achieves a higher fooling rate while requires fewer queries and occludes smaller areas in the video. Moreover, we also demonstrated that BSCs still have a higher fooling rate than the same area square patch against the LGS defense method.

## Acknowledgements

This work was supported in part by NSFC project (#62032006), Science and Technology Commission of Shanghai Municipality Project (20511101000), and in part by Shanghai Pujiang Program (20PJ1401900).

## References

- Brown, T. B.; Mané, D.; Roy, A.; Abadi, M.; and Gilmer, J. 2017. Adversarial patch. *arXiv preprint arXiv:1712.09665*.
- Carlini, N.; and Wagner, D. 2017. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, 39–57. IEEE.
- Carreira, J.; and Zisserman, A. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6299–6308.
- Chen, P.-Y.; Zhang, H.; Sharma, Y.; Yi, J.; and Hsieh, C.-J. 2017. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, 15–26.
- 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.
- Chiang, P.-y.; Ni, R.; Abdelkader, A.; Zhu, C.; Studer, C.; and Goldstein, T. 2020. Certified defenses for adversarial patches. *arXiv preprint arXiv:2003.06693*.
- Donahue, J.; Anne Hendricks, L.; Guadarrama, S.; Rohrbach, M.; Venugopalan, S.; Saenko, K.; and Darrell, T. 2015. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2625–2634.
- Fawzi, A.; and Frossard, P. 2016. Measuring the effect of nuisance variables on classifiers. In *British Machine Vision Conference (BMVC), CONF*.
- Feichtenhofer, C.; Fan, H.; Malik, J.; and He, K. 2019. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, 6202–6211.
- Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.
- Han, N.; Chen, J.; Xiao, G.; Zhang, H.; Zeng, Y.; and Chen, H. 2021. Fine-grained Cross-modal Alignment Network for Text-Video Retrieval. In *Proceedings of the 29th ACM International Conference on Multimedia*, 3826–3834.
- Hara, K.; Kataoka, H.; and Satoh, Y. 2018. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 6546–6555.
- Ilyas, A.; Engstrom, L.; Athalye, A.; and Lin, J. 2018. Black-box adversarial attacks with limited queries and information. In *International Conference on Machine Learning*, 2137–2146. PMLR.
- 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, 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.
- Karpathy, A.; Toderici, G.; Shetty, S.; Leung, T.; Sukthankar, R.; and Fei-Fei, L. 2014. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 1725–1732.
- Kay, W.; Carreira, J.; Simonyan, K.; Zhang, B.; Hillier, C.; Vijayanarasimhan, S.; Viola, F.; Green, T.; Back, T.; Natsev, P.; et al. 2017. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*.
- Kuehne, H.; Jhuang, H.; Garrote, E.; Poggio, T.; and Serre, T. 2011. HMDB: a large video database for human motion recognition. In *2011 International conference on computer vision*, 2556–2563. IEEE.
- 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 *NDSS*.
- Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, 740–755. Springer.
- Liu, A.; Liu, X.; Fan, J.; Ma, Y.; Zhang, A.; Xie, H.; and Tao, D. 2019. Perceptual-sensitive gan for generating adversarial patches. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, 1028–1035.
- Liu, S.; Ren, Z.; and Yuan, J. 2020. Sibnet: Sibling convolutional encoder for video captioning. *IEEE transactions on pattern analysis and machine intelligence*.
- Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.
- 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.
- Nilsson, D.; and Sminchisescu, C. 2018. Semantic video segmentation by gated recurrent flow propagation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 6819–6828.
- Ranjan, A.; Janai, J.; Geiger, A.; and Black, M. J. 2019. Attacking optical flow. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2404–2413.
- 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.



- Shen, B.; Sethi, I. K.; and Bhaskaran, V. 1998. DCT domain alpha blending. In *Proceedings 1998 International Conference on Image Processing, ICIP98 (Cat. No. 98CB36269)*, volume 1, 857–861. IEEE.
- Shi, Y.; and Han, Y. 2021. Decision-based Black-box Attack Against Vision Transformers via Patch-wise Adversarial Removal. *arXiv preprint arXiv:2112.03492*.
- Song, X.; Chen, J.; Wu, Z.; and Jiang, Y.-G. 2021. Spatial-temporal Graphs for Cross-modal Text2Video Retrieval. *IEEE Transactions on Multimedia*.
- Su, D.; Su, Z.; Wang, J.; Yang, S.; and Ma, J. 2009. UCF-101, a novel Omi/HtrA2 inhibitor, protects against cerebral ischemia/reperfusion injury in rats. *The Anatomical Record: Advances in Integrative Anatomy and Evolutionary Biology*, 292(6): 854–861.
- Su, Z.; Shang, X.; Chen, J.; Jiang, Y.-G.; Qiu, Z.; and Chua, T.-S. 2020. Video Relation Detection via Multiple Hypothesis Association. In *Proceedings of the 28th ACM International Conference on Multimedia*, 3127–3135.
- Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2818–2826.
- Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.
- Wales, D. J.; and Doye, J. P. 1997. Global optimization by basin-hopping and the lowest energy structures of Lennard-Jones clusters containing up to 110 atoms. *The Journal of Physical Chemistry A*, 101(28): 5111–5116.
- Wang, W.; Song, H.; Zhao, S.; Shen, J.; Zhao, S.; Hoi, S. C.; and Ling, H. 2019. Learning unsupervised video object segmentation through visual attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3064–3074.
- Wang, Z.; Chen, J.; and Jiang, Y.-G. 2021. Visual Co-Occurrence Alignment Learning for Weakly-Supervised Video Moment Retrieval. In *Proceedings of the 29th ACM International Conference on Multimedia*, 1459–1468.
- Wei, X.; Zhu, J.; Yuan, S.; and Su, H. 2019. Sparse adversarial perturbations for videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 8973–8980.
- Wei, Z.; Chen, J.; Goldblum, M.; Wu, Z.; Goldstein, T.; and Jiang, Y.-G. 2021a. Towards transferable adversarial attacks on vision transformers. *arXiv preprint arXiv:2109.04176*.
- 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*, volume 34, 12338–12345.
- Wei, Z.; Chen, J.; Wu, Z.; and Jiang, Y.-G. 2021b. Boosting the Transferability of Video Adversarial Examples via Temporal Translation. *arXiv preprint arXiv:2110.09075*.
- Wei, Z.; Chen, J.; Wu, Z.; and Jiang, Y.-G. 2021c. Cross-Modal Transferable Adversarial Attacks from Images to Videos. *arXiv:2112.05379*.
- Williams, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4): 229–256.
- Wu, Z.; Jiang, Y.-G.; Wang, X.; Ye, H.; and Xue, X. 2016. Multi-stream multi-class fusion of deep networks for video classification. In *Proceedings of the 24th ACM international conference on Multimedia*, 791–800.
- Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhudinov, R.; Zemel, R.; and Bengio, Y. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, 2048–2057. PMLR.
- Yang, C.; Kortylewski, A.; Xie, C.; Cao, Y.; and Yuille, A. 2020. Patchattack: A black-box texture-based attack with reinforcement learning. In *European Conference on Computer Vision*, 681–698. Springer.
- Yang, Z.; Han, Y.; and Wang, Z. 2017. Catching the temporal regions-of-interest for video captioning. In *Proceedings of the 25th ACM international conference on Multimedia*, 146–153.
- Zhang, H.; Zhu, L.; Zhu, Y.; and Yang, Y. 2020. Motion-Excited Sampler: Video Adversarial Attack with Sparked Prior. *arXiv preprint arXiv:2003.07637*.
- Zhang, X.; Wu, Z.; Weng, Z.; Fu, H.; Chen, J.; Jiang, Y.-G.; and Davis, L. 2021. VideoLT: Large-scale Long-tailed Video Recognition. *arXiv preprint arXiv:2105.02668*.