Heuristic Black-box Adversarial Attacks on Video Recognition Models

Zhipeng Wei\textsuperscript{1}, Jingjing Chen\textsuperscript{2}, Xingxing Wei\textsuperscript{3}, Linxi Jiang\textsuperscript{2}, Tat-Seng Chua\textsuperscript{4}, Fengfeng Zhou\textsuperscript{5}, Yu-Gang Jiang\textsuperscript{2}

\textsuperscript{1}Jilin University, \textsuperscript{2}Fudan University, \textsuperscript{3}Beihang University, \textsuperscript{4}National University of Singapore, \textsuperscript{5}Health Informatics Lab, College of Computer Science and Technology, and Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun, Jilin, China, 130012

weizp17@mails.jlu.edu.cn, chenjingjing@fudan.edu.cn, xxwei@buaa.edu.cn, lxjiang18@fudan.edu.cn, chuats@comp.nus.edu.sg, FengfengZhou@gmail.com, ygj@fudan.edu.cn

Abstract

We study the problem of attacking video recognition models in the black-box setting, where the model information is unknown and the adversary can only make queries to detect the predicted top-1 class and its probability. Compared with the black-box attack on images, attacking videos is more challenging as the computation cost for searching the adversarial perturbations on a video is much higher due to its high dimensionality. To overcome this challenge, we propose a heuristic black-box attack model that generates adversarial perturbations only on the selected frames and regions. More specifically, a heuristic-based algorithm is proposed to measure the importance of each frame in the video towards generating the adversarial examples. Based on the frames’ importance, the proposed algorithm heuristically searches a subset of frames where the generated adversarial example has strong adversarial attack ability while keeps the perturbations lower than the given bound. Besides, to further boost the attack efficiency, we propose to generate the perturbations only on the salient regions of key frames. In this way, the generated perturbations are sparse in both temporal and spatial dimensions. Experimental results of attacking two mainstream video recognition methods on the UCF-101 dataset and the HMDB-51 dataset demonstrate that the proposed heuristic black-box adversarial attack method can significantly reduce the computation cost and lead to more than 28\% reduction in query numbers for the untargeted attack on both datasets.

Introduction

Deep neural networks are vulnerable to adversarial samples (Goodfellow, Shlens, and Szegedy 2014; Szegedy et al. 2013). Recent works have shown that adding a small human-imperceptible perturbation to a clean sample can fool the deep models, leading them to make wrong predictions with high confidence (Moosavi-Dezfooli et al. 2017). As results, it has raised serious security concerns for the deployment of deep models in security-critical applications, such as face recognition (Kurakin, Goodfellow, and Bengio 2016), video surveillance (Sultani, Chen, and Shah 2018), etc. Therefore, it is crucial to study the adversarial examples for deep neural networks. In addition, investigating adversarial samples also helps to understand the working mechanism of deep models and provides opportunities to improve networks’ robustness.

Nevertheless, most of the existing works focus on exploring adversarial samples for image recognition models under white-box or black-box settings (Kurakin, Goodfellow, and Bengio 2016; Szegedy et al. 2013; Papernot et al. 2016; Carlini and Wagner 2017; Su, Vargas, and Sakurai 2019). Adversarial attacks on video models, especially the black-box attacks, have been seldom explored. Indeed, deep neural network based real-time video classification systems, e.g., video surveillance systems, are being increasing deploys in real-world scenarios. Therefore, it is crucial to investigating the adversarial samples for video models.

This paper studies the problem of generating adversarial samples to attack video recognition models in the black-box settings, where the model is not revealed and we can only make queries to detect the predicted top-1 class and its probability. The major obstacle for solving this problem is how to reduce the number of queries to improve attack efficiency. Compare to statistic images, the dimensionality of video data is much higher as videos have both temporal and spatial dimensions. As results, directly extending the existing black-box attack methods proposed for image models to video models will consume much more queries for gradient estimation, leading to the low attack efficiency.

To this end, we propose an efficient black-box attack method specially for video models in this paper. Considering the fact that video data contains both temporal and spatial redundancy, it is therefore not necessary to search the adversarial perturbations for all the frames (Wei, Zhu, and Su 2019). Besides, for a given video, different frames and different regions in each frame contribute differently towards video classification results (Peng, Zhao, and Zhang 2018). Intuitively, the key frames which contain the key evidence of one specific event, or the salient regions in the key frame (usually the foreground regions), play a vital role for the video classification results. As results, generating adversarial perturbations on the salient regions of key frames is the most effective way to attack video models. Motivated by this, we propose a heuristic-based method to select the key frames for adversarial samples generation.
1 Overview the proposed method. In the proposed method, the perturbation directions are initialized with the samples from other classes different from the class of the clean video. Meanwhile, it heuristically searches a subset of frames from the clean video with the initialized directions, and the salient regions detected from the selected frames are utilized to update the initial directions. The initial directions as well as the selected key frames are updated iteratively to make the number of selected frames as small as possible while keeps the perturbation noise lower than the given bound. Then the initial direction which has the minimum distance to the decision boundary is chosen as the final initial direction and zeroth order optimization (Chen et al. 2017) is next utilized for direction updating. As the perturbations are generated only for the salient regions of the selected frames, the proposed method significantly reduces the number of queries.

Our major contributions can be summarized as follows:

• We propose an efficient black-box adversarial attack model that heuristically select a subset consisting of the key frames to generate adversarial perturbations.

• We introduce saliency detection in the black-box adversarial attack model to generate adversarial perturbations only on the salience regions of the key frames, which further reduces query numbers. As far as we know, this is the first attempt to introduce prior knowledge for black-box video attack.

• Extensive experiments on two benchmark data sets demonstrate that the proposed method is efficient and effective. It reduces more than 28% reduction in query numbers for the untargeted attack.

Related Work

Adversarial Attack on Image Models

Recent works on adversarial attack are mostly focus on image models, including both white-box (Szegedy et al. 2013; Goodfellow, Shlens, and Szegedy 2014; Kurakin, Goodfellow, and Bengio 2016; Dong et al. 2018; Carlini and Wagner 2017; Moosavi-Dezfooli, Fawzi, and Frossard 2016; Papernot et al. 2016; Sarkar et al. 2017) and black-box attacks (Chen et al. 2017; Tu et al. 2018; Cheng et al. 2018; Brendel, Rauber, and Bethge 2017; Papernot et al. 2017; Liu et al. 2016). This section reviews recent works on generating adversarial samples for image models.

White-box Attack

White-box attack assumes that the structure as well as the parameters of the targeted model are known to the attacker. In this setting, attacks can be done easily as the gradient of attack objective function can be computed via backpropagation. In recent years, different white box attack methods have been proposed. For example, L-BFGS attack (Szegedy et al. 2013) first crafts adversarial examples against deep neural networks towards maximizing the network’s prediction error, and demonstrates the transferability of adversarial examples. Fast Gradient Sign Method (FGSM) (Goodfellow, Shlens, and Szegedy 2014) updates gradient once along the direction of the sign of gradient at each pixel. FGSM can be applied in multiple times with small step size for better adversarial examples in (Kurakin, Goodfellow, and Bengio 2016), and also be combined with momentum to increase adversarial intensity (Dong et al. 2018). C&W’s Attack (Carlini and Wagner 2017) generates the targeted adversarial examples by defining effective objective functions, and provides three types of attacks: $l_0$ attack, $l_2$ attack, and $l_{\infty}$ attack. Other algorithms such as DeepFool (Moosavi-Dezfooli, Fawzi, and Frossard 2016), Jacobian-based Saliency Map Attack (JSMA) (Papernot et al. 2016), UPSET and ANGRI (Sarkar et al. 2017), etc, have carried out white box attacks from different aspects.

Black-box Attack

Compare with the white-box attack, the black-box attack is a more realistic but more challenging setting as the model information remains unknown to the attacker. Existing efforts on black-box attack for image models include Zeroth Order Optimization (ZOO) (Chen et al. 2017), Autoencoder-based Zeroth Order Optimization Method (AutoZOOM) (Tu et al. 2018), decision based black-box attack (Brendel, Rauber, and Bethge 2017), Opt-attack (Cheng et al. 2018). Zeroth Order Optimization (ZOO) (Chen et al. 2017) uses ADAM’s update rule or New-
Adversarial Attack on Video Models

There is much less related work on generating adversarial samples to attack video models compared with adversarial attacks on image models. The first white-box attack method for video recognition models is proposed in (Wei, Zhu, and Su 2019), where an $l_2$-1-norm regularization based optimization algorithm is proposed to compute the sparse adversarial perturbations for videos. Different from (Wei, Zhu, and Su 2019), (Li et al. 2018) took advantages of Generative Adversarial Networks (GANs) to do a white-box attack. Their attack method generates the universal perturbation offline and works with unseen input for the real-time video recognition model. Nevertheless, the above mentioned methods assume the complete knowledge of the video recognition models which is different from our settings. Another work is (Inkawichay et al. 2019) which proposes an untargeted adversarial attack that based on FGSM (Goodfellow, Shlens, and Szegedy 2015) and iterative FGSM (Kurakin, Goodfellow, and Bengio 2017) for flow-based video recognition models in both white-box and black-box settings. The proposed white-box attack leverages Flownet2 and chain rule to obtain the gradients through the whole video recognition model, where Flownet2 is required to estimate optical flow frames while also providing gradient information easily. They used the transferability of adversarial examples generated by the white-box model to implement the black-box attack. Different to these two works which generate the adversarial perturbations for all the frames in the video, our work heuristically searches a subset of frames and the adversarial perturbations are only generated on the salient regions of the selected frames, which effectively reduce the number of queries.

Methodology

In this section, we introduce the proposed heuristic black-box attack algorithm for video recognition models. We assume that the predicted top-1 class and its probability are known to an attacker.

Denote the video recognition model as a function $F$. Specifically, the DNN $F(x)$ takes a clean video $x \in R^{C \times W \times H \times T}$ as an input and output the top-1 class $\hat{y}$ and its probability $P(\hat{y} | x)$, where $T$, $W$, $H$, $C$ denote the number of frames, width, height, and the number of channels respectively. It suffices to denote its associated true class label by $y \in Y = \{1, \cdots, K\}$, where $K$ is the number of classes, and its adversarial example by $x_{adv}$. In the untargeted attack, the goal is to make $F(x_{adv}) \neq \hat{y}$, while for the targeted attack with the targeted adversarial class $y_{adv}$, the goal is to satisfy $F(x_{adv}) = y_{adv}$.

Our attack algorithm is built based on the Opt-attack (Cheng et al. 2018), which is originally proposed for attacking image models by formulating the hard-label black-box attack as a real-valued optimization problem that can be solved by zeroth order optimization algorithms. We extend Opt-attack (Cheng et al. 2018) from image models to video models. Specifically, to deal with the high dimensional video data and improve the attack efficiency, we propose a heuristic algorithm to select a subset of video frames and only search the adversarial perturbations for the selected frames. Following the Opt-attack, we denote $\theta$ as the search direction of the video. The function of the distance from $x$ to the decision boundary along the direction $\theta$ is denoted as $g(\theta)$, which is calculated by a fine-grained search and a binary search function defined in (Cheng et al. 2018). The objective of Opt-attack’s is to find the direction that minimizes $g(\theta)$ by zeroth order optimization proposed in (Cheng et al. 2018). In Opt-attack algorithm, the optimization process can be divided into two parts: direction initialization and direction updating. The direction initialization is done by the function $\theta = x + \frac{p}{\|p\|}$, where $p = \hat{x} - x$, and $\hat{x}$ denotes the video from other classes; Then update $\theta$ with the zeroth order optimization and $g(\theta)$. Finally, we find the adversarial example by $x_{adv} = x + g(\theta^*) \times \theta^*$, where $\theta^*$ is the optimal solution of the zeroth order optimization.

Here, we focus on the direction initialization part, and propose heuristic temporal and spatial selection methods in direction initialization for generating barely noticeable adversarial videos that result in misclassification of DNN $F$. Specifically, we attempt to find the sparse $\theta$ by modifying $p$ with the equation $p = p \times M$, where $M \in \{0,1\}^{T \times W \times H \times C}$ is the mask introducing temporal and spatial sparsity. For each element in $M$, if its value equals 1, the corresponding pixel in the video will be included in the process of perturbation computation. The selected regions with $M$ are named as masked regions. During the adversarial example generation, the computed perturbations are only added to the masked regions. Next, we describe in detail the proposed heuristic sparse methods.

It worth to mention that it has been demonstrated in Uesato et al., the type of gradient-free optimizer has relatively small influence towards the attack efficiency. Hence, we use the zero-order optimization here. The estimated gradient is defined as

$$\hat{g} = \frac{g(\theta + \beta u) - g(\theta)}{\beta} \cdot u \quad (1)$$

where $u$ is a random Gaussian vector of the same size as $\theta$, and $\beta > 0$ is a smoothing parameter which will reduce by 10 times if the estimated gradients can’t provide useful information for updating $\theta$. Following (Cheng et al. 2018), we set $\beta = 0.005$ in all experiments. Besides, we sample...
u from Gaussian distribution for 20 times to calculate their estimators and average them to get more stable $\hat{g}$. In each iteration, $\theta$ is updated by

$$\theta \leftarrow \theta - \eta \hat{g}$$

where $\eta$ is the step size, which is adjusted at each iteration by a backtracking line-search approach.

### Heuristic Temporal Sparsity

Videos have successive frames in the temporal domain, thus, we consider to search a subset of frames that contributes the most to the success of an adversarial attack. Here we introduce the concept of temporal sparsity which refers to the adversarial perturbations generated only on the selected key frames during the direction initialization process.

In order to achieve temporal sparsity, we first propose a heuristic-based algorithm to evaluate the importance of each frame. Given a frame $t$, $M_t$ means that all frames except the $t$-th frame are equal to 1. Based on $F$, we can get top-1 class label and its probability by $\hat{y}_t$, $P(\hat{y}_t|(p \times M_t + x)) = F(p \times M_t + x), \forall t \in \{1, \cdots, T\}$. We then sort the sequence of video frames according to the descending order of $P(\hat{y}_t|(p \times M_t + x))$ under the condition that $\hat{y}_t$ is an adversarial label. Note that, the larger value of $P(\hat{y}_t|(p \times M_t + x))$ means the less importance of $t$-th frame towards generating the adversarial sample.

We search a set of key frames based on the sorted video frame sequence. This searching process is performed during the direction initialization for all randomly selected $\hat{x}$. Denote $\omega$ as the bound of mean absolute perturbation of each pixel in videos that determines when to stop toward searching a smaller set of key frames. When the mean absolute perturbation on the selected key frames lower than the bound $\omega$, our method will continue searching a smaller set of key frames for adversarial perturbation generation. The value of $\omega$ is selected according to the experiments on validation set, which we will discuss in the experimental part. Algorithm 1 summarizes the whole procedure of searching the key frames for the targeted attack. In Algorithm 1, DELFRAME sets the values of $i$-th frame in $M$ to 0; SORTED sorts the indexes of frames in descending order by the associated top-1 probabilities; LENS calculates the number of selected key frames; and MAP computes the mean absolute perturbation for $g(\theta) \times \theta$.

### Heuristic Spatial Sparsity

Intuitively, salient regions, for example, the foreground of the frames, contribute more to the video classification results. Generating adversarial perturbations on the salient regions will be more likely to fool the deep models. Therefore, we introduce the saliency maps of video frames as prior knowledge for the generation of adversarial perturbations. As the perturbation are only generated for salient regions, it hence introduces spatial sparsity. To generate the saliency map for each frame, an efficient saliency detection approach proposed in (Hou and Zhang 2007) and implement by OpenCV (Bradski 2000) is applied. To control the area ratio of the salient region in the frame, we introduce a parameter $\varphi \in (0, 1]$, and smaller $\varphi$ leads to smaller portion of salient regions. In the untarget attack, we initialize the mask $M$ to introduce spatial sparsity for all video frames based on $\varphi$. In the targeted attack, we also use the heuristic-based algorithm to obtain the descending order sequence of frames. After that, we only add spatial sparsity on the selected frame in $M$. In $M$, the pixels of salient regions is as while the rest is set 0. We investigate different values of $\varphi$ in Section 10.

### Overall Framework

The whole process of the our method for the targeted attack is describe in Algorithm 2, where SPATIAL function performs a saliency detection with $\varphi$ for each frame, and initializes $M$ by setting the values of the salient regions to 1, the others to 0; UPDATE represents the zeroth order optimization to update $\theta$, . In the overall framework, we firstly initialize $M$ by SPATIAL to implement spatial sparsity, then

### Algorithm 1: Heuristic temporal selection algorithm for the targeted attack.

<table>
<thead>
<tr>
<th>Input</th>
<th>DNN $F$, clean video $x$, true label $y$, target class $y_{adv}$, initial mask $M \in {1}^{T \times W \times H \times C}$, an empty array $A$.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output</td>
<td>Mask of key frames $M$.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Bound $\omega$.</td>
</tr>
</tbody>
</table>

1. $\hat{x}$ $\leftarrow$ a video sample of target class $y_{adv}$;
2. $p, k \leftarrow \hat{x} - x, 0$;
3. for $t \leftarrow 1$ to $T$ do
4. $M_t \leftarrow$ DELFRAME($M, t$); // the values of $i$-th frame are equal to 0.
5. $\hat{y}_t, P(\hat{y}_t|(p \times M_t + x)) \leftarrow F(p \times M_t + x)$;
6. if $\hat{y} = y_{adv}$ then
7. $A[k], k \leftarrow (t, P(\hat{y}|(p \times M_t + x))), k + 1$;
8. end
9. $A \leftarrow$ SORTED($A$); // indexes of frames are sorted in descending order by $P(\hat{y}|(p \times M_t + x))$.
10. $\theta_{init} \leftarrow \frac{p}{\|p\|}$ for $i \leftarrow 1$ to $k$ do
11. $\hat{M} \leftarrow$ DELFRAME($M, A[i]$) ;
12. $\hat{p} \leftarrow p \times \hat{M}$;
13. $\theta \leftarrow \frac{\hat{p}}{\|\hat{p}\|}$;
14. $\hat{y}_t, P(\hat{y}((x + \hat{p})) \leftarrow F(x + \hat{p})$;
15. if $\hat{y} = y_{adv}$ then
16. if MAP$(g(\theta) \times \theta) \leq \omega$ then
17. if LENS($\hat{M}$) $<$ LENS($M$) then // the number of key frames.
18. $M, \theta_{init} \leftarrow \hat{M}, \theta$;
19. else
20. if MAP$(g(\theta) \times \theta) < MAP(g(\theta_{init}) \times \theta_{init})$ then
21. $M, \theta_{init} \leftarrow \hat{M}, \theta$;
22. end
23. end
24. return $M$
Algorithm 2: Heuristic-based targeted attack algorithm.

Input: DNN $F$, clean video $x$, true label $y$, target class $y_{adv}$, an empty array $A$
Output: Adversarial example $x_{adv}$.
Parameters: $\omega, \varphi$, the number of update iterations $I$.

1. $M \leftarrow$ SPATIAL$(x, \varphi)$;
2. $M \leftarrow$ Algorithm 1$(F, x, y, y_{adv}, M, A, \omega)$;
3. $\theta = \frac{\omega}{\theta M}$;
4. $\theta = \theta \times M$;
5. for $t \leftarrow 1$ to $I$ do
6. $g = g(\theta + \beta \cdot \nabla_{\theta} F(x; y, y_{adv}, M))$;
7. $\theta = \theta - \eta g$;
8. end
9. $x_{adv} = x + g(\theta) \times \theta$;
10. return $x_{adv}$

combine the spatial sparsity with the temporal sparsity by Algorithm 1 to get the final $M$. After that, we obtain the temporal and spatial sparse $\theta$, and use UPDATE to update $\theta$ iteratively. Finally, the adversarial example can be found by the optimal direction.

**Experiments**

In this section, we test the performance of our heuristic based black-box attack algorithm with various parameters $\omega$ and $\varphi$ in reducing overall perturbations and the number of queries. Furthermore, we show a comprehensive evaluation of our method on multiple video recognition models.

**Experimental Setting**

**Datasets.** We consider two widely used datasets for video recognition: UCF-101 (Su et al. 2009) and HMDB-51 (Kuehne et al. 2011). UCF-101 is an action recognition dataset that 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. During the evaluation, we use 16-frame snippets that uniform sampled from each video as input samples of target models.

**Metrics.** Four 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) Median queries (MQ) (Ilyas et al. 2018): the median number of queries. 3) Mean absolute perturbation (MAP): denotes the mean perturbation of each pixel in the entire video. We use MAP to denote the MAP of each pixel in the masked region. 4) Sparsity (S): represents the proportion of pixels with no perturbations versus all pixels in a specific video. $S = 1 - \frac{1}{T} \sum_{t=1}^{T} \varphi_t$, where $\varphi$ is the area of the selected salient region in the corresponding frame, $\varphi_t$ is $\varphi$ of the $t$-th frame and $T$ is the total number of frames.

<table>
<thead>
<tr>
<th>Model</th>
<th>UCF-101</th>
<th>HMDB-51</th>
</tr>
</thead>
<tbody>
<tr>
<td>C3D</td>
<td>85.88</td>
<td>59.57</td>
</tr>
<tr>
<td>LRCN</td>
<td>64.92</td>
<td>37.42</td>
</tr>
</tbody>
</table>

**Table 1: Test Accuracy(%) of the target models.**

**Threat Model** We use the special case of the Partial-information setting (Ilyas et al. 2018). In this setting, the attacker only has access to the top-1 class $\hat{y}$ and its probability $P(\hat{y}|x)$, given a video $x$. Both untargeted and targeted black-box attacks are considered. In the experiments, two video recognition models, Long-term Recurrent Convolutional Networks (LRCN) (Donahue et al.) and C3D (Hara, Kataoka, and Satoh 2018) are used as target models. LRCN (Donahue et al.) model uses Recurrent Neural Networks (RNNs) to encode the temporal information and long-range dependencies on the features generated by CNNs. In our implementation, Inception V3 (Szegedy et al. 2016) is utilized to extract features for video frames and LSTM is utilized for video classification; C3D model (Hara, Kataoka, and Satoh 2018) applies 3D convolution to learn spatio-temporal features from videos with spatio-temporal filters for video classification. These models are the main methods of video classification. Table 1 summarizes the test accuracy of 16-frame snippets with these two models.

**Parameter Setting**

Note that we have two parameters need to set in Algorithm 2, one is the perturbations bound $\omega$ and the other one is the area ratio of salient regions $\varphi$. The parameter tuning is done on 30 videos that randomly sampled from the test set of UCF-101 and can be correctly classified by the target models. We do a grid search to find the most appropriate values for these two parameters. For $\omega$, we set it as $\{0, 3, 6, 9, 12, 15, \infty\}$ in the untargeted attack, as $\{0, 15, 30, 45, \infty\}$ in the targeted attack since the targeted attack has larger perturbations than the untargeted attack, and evaluate the attack performance on C3D model. Table 2 shows the performance on the untargeted attack. Basically, large $\omega$ leads to sparse perturbations When $\omega$ is set to $\infty$, the sparsity value can be as low as 83.75%, which means that only $2.6 \times (1 - 83.75\%) \approx 2.6$ video frames will be selected to generate the adversarial perturbations. Therefore, to strikes a balance between the MAP and temporal sparsity, we set $\omega = 3$ in the untargeted attack to conduct subsequent experiments. Table 4 lists the results on the targeted attack. The values of $\omega$ and $MQ$ is greater than those in the untargeted attack since the targeted attack has larger perturbations than the untargeted attack. We set $\omega = 30$ in experiments of the targeted attack.

Similarly, we perform grid search to decide the value of $\varphi$. For the untargeted attack, we fix $\omega = 3$ and set parameter $\varphi$ as $\{0.2, 0.4, 0.6, 0.8, 1.0\}$ to evaluate the performance. Table 3 lists the results. Note that, $\varphi = 1$ means that all the regions in the frame are selected. As can be seen, when the value of $\varphi$ is small, the median query number (MQ), mean absolute perturbations (MAP) will be reduced, while the mean absolute perturbations (MAP) on the selected regions will be increased. When $\varphi = 0.2$, the sparsity value can be as high
Table 2: Results of our algorithm with various $\omega$ in the untargeted attack.

<table>
<thead>
<tr>
<th>$\omega$</th>
<th>$FR(%)$</th>
<th>$MQ$</th>
<th>$MAP$</th>
<th>$MAP^*$</th>
<th>$S(%)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>100</td>
<td>16085.0</td>
<td>3.7033</td>
<td>3.8449</td>
<td>17.69</td>
</tr>
<tr>
<td>3</td>
<td>100</td>
<td>16085.0</td>
<td>3.6858</td>
<td>3.9667</td>
<td>25.19</td>
</tr>
<tr>
<td>6</td>
<td>100</td>
<td>15996.0</td>
<td>3.7471</td>
<td>4.0328</td>
<td>23.94</td>
</tr>
<tr>
<td>9</td>
<td>100</td>
<td>17527.0</td>
<td>3.7757</td>
<td>4.2862</td>
<td>22.58</td>
</tr>
<tr>
<td>12</td>
<td>100</td>
<td>191216.0</td>
<td>10.7961</td>
<td>13.4766</td>
<td>22.58</td>
</tr>
<tr>
<td>15</td>
<td>100</td>
<td>302230.50</td>
<td>9.7547</td>
<td>10.5442</td>
<td>8.54</td>
</tr>
<tr>
<td>$\infty$</td>
<td>100</td>
<td>16085.0</td>
<td>3.7033</td>
<td>3.8449</td>
<td>17.69</td>
</tr>
</tbody>
</table>

Table 3: Results of our algorithm with various $\varphi$ in the untargeted attack.

<table>
<thead>
<tr>
<th>$\varphi$</th>
<th>$FR(%)$</th>
<th>$MQ$</th>
<th>$MAP$</th>
<th>$MAP^*$</th>
<th>$S(%)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2</td>
<td>90</td>
<td>8770.0</td>
<td>1.5890</td>
<td>3.7153</td>
<td>85.00</td>
</tr>
<tr>
<td>0.4</td>
<td>100</td>
<td>12336.0</td>
<td>2.6273</td>
<td>3.0203</td>
<td>68.84</td>
</tr>
<tr>
<td>0.6</td>
<td>100</td>
<td>14125.0</td>
<td>3.2194</td>
<td>5.7604</td>
<td>54.25</td>
</tr>
<tr>
<td>0.8</td>
<td>100</td>
<td>13845.0</td>
<td>3.4507</td>
<td>4.6347</td>
<td>40.33</td>
</tr>
<tr>
<td>1.0</td>
<td>100</td>
<td>16085.0</td>
<td>3.6858</td>
<td>3.9667</td>
<td>25.19</td>
</tr>
</tbody>
</table>

as 85.00%, which results in a large $MAP^*$ and lower fooling rate (FP). To balance the metrics $MAP^*$, $MQ$ and $MAP$, we set $\varphi$ as 0.6 for the untargeted attack. When $\varphi = 0.6$, it reduces the number of queries by 22.17% and the overall perturbations (MAP) by 25.79%. The results also suggest that the incursion of spatial sparsity brings to our method not just a reduction in the number of queries but a reduction in MAP. Table 4 lists the results for the targeted attack. Similarly, in order to balance the metrics $MAP^*$, $MQ$ and $MAP$, we set $\varphi$ as 0.8 for the targeted attack.

### Performance Comparison

We compare our method with the Opt-attack (Cheng et al. 2018) which originally proposed to attack image classification models under black-box setting. We directly extend Opt-attack to attack video models for comparison. Besides, we also compare with one variant of our method by removing the spatial sparsity module. In this setting, only temporal sparsity is considered. The evaluations are performed with two video recognition models on two datasets.

Table 6 lists the performance comparisons regarding to the untargeted attack as well the targeted attack on UCF-101 and HMDB-51 dataset. For untargeted attack, we have the following observations. First, compared to Opt-attack, our method that considers both temporal and spatial sparsity significantly reduces the number of queries. For LRCN and C3D model, the number of queries has been reduced by more than 28% on both datasets. Second, compared to introducing temporal sparsity, introducing spatial sparsity is more effective in reducing the query numbers. For example, for C3D, introducing temporal sparsity alone helps to reduce the query numbers around 9% while introducing both spatial and temporal sparsity reduce the query numbers for more than 28% on UCF-101. By looking into the results, we found that around 12% of the queries are spent on searching a set of key frames that maximize the sparsity value while keeping the mean absolute perturbations (MAP) lower than the given bound. Third, on most cases, introducing temporal and spatial sparsity helps to make the adversarial perturbations much sparser and hence reduce the mean absolute perturbations (MAP). In summary, introducing temporal and spatial sparsity increases the query efficiency and helps to achieve human-imperceptible perturbations.

For the targeted attack, similar trends are observed as the untargeted attack. By introducing both spatial and temporal sparsity, our method significantly reduces the query numbers. On UCF-101 dataset, the query numbers can be reduced from 445,279 to 399,655 for LRCN model. For C3D model, the query numbers have been reduced by more than 19.59% on both datasets. Although spatial sparsity significantly reduces the number of queries, it increases $MQ$, $MAP$ and $MAP^*$ a large margin on the targeted attack. Compared to the untargeted attack, $MQ$, $MAP$ as well $MAP^*$ on the targeted attack are much higher. One major reason is that compared to the untargeted attack, the target attack are more difficult. Besides, to make sure that it achieves 100% of the fooling rate, the MAP as well $MAP^*$ increased significantly when introducing the spatial sparsity. The results basically suggest that for the targeted attack, the selected saliency regions from the clean video frames may not contribute much to the classification of targeted classes, hence it increases the perturbations in order to successfully fool the recognition model, the generated sparse directions cause the targeted attack to fall into local optimal solutions and stop the direction updating early. In addition, introducing temporal sparsity increases the query numbers because 39.75% of the queries are spent on searching a small set of keyframes in order to achieve temporal sparsity.

Figure 2 further shows four examples of adversarial frames generated by the proposed method. For all the examples, our method successfully fools the recognition model. For example, in the first example, the ground-truth label for the video is “Apply eye makeup”, by adding the generated human-imperceptible adversarial perturbations (sec-
Table 6: Untargeted and targeted attacks against C3D/LRCN Models. For all attack models, the Fooling Rate (FR) is 100%.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Target Model</th>
<th>Attack Model Setting/Metric</th>
<th>Untargeted attacks</th>
<th>Targeted attacks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>C3D</td>
<td>Opt-attack (Cheng et al. 2018)</td>
<td>MQ MAP MAP ∗ S(%)</td>
<td>MQ MAP MAP ∗ S(%)</td>
</tr>
<tr>
<td>UCF-101</td>
<td></td>
<td>Our (Temp.)</td>
<td>17997.5 4.2540 2.540 0.00</td>
<td>207944.5 9.0906 9.0906 0.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Our (Temp. + Spat.)</td>
<td>16292.0 4.0895 4.364 21.19</td>
<td>313229.0 7.8069 10.4700 28.00</td>
</tr>
<tr>
<td></td>
<td>LRCN</td>
<td>Opt-attack (Cheng et al. 2018)</td>
<td>12359.5 1.8320 1.8320 0.00</td>
<td>445279.0 13.4795 13.4795 0.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Our (Temp.)</td>
<td>14713.5 1.8754 1.8794 17.19</td>
<td>167217.0 10.8958 15.4904 34.28</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Our (Temp. + Spat.)</td>
<td>8421.5 1.8383 3.0848 47.50</td>
<td>396550.5 11.2066 19.8620 46.92</td>
</tr>
<tr>
<td>HMDB-51</td>
<td>C3D</td>
<td>Opt-attack (Cheng et al. 2018)</td>
<td>14509.5 2.8930 2.8930 0.00</td>
<td>205286.5 6.5704 6.5704 0.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Our (Temp.)</td>
<td>13536.5 2.9214 3.2010 26.94</td>
<td>196371.5 8.3599 10.6761 21.88</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Our (Temp. + Spat.)</td>
<td>10616.0 2.3765 4.4574 57.04</td>
<td>144917.5 9.6109 12.2993 28.70</td>
</tr>
<tr>
<td></td>
<td>LRCN</td>
<td>Opt-attack (Cheng et al. 2018)</td>
<td>18655.0 2.7586 2.7586 0.00</td>
<td>224414.0 3.8598 3.8598 0.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Our (Temp.)</td>
<td>15369.5 2.8011 2.8923 24.22</td>
<td>339367.0 4.0618 5.5601 28.75</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Our (Temp. + Spat.)</td>
<td>13311.5 1.5390 2.8302 62.03</td>
<td>206120.0 12.7966 18.1835 42.87</td>
</tr>
</tbody>
</table>

Figure 2: Examples of adversarial frames generated in the untargeted attack by our method. The clean frames are show in the top row; the corresponding adversarial frames are in the middle row and the perturbations are show in the bottom row. The perturbations are visualized by re-scaling them into the range of 0-255.

Conclusion

In this paper, we proposed a heuristic black-box adversarial attack algorithm for video recognition models. To reduce query numbers and improve attack efficiency, our method explores the sparsity of adversarial perturbations in both temporal and spatial domains. Our algorithm is adaptive to multiple target models and video datasets and enjoys global sparsity and query efficiency improvement. Moreover, the experimental results demonstrate that video recognition models are vulnerable to adversarial attack, and our algorithm achieves small human-imperceptible perturbation using fewer queries. The most pertinent area of future work is to further investigate the black-box attack for the targeted attack using fewer queries.

Acknowledgments

The work was funded by the National Research Foundation, Prime Ministers Office, Singapore under its IRC@Singapore Funding Initiative, and the NSFC Projects (No.61806109). The work was also funded by the Jilin Provincial Key Laboratory of Big Data Intelligent Computing (20180622002JC), the Education Department of Jilin Province (JJKH20180145KJ), and the startup grant of the Jilin University, the Bioknow MedAI Institute (BMCPP-2018-001), the High Performance Computing Center of Jilin University, and the Fundamental Research Funds for the Central Universities, JLU.
References


