# Motion-Excited Sampler: Video Adversarial Attack with Sparked Prior

Hu Zhang[1], Linchao Zhu[1], Yi Zhu[2], and Yi Yang[1]

[1] ReLER, University of Technology Sydney, NSW
[2] Amazon Web Services
Hu.Zhang-1@student.uts.edu.au; zhulinchao7@gmail.com;
yzaws@amazon.com; Yi.Yang@uts.edu.au

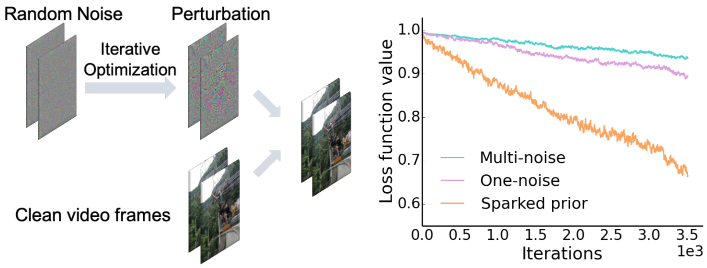
**Abstract.** Deep neural networks are known to be susceptible to adversarial noise, which is tiny and imperceptible perturbation. Most of previous works on adversarial attack mainly focus on image models, while the vulnerability of video models is less explored. In this paper, we aim to attack video models by utilizing intrinsic movement pattern and regional relative motion among video frames. We propose an effective motion-excited sampler to obtain motion-aware noise prior, which we term as sparked prior. Our sparked prior underlines frame correlations and utilizes video dynamics via relative motion. By using the sparked prior in gradient estimation, we can successfully attack a variety of video classification models with fewer number of queries. Extensive experimental results on four benchmark datasets validate the efficacy of our proposed method.

**Keywords:** Video Adversarial Attack, Video Motion, Noise Sampler.

## 1 Introduction

Despite the superior performance achieved in a variety of computer vision tasks, i.e., image classification [12], object detection [25], segmentation [11,4], Deep Neural Networks (DNNs) are shown to be susceptible to adversarial attacks that a well-trained DNN classifier may make severe mistakes due to a single invisible perturbation on a benign input and suffer dramatic performance degradation. To investigate the vulnerability and robustness of DNNs, many effective attack methods have been proposed on image models. They either consider a white-box attack setting where the adversary can always get the full access to the model including exact gradients of given input, or a black-box one, in which the structure and parameters of the model are blocked that the attacker can only access the $(input, output)$ pair through queries.

DNNs have also been widely applied in video tasks, such as video action recognition [31,18], video object detection [37], video segmentation [32], video inpainting [19] etc. However, limited work has been done on attacking video models. A standard pipeline of video adversarial attack is shown in Fig. 1(a). Specially designed perturbations are estimated from prior, which normally is



(a) (b)

**Fig. 1.** (a) A pipeline of generating adversarial examples to attack a video model. (b) Loss curve comparison: i) Multi-noise: sample noise prior individually for each frame; ii) One-noise: sample one noise prior for all frames; iii) Sparked prior (ours): sample one noise prior for all frames and sparked by motion information. Loss is computed in attacking an I3D model on Kinetics-400 dataset. The lower loss indicates the better attacking performance. Our proposed sparked prior clearly outperforms (i) and (ii) in terms of attacking video models. The figure is best viewed in color.

random noise, and imposed on the clean video frames to generate the adversarial examples. The goal is using the adversarial examples to trick the model into giving a wrong prediction. Most literature [34,21] focuses on the white-box attack setting and simply transfer the methods used in image domain to video domain. Recently [17] proposes a black-box attack method, which simply decodes a video into frames and transfers gradients from a pretrained image model for each frame. All the aforementioned methods ignore the intrinsic difference between images and videos, e.g., the extra temporal dimension. This naturally leads to a question: should we use motion information in video adversarial attack?

In this work, we propose to use motion information for black-box attack on video models. In each optimization step, instead of directly using random noise as prior, we first generate a motion map (i.e., motion vector or optical flow) between frames and construct a motion-excited sampler. The random noise will then be selected by the sampler to obtain motion-aware noise for gradient estimation, which we term as sparked prior. In the end, we feed the original video frames and the sparked prior to gradient estimator and use the estimated gradients to iteratively update the video. To show the effectiveness of using motion information, we perform a proof-of-concept comparison to two baselines. One is initializing noises separately for each frame extending [14] (multi-noise), the other is initializing one noise and replicating it across all video frames (one-noise). We use the training loss curve to reflect the effectiveness of video attack methods, in which the loss measures the distance to a fake label. As we can see in Fig. 1(b), the loss of our proposed method drops significantly faster (orange curve) than one-noise and multi-noise method. This indicates that our method takes fewer queries to successfully attack a video model. This answers our previous question that we should use motion information in video adversarial attack. Our main contributions can be summarized as follows:

- We find that simply transferring attack methods on image models to video models is less effective. Motion plays a key role in video attacking.
- We propose a motion-excited sampler to obtain sparked prior, which leads to more effective gradient estimation for faster adversarial optimization.
- We perform thorough experiments on four video action recognition datasets against two kinds of models and show the efficacy of our proposed algorithm.

## 2 Related Work

**Adversarial Attack.** Adversarial examples have been well studied on image models. [28] first shows that an adversarial sample, computed by imposing small noise on the original image, could lead to a wrong prediction. By defining a number of new losses, [1] demonstrates that previous defense methods do not significantly increase the robustness of neural networks. [23] first studies the black-box attack in image model by leveraging the transferability of adversarial examples, however, their success rate is rather limited. [13] extends Natural Evolutionary Strategies (NES) to do gradient estimation and [14] proposes to use time and data-dependent priors to reduce queries in black-box image attack. More recently, [6] proposes a meta-based method for query-efficient attack on image models.

However, limited work have been done on attacking video models. In terms of white-box attack, [34] proposes to investigate the sparsity of adversarial perturbations and their propagation across video frames. [21] leverages a Generative Adversarial Network (GAN) to account for temporal correlations and generate adversarial samples for a real-time video classification system. [15] focuses on attacking the motion stream in a two-stream video classifier by extending [9]. [5] proposes to append a few dummy frames to attack different networks by optimizing specially designed loss. The first black-box video attack method is proposed in [17], where they utilize the ImageNet pretrained models to generate a tentative gradient for each video frame and use NES to rectify it. More recently, [35] and [38] focus on sparse perturbations only on the selected frames and regions, instead of the whole video.

Our work is different from [17] because we leverage the motion information directly in generating adversarial videos. We do not utilize the ImageNet pretrained models to generate gradient for each video frame. Our work is also different from [35,38] in terms of both problem setting and evaluation metric. We follow the setting of [17] to treat the entire video as integrity, instead of attacking the video model from the perspective of frame difference in a sparse attack setting. We use attack success rate and consumed queries to evaluate our method instead of mean absolute perturbation. By using our proposed motion-aware sparked prior, we can successfully attack a number of video classification models using much fewer queries.

**Video Action Recognition.** Recent methods for video action recognition can be categorized into two families depending on how they reason motion information, i.e., 3D Convolutional Neural Networks (CNNs) [16,29,2,33,7,39] and two-stream networks [26,8,31,41,22]. 3D CNNs simply extend 2D filters to 3D filters in order to learn spatio-temporal representations directly from videos. Since early 3D models [16,29] are hard to train, many follow-up work have been proposed [2,24,30,7]. Two-stream methods [26] train two separate networks, a spatial stream given input of RGB images and a temporal stream given input of stacked optical flow images. An early [8] or late [31] fusion is then performed to combine the results and make a final prediction. Optical flow information has also been found beneficial in few-shot video classification [40]. Although 3D CNNs and two-stream networks are two different family of methods, they are not mutually exclusive and can be combined together. All aforementioned methods indicate the importance of motion information in video understanding.

Based on this observation, we believe motion information should benefit video adversarial attack. We thus propose the motion-excited sampler to generate a better prior for gradient estimation in a black-box attack setting. By incorporating motion information, our method shows superior attack performance on four benchmark datasets against two widely adopted video classification models.

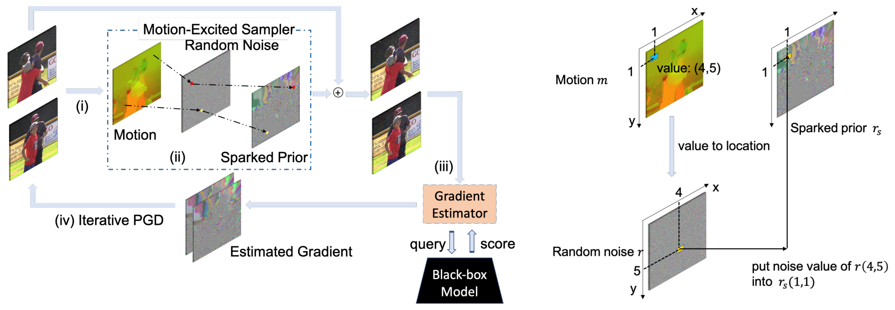
# 3 Method

## 3.1 Problem Formulation

We consider a standard video action recognition task for attacking. Suppose the given videos have an underlying distribution denoted as $\mathcal{X}$, sample $\boldsymbol{x} \in \mathbb{R}^{V \times H \times W \times C}$ and its corresponding label $y \in \{1, 2, ..., K\}$ are a pair in $\mathcal{X}$, where $V, H, W, C$ denote the number of video frames, height, width and channels of each frame respectively. $K$ represents the number of categories. We denote DNN model as function $f_{\boldsymbol{\theta}}$, where $\boldsymbol{\theta}$ represents the model's parameters. The goal of a black-box adversarial attack is to find an adversarial example $\boldsymbol{x}_{adv}$ with imperceivable difference from $\boldsymbol{x}$ to fail the target model $f_{\boldsymbol{\theta}}$ through querying the target model multiple times. It can be mathematically formulated as:

$$\begin{aligned} &\mathop{\arg\min}_{\boldsymbol{x}_{adv}} L(f_{\boldsymbol{\theta}}(\boldsymbol{x}_{adv}), y) \\ &\text{s.t. } \|\boldsymbol{x}^i_{adv} - \boldsymbol{x}^i\| \leq \kappa, i = 0, 1, ..., V-1 \\ &\quad\; \#\text{queries} \leq \text{Q} \end{aligned} \tag{1}$$

Here $i$ is the video frame index, starting from 0 to $V-1$. $\|\cdot\|$ denotes the $\ell_p$ norm that measures how much perturbation is imposed, and $\kappa$ indicates the maximum perturbations allowed. $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ is the returned logits or probability by the target model $f_{\boldsymbol{\theta}}$ when given an input video $\boldsymbol{x}$. The loss function $L(f_{\boldsymbol{\theta}}(\boldsymbol{x}_{adv}), y)$ measures the degree of certainty for the input $\boldsymbol{x}_{adv}$ maintaining true class $y$. For simplicity, we shorten the loss function as $L(\boldsymbol{x}_{adv}, y)$ in the rest of the paper



(a) (b)

**Fig. 2. (a)**: Overview of our framework for black-box video attack. i) Compute motion maps from given video frames; ii) Generate sparked prior from random noise by the proposed motion-excited sampler; iii) Estimate gradients by querying the black-box video model; iv) Use the estimated gradient to perform iterative projected gradient descent (PGD) optimization on the video. **(b)**: Illustration of Motion-Excited Sampler.

since model parameters $\boldsymbol{\theta}$ remain unchanged. The goal is to minimize the certainty and successfully fool the classification model. The first constraint enforces high similarity between clean video $\boldsymbol{x}$ and its adversarial version $\boldsymbol{x}_{adv}$. The second constraint imposes a fixed budget Q for the number of queries used in the optimization. Hence, the fewer queries required for adversarial video and the higher overall success rate within $\kappa$ perturbation, the better the attack method. The overview of our method is shown in Fig. 2(a).

### 3.2 Motion Map Generation

In order to incorporate motion information for video adversarial attack, we need to find an appropriate motion representation. There are two widely adopted motion representations in video analysis domain [26,31], motion vector and optical flow. Both of them can reflect the pixel intensity changes between two adjacent video frames. In this work, we adopt the accumulated motion vector [36] and the TVL1 flow [3] as the motion representation. We first describe accumulated motion vector as below.

Most modern codecs in video compression divide video into several intervals and split frames into $I$-frames (intracoded frames) and $P$-frames (predictive frames). Here, we denote the number of intervals as $N$ and the length of an interval as $T$. In each interval, the first frame is $I$-frame and the rest $T-1$ are $P$-frames. The accumulated motion vector is formulated as the motion vector of each $P$-frame, that can trace back to the initial $I$-frame instead of depending on previous $P$-frames. Suppose the accumulated motion vector in frame $t$ of interval $n$ is denoted as $\boldsymbol{m}^{(t,n)} \in \mathbb{R}^{H\times W\times 2}$ and each pixel value at location $i$ in $\boldsymbol{m}^{(t,n)}$ can be computed as:

$$\boldsymbol{m}_i^{(t,n)} = i - \mathcal{B}_i^{(t,n)}, 1 \le t \le T-1, 0 \le n \le N-1 \tag{2}$$

where $\mathcal{B}_i^{(t,n)}$ represents the location traced back to initial $I$-frame from $t$-th frame in interval $n$. We refer the readers to [36] for more details.

For an interval with $T$ frames, we can obtain $T-1$ accumulated motion vectors. We only choose the last one $\boldsymbol{m}^{(T-1,n)}$ since it traces all the way back to the $I$-frame in this interval and is most motion-informative. We abbreviate $\boldsymbol{m}^{(T-1,n)}$ as $\boldsymbol{m}^{(n)}$ and thus, we have a set of $N$ accumulated motion vectors for the whole video, denoted as $\mathcal{M} = \{\boldsymbol{m}^{(0)}, \boldsymbol{m}^{(1)}, ..., \boldsymbol{m}^{(N-1)}\}$.

Optical flow is a motion representation that is similar to motion vector with the same dimension $\mathbb{R}^{H\times W\times 2}$. It also contains spatial details but is not directly available in compressed videos. Here we use TVL1 algorithm [3] to compute the flow given its wide adoption in video-related applications [31,2,7]. We apply the same strategy as motion vectors to select optical flow and will also obtain $N$ flow vectors. The set of flow vectors is also denoted as $\mathcal{M}$ for simplicity.

### 3.3 Motion-Excited Sampler

In a black-box setting, random noise is normally employed in generating adversarial examples. However, as stated before in Fig. 1(b), direct usage of random noise is not promising in video attack. To tackle this problem, we propose to involve motion information in the process of generating adversarial examples, and thus propose motion-excited sampler.

First, we define the operation of motion-excited sampler (ME-Sampler) as

$$\boldsymbol{r}_s = \text{ME-S}\textsc{ampler}(\boldsymbol{r}, \boldsymbol{m}), \tag{3}$$

where $\boldsymbol{r} \in \mathbb{R}^{V\times H\times W\times 3}$ denotes the initial random noise, motion maps $\boldsymbol{m} \in \mathbb{R}^{V\times H\times W\times 2}$ are selected with replacement from set $\mathcal{M}$ introduced in Section 3.2. $\boldsymbol{r}_s \in \mathbb{R}^{V\times H\times W\times 3}$ will be the transformed motion-aware noise, which we term as **sparked prior** afterwards.

To be specific, we use the motion-excited sampler to "warp" the random noise by motion. It is not just rearranging the pixels in the random noise, but constructing a completely new prior given the motion information. For simplicity, we only consider the operation for one frame here. It is straightforward to extend to the case of multiple frames. Without abuse of notation, we still use $\boldsymbol{r}, \boldsymbol{r}_s, \boldsymbol{m}$ for clarification in this section.

At the $i$-th location of the motion map $\boldsymbol{m}$, we denote the motion vector value as $p_i \in \mathbb{R}^2$ and its coordinate is denoted $(x_i,\ y_i)$, i.e., $p_i = \boldsymbol{m}[x_i, y_i]$. Here, $p_i = (u_i, v_i)$ has two values, which indicate the horizontal and vertical relative movements, respectively. When computing the value of position $(x_i, y_i)$ in sparked prior $\boldsymbol{r}_s$, $(u_i, v_i)$ will serve as the new coordinates for searching in original random noise. The corresponding noise value of $\boldsymbol{r}[u_i, v_i]$ will be assigned as the value in $\boldsymbol{r}_s[x_i, y_i]$. Thus, we have:

$$\begin{aligned} (u_i, v_i) &= \boldsymbol{m}[x_i, y_i], \\ \boldsymbol{r}_s[x_i, y_i] &= \boldsymbol{r}[u_i, v_i]. \end{aligned} \tag{4}$$

**Algorithm 1** GRAD-EST$(\boldsymbol{x}, y, \boldsymbol{g}, \text{loop})$: Estimate $\ell(\boldsymbol{g}) = -\nabla_{\boldsymbol{g}}\langle \nabla_{\boldsymbol{x}} L(\boldsymbol{x}, y), \boldsymbol{g}\rangle$.

```
Input: video $\boldsymbol{x}$, its label $y$, number of frame of video $\boldsymbol{x}$ is $V$. initialized $\boldsymbol{g}$, interval t
    for sampling new motion map, $\delta$ for loss variation and $\epsilon$ for approximation.
if loop % t = 0 then
    motion map $\boldsymbol{m}_1, \boldsymbol{m}_2, ..., \boldsymbol{m}_{V-1}$ are chosen from $\mathcal{M} = \{\boldsymbol{m}^{(0)}, \boldsymbol{m}^{(1)}, ..., \boldsymbol{m}^{(N-1)}\}$
    with replacement and are concatenated to be $\boldsymbol{m}$;
end if
$\boldsymbol{r} \leftarrow \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$;
$\boldsymbol{r}_s = \text{ME-SAMPLER}(\boldsymbol{r}, \boldsymbol{m})$;
$\boldsymbol{w}_1 = \boldsymbol{g} + \delta \boldsymbol{r}_s$;
$\boldsymbol{w}_2 = \boldsymbol{g} - \delta \boldsymbol{r}_s$;
$\ell(\boldsymbol{w}_1) = -\langle \nabla_{\boldsymbol{x}} L(\boldsymbol{x}, y), \boldsymbol{w}_1\rangle \approx \frac{L(\boldsymbol{x},y) - L(\boldsymbol{x}+\epsilon\cdot\boldsymbol{w}_1, y)}{\epsilon}$;
$\ell(\boldsymbol{w}_2) = -\langle \nabla_{\boldsymbol{x}} L(\boldsymbol{x}, y), \boldsymbol{w}_2\rangle \approx \frac{L(\boldsymbol{x},y) - L(\boldsymbol{x}+\epsilon\cdot\boldsymbol{w}_2, y)}{\epsilon}$;
$\boldsymbol{\Delta} = \frac{L(\boldsymbol{x}+\epsilon\boldsymbol{w}_2, y) - L(\boldsymbol{x}+\epsilon\boldsymbol{w}_1, y)}{\delta\epsilon}\boldsymbol{r}_s$;
Output: $\boldsymbol{\Delta}$.
```

We give a simplified example in Fig. 2(b) to show how our motion-excited sampler works. To determine pixel value located in $(1, 1)$ of $\boldsymbol{r}_s$, we first get motion value $(4, 5)$ from motion map $\boldsymbol{m}$ at its $(1, 1)$ location. We then select pixel value located in $(4, 5)$ of $\boldsymbol{r}$ and put its value into location $(1, 1)$ of $\boldsymbol{r}_s$.

Generally speaking, sparked prior is still a noise map. Note that, initial noise is completely random and irrelevant to the input video. With motion-excited sampler (operation in Eq. 4 and Fig. 2(b)), pixels with the same movements will have the same noise values in sparked prior. Then, sparked prior connects different pixels on the basis of motion map and is thus block-wised. Compared to the initial noise which is completely random, sparked prior is more informative and relevant to the video because of the incorporated motion information. It thus helps to guide the direction of estimated gradients towards attacking video models in a black-box setting and enhances the overall performance.

### 3.4 Gradient Estimation and Optimization

Once we have the sparked prior, we incorporate it with the input video and feed them to the black-box model to estimate the gradients. We consider $\ell_{inf}$ noise in our paper following [17], but our framework also applies to other norms.

Similar to [14], rather than directly estimating the gradient $\nabla_{\boldsymbol{x}} L(\boldsymbol{x}, y)$ for generating adversarial video, we perform iterative updating to search. The new loss function designed for such optimization is,

$$\ell(\boldsymbol{g}) = -\langle \nabla_{\boldsymbol{x}} L(\boldsymbol{x}, y), \boldsymbol{g}\rangle, \tag{5}$$

$\nabla_{\boldsymbol{x}} L(\boldsymbol{x}, y)$ is the groundtruth gradient we desire and $\boldsymbol{g}$ is the gradient to be estimated. An intuitive observation from this loss function is that iterative minimization of Eq. 5 will drive our estimated gradient $\boldsymbol{g}$ closer to the true gradient.

We denote $\varDelta$ to be the gradient $\nabla_{\boldsymbol{g}}\ell(\boldsymbol{g})$ of loss $\ell(\boldsymbol{g})$. We perform a two-query estimation to the expectation and apply the authentic sampling to get

$$\varDelta = \frac{\ell(\boldsymbol{g} + \delta\boldsymbol{r}_s) - \ell(\boldsymbol{g} - \delta\boldsymbol{r}_s)}{\delta}\boldsymbol{r}_s, \tag{6}$$

where $\delta$ is a small number adjusting the magnitude of loss variation. By substituting Eq. (5) to Eq. (6), we have

$$\varDelta = \frac{\langle \nabla_{\boldsymbol{x}} L(\boldsymbol{x}, y), \boldsymbol{g} - \delta\boldsymbol{r}_s\rangle - \langle \nabla_{\boldsymbol{x}} L(\boldsymbol{x}, y), \boldsymbol{g} + \delta\boldsymbol{r}_s\rangle}{\delta}\boldsymbol{r}_s. \tag{7}$$

In the context of finite difference method, we notice, given the function $L$ at a point $\boldsymbol{x}$ in the direction of vector $\boldsymbol{g}$, the directional derivative $\langle \nabla L(\boldsymbol{x}, y), \boldsymbol{g}\rangle$ can be transferred as:

$$\langle \nabla_{\boldsymbol{x}} L(\boldsymbol{x}, y), \boldsymbol{g}\rangle \approx \frac{L(\boldsymbol{x} + \epsilon\boldsymbol{g}, y) - L(\boldsymbol{x}, y)}{\epsilon}, \tag{8}$$

$\epsilon$ is a small constant for approximation. By combining Eq. (7)-(8), we have,

$$\varDelta = \frac{L(\boldsymbol{x} + \epsilon\boldsymbol{w}_2, y) - L(\boldsymbol{x} + \epsilon\boldsymbol{w}_1, y)}{\delta\epsilon}\boldsymbol{r}_s, \tag{9}$$

with $\boldsymbol{w}_1 = \boldsymbol{g} + \delta\boldsymbol{r}_s$ and $\boldsymbol{w}_2 = \boldsymbol{g} - \delta\boldsymbol{r}_s$. The resulting algorithm for generating gradient for $\boldsymbol{g}$ is shown in Algorithm 1.

Once we have Algorithm 1, we can use it to update estimated gradient and optimize the adversarial video. To be specific, in iteration $t$, $\varDelta_t$ is returned by Algorithm 1. We update $\boldsymbol{g}_t$ by simply applying one-step gradient descent: $\boldsymbol{g}_t = \boldsymbol{g}_{t-1} - \eta\varDelta_t$, $\eta$ is a hyperparameter to update $\boldsymbol{g}_t$. The updated $\boldsymbol{g}_t$ is the gradient we want to use for generating adversarial videos. Finally, we combine our estimated $\boldsymbol{g}_t$ with projection gradient descent (PGD) to translate our gradient estimation algorithm into an efficient video adversarial attack method. The detailed procedure is shown in Algorithm 2, in which $\arg\max[f_{\boldsymbol{\theta}}(\boldsymbol{x}_t)]$ returns top predicted class label, $\textsc{Clip}(\cdot)$ constrain the updated video $\boldsymbol{x}_t$ close to the original video $\boldsymbol{x}_0$, where $\boldsymbol{x}_0 - \kappa$ is the lower bound and $\boldsymbol{x}_0 + \kappa$ the upper bound. $\kappa$ is the noise constraint in Eq. (1).

### 3.5 Loss Function

Different from applying cross-entropy loss directly, we adopt the idea in [1] and design a logits-based loss. Here, the logits returned from the black-box model is denoted as $l \in \mathbb{R}^K$, where $K$ is the number of classes. We denote the class of largest value in logits $l$ as $y$, the largest logits value is $l_y$. The final loss can be obtained as $L = \max(l_y - \max_{k\neq y} l_k, 0)$. Minimizing $L$ is expected to confuse the model with the second most confident class prediction so that our adversarial attack could succeed.

**Algorithm 2** Adversarial Example Optimization for $\ell_{inf}$ norm perturbations.

```
Input: original video $\boldsymbol{x}$, its label $y$, learning rate $h$ for updating adversarial video.
$\boldsymbol{x}_0 \leftarrow \boldsymbol{x}$, initially estimated $\boldsymbol{g}_0 \leftarrow \mathbf{0}$, initial loop $t = 1$;
while $\arg\max[f_{\boldsymbol{\theta}}(\boldsymbol{x}_t)] = y$ do
    $\boldsymbol{\Delta}_t = \textsc{Grad-Est}(\boldsymbol{x}_{t-1}, y, \boldsymbol{g}_{t-1}, t-1)$;
    $\boldsymbol{g}_t = \boldsymbol{g}_{t-1} - \eta\boldsymbol{\Delta}_t$;
    $\boldsymbol{x}_t = \boldsymbol{x}_{t-1} - h \cdot sign(\boldsymbol{g}_t)$;
    $\boldsymbol{x}_t = \textsc{Clip}(\boldsymbol{x}_t, \boldsymbol{x}_0 - \kappa, \boldsymbol{x}_0 + \kappa)$;
    $t = t + 1$;
end while
Output: $\boldsymbol{x}_t$.
```

**Table 1.** Test accuracy (%) of the video models.

| Model | Kinetics-400 | UCF-101 | HMDB-51 | SthSth-V2 |
|---|---|---|---|---|
| I3D | 70.11 | 93.55 | 68.30 | 50.25 |
| TSN2D | 68.87 | 86.04 | 54.83 | 35.11 |

# 4 Experiments

## 4.1 Experimental Setting

**Datasets.** We perform video attack on four video action recognition datasets: UCF-101 [27], HMDB-51 [20], Kinetics-400 [18] and Something-Something V2 [10]. UCF-101 consists of 13,200 videos spanned over 101 action classes. HMDB-51 includes 6,766 videos in 51 classes. Kinetics-400 is a large-scale dataset which has around 300K videos in 400 classes. Something-Something V2 is a recent crowd-sourced video dataset on human-object interaction that needs more temporal reasoning. It contains 220,847 video clips in 174 classes. For notation simplicity, we use SthSth-V2 to represent Something-Something V2.

**Video Models.** We choose two video action recognition models, I3D [18] and TSN2D [31], as our black-box models. For I3D training on Kinetics-400 and SthSth-V2, we train it from ImageNet initialized weights. For I3D training on UCF-101 and HMDB-51, we train it with Kinetics-400 pretrained parameters as initialization. For TSN2D training, we use ImageNet initialized weights on all four datasets. The test accuracy of two models can be found in Table 1.

**Attack Setting.** We perform both untargeted and targeted attack under limited queries. Untargeted attack requires the given video to be misclassified to any wrong label and targeted attack requires classifying it to a specific label. For each dataset, we randomly select one video from each category following the setting in [17]. All attacked videos are correctly classified by the black-box model. We impose $\ell_{inf}$ noise on video frames whose pixels are normalized to 0-1. We constrain the maximum perturbation $\kappa = 0.03$, maximal queries Q = 60,000 for untargeted attack and $\kappa = 0.05$, Q = 200,000 for targeted attack. If one video is failed to attack within these constraints, we record its consumed queries as Q.

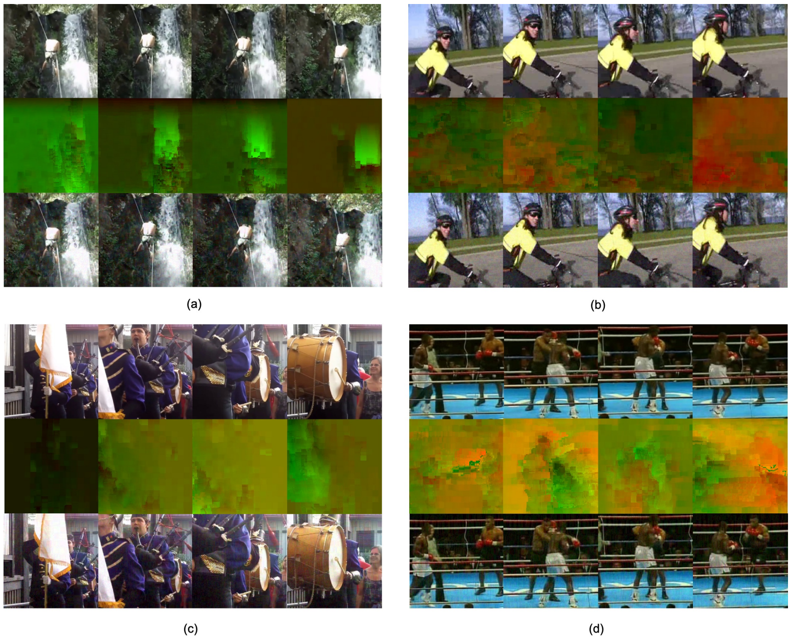


**Fig. 3.** Examples of motion vectors in generating adversarial samples. In (a)-(d), the first row is the original video frame, the second row is the motion vector and the third row is generated adversarial video frame. a) Kinetics-400 on I3D: Abseling→Rock climbing; b) UCF-101 on I3D: Biking→Walking with dog; c) Kinetics-400 on TSN: Playing bagpipes→Playing accordion; d) UCF-101 on TSN: Punching→Lunges.

**Evaluation Metric.** We use the average number of queries (ANQ) required in generating effective adversarial examples and the attack success rate (SR). ANQ measures the average number of queries consumed in attacking across all videos and SR shows the overall success rate in attacking within query budget Q. A smaller ANQ and higher SR is preferred. For now, there is not a balanced metric that takes both ANQ and SR into account.

### 4.2 Comparison to State-of-the-Art

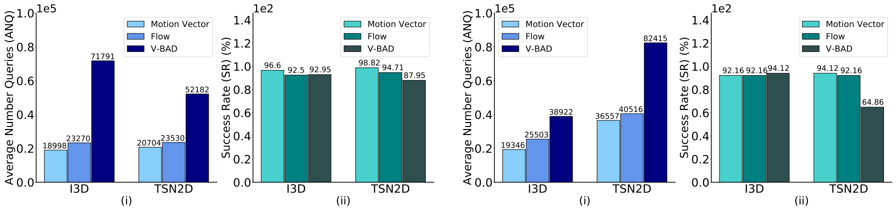
We report the effectiveness of our proposed method in Table 2. We present the results of leveraging two kinds of motion representations: Motion Vector (MV) and Optical Flow (OF) in our proposed method. In comparison, we first show the attacking performance of V-BAD [17] under our video models since V-BAD is the only directly comparable method. We also extend two image attack methods [13,14] as strong baselines to video to demonstrate the advantage of using motion information. They are denoted as E-NES and E-Bandits respectively and their attacking results are shown in Table 2.

**Table 2.** Untargeted attacks on SthSth-V2, HMDB-51, Kinetics-400, UCF-101. The attacked models are I3D and TSN2D. “ME-Sampler” denotes the results of our method. “OF” denotes Optical Flow. “MV” denotes Motion Vector.

| Dataset / Model | Method | I3D | | TSN2D | |
|---|---|---|---|---|---|
| | | ANQ | SR(%) | ANQ | SR(%) |
| SthSth-V2 | E-NES [13] | 11,552 | 86.96 | 1,698 | 99.41 |
| | E-Bandits [14] | 968 | 100.0 | 435 | 99.41 |
| | V-BAD [17] | 7,239 | 97.70 | 495 | 100.0 |
| | ME-Sampler (OF) | 735 | 98.90 | 315 | 100.0 |
| | ME-Sampler (MV) | **592** | **100.0** | **244** | **100.0** |
| HMDB-51 | E-NES [13] | 13,237 | 84.31 | 19,407 | 76.47 |
| | E-Bandits [14] | 4,549 | 99.80 | 4,261 | 100.0 |
| | V-BAD [17] | 5,064 | 100.0 | 2,405 | 100.0 |
| | ME-Sampler (OF) | **3,306** | **100.0** | 842 | 100.0 |
| | ME-Sampler (MV) | 3,915 | 100.0 | **831** | **100.0** |
| Kinetics-400 | E-NES [13] | 11,423 | 89.30 | 20,698 | 71.93 |
| | E-Bandits [14] | 3,697 | 99.00 | 6,149 | 97.50 |
| | V-BAD [17] | 4,047 | **99.75** | 2,623 | 99.75 |
| | ME-Sampler (OF) | 3,415 | 99.30 | 2,631 | 98.80 |
| | ME-Sampler (MV) | **2,717** | 99.00 | **1,715** | **99.75** |
| UCF-101 | E-NES [13] | 23,531 | 69.23 | 41,328 | 34.65 |
| | E-Bandits [14] | 10,590 | 89.10 | 24,890 | 66.33 |
| | V-BAD [17] | 8,819 | 97.03 | 17,638 | 91.09 |
| | ME-Sampler (OF) | 6,101 | 96.00 | 6,598 | 97.00 |
| | ME-Sampler (MV) | **4,748** | **98.02** | **5,353** | **99.00** |

Overall, our method using motion information achieves promising results on different datasets and models. On SthSth-V2 and HMDB-51, we even achieve 100% SR. On Kinetics-400 and UCF-101, we also get over 97% SR. The number of queries used in attacking is also encouraging. One observation worth noticing is that it only takes hundreds of queries to completely break the models on SthSth-V2. For the rest of models, we just consume slightly more queries. To analyze this, we observe that models consuming slightly more queries often have higher recognition accuracy from Table 1. From this result, we conclude that a model is likely to be more robust if its performance on the clean video is better.

In terms of using motion vector and optical flow, we find that motion vector outperforms optical flow in most cases, e.g., the number of queries used 5,353 (MV) vs 6,598 (OF) on UCF-101 against TSN2D. The reason is that motion vector always has a clearer motion region since it is computed by a small block of size 16 × 16. However, optical flow is always pixel-wisely calculated. It is not difficult to imagine that when the region used for describing is relatively larger, it is easier and more accurate to portray the overall motion. When only considering each pixel, the movement is likely to be lost in tracking and make some mistakes.



(a) SthSth-V2

(b) HMDB-51

**Fig. 4. (a)**: Comparisons of targeted attack on SthSth-V2 with V-BAD: i) Average queries consumed by I3D and TSN2D; ii) Success rate achieved by I3D and TSN2D. **(b)**: Comparisons of targeted attack on HMDB-51 with V-BAD: i) Average queries consumed by I3D and TSN2D; ii) Success rate achieved by I3D and TSN2D.

Compared to E-NES and E-Bandits, we achieve better results, either on consumed queries or success rate, e.g., when attacking a TSN2D model on UCF-101, our success rate is 99.00%, which is much higher than 34.65% for E-NES and 66.33% for E-Bandits. The query 5,353 is also much smaller than 41,328 and 24,890. When compared to V-BAD, our method requires much fewer queries. For example, we save at least 1,758 queries on HMDB-51 against I3D models. Meanwhile, we achieve better success rate 100.0% vs 97.70% on SthSth-V2.

Finally, we show the visualizations of adversarial frames on Kinetics-400 and UCF-101 in Fig. 3. We note that the generated video has little difference from the original one but can lead to a failed prediction. More visualization can be found in the supplementary materials.

### 4.3 Targeted Attack

In this section, we report the results of targeted attack on dataset HMDB-51 and SthSth-V2 in Fig. 4. For dataset SthSth-V2 from Fig. 4(a), our method consumes less than 25,000 queries using either motion vector or optical flow. However, it costs V-BAD 71,791 against I3D model and 52,182 against TSN2D model. The success rate is about 6% higher in TSN2D model but with much fewer queries. For dataset HMDB-51 against I3D from Fig. 4(b), we also outperform V-BAD by saving more than 10,000 queries and achieve comparable success rate. For TSN2D, we only require half of the queries as V-BAD consumes but achieve a much higher success rate meanwhile, i.e., 92.16% vs 64.86%.

Combining with the untargeted results, we conclude that our method is more effective in generating adversarial videos than the comparing baselines.

### 4.4 Ablation Study

In this section, we first show the necessity of motion maps and then demonstrate that it is the movement pattern in the motion map that contributes to the attacking. We also study the effect of different losses. Experiments in this section

**Table 3.** Compare to cases without introducing motion information.

| Dataset / Model | Method | I3D | | TSN2D | |
|---|---|---|---|---|---|
| | | ANQ | SR(%) | ANQ | SR(%) |
| Kinetics-400 | Multi-noise | 11,416 | 95.00 | 15,966 | 89.87 |
| | One-noise | 8,258 | 96.25 | 8,392 | 96.25 |
| | Ours | **3,089** | **100.0** | **2,494** | **100.0** |
| UCF-101 | Multi-noise | 15,798 | 90.00 | 30,337 | 70.00 |
| | One-noise | 22,908 | 93.33 | 16,620 | 90.00 |
| | Ours | **6,876** | **100.0** | **8,399** | **100.0** |

**Table 4.** Comparison of motion map with two handcrafted maps.

| Dataset / Model | Method | I3D | | TSN2D | |
|---|---|---|---|---|---|
| | | ANQ | SR(%) | ANQ | SR(%) |
| Kinetics-400 | U-Sample | 10,250 | 96.25 | 9,166 | 96.20 |
| | S-Vaule | 8,610 | 98.75 | 8,429 | 96.20 |
| | Our | **3,089** | **100.0** | **2,494** | **100.0** |
| UCF-101 | U-Sample | 13,773 | 93.33 | 17,718 | 83.33 |
| | S-Vaule | 11,471 | 96.67 | 17,116 | 86.67 |
| | Our | **6,876** | **100.0** | **8,399** | **100.0** |

are conducted on a subset of 30 randomly selected categories from UCF-101 and 80 from Kinetics-400 by following the setting in [17].

**The necessity of motion maps.** As mentioned in Section 1, we show motion is indeed important for the attack and evaluate two cases without using motion maps: 1) Multi-noise: Directly introducing random noise for each frame independently; 2) One-noise: Introducing only one random noise and replicated to all frames. The results are in Table 3. The results show that methods without using motion are likely to spend more queries and suffer a lower success rate. For example, on UCF-101 against I3D model, 'Multi-noise' consumes queries more than twice as our result and the success rate is 10% lower. Such big gaps between methods without motion and ours indicate that our designed mechanism to utilize motion maps plays an important role in directing effective gradient generation for an improved search of adversarial videos.

**Why motion maps helps?** Here, we further replace the motion map with two handcrafted maps to reveal that the intrinsic movement pattern in a motion map matters. We show that without the correct movement pattern, the attacking performance drops significantly even using the same operation in Eq. 4.

We first define a binary map $\mathcal{R}$ whose pixel values are 1 when the corresponding pixels in original motion map are nonzero, the rest pixel values are set as 0 and then define the two new maps here. "Uniformly Sample" (U-Sample): A map $\mathcal{U}$ is created whose pixel values are uniformly sampled from [0, 1] and

**Table 5.** Comparison of losses based on Cross-Entropy, Probability, Logits.

| Dataset / Model | Method | I3D | | TSN2D | |
|---|---|---|---|---|---|
| | | ANQ | SR(%) | ANQ | SR(%) |
| Kinetics-400 | Cross-Entropy | 3,452 | 98.75 | 2,248 | 100.0 |
| | Probability | 3,089 | 100.0 | 2,494 | 100.0 |
| | Logits | **2,423** | **100.0** | **1,780** | **100.0** |
| UCF-101 | Cross-Entropy | 17,362 | 80.00 | 17,992 | 73.26 |
| | Probability | 13,217 | 90.00 | 14,842 | 81.19 |
| | Logits | **6,876** | **100.0** | **6,182** | **100.0** |

scaled to [-50,50]. Binary map $\mathcal{R}$ and $\mathcal{U}$ are multiplied together to replace original motion map. “Sequenced Value” (S-Value): A map $\mathcal{S}$ whose pixel values are ranged in order starting from 0,1,2 from left to right and from top to bottom. Binary map $\mathcal{R}$ and $\mathcal{S}$ are multiplied together replace original motion map.

The results are in Table 4. We first notice that ‘S-Value’ slightly outperforms ‘U-Sample’. For example, on Kinetics-400 against I3D, ‘S-Value’ saves more than 1,000 queries but gets 2% higher success rate. We analyze the reason to be the gradual change of pixel values in ‘S-Value’, rather than irregular change. As for our method, on UCF-101 against TSN2D, 9,319 queries are saved and 16.77% higher success rate is obtained when compared to ‘U-sample’. Through such comparison, we conclude that the movement pattern in motion map is the key factor to improve the attacking performance.

**Comparison of different losses.** We study the effect of three different losses for optimization here: cross-entropy loss, logits-based in Section 3.5. We further transfer logits $l$ to probability $p$ by Softmax and construct probability-based loss: $L = \max(\sum_k p_k - \max_{k \neq y} p_k, 0)$, $y$ is the class with the largest probability value. From the results in Table 5, we conclude that logits-based loss always performs better while the other two are less effective. We also observe that Kinectics-400 is less restrictive to the selection of optimization loss, compared to UCF-101.

## 5 Conclusion

In this paper, we study the black-box adversarial attack on video models. We find that direct transfer of attack methods from image to video is less effective and hypothesize motion information in videos plays a big role in misleading video action recognition models. We thus propose a motion-excited sampler to generate sparked prior and obtain significantly better attack performance. We perform extensive ablation studies to reveal that movement pattern matters in attacking. We hope that our work will give a new direction to study adversarial attack on video models and some insight into the difference between videos and images.

**Acknowledgement.** This work is partially supported by ARC DP200100938. Hu Zhang (No. 201706340188) is partially supported by the Chinese Scholarship Council.

## References

1. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: 2017 IEEE Symposium on Security and Privacy (SP). pp. 39–57. IEEE (2017)
2. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6299–6308 (2017)
3. Chambolle, A.: An algorithm for total variation minimization and applications. Journal of Mathematical imaging and vision **20**(1-2), 89–97 (2004)
4. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In: European conference on computer vision (ECCV) (2018)
5. Chen, Z., Xie, L., Pang, S., He, Y., Tian, Q.: Appending adversarial frames for universal video attack. arXiv preprint arXiv:1912.04538 (2019)
6. Du, J., Zhang, H., Zhou, J.T., Yang, Y., Feng, J.: Query-efficient meta attack to deep neural networks. In: International Conference on Learning Representations (2020), https://openreview.net/forum?id=Skxd6gSYDS
7. Feichtenhofer, C., Fan, H., Malik, J., He, K.: SlowFast Networks for Video Recognition. In: International Conference on Computer Vision (ICCV) (2019)
8. Feichtenhofer, C., Pinz, A., Zisserman, A.: Convolutional Two-Stream Network Fusion for Video Action Recognition. In: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
9. Goodfellow, I., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: International Conference on Learning Representations (2015), http://arxiv.org/abs/1412.6572
10. Goyal, R., Kahou, S.E., Michalski, V., Materzynska, J., Westphal, S., Kim, H., Haenel, V., Fruend, I., Yianilos, P., Mueller-Freitag, M., et al.: The" something something" video database for learning and evaluating visual common sense. In: ICCV. vol. 1, p. 3 (2017)
11. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. Proceedings of the IEEE International Conference on Computer Vision (2017)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition (2016)
13. Ilyas, A., Engstrom, L., Athalye, A., Lin, J.: Black-box adversarial attacks with limited queries and information. In: International Conference on Machine Learning. pp. 2137–2146 (2018)
14. Ilyas, A., Engstrom, L., Madry, A.: Prior convictions: Black-box adversarial attacks with bandits and priors. arXiv preprint arXiv:1807.07978 (2018)
15. Inkawhich, N., Inkawhich, M., Chen, Y., Li, H.: Adversarial attacks for optical flow-based action recognition classifiers. arXiv preprint arXiv:1811.11875 (2018)
16. Ji, S., Xu, W., Yang, M., Yu, K.: 3d convolutional neural networks for human action recognition. IEEE transactions on pattern analysis and machine intelligence **35**(1), 221–231 (2012)
17. Jiang, L., Ma, X., Chen, S., Bailey, J., Jiang, Y.G.: Black-box adversarial attacks on video recognition models. In: Proceedings of the 27th ACM International Conference on Multimedia. pp. 864–872 (2019)
18. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al.: The kinetics human action video dataset. arXiv preprint arXiv:1705.06950 (2017)

19. Kim, D., Woo, S., Lee, J.Y., So Kweon, I.: Deep Video Inpainting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5792–5801 (2019)
20. Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T.: Hmdb: a large video database for human motion recognition. In: 2011 International Conference on Computer Vision. pp. 2556–2563. IEEE (2011)
21. Li, S., Neupane, A., Paul, S., Song, C., Krishnamurthy, S.V., Chowdhury, A.K.R., Swami, A.: Adversarial perturbations against real-time video classification systems. arXiv preprint arXiv:1807.00458 (2018)
22. Lin, J., Gan, C., Han, S.: TSM: Temporal Shift Module for Efficient Video Understanding. In: Proceedings of the IEEE International Conference on Computer Vision (2019)
23. Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z.B., Swami, A.: Practical black-box attacks against machine learning. In: Proceedings of the 2017 ACM on Asia conference on computer and communications security. pp. 506–519. ACM (2017)
24. Qiu, Z., Yao, T., Mei, T.: Learning spatio-temporal representation with pseudo-3d residual networks. In: International Conference on Computer Vision (ICCV) (2017)
25. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In: Advances in Neural Information Processing Systems (NeurIPS) (2015)
26. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Advances in neural information processing systems. pp. 568–576 (2014)
27. Soomro, K., Zamir, A.R., Shah, M.: Ucf101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402 (2012)
28. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)
29. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: Proceedings of the IEEE international conference on computer vision. pp. 4489–4497 (2015)
30. Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., Paluri, M.: A Closer Look at Spatiotemporal Convolutions for Action Recognition. In: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
31. Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Gool, L.V.: Temporal Segment Networks: Towards Good Practices for Deep Action Recognition. In: European Conference on Computer Vision (ECCV) (2016)
32. Wang, Q., Zhang, L., Bertinetto, L., Hu, W., Torr, P.H.: Fast online object tracking and segmentation: A unifying approach. In: Proceedings of the IEEE conference on computer vision and pattern recognition (2019)
33. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local Neural Networks. In: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
34. Wei, X., Zhu, J., Su, H.: Sparse adversarial perturbations for videos. arXiv preprint arXiv:1803.02536 (2018)
35. Wei, Z., Chen, J., Wei, X., Jiang, L., Chua, T.S., Zhou, F., Jiang, Y.G.: Heuristic black-box adversarial attacks on video recognition models. arXiv preprint arXiv:1911.09449 (2019)

36. Wu, C.Y., Zaheer, M., Hu, H., Manmatha, R., Smola, A.J., Krähenbühl, P.: Compressed video action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6026–6035 (2018)
37. Wu, H., Chen, Y., Wang, N., Zhang, Z.: Sequence Level Semantics Aggregation for Video Object Detection. Proceedings of the IEEE International Conference on Computer Vision (2019)
38. Yan, H., Wei, X., Li, B.: Sparse black-box video attack with reinforcement learning. arXiv preprint arXiv:2001.03754 (2020)
39. Zhu, L., Tran, D., Sevilla-Lara, L., Yang, Y., Feiszli, M., Wang, H.: Faster recurrent networks for efficient video classification. In: AAAI (2020)
40. Zhu, L., Yang, Y.: Label independent memory for semi-supervised few-shot video classification. IEEE Transactions on Pattern Analysis and Machine Intelligence (2020). https://doi.org/10.1109/TPAMI.2020.3007511
41. Zhu, Y., Lan, Z., Newsam, S., Hauptmann, A.G.: Hidden Two-Stream Convolutional Networks for Action Recognition. In: Asian Conference on Computer Vision (ACCV) (2018)