Efficient Robustness Assessment via Adversarial Spatial-Temporal Focus on Videos

Xingxing Wei, *Member, IEEE*, Songping Wang, and Huanqian Yan

***Abstract*—Adversarial robustness assessment for video recognition models has raised concerns owing to their wide applications on safety-critical tasks. Compared with images, videos have much high dimension, which brings huge computational costs when generating adversarial videos. This is especially serious for the query-based black-box attacks where gradient estimation for the threat models is usually utilized, and high dimensions will lead to a large number of queries. To mitigate this issue, we propose to *simultaneously* eliminate the temporal and spatial redundancy within the video to achieve an effective and efficient gradient estimation on the reduced searching space, and thus query number could decrease. To implement this idea, we design the novel Adversarial spatial-temporal *Focus* (*AstFocus*) attack on videos, which performs attacks on the simultaneously focused key frames and key regions from the inter-frames and intra-frames in the video. AstFocus attack is based on the cooperative Multi-Agent Reinforcement Learning (MARL) framework. One agent is responsible for selecting key frames, and another agent is responsible for selecting key regions. These two agents are jointly trained by the common rewards received from the black-box threat models to perform a cooperative prediction. By continuously querying, the reduced searching space composed of key frames and key regions is becoming precise, and the whole query number becomes less than that on the original video. Extensive experiments on four mainstream video recognition models and three widely used action recognition datasets demonstrate that the proposed AstFocus attack outperforms the SOTA methods, which is prevenient in fooling rate, query number, time, and perturbation magnitude at the same time.**

***Index Terms*—Adversarial examples, black-box attacks, reinforcement learning, spatial-temporal analysis, video recognition.**

I. Introduction

DEEP Neural Networks (DNNs) have made remarkable achievements in various tasks such as object detection [1], action recognition [2], scene understanding [3], and so on. Recent studies illustrate the DNNs' vulnerability to the so-called adversarial examples [4], [5], [6]. Afterwards, a series of methods are proposed to evaluate the adversarial robustness of DNNs. Among these works, the attack-based robustness evaluation methods [7], [8], [9] are more popular and practical because of their good implementability. They mainly seek for the minimum adversarial perturbations of successful attacks to measure the robustness [10]. On one hand, accurate assessment for adversarial robustness can help to deploy DNNs into safety-critical systems. On the other hand, it provides a quantitative metric to design more robust DNNs. Therefore, adversarial robustness assessment is important in both theoretical and practical values.

Video recognition [11], [12], [13] is a major branch in computer vision. Leveraging the temporal and spatial relationship within the video data can effectively locate and classify the objects or behaviors in videos, and thus help to perform video analysis. Owing to the DNNs' advantage, current video recognition models are usually designed based on DNNs. The DNNs' vulnerability is inevitably inherited by video recognition models. Owing to the wide applications in some safety-critical tasks like security surveillance, evaluating their adversarial robustness becomes necessary. Currently, more and more users begin to employ the video recognition APIs released by commercial cloud platforms because of their easy accessibility. In such cases, the APIs' details are not public, we can only assess their adversarial robustness according to the outputs obtained by querying the systems. So these methods are called as query-based black-box attacks, which mainly rely on the estimated gradients for the APIs [14], [15].

Compared with images, videos have much high dimensions owing to the additional temporal information, which brings huge computational costs when generating adversarial videos. This is especially serious for the query-based black-box attacks because the high-dimension video data needs a large number of queries to obtain an accurate gradient estimation. Thus, seeking for the minimum adversarial perturbations on videos is more challenging than that on images, a reasonable attack algorithm should reduce the video dimensions first, so as to improve the attacks' efficiency and reduce the perturbations' magnitude. To meet this goal, temporally sparse video attacks [16], [17], [18] are proposed to eliminate the redundancy in the temporal domain, and spatial video attacks [19] try to eliminate the redundancy in the spatial domain. More importantly, the spatial and temporal redundancy should be jointly considered, i.e, modeling the key regions within key frames, and then evaluating the robustness on these areas. The current related methods [20], [21] both regard the selecting key frames and selecting key regions as two separate steps, and don't simultaneously consider their interaction, thus leading to the sub-optimal attacking efficiency and performance.

Manuscript received 10 July 2022; revised 16 March 2023; accepted 26 March 2023. Date of publication 29 March 2023; date of current version 4 August 2023. This work was supported in part by National Key R & D Program of China under Grant 2020AAA0104002, and in part by National Natural Science Foundation of China under Grant 62076018. Recommended for acceptance by G. Hua. *(Corresponding author: Xingxing Wei.)*

The authors are with the Institute of Artificial Intelligence, Beihang University, Beijing 100191, China (e-mail: xxwei@buaa.edu.cn; theone@buaa.edu.cn; yanhq@buaa.edu.cn).

Digital Object Identifier 10.1109/TPAMI.2023.3262592

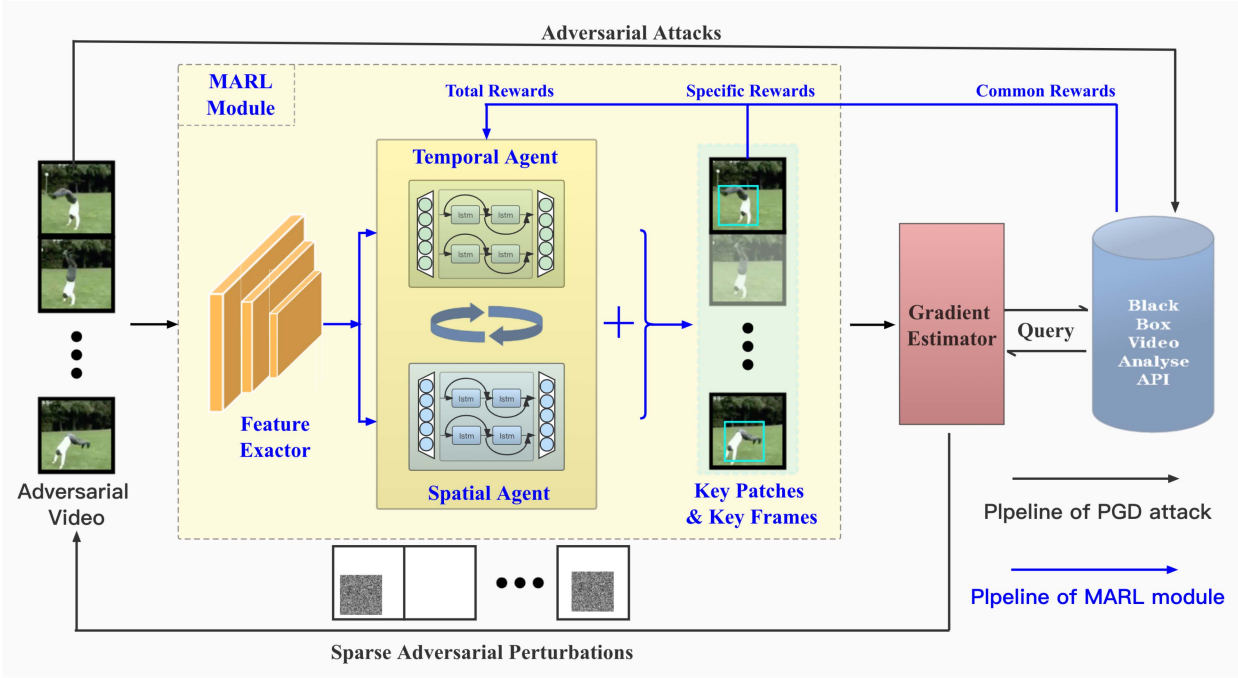


Fig. 1. Overview of the proposed AstFocus attack. It integrates a cooperative Multi-Agent Reinforcement Learning (MARL) module into the PGD attack with NES gradient estimation [22], and thus selects key frames and key patches within the video to reduce dimensions. In this way, an effective and efficient gradient estimation on the reduced space is achieved, and the evaluation's efficiency and accuracy are improved at the same time.

However, simultaneously optimizing the key frames and key regions is difficult. Because they belong to different domains, and are closely coupled, i.e, changing the key frames also affects the selection of key regions. This is more challenging in the query-based black-box attacks, where only the feedback from the threat model can be used to perform the optimization. Considering the above points, this paper mainly addresses the following problem: *How to simultaneously learn the precise key frames and key regions to efficiently and accurately assess the adversarial robustness of video recognition in the query-based black-box setting?*

To answer this question, in this paper, we design the novel **A**dversarial **s**patial-**t**emporal *Focus* (*AstFocus*) attack on videos, which performs attacks on the simultaneously focused key frames and key regions from the inter-frames and intra-frames in the video. The key frames and key regions are dynamically adjusted by the interaction with the threat model. Technically, this process is achieved based on the cooperative Multi-Agent Reinforcement Learning (MARL) [23]. One agent is responsible for selecting key frames (temporal agent), and another agent is responsible for selecting key regions (spatial agent). These two agents use one backbone network, and are jointly trained by the common rewards received from the black-box threat models to perform a cooperative prediction. By continuously querying, the focused space composed of key frames and key regions is becoming precise, and the whole query number becomes less than that on the original video.

More specifically, AstFocus attack is constructed based on the PGD+NES baseline, which extends PGD [24] to the black-box attack with Natural Evolution Strategy (NES) [25] gradient estimator. We attach two agents before the gradient estimator module to reduce the video dimension. In each PGD iteration, NES gradient estimator is first performed on the selected key frames and key regions predicted by agents. Then the local adversarial perturbations are generated to attack the threat model. Finally, these two agents are updated according to the computed rewards to predict better key frames and key regions in the next iteration. This process is continuously repeated until the successful attack is achieved. These two agents have similarities and also differences. For policy networks, we apply the same one backbone network to extract the feature maps from the input video frames for both of them, but design the distinct LSTM-based [26] structures according to their own characteristic to predict the optimal actions. For actions, the temporal agent's actions are defined as the sets composed of different key frames. For the spatial agent, actions are defined as the sets composed of different patch regions located in each frame. For rewards, three rewards are carefully designed to train the agents. The first one is the common reward from the feedback of black-box threat models, which is used to simultaneously guide two agents. The following two rewards are specially designed for temporal and spatial agent, respectively. And they mainly measure the actions from the view of appearance. The whole flowchart of AstFocus attack is shown in Fig. 1, and the code is released in https://github.com/DeepSota/AstFocus.

This paper is an extended work based on our conference version [27] and has the following major improvements. First, we consider the spatial redundancy besides the temporal redundancy in the previous version, and further propose the novel AstFocus attack to simultaneously learn key frames and key

regions and generate perturbations. This is a major change in the idea, which comprehensively makes use of the videos' spatial-temporal character to perform attacks. Second, we design a cooperative multi-agent RL based method to implement the new idea, while the previous version uses single-agent RL. Thus the rewards, actions, and policies are carefully re-designed. Third, more experiments are given and discussed involving the parameter tuning, ablation study, and comparisons with SOTA methods. We also re-write the abstract, introduction, methodology, and experiment sections to better introduce our motivation and methods. We believe these modifications can significantly improve the quality of our work.

In summary, this paper has the following contributions:

- We propose AstFocus attack, a novel query-based black-box attack method to assess the adversarial robustness for video recognition models, the adversarial perturbations are only added on the key spatial-temporal focused spaces, which can help reduce attack query numbers and perturbations significantly.
- A cooperative multi-agent reinforcement learning module is adopted for identifying the key frames and key regions at the same. For that, we carefully design the actions, policy networks, and rewards for both the agents according to the specific task. The agents are updated in each iteration rather than after each round of successful attack, so is efficient to converge.
- Compared with the state-of-the-art video attack algorithms, the proposed AstFocus attack can achieve less query number and smaller adversarial perturbations. Specifically, it reduces at least 10% query number, and improves at least 5% fooling rate with the smallest perturbations, which verifies the efficiency and effectiveness of AstFocus attacks.

The rest of this paper is organized as follows: we briefly review the related works in Section II. The proposed AstFocus attack algorithm is described in Section III. Experimental results and analysis are presented in Section IV. Finally, we conclude the whole paper in Section V.

II. Related Works

A. Adversarial Attacks on Videos

Adversarial example [4], [24], [28] is a maliciously crafted input designed for making the classifier produce wrong output. To make human imperceptible of its existence, the generation of adversarial examples is often limited by some deliberate conditions, such as noise size and query numbers. Adversarial video attack and adversarial image attack are similar, the difference is that the attack space of the video is much larger than that of images. It is not easy to directly extend some image attack algorithms to attack such high-dimension video data. High dimensions usually bring huge search space, leading to high costs to achieve successful attacks. Especially in the black-box setting, a huge search space will bring a large number of queries.

Some video attack techniques have been proposed to find adversarial videos. Wei et al. [16] generate sparse 3D adversarial perturbations to add on the whole video. To reduce the attacking space, an $l_{2,1}$-norm regularization based optimization is designed for making the adversarial perturbations more concentrated in some key frames of the input video. This method shows the sparse ability of adversarial video noises. Similarly, [18] propose "one frame attack," they only add adversarial noise on one video frame. The perturbation can easily defeat deep learning-based action recognition systems. The vulnerable frame is perturbed with a gradient-based adversarial attack method. In addition, [29] finds that the temporal structure is key to generating adversarial videos. They have used generative adversarial network to generate adversarial examples that can cause large misclassification rate for the video recognition models.

Not only white-box video attacks, but also black-box video attacks are explored. One class of such methods is based on transferablity across different models. For example, Wei et al. [32] perform black-box video attacks based on adversarial perturbations generated on image models.

Another black-box video attacks belong to query-based methods. They generate perturbations via querying the target video recognition system. Among them, Jiang et al. [19] extend PGD algorithm to video attack with gradient estimators computed using super-pixels. To reduce attacking costs, some efficient black-box video attack algorithms are proposed. [30] argues the initialized random noises in [19] are less effective, they utilize the intrinsic movement pattern and regional relative motion, and propose the motion-aware noises to replace random noises. By using this prior in gradient estimation, fewer queries are needed to perform video attacks. Wei et al. [20] search for a subset of frames based on the importance of each video frame to the recognition model. Besides, they also limit the adversarial perturbations only on some salient regions. Because the temporal and spatial reductions are separately formulated, the method usually needs hundreds of thousands of queries. To mitigate this defect. Wei et al. [17] have proposed a sparse video attack algorithm based on reinforcement learning. An agent is designed to identify key frames through some interactions with the threat model. It can significantly reduce the adversarial perturbations, but update the agent only after each round of successful attack. This poor update mechanism leads to many unnecessary queries and a weak fooling rate. RLSB attack [21] explores to select key frames and key regions to reduce the high computation cost. However, the reinforcement learning is only applied to select key frames, which is similar to [17]. The process of selecting key regions is based on the saliency maps, it is independent to the process of selecting key frames, and not integrated into the reinforcement learning framework. Thus, the selecting key frames and key regions are separately formulated. Recently, [31] presents to parameterize the temporal structure of the search space using geometric transformations, and then reduce the temporal search space. Thus, they can efficiently estimate the gradients.

In this paper, we also explore important searching space, which is different from the previous work focusing only on key frames in the temporal domain. We jointly consider the identification of key regions in the spatial domain besides the temporal domain. For that, a multi-agent reinforcement learning

TABLE I
COMPARISONS WITH QUERY-BASED BLACK-BOX VIDEO ATTACK METHODS. "TEMPORAL" DENOTES REDUCING TEMPORAL REDUNDANCY IN THE VIDEO, "SPATIAL" DENOTES REDUCING SPATIAL REDUNDANCY, "JOINTLY" DENOTES JOINTLY LEARNING FOR REDUCING THE SPATIAL AND TEMPORAL REDUNDANCY IN THE VIDEO

	Temporal	Spatial	Jointly
VBAD attack [19]	✘	✔	✘
Sparse attack [17]	✔	✘	✘
Motion-sampler attack [30]	✔	✘	✘
GEO-TRAP attack [31]	✔	✘	✘
Heuristic attack [20]	✔	✔	✘
RLSB attack [21]	✔	✔	✘
AstFocus attack (ours)	✔	✔	✔

is designed to identify a reduced space through rewards on the inherent property of video and interactions with the threat model. The comparisons with query-based black-box video attack methods are summarized in Table I.

B. Spatial-Temporal Property for Videos

Video can be regarded as multiple continuous images, therefore video processing often needs to consider both spatial and temporal correlations. The simultaneous consideration of temporal and spatial correlation of video is the key of video related tasks. Video action recognition is a longstanding research topic in multimedia and computer vision. Many mainstream algorithms are motivated by the advances in image classification, and improved through utilizing the temporal dimension of the video data. To facilitate the classification performance, Wu et al. [33] have proposed a hybrid deep learning framework for video classification, which is able to harness not only the spatial and short-term motion features, but also the long-term temporal clues. They integrate the spatial and temporal features in deep neural model with elaborately designed regularizations to explore feature correlations. The method can produce competitive classification performance. Some works based on the spatial-temporal property can be found in [11], [12], [13].

Unlike the above methods, we consider the spatial-temporal property of videos in the video attack task. The temporal and spatial redundancy within videos are reduced to improve the efficiency of video attack, which extends the application scope of spatial-temporal property of videos.

III. METHODOLOGY

In this section, we first give the baseline video attack algorithm: PGD [24] attack with NES [25] gradient estimator. Then the details of integrating cooperative Multi-Agent Reinforcement Learning (MARL) [23] into the baseline are introduced. Finally, the whole algorithm is summarized.

A. Preliminaries

We assume $F(\cdot)$ is a black-box video recognition model only whose *top-1* information including the category label and confidence score can be required. Given a video $X = \{x_i | i = 1, \ldots, M\}$ with ground-truth label y where $x_i \in \mathbb{R}^{H\times W\times 3}$ denotes the i-th frame, and M is the total frame number, the predicted category label is $y = F(X)$, and the corresponding confidence score is $P(y|X)$.

To attack the video recognition model, we extend Projected Gradient Descent (PGD) [24] to adapt the video data. The adversarial video X' under the un-targeted attack is defined as:

$$X'_{t+1} = Proj\left(X'_t + \alpha \cdot sign\left(\nabla_X l(X'_t, y)\right)\right), \tag{1}$$

where $Proj(\cdot)$ projects the updated adversarial example to a valid range. α is the attack step, and is used to control the magnitude of the added adversarial noise per each iteration. The $sign(\cdot)$ is the sign function, and $l(\cdot)$ is the cross-entropy loss function. Due to the limitation of black-box settings, we cannot obtain the accurate gradient g by directly computing $g = \nabla_X l(X'_t, y))$. Instead, [22] proposes to utilize Natural Evolution Strategy (NES) [25] to estimate g by querying the threat model. Specifically, NES can be described as:

$$g \approx \frac{1}{\Delta n}\sum_{i=1}^{n} \sigma_i \cdot P\left(y|X'_t + \Delta \cdot \sigma_i\right). \tag{2}$$

It first samples $n/2$ values $\delta_i \backsim N(0, I)$, and then sets $\delta_j = -\delta_{n-j+1}, j \in \{(n/2+1), ..n\}$. Finally, the gradient g is estimated through averaging the ratio of the predicted results to search variance Δ.

For the targeted attack, (1) is modified as follows:

$$X'_{t+1} = Proj\left(X'_t - \alpha \cdot sign\left(\nabla_X l(X'_t, y')\right)\right), \tag{3}$$

where y' is a target category label pred-defined by the adversary in advance. In (2), the ground-truth y should also be modified as the target label y' to estimate the gradients versus the target label.

In practial application, directly performing (2) is inefficient. Because the number of sample points n is related with the dimension of $X'_t \in \mathbb{R}^{M\times H\times W\times 3}$. Owing to the high dimension of video data X'_t, we need to set a large value of n to compute an accurate gradient in each iteration t, which will lead to a large number of queries with the threat model. To improve the attack efficiency, the video dimension should be reduced by selecting the key frames and key regions, obtaining a reduced M, H and W, and thus a small value of n can be available. Technically, we hope to replace X'_t in (2) with $\hat{X}'_t = \Gamma(X'_t)$, where $\Gamma(\cdot)$ denotes the reduced operation, and $\hat{X}'_t$ is the reduced video.

B. The Proposed AstFocus Attack

To implement the above idea, we build the so-called AstFocus attack based on a cooperative multi-agent reinforcement learning (MARL) to jointly solve for the key frames and key regions during the black-box attack process. In AstFocus attack, one agent is responsible for selecting key frames (temporal agent), and another agent is responsible for selecting key regions (spatial agent). These two agents are cooperative to achieve the same goal. The processes of selecting key frames and key regions in each iteration of PGD are formulated into the Markov Decision Processes (MDP). The details of these two agents as well as the optimization algorithm are given below.

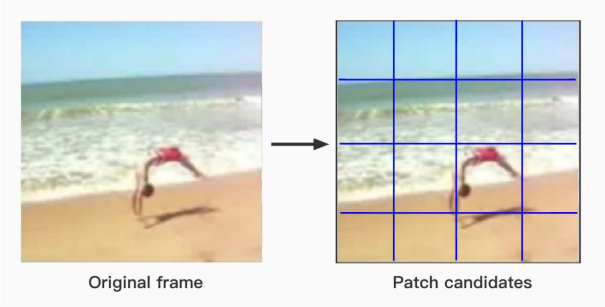


Fig. 2. Designed actions of the spatial agent. In each frame, we uniformly divide the frame into overlapped patches according to a predefined stride. All the patch candidates constitute the actions. For simplicity, the stride equals to the patch size in this example.

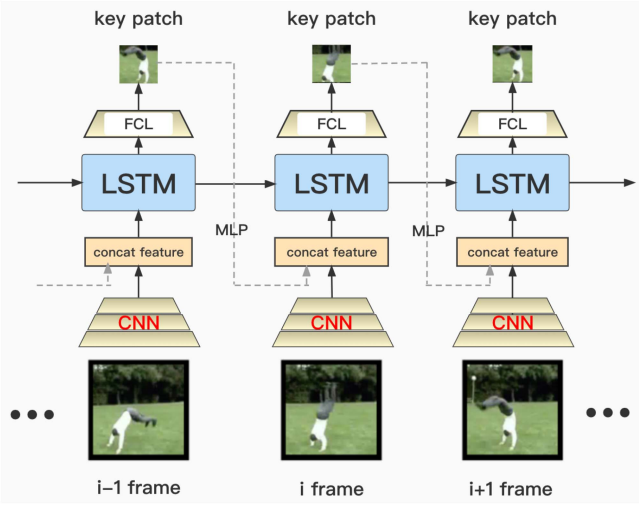


Fig. 3. Flowchart for the Policy network of the proposed spatial agent. It is used to identify the crucial regions of each video frame.

1) Spatial Agent: Spatial agent actually aims at solving an object localization problem (detecting key regions), we detail from three parts.

Action Design. To construct the actions of the spatial agent, we uniformly divide each video frame into overlapped patches inspired by the Vision Transformer [34]. In this way, we obtain a candidate patch set for the i-th frame x_i: $B_i = \{b_i^j | j = 1, \ldots, D\}$ where b_i^j denotes the j-th patch region within x_i, and D is the total number of candidate patches in this frame. $b_i^j \in \mathbb{R}^{h \times w}$ denotes that the patch's size is h and w, and their values will be tuned in the experiments. The goal of spatial agent is to select an optimal patch $b_i^* \in B_i$ in each frame as the key region, and thus the final selected action is a sequence set $a^p = \{b_i^* | i = 1, \ldots, M\}$. From the definition, we can see that there are totally D^M action combinations for the given video X, which implies the search space is huge. An example of actions in one frame is listed in Fig. 2, where $D = 16$.

Policy Network Design. Spatial policy network $\pi^p(a^p|s^p)$ is used to predict the spatial action a^p when the state s^p is given. The flowchart of our policy network is listed in Fig. 3. Overall speaking, because we need to tackle with the sequence video data, a LSTM-based [26] structure is used to construct the policy network $\pi^p(a^p|s^p)$. For the i-th frame x_i, a lightweight convolution neural network (CNN) $f(x_i)$ is first to extract the frame-level feature maps e_i. In our experiments, we use MobileNet V2 as the lightweight CNN backbone for simplicity. Users can also apply other lightweight CNNs. Then they are fed into the LSTM unit to predict the logits for each patch. Next, a Softmax with Fully Connected Layer (FCL) is attached to output each patch's probability $p_{b_i^j}$. Finally, we utilize the categorical sampling to obtain the optimal patch region b_i^* according to their probability values $p(B_i) = \{p_{b_i^j} | j = 1, \ldots, D\}$. To guarantee the smooth change of selected patch between adjacent frames, we concat the local patch features $e_{i-1}^{b^*}$ of the previous selected patch b_{i-1}^* with the current frame-level features e_i to jointly predict the current patch region, and e_{i-1}^* is extracted via a simple multilayer perceptron (MLP) on the corresponding patch features of e_{i-1}.

Formally, the frame-level feature maps are extracted by:

$$e_i = f(x_i), i = 1, 2, \ldots, M, \tag{4}$$

next, the optimal action for each frame is achieved by:

$$p(B_i) = \pi_\theta^p \left(\cdot | concat(e_i, e_{i-1}^{b^*}), h_{i-1}^\pi \right), i = 1, 2, \ldots, M, \tag{5}$$

$$b_i^* = categorical(p(B_i)), i = 1, 2, \ldots, M, \tag{6}$$

where h_{i-1}^π denotes hidden states output by LSTM unit in the i-1-th frame. Thus, the state s^p in our method is defined as the concat feature $concat(e_i, e_{i-1}^{b^*})$. Equation (6) is repeated M times to achieve the optimal action $a^p = \{b_i^* | i = 1, \ldots, M\}$.

In our method, the policy network is updated in each iteration t of PGD attack, therefore, the optimal action will be updated in each iteration until the PGD attack stops.

Reward Design. In each iteration, the spatial policy network will receive the feedback from the environment to update its parameters θ^p. Therefore, we need to design the reasonable rewards to guide the update of policy network. Because AstFocus attacks are based on the Multi-Agent Reinforcement Learning (MARL) framework, we design two kinds of rewards: one is specific for the spatial agent, and another is the common rewards shared with temporal agent.

For the special reward, an intuitive idea to evaluate the patch's importance is the area covered by the foreground objects. Because video recognition model mainly performs predictions based on the foreground objects like person, car, etc. Therefore, if the policy network $\pi^p(a^p|s^p)$ selects the foreground patch, the specific reward should be enlarged, and thus the policy network will be encouraged to select the foreground object in the next iteration. Based on this idea, we need a metric to measure the objectness score for a given patch. We here choose a classic objectness model: edgeboxes [35]. It calculates the edge response of each pixel and determines the boundary of the object by using the structured edge detector.

More concretely, the $r_{edgebox}^i$ reward for the selected patch b_i^* can be described as following:

$$r_{edgebox}^i = \frac{\prod_k w_b(s_k) \cdot u_k}{2 \cdot (w + h)^2}, i = 1, 2, \ldots, M, \tag{7}$$

The edgebox reward for the whole video is defined:

$$r_{edgebox} = \sum_{i=1}^{M} r^i_{edgebox}, i = 1, 2, \ldots, M, \tag{8}$$

where w and h are the patch's width and height. $w_b(s_k)$ is used to measure the affinity of the k-th edge groups in the selected patch. The u_k is the sum of the k-th edge groups in the selected patch. In general, a large patch often results in a large edgebox value. More detailed information about edgebox function can be found in [35].

For the common reward, it comes from the feedback of the black-box threat models. If the selected patch is reasonable, the generated adversarial patch should have a strong attacking ability, and thus will make the confidence score output by threat models have a big drop. Therefore, we can use the confidence drop of the ground-truth label as a metric to compute this reward. Because this is also useful to the temporal agent, it is called as common reward. Specifically, the common reward r_{common} is defined as follows:

$$V(X') = exp(P(y'|X') - P(y|X')); \tag{9}$$

$$r_{common} = \frac{V(X'_{t+1}) - V(X'_t)}{V(X'_t)}, \tag{10}$$

where $exp(\cdot)$ is the exponential function, and $P(y|X')$ represents the ground-truth label's confidence when X' is fed into video recognition model. In the un-targeted attack, $P(y'|X')$ represents the second-ranked label's confidence which is considered as the most competitive label to replace the ground-truth label. Only if the second-ranked label's confidence becomes larger than that of the ground-truth label, $V(X')$ becomes large. Then, we use the relative change of $V(X')$ at different iteration t as the metric for common reward. Equation (10) is designed to encourage the agent to add perturbations on the selected regions that make the second-ranked label's confidence gradually reach the ground-truth label and finally exceed it. In targeted attack, $P(y'|X')$ is the confidence of pre-defined target label by the adversary.

In summary, the t-iteration reward for spatial agent is:

$$r^t_{spatial} = r^t_{common} + \lambda_1 r^t_{edgebox}, \tag{11}$$

where λ_1 denotes a weight to balance the two terms.

2) Temporal Agent: There exists a major distinction between temporal agent and spatial agent. Spatial agent aims at solving the object localization problem, while temporal agent aims at solving the binary classification problem (selecting or not selecting a frame). Thus, the actions, rewards, and policy networks of temporal agent should be re-designed.

Action Design. Key frames refer to those video frames that are conducive to successful attack, and their number is less than that of the whole video. The goal of the temporal agent is to select some key frames from the whole input video X, and thus the final selected action is also a sequence set $a^f = \{o^*_i | i = 1, \ldots, M\}$ just like spatial agent. The $o^*_i \in \{0, 1\}$ indicates whether the i-th frame is selected or not. Therefore, there are totally 2^M different actions, which is not friendly to direct optimization learning.

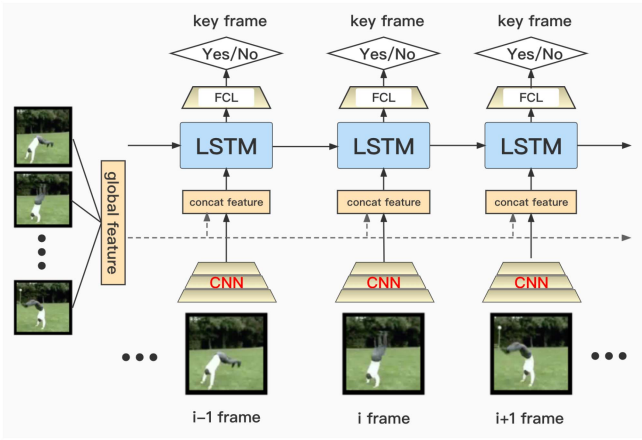


Fig. 4. Flowchart for the policy network of the proposed temporal agent. It is used to select the key video frames from the input video.

Policy Network Design. The temporal policy network $\pi^f(a^f|s^f)$ is used to predict the spatial action a^f when the state s^f is given. It is constructed with a LSTM structure. The skeleton diagram of the temporal policy network is shown in Fig. 4. The input of the policy network is the concat features composed of current frame-level features e_i and a video-level global features e^g. Combining these two features can better select the key frames by considering the global video information. The global features e^g is achieved by a fully connected layer on all the frame-level features e_i, $i = 1, \ldots, M$. The output of LSTM network is then fed to a Softmax with Fully Connected Layer (FCL) to predict the probability p_i to indicate o_i=1. Technically, the temporal policy network can be expressed as:

$$p_i = \pi^f_\theta \left(\cdot|concat(e_i, e^g), h^\pi_{i-1}\right), i = 1, 2, \ldots, M, \tag{12}$$

$$o^*_i = Bernoulli(p_i), i = 1, 2, \ldots, M, \tag{13}$$

where $Bernoulli(\cdot)$ is the Bernouli function. h^π_{i-1} denotes the hidden states output by LSTM unit in the i-1-th frame. The state s^f is defined as the concat feature $concat(e_i, e^g)$. Equation (13) is repeated M times to get $a^f = \{o^*_i | i = 1, \ldots, M\}$.

Reward Design. To make the temporal agent intelligent, the temporal policy network interact with the environment for updating its parameters θ^f. Similar to the training of the spatial agent, in addition to the common reward function which shared with the spatial agent, we also have designed two special rewards to guide the temporal agent. The first specific reward function is the sparse reward r_{sparse}:

$$r_{sparse} = exp\left(-\frac{1}{M}\left|\sum_{i=1}^{M} o_i - L\right|\right), \tag{14}$$

where L here is used to control the number of key frames selected by the temporal agent, and $L < M$. The second specially designed reward function is mainly used to evaluate the representative ability of the video frames selected by the temporal agent. Because the selected video frames need to be sparse but effectively represent the semantic information of the whole input video. The representative reward function [36] r_{rep} is defined

as:

$$r_{rep} = exp\left(-\frac{1}{M}\sum_{i=1}^{M}\min_{t'\subset\mathcal{K}}||e_i - e_{t'}||_2\right), \quad (15)$$

where $\mathcal{K}$ is a set of selected frame, i.e., frames with o_i=1. Through those reward functions, the temporal can be forced to recognize few and critical video frames. The selected video frames can be effectively reduce the temporal redundancy of the entire video and effectively improve the following attack efficiency.

To make key video frames conducive to successful attacks, r_{common} also join the learning of the temporal agent. For the t-th iteration, the corresponding reward is:

$$r^t_{temporal} = r^t_{common} + \lambda_2 r^t_{sparse} + \lambda_3 r^t_{rep}, \quad (16)$$

where λ_2 and λ_3 are two balance coefficients, and they will be discussed and set up in the experimental section.

So far, through the cooperation of spatial agent and temporal agent, the key regions in the key video frames of the input video can be identified. In the procedure of multi-agent reinforcement learning, the agents interact with the threat model for many times, and the predicted results of the agents are more inclined to the rapid and successful attack. Therefore, the critical attacking spaces selected by multi-agent reinforcement learning is the space sensitive to attack, which can effectively improve the efficiency of attack.

3) Optimization Algorithm: There are two parts to optimize: one is the lightweight CNN backbone $f(\cdot)$, and another is the policy network $\pi(\cdot)$.

CNN Backbone. In our method, the CNN backbone $f(\cdot)$ is used for both temporal agent and spatial agent. It extracts the frames' feature maps to construct state s. To decouple the training process of CNN backbone and policy network, we directly apply a pre-trained MobileNet V2 backbone on ImageNet dataset as the feature extractor. In this way, we can focus on the optimization of two policy networks.

Policy Network. The policy gradient methods are used to optimize the temporal and spatial policy network. They are to directly adjust the parameters θ in order to maximize the objective $J(\theta) = \mathbb{E}_{s\backsim\rho^\pi, a\backsim\pi_\theta}[R]$ by tacking steps in the direction of $\nabla_\theta J(\theta)$. By introducing an action-value function $Q^\pi(s,a)$, the policy gradient can be changed as:

$$\nabla_\theta J(\theta) = \mathbb{E}_{s\backsim\rho^\pi, a\backsim\pi_\theta}\left[\nabla_\theta log\pi_\theta(a|s) Q^\pi(s,a)\right]. \quad (17)$$

To solve (17), we utilize the actor-critic reinforcement learning framework [37] where a critic network is applied to approximate the action-value function $Q^\pi(s,a)$. Actor network is the policy network in Figs. 3 and 4.

Because our method focuses on the cooperative multi-agent tasks, in which the two agents are trying to optimize a shared reward function. Each agent is decentralized and only has access to locally available information. For example, temporal agent can only observe the change of key frames, and spatial agent can only observe the change of key patches. Therefore, our method can be described as Decentralized Partially Observable Markov Decision Processes (Dec-POMDP) [38]. To solve this problem, [23] presents the multi-agent decentralized actor, centralized critic approach, thus (17) is reformulated as:

$$\nabla_{\theta_k} J(\theta_k) = \mathbb{E}_{s\backsim\rho^\pi, a_k\backsim\pi_k} \times \left[\nabla_{\theta_k} log\pi_k(a_k|s_k) Q^\pi_k(\mathbf{s}, a_1, \ldots, a_K)\right]. \quad (18)$$

where π_k denotes the policy network of the k-th agent, and θ_k is the corresponding parameters. In our method, there are totally two agents. The corresponding policy networks are $\pi^p(a^p|s^p)$ with parameter θ^p and $\pi^f(a^f|s^f)$ with parameter θ^f. Here $Q^\pi_k(\mathbf{s}, a_1, a_2)]$ is a centralized action-value function that takes as input the actions of all agents (a_1, a_2) in addition to some state information $\mathbf{s} = [s^p, s^f]$, and outputs the Q-value for agent k. In this way, we can perform a communication between two agents. In cooperative MARL, each agent is expected to maximize the common reward and its specific reward, therefore, we just need to solve (18) according to the rewards for spatial agent and temporal agent, respectively.

To solve (18) for the spatial agent $\pi^p(a^p|s^p)$ and temporal agent $\pi^f(a^f|s^f)$, we use the Proximal Policy Optimization (PPO), a popular single-agent on-policy RL algorithm [39] to obtain the θ_f and θ_p. For the details of PPO algorithms, please refer to [39].

Algorithm 1: AstFocus Black-Box Video Attack Algorithm.

Input: Clean video:X; ground-truth label:y; feature extractor: $f(\cdot)$; black-box video recognition model:$F(\cdot)$. Max PGD iterations:T; PGD attack step:α; learning rate:ϵ.
Output: Adversarial video X'

Initialize parameters θ^f and θ^p for temporal policy network $\pi^f(\cdot)$ and spatial policy network $\pi^p(\cdot)$;
Extract frame-level features $\{e_i|i$=1,...,$M\}$ via (4);
while $t < T$ **do**
  Compute key regions $a^p = \{b^*_i | i = 1, \ldots, M\}$ via (5) and (6);
  Compute key frames $a^f = \{o^*_i | i = 1, \ldots, M\}$ via (12) and (13);
  Obtain the core video $\hat{X} = \{o^*_i \cdot b^*_i | i = 1, \ldots, M\}$;
  Estimate the gradients on $\hat{X}$ via (2);
  Generate adversarial video X'_{t+1} via (1);
  if $F(X'_{t+1}) \neq y$ **then**
    $X' \leftarrow X'_{t+1}$; Break;
  else
    Compute spatial reward $r^t_{spatial}$ via (11) and temporal reward $r^t_{temporal}$ via (16);
    Compute $\nabla_{\theta^f} J(\theta^f)$ and $\nabla_{\theta^p} J(\theta^p)$ via (18);
    Update $\theta^f \leftarrow \theta^f + \epsilon \cdot \nabla_{\theta^f} J(\theta^f)$;
  Update $\theta^p \leftarrow \theta^p + \epsilon \cdot \nabla_{\theta^p} J(\theta^p)$;
  end if
end while
return X'

C. The Overall Framework

After the MARL module, the key frames and key regions are obtained. The video $X'_t \in \mathbb{R}^{M\times H\times W\times 3}$ in (1) and (2) is

TABLE II
THE ACCURACY OF FOUR DIFFERENT MODES ON THREE DATASETS

Models	Datasets		
	UCF-101	HMDB-51	Kinetics-400
C3D [11]	85.88%	59.57%	54.20%
TSN [12]	83.03%	56.08 %	70.42%
TSM [13]	94.58%	74.77%	71.90%
SlowFast [43]	92.78%	65.95%	74.42%

reduced to the video $\hat{X}_t' \in \mathbb{R}^{m\times h\times w\times 3}$ composed of key frames and key regions, where m denotes the number of key frames, and h, w denote the key patches' height and width. It is clear that $m \ll M, h \ll H, w \ll W$. AstFocus attack finally utilizes $\hat{X}_t' \in \mathbb{R}^{m\times h\times w\times 3}$ to compute (2). Because of the reduced dimension, the gradient estimation can be efficient.

We now give the overall algorithm of AstFocus attack, which is illustrated under the un-targeted attack. The process of agent learning is an unsupervised process. Through continuous interaction with the threat model, the agent gets the feedback from the attack effect and external evaluation indicators from the video itself to update agents to encourage them to perform better. The whole algorithm is summarized in Algorithm 1.

IV. EXPERIMENTS AND RESULTS

A. Datasets and Recognition Models

Datasets. In our experiments, three public action recognition datasets: UCF-101 [40], HMDB-51 [41], and Kinetics-400 [42] are used. The UCF-101 contains 13,320 videos with 101 action categories, HMDB-51 is a dataset for human motion recognition, which contains 51 action categories with a total of 70,00 videos. Kinetics-400 contains 400 human action classes, with at least 400 video clips for each action. All of these datasets divides 70% of the video into training sets and 30% of the test sets. We randomly sample 100 videos from UCF-101 test set, 50 videos from HMDB-51 test set, and 400 videos from Kinetics-400 test set. All sampled videos can be classified by the recognition models correctly.

Recognition Models. For recognition models, four representative methods are used in our experiments. They are C3D [11], Temporal Segment Network (TSN) [12], Temporal Shift Module (TSM) [13], and SlowFast network [43]. These models are all mainstream methods for video classification task. For TSN, TSM, and SlowFast on three datasets, we utilize the corresponding pre-trained weights released by MMAction2 [44], a widely used open-source toolbox for video understanding based on PyTorch. For C3D, because MMAction2 only releases the pre-trained weights on UCF101, to ensure the consistency, we utilize the officially pre-trained weights on three datasets released by the authors.[1] Table II lists their accuracy values under the test set.

[1]https://github.com/kenshohara/3D-ResNets-PyTorch

B. Evaluation Metrics

There are four metrics to test the performance of our method on various sides. Specifically, Fooling Rate, Query Number, Mean Absolute Perturbation, and Time are explored.

Fooling Rate (FR): indicates the percentage of adversarial videos, which successfully fool the threat model, out of all the tested videos. FR reflects the probability of successfully generating adversarial examples. A higher FR value means the better performance on the task of attacks.

Mean Absolute Perturbation (MAP): denotes the magnitude value of the generated adversarial perturbation $\mathbf{r}$. For a given video: MAP=$\frac{1}{M}\sum_i |\mathbf{r}_i|$, where M is the number of frames in a video, and $\mathbf{r}_i$ is the perturbation intensity vector on the i-th frame. To be intuitive, the value of MAP is resized to 0-255. In the experiments, we report the average MAP across the test videos. A lower MAP value means the better imperceptibility.

Query Number (QN): denotes the used query times to successfully fool the threat model for a given adversarial video. It reflects the efficiency of different video attack methods. In the experiments, we set an upper bound for the query number, if the queries reach the upper bound but the threat video model is still not fooled successfully, we think this adversarial video is not successfully generated. The average query number across the test videos is reported. A lower QN value means the higher efficiency.

Time (T): denotes all the cost time when the successful attack is finished. We use seconds to measure the time. In the experiments, we report the average seconds across the test videos. A lower time value means the higher efficiency.

Note that previous works [17], [20], [27] have also used these metrics. But this paper has a slight difference with them. In [17], [20], [27], they compute MAP and NQ values only for the adversarial videos which can successfully perform the attacks, the MAP and NQ values of failed videos don't be considered. In contrast, this paper computes MAP and NQ values for all the test videos. We think this is more reasonable because the failed videos also generate perturbations and cost queries with the threat models.

C. State-of-the-Art Attack Competitors

Here, we use six state-of-the-art black-box video attack methods as comparisons with our method in effect and speed, named VBAD attack [19], Heuristic attack [20], Sparse attack [17], GEO-TRAP attack [31], RLSB attack [21], and Motion-sampler attack [30]. The detailed introductions about these competitors can be found in the related works section. We use their own officially released codes to conduct comparisons (for Sparse attack, we directly use the well-trained agent to predict key frames and then perform attacks. There is no released code for RLSB attack, we implement it according to the paper). For fair comparisons, all the settings are the same.

D. Implementation Details

In the query-based black-box attacks, the query number is a key metric to evaluate the attacks' performance. Thus, given a

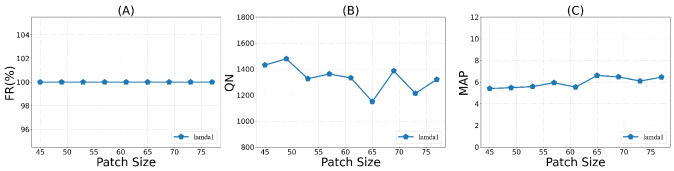


Fig. 5. Parameter tuning results of AstFocus attacks with different patch sizes. (A) The effects for fooling rate. (B) The effects for query number. (C) The effects for perturbation magnitude.

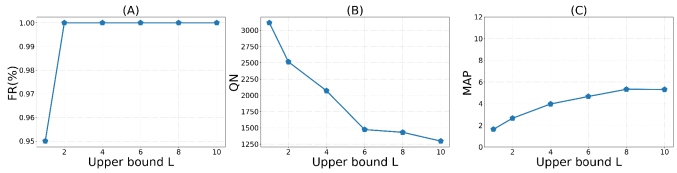


Fig. 6. Parameter tuning results of AstFocus attacks with different upper bounds of key frames. (A) The effects for fooling rate. (B) The effects for query number. (C) The effects for perturbation magnitude.

video, we set a maximum query number for all the compared methods. If the used query number is above the maximum query number, the adversarial attack is regarded as failure for this given video. We here set the maximum query number to 1.5×10^4 in the un-targeted attack and 3×10^4 in the targeted attack. In the NES, we set the variance Δ in NES to 10^{-3} for the un-targeted attack and 10^{-6} for the targeted attack according to our experience.

E. Parameter Tuning

There are some hyperparameters in our method. In this section, we will determine their values via parameter tuning on the validate set. Specifically, we randomly selected 20 videos from HMDB-51 to construct the validation set, and then perform parameter tuning versus C3D model.

1) Patch Size for Spatial Agent: The first hyperparameter is the patch size h and w when designing the spatial agent's action. A reasonable patch size will lead to less queries and less perturbations. The parameter tuning results for patch size is given in Fig. 5, where we explore its effects for the fooling rate, query number and perturbations, respectively. From the figure, we see that patch size mainly affects the query number but shows slight changes for fooling rate and perturbation magnitude. Moreover, Fig. 5(B) shows the query number is relatively sensitive to the patch size. This is reasonable because the pre-defined patch size determines the proportion of selected key regions out of the whole image, thus affects the query number.[2] Overall, when the patch size is set to 65, the query number reaches the smallest value. Therefore, we set $h = w = 65$.

2) Upper Bound of Key Frames: The second hyperparameter is the upper bound L of selected key frames in (14). A reasonable L can help our method select the minimal key frames to perform a successful video attack, and thus query number can be reduced. The parameter tuning results for upper bound L is given in Fig. 6, where we also explore its effects for the fooling rate, query number and perturbations, respectively. We can see that with the increase of L value, the fooling rate will gradually stabilize to 100% and query number is slowly decreasing, but it would cause a big increase in perturbation magnitude. To balance three different evaluation metrics, we set $L = 10$ in the following experiments.

[2]From the last column in Table III, we see that AstFocus attack has smaller variance when performing multiple times. For attacking C3D model on HMDB-51, the variance has no changes for FR, and only changes 1% around the mean for MAP, 4% around the mean for NQ. Therefore, the unsmooth curve is not caused by the significant variance.

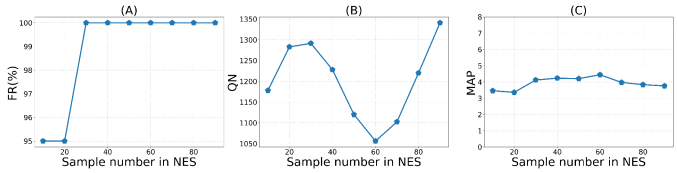


Fig. 7. Parameter tuning results of AstFocus attacks with different sample numbers n. (A) The effects for fooling rate. (B) The effects for query number. (C) The effects for perturbation magnitude.

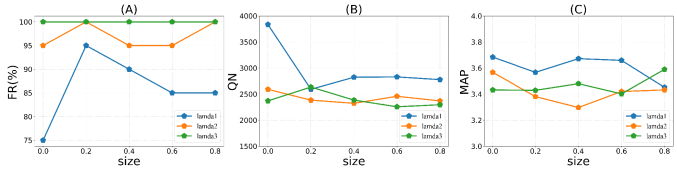


Fig. 8. Parameter tuning results of AstFocus attacks with different reward weights. (A) The effects for fooling rate. (B) The effects for query numbers. (C) The effects for perturbation magnitude.

3) Sample Number in NES: The third hyperparameter is the number of sampled points n in (2). The sample number n per each iteration has a great influence on the accuracy of the estimated gradient, especially when the attacking space changes. To explore the impact of the sample number n on the attack effect, we have conducted a series of experiments. The parameter tuning results are given in Fig. 7. We can see that with the increase of n value, the fooling rate will gradually stabilize to 100%, but query number and perturbation magnitude achieve their optimal performance when n is located in 60. Therefore, we set $n = 60$ in the following experiments.

4) Weights for Various Rewards: There are three weights to tune in the reward functions. They are λ_1 in (11), λ_2 and λ_3 in (16), which measures the importance of their own rewards. The parameter tuning results are given in Fig. 8. According to the figure, we set $\lambda_1 = 0.2, \lambda_2 = 0.4$, and $\lambda_3 = 0.6$, respectively. It means there exists more redundancy to reduce in the temporal domain than spatial domain, thus needing to set large rewards in (16) to guide agent for learning key frames.

F. Ablation Study

To explore the effectiveness of different components in the proposed algorithm, a series of experiments are conducted here. Specifically, we investigate the effects of various agents and various rewards, respectively. Similarly, we randomly select 20 videos from HMDB-51 to construct the validation set, and then

TABLE III
EFFECTS OF VARIOUS AGENTS TO ASTFOCUS ATTACKS IN AN UN-TARGETED SETTING

Metrics	Different agent versions			
	Baseline	Spatial	Temporal	Spatial&Temporal
FR(%)	73.3±2.9	88.3±2.9	93.3±2.9	100±0.0
QN	3662±244	2670±155	2757±125	2227±92
MAP	6.37±0.08	4.21±0.05	4.53±0.07	3.35±0.03

perform the ablation study versus C3D video recognition model. Because the gradient estimator module introduces randomness, to consider this factor, we perform each ablation study for five times, and then report the mean ± variance for different metrics.

1) Effects of Various Agents: In our method, the baseline is PGD+NES algorithm. Then we integrate two agents into the PGD+NES to reduce the video dimension from the temporal and spatial domains, respectively. Here we perform the ablation study about whether these two agents work in the video attack. The results are given in Table III, where "Baseline" denotes the PGD+NES. In this setting, because there is no dimension reduction module, the perturbations are added on the whole video, which can be called as "dense attack". The term "Spatial" denotes integrating the spatial agent into the baseline. In this setting, we reduce the spatial redundancy by selecting the key patches in each frame. The term "Temporal" denotes integrating the temporal agent into the baseline, which reduces the temporal redundancy by selecting the key frames. The term "Spatial & Temporal" denotes the full version of AstFocus attack, i.e., simultaneously reducing the temporal and spatial redundancy via two agents.

We show the effects versus fooling rate (FR), number query (NQ), and perturbation magnitude (MAP). From the table, we can see that the dimension reduction is indeed useful to the attacking performance, i.e., the "Spatial" and "Temporal" achieve higher FR and smaller QN and MAP than the "Baseline". By simultaneous reducing the temporal and spatial redundancy, "Spatial & Temporal" achieves the highest FR (100%), and smallest QN and MAP. The average FR increases 26.7% (73.3%→100%), average QN and MAP decrease about 36% (3662→2227) and 47% (6.37→3.35) versus the baseline, respectively. In addition, "Spatial & Temporal" has smaller variance than baseline. For example, the variance has no changes for FR metric, and only changes 1% around the mean for MAP metric, 4% around the mean for NQ metric. For this reason, we neglect the variance value in the following comparison experiments. This verifies the important role of dimension reduction when performing video attacks.

We also compare our RL-based agents with random agents, where the key frames and key patches are randomly selected in each PGD iteration, other settings are the same. Fig. 9 shows the comparison results. We can see that for all the evaluation metrics, our RL-based agent outperforms the random agent (100%±0% versus 86.7%±2.89%, 2227±92 versus 2632±114, 3.35±0.03 versus 3.62±0.04), which verifies the temporal and spatial agents in our method are jointly well-trained under the guidance of the carefully designed rewards. With the feedback

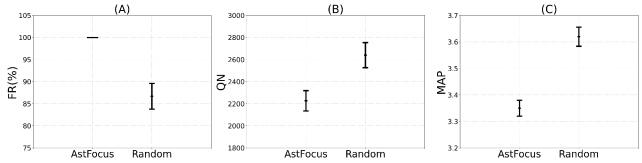


Fig. 9. Comparison between RL-based agent and random agents.

TABLE IV
EFFECTS OF VARIOUS REWARDS TO ASTFOCUS ATTACKS IN AN UN-TARGETED SETTING

Metrics	Different reward versions			
	Common	+Edgebox	+Sparse	+Representative
FR(%)	76.7±2.9	91.7±2.9	96.7±2.9	100±0.0
QN	3780±183	2690±122	2490±87	2227±92
MAP	3.62±0.07	3.58±0.04	3.39±0.04	3.35±0.03

of threat models, the temporal and spatial agents become intelligent.

2) Effects of Various Rewards: To better guide the agent learning, we carefully design the rewards. Specifically, one common reward (10) coming from the black-box threat model, and two kinds of specific rewards (8) as well as (15) and (14). The common reward is shared by the spatial agent and temporal agent, and the special rewards only belong to their own agents. In this section, we explore the effects of these rewards to the fooling rate, query number and perturbations.

Table IV lists the ablation study for effects of various rewards. The term "Common" denotes the guidance of both temporal and spatial agents only using the common reward in (10). The terms "+Edgebox," "+Sparse," and '+Representative" denote adding the corresponding rewards ((8), (14), and (15), respectively) on the former basis to guide the agent learning. From the table, we see that with the addition of more and more rewards, average FR gradually increases, and average QN and average MAP gradually decrease. Compared with the solo common reward, the full version (the rightmost column) with all the rewards improves 23.3% for average FR (76.7%→100%), and reduces 43% for average QN (3780→2227), and 8% for average MAP (3.62→3.35). The variance also becomes smaller and smaller. The contrast verifies the rationality of the designed rewards.

3) Convergence of AstFocus Attacks: Because our agents are updated by the rewards in each iteration, it is necessary to investigate whether the agents are under the convergence with the increasing iterations. For that, we list the values' change of (9) with the increasing PGD iteration in Fig. 10. Equation (9) directly reflects the success or failure of an attack. If the target class's confidence score is above the ground-truth class's confidence score, the value of (9) will be above 1, representing that the attack is successful, and vice versa. From the figure, we can see that (9)'s values for all the threat models are gradually increasing until the stable situation. When the iteration reaches 400, all the models achieves the convergence. This verifies the good convergence of AstFocus attack. Actually, the attack usually stops when (9)'s value is above 1, i.e., the step 9 in Algorithm 1. Therefore, we only need few iterations in

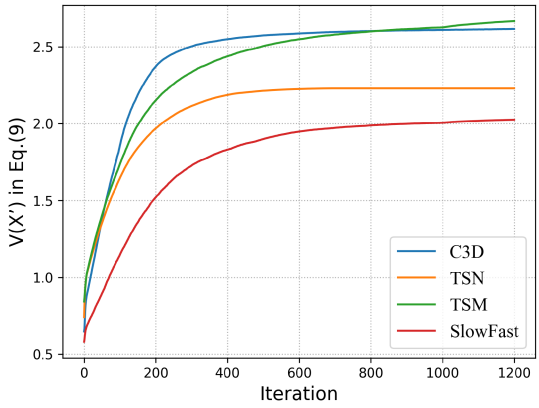


Fig. 10. Convergence of the proposed AstFocus attacks.

Fig. 11. Qualitative example for selecting key frames and key regions in AstFocus attacks. From top to bottom denotes the selected key frames and key patches in the 5th, 10th, and 20th PGD iteration.

application. Fig. 11 gives a qualitative example of the agents in AstFocus attacks, where the key frames and key patches in different iterations are illustrated by the bounding boxes. We can see that the spatial agent gradually focuses on the foreground objects. This is reasonable because these areas are key cues for video recognition task. In addition, the temporal agent tends to select the frames with big changes in the actions. These frames have a strong representative ability for the whole video from the appearance, which shows key frames are sensitive to attacks.

G. Comparisons With SOTA Methods

Here, we compare the proposed AstFocus attack with six state-of-the-art black-box video attack methods on three public datasets and four widely used video recognition models. The comparative results in the un-targeted and targeted settings are recorded in Tables V, VI, and VII (for fair comparison, the target label for all the methods are the same when performing targeted attacks). From the tables, we see that: *(1) For attack effect (FR and MAP)*, our method significantly outperforms other six SOTA methods for FR metric (at least 5%) versus all the threat models on all the datasets, showing the big advantage in attacking ability. For MAP metric, AstFocus attack only slightly loses to Sparse attack in the un-targeted attack but obviously outperforms other five video attacks. Because Sparse attack adds adversarial perturbations only on the fixed key frames in each PGD iteration. In (1), there exists a clip operation $Proj(\cdot)$ to project the perturbations

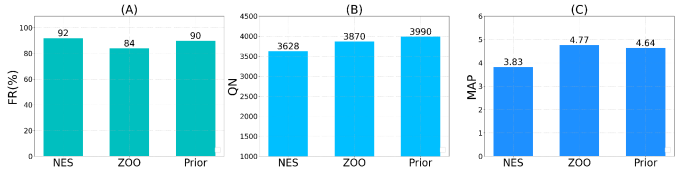


Fig. 12. Comparisons of different gradient estimators within AstFocus.

to a small range. So the upper bound of adversarial perturbations generated by Sparse attack is small. This design also limits the attacking efficiency and effectiveness, for example, Sparse attack only has almost 40% FR but needs almost 9000 NQ on average for un-targeted attacks, far less than AstFocus attacks. Overall, AstFocus attack is better than Sparse attack. A small MAP under a high FR means an accurate evaluation for the models' adversarial robustness. From this viewpoint, AstFocus attack is more suitable to evaluate different video models. *(2) For attack efficiency (NQ and T)*, AstFocus also significantly beats other six SOTA methods for NQ versus all the threat models on all the datasets, reducing at least 10% queries compared with the second best video attacks. For time metric, AstFocus attack only slightly loses to VBAD attack but still beats other five video attacks. This is reasonable because AstFocus attack integrates two additional agents to reduce dimensions during attacks while VBAD does not involve this step. In return, AstFocus greatly outperforms VBAD versus the other three metrics. Overall, AstFocus has the high efficiency. *(3) For simultaneous modeling*, AstFocus attack remarkably outperforms RLSB attack on all the settings, showing simultaneously modeling the key frames and key regions is indeed more effective than separately modeling them. This also demonstrates the core idea in this paper. *(4) From the view of robustness evaluation*, all the seven black-box video attacks show C3D has better adversarial robustness than the other models. The C3D has lower FR values but higher NQ values, which shows C3D is harder to attack. This may motivates us an in-depth study for the C3D's structure to design robust video recognition models.

H. Integrated With Other Gradient Estimators

In our AstFocus attack, the current gradient estimator is NES. Actually, we can replace NES with other state-of-the-art gradient estimators. To test this point, we conduct experiments. Here we choose two SOTA gradient estimators: Prior convictions [49] and ZOO [50]. Fig. 12 gives the results. We can see that when the gradient estimators are changed, the fooling rate, query number, and perturbation magnitudes only show a slight variation. Relatively speaking, NES achieves the better performance versus three metrics. AstFocus attack is a flexible framework, which implies other modules can be replaced except the MARL module. The PGD can also be replaced with its improved versions.

I. Qualitative Results of AstFocus Attacks

We list two adversarial videos and the perturbations generated by AstFocus attacks in Fig. 13, we see adversarial video is consistent with original video from the appearance, showing the

TABLE V
THE COMPARATIVE RESULTS VERSUS FOUR DIFFERENT THREAT MODELS ON UCF-101 DATASET. THE BEST RESULTS ARE HIGHLIGHTED IN RED. THE SYMBOL "-" MEANS THE USED NQ EXCEEDS THE MAXIMUM NQ. ↑ DENOTES THE LARGER, THE BETTER, AND ↓ DENOTES THE SMALLER, THE BETTER

Datasets	Threat Models	Attack Methods	Un-targeted attacks				Targeted attacks			
			MAP ↓	NQ ↓	FR ↑	T(s)↓	MAP↓	NQ↓	FR ↑	T(s)↓
UCF-101	TSM [13]	VBAD attack [19]	6.023	2803	84%	31.2	9.208	15304	63%	**201.7**
		Heuristic attack [20]	5.956	10657	40%	212.9	–	–	–	–
		Sparse attack [17]	**3.417**	8529	58%	147.2	–	–	–	–
		Motion-sampler attack [30]	7.237	5187	83%	124.6	7.415	13577	79%	288.6
		GEO-TRAP attack [31]	5.865	3782	88%	87.2	6.265	11494	84%	247.6
		RLSB attack [21]	4.823	4898	87%	101.3	7.274	20532	40%	365.4
		AstFocus attack (ours)	3.355	**1138**	**96%**	**24.6**	**4.546**	**8064**	**100%**	274.5
	TSN [12]	VBAD attack [19]	6.168	2450	84%	**13.8**	9.391	18960	47%	394.6
		Heuristic attack [20]	5.265	9135	51%	141.7	–	–	–	–
		Sparse attack [17]	**3.131**	6916	64%	181.4	–	–	–	–
		Motion-sampler attack [30]	6.895	4744	78%	95.6	6.903	19626	59%	316.8
		GEO-TRAP attack [31]	5.472	3782	75%	87.3	5.952	18585	55%	301.2
		RLSB attack [21]	5.238	3504	93%	48.2	8.448	20668	44%	289.5
		AstFocus attack (ours)	3.265	**2015**	**99%**	37.4	**4.495**	**8483**	**76%**	**272.9**
	C3D [11]	VBAD attack [19]	6.800	4890	75%	**43.4**	10.760	20234	60%	**139.2**
		Heuristic attack [20]	6.295	14160	30%	143.2	–	–	–	–
		Sparse attack [17]	**3.009**	9507	42%	86.0	–	–	–	–
		Motion-sampler attack [30]	6.153	8132	62%	97.9	7.242	20690	47%	252.9
		GEO-TRAP attack [31]	5.877	7045	75%	74.4	6.332	17340	77%	205.4
		RLSB attack [21]	5.326	6568	68%	72.0	7.225	22018	35%	207.1
		AstFocus attack (ours)	4.015	**4224**	**90%**	66.8	**4.225**	**13470**	**88%**	236.4
	SlwoFast [43]	VBAD attack [19]	6.302	4089	77%	42.5	9.118	19423	53%	499.2
		Heuristic attack [20]	5.869	12776	34%	260.5	–	–	–	–
		Sparse attack [17]	**3.164**	8642	58%	168.8	–	–	–	–
		Motion-sampler attack [30]	7.086	5166	77%	136.3	7.275	17928	56%	451.8
		GEO-TRAP attack [31]	5.712	4334	84%	120.5	6.273	16506	62%	532.1
		RLSB attack [21]	5.586	4563	85%	94.5	7.655	22552	43%	448.4
		AstFocus attack (ours)	4.286	**1435**	**93%**	**35.7**	**4.436**	**13660**	**85%**	**385.3**

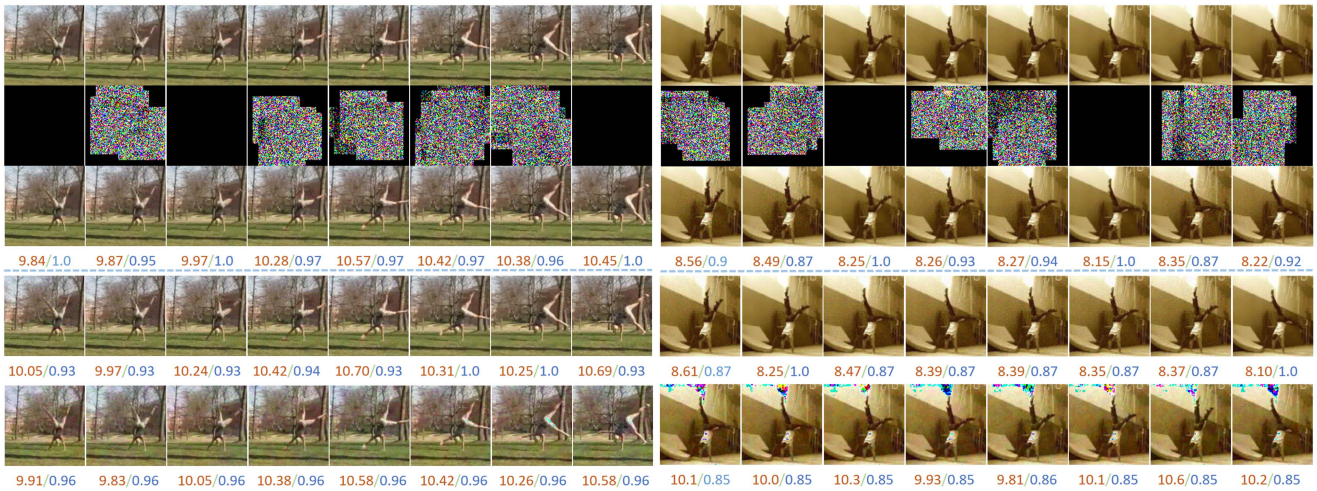


Fig. 13. Two qualitative examples output by AstFocus attacks. For each example, from top to bottom rows are clean video, adversarial perturbations, and our adversarial video, respectively. Two adversarial videos generated by RLSB attack and GEO-TRAP attack are listed below the dotted line as a reference. We compute two metrics as (blurriness/SSIM) for each video frame. For the left blurriness degree [45], the smaller the better, and for the right SSIM [46], the larger the better.

TABLE VI
THE COMPARATIVE RESULTS VERSUS FOUR DIFFERENT THREAT MODELS ON HMDB-51 DATASET. THE BEST RESULTS ARE HIGHLIGHTED IN RED. THE SYMBOL "-" MEANS THE USED NQ EXCEEDS THE MAXIMUM NQ. ↑ DENOTES THE LARGER, THE BETTER, AND ↓ DENOTES THE SMALLER, THE BETTER

Datasets	Threat Models	Attack Methods	Un-targeted attacks				Targeted attacks			
			MAP ↓	NQ ↓	FR ↑	T(s)↓	MAP↓	NQ↓	FR ↑	T(s)↓
HMDB-51	TSM [13]	VBAD attack [19]	6.361	1818	92%	**22.4**	9.057	17550	60%	495.6
		Heuristic attack [20]	5.043	10385	58%	211.4	–	–	–	–
		Sparse attack [17]	**3.334**	6244	62%	102.5	–	–	–	–
		Motion-sampler attack [30]	7.229	3911	90%	95.8	8.012	19508	58%	686.7
		GEO-TRAP attack [31]	5.919	3164	92%	84.8	6.222	10836	84%	**337.7**
		RLSB attack [21]	5.323	5950	82%	112.9	7.754	20171	28%	575.8
		AstFocus attack (ours)	3.411	**1529**	**100%**	34.7	**4.326**	**7319**	**92%**	419.1
	TSN [12]	VBAD attack [19]	5.873	2373	90%	**26.4**	9.244	21795	46%	659.1
		Heuristic attack [20]	5.395	10146	58%	172.5	–	–	–	–
		Sparse attack [17]	**3.271**	6765	74%	105.9	–	–	–	–
		Motion-sampler attack [30]	7.275	3667	88%	74.1	8.087	24332	28%	964.6
		GEO-TRAP attack [31]	5.192	3392	88%	61.6	6.344	21492	36%	744.6
		RLSB attack [21]	5.312	4217	92%	68.6	6.494	22718	22%	719.1
		AstFocus attack (ours)	3.52	**2198**	**96%**	51.1	**4.090**	**9953**	**74%**	**522.4**
	C3D [11]	VBAD attack [19]	6.743	4107	78%	**36.5**	10.528	22302	64%	**361.2**
		Heuristic attack [20]	4.838	10534	42%	117.4	–	–	–	–
		Sparse attack [17]	**2.983**	8545	46%	73.8	–	–	–	–
		Motion-sampler attack [30]	7.035	6491	68%	89.5	7.973	22199	44%	558.4
		GEO-TRAP attack [31]	5.666	5082	84%	48.2	6.324	16374	74%	518.2
		RLSB attack [21]	4.688	7279	62%	77.1	7.212	18190	60%	530.8
		AstFocus attack (ours)	3.835	**3628**	**92%**	45.6	**4.025**	**9997**	**86%**	473.6
	SlwoFast [43]	VBAD attack [19]	6.528	5442	72%	**46.7**	10.615	22955	36%	693.3
		Heuristic attack [20]	5.875	9094	54%	162.9	–	–	–	–
		Sparse attack [17]	**3.228**	8977	56%	138.0	–	–	–	–
		Motion-sampler attack [30]	7.163	6553	74%	156.4	7.956	18513	52%	652.3
		GEO-TRAP attack [31]	6.242	5741	78%	122.8	6.179	17636	46%	666.1
		RLSB attack [21]	5.68	4495	84%	81.9	7.416	20378	34%	650.0
		AstFocus attack (ours)	4.078	**2295**	**96%**	47.1	**4.682**	**13970**	**78%**	**567.9**

TABLE VII
THE COMPARATIVE RESULTS VERSUS FOUR DIFFERENT THREAT MODELS ON KINETICS-400 DATASET. THE BEST RESULTS ARE HIGHLIGHTED IN RED. THE SYMBOL "-" MEANS THE USED NQ EXCEEDS THE MAXIMUM NQ. ↑ DENOTES THE LARGER, THE BETTER, AND ↓ DENOTES THE SMALLER, THE BETTER

Datasets	Threat Models	Attack Methods	Un-targeted attacks				Targeted attacks			
			MAP ↓	NQ ↓	FR ↑	T(s)↓	MAP↓	NQ↓	FR ↑	T(s)↓
Kinetics-400	TSM [13]	VBAD attack [19]	6.480	3626	78%	**16.8**	10.338	23670	34%	593.6
		Heuristic attack [20]	5.744	11918	56%	202.1	–	–	–	–
		Sparse attack [17]	**2.725**	9105	60%	147.6	–	–	–	–
		Motion-sampler attack [30]	7.234	4494	88%	105.9	8.033	21032	46%	803.1
		GEO-TRAP attack [31]	6.007	3962	92%	87.1	6.217	15585	72%	**536.0**
		RLSB attack [21]	5.856	4422	84%	91.4	7.200	21724	26%	653.2
		AstFocus attack (ours)	3.658	**2416**	**96%**	44.1	**4.482**	**9758**	**88%**	556.5
	TSN [12]	VBAD attack [19]	5.764	1668	92%	**22.6**	9.414	17560	58%	**427.9**
		Heuristic attack [20]	4.806	10080	52%	165.6	–	–	–	–
		Sparse attack [17]	**2.579**	8212	54%	117.8	–	–	–	–
		Motion-sampler attack [30]	6.982	3422	90%	80.3	7.979	22119	36%	735.2
		GEO-TRAP attack [31]	5.636	2684	90%	53.1	5.789	16402	50%	459.7
		RLSB attack [21]	5.395	3774	94%	63.7	7.140	23574	22%	622.4
		AstFocus attack (ours)	3.349	**1021**	**100%**	26.7	**4.684**	**9940**	**90%**	558.4
	C3D [11]	VBAD attack [19]	5.640	3444	90%	13.2	10.230	22070	52%	135.2
		Heuristic attack [20]	5.805	11384	48%	44.7	–	–	–	–
		Sparse attack [17]	**2.769**	5045	78%	20.7	–	–	–	–
		Motion-sampler attack [30]	6.895	2485	96%	14.1	7.808	15679	70%	163.2
		GEO-TRAP attack [31]	6.135	3436	96%	15.6	6.334	12260	90%	**120.5**
		RLSB attack [21]	4.925	5915	76%	25.2	8.053	20975	36%	179.6
		AstFocus attack (ours)	3.858	**1055**	**100%**	**9.1**	**4.728**	**10840**	**92%**	152.9
	SlwoFast [43]	VBAD attack [19]	6.667	2732	86%	33.6	10.532	19970	44%	599.2
		Heuristic attack [20]	4.723	9154	54%	159.5	–	–	–	–
		Sparse attack [17]	**3.043**	5901	70%	97.8	–	–	–	–
		Motion-sampler attack [30]	7.145	2282	92%	55.4	7.953	20264	40%	878.3
		GEO-TRAP attack [31]	5.832	1646	94%	37.5	6.197	9594	86%	366.8
		RLSB attack [21]	4.636	4802	88%	88.1	7.405	21137	34%	705.4
		AstFocus attack (ours)	3.356	**851**	**100%**	**24.4**	**4.015**	**7572**	**98%**	**386.2**

TABLE VIII
RESULTS OF ASTFOCUS ATTACK AGAINST DEFENDED C3D METHOD ON HMDB-51

Metrics	No defense	PGD-AT [24]	OUD [47]	AdvIT [48]
FR(%)	92.0	60.0 (↓32)	72.0 (↓20)	70.0 (↓22)
QN	3628	7005 (↑3377)	6283 (↑2655)	2802 (↓826)
MAP	3.835	4.814 (↑0.979)	5.090 (↑1.255)	3.512 (↓0.323)

imperceptibility of adversarial perturbations. To understand the perturbations, we enlarge their values to give a display. We see the final adversarial perturbations are sparse both in inter-frames and intra-frames. They show a superposition phenomenon by many noise patches generated in each PGD iteration. These adversarial perturbations cover the foreground regions in key frames. We also give two adversarial videos generated by other recently published attack methods (RLSB attack and GEO-TRAP attack) as a reference, where we can see our method has better imperceptible perturbations than the other methods.

To better show the advantage, we compute two metrics to quantitatively measure the image quality. The first metric is to measure the blurriness degree [45]. For this metric, the smaller the better. And another metric is SSIM [46]. For this metric, the larger the better. We list these two metric values below each video frame as (blurriness/SSIM), where we see that our adversarial videos show better image quality than RLSB and GEO-TRAP attacks.

J. AstFocus Attack Against Defense Methods

We evaluate the performance of AstFocus attack against defense methods. Three kinds of representative video defense methods are chosen: Adversarial Training method (PGD-AT [24]), modifying network architecture method (OUD [47]), and pre-processing method (AdvIT[3] [48]). The results for C3D model on HMDB-51 dataset are reported in Table VIII, where the changes compared with the un-defended C3D are listed in the brackets. We can see that both the attacking performance and efficiency decrease. Specifically, the maximum drop of FR after defense is 32%, QN increases by 3377 at most, and MAP increases by 33% at most. This is reasonable because the defended model will be harder to attack, but the FR, QN, and MAP are still acceptable. This shows that AstFocus attack is effective to evaluate the adversarial robustness even for the defended action recognition models.

V. CONCLUSION

In this article, we designed the novel adversarial spatial-temporal focus attack on videos to simultaneously identify the key frames and key regions in the video. AstFocus attack was based on the cooperative multi-agent reinforcement learning framework. One agent was responsible for selecting key frames, and another agent was responsible for selecting key regions. These two agents were jointly trained by the common rewards received from the black-box threat models. By continuously querying, the reduced space composed of key frames and key regions was becoming precise, and the whole query number was less than that on the original video. Extensive experiments on four famous video recognition models and three public action recognition datasets verified our efficiency and effectiveness, which was prevenient in fooling rate, query number, time, and perturbation magnitude at the same time.

[3]AdvIT is proposed to detect the adversarial example. To adopt it to perform defense, we attach it before the threat model. If the input is detected as adversarial example, it will not be fed into the threat model. For this reason, the QN and MAP may decrease rather than increase.

REFERENCES

[1] W. Wang, Q. Lai, H. Fu, J. Shen, H. Ling, and R. Yang, "Salient object detection in the deep learning era: An in-depth survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 6, pp. 3239–3259, Jun. 2022.

[2] Y. Zhu et al., "A comprehensive study of deep video action recognition," 2020, *arXiv:2012.06567*.

[3] S. Yang, W. Wang, C. Liu, and W. Deng, "Scene understanding in deep learning-based end-to-end controllers for autonomous vehicles," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 49, no. 1, pp. 53–63, Jan. 2019.

[4] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014, *arXiv:1412.6572*.

[5] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, "Adversarial examples are not bugs, they are features," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 125–136.

[6] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 86–94.

[7] H. Wang et al., "Understanding the robustness of skeleton-based action recognition under adversarial attack," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 14651–14660.

[8] S. Tang et al., "RobustART: Benchmarking robustness on architecture design and training techniques," 2021, *arXiv:2109.05211*.

[9] S. Geisler, T. Schmidt, H. Şirin, D. Zügner, A. Bojchevski, and S. Günnemann, "Robustness of graph neural networks at scale," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 7637–7649.

[10] U.-A. M. Chapman-Rounds, U. Bhatt, E. Pazos, M.-A. Schulz, and K. Georgatzis, "FIMAP: Feature importance by minimal adversarial perturbation," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 11433–11441.

[11] K. Hara, H. Kataoka, and Y. Satoh, "Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and ImageNet?," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6546–6555.

[12] L. Wang et al., "Temporal segment networks: Towards good practices for deep action recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 20–36.

[13] J. Lin, C. Gan, and S. Han, "TSM: Temporal shift module for efficient video understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 7083–7093.

[14] Y. Dong, S. Cheng, T. Pang, H. Su, and J. Zhu, "Query-efficient black-box adversarial attacks guided by a transfer-based prior," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 12, pp. 9536–9548, Dec. 2022.

[15] X. Wei, Y. Guo, and J. Yu, "Adversarial sticker: A stealthy attack method in the physical world," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 3, pp. 2711–2725, Mar. 2023.

[16] X. Wei, J. Zhu, S. Yuan, and H. Su, "Sparse adversarial perturbations for videos," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 8973–8980.

[17] X. Wei, H. Yan, and B. Li, "Sparse black-box video attack with reinforcement learning," *Int. J. Comput. Vis.*, vol. 130, pp. 1459–1473, 2022.

[18] J. Hwang, J.-H. Kim, J.-H. Choi, and J.-S. Lee, "Just one moment: Structural vulnerability of deep action recognition against one frame attack," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 7668–7676.

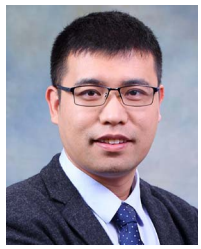
[19] L. Jiang, X. Ma, S. Chen, J. Bailey, and Y.-G. Jiang, "Black-box adversarial attacks on video recognition models," in *Proc. ACM Int. Conf. Multimedia*, 2019, pp. 864–872.

[20] Z. Wei et al., "Heuristic black-box adversarial attacks on video recognition models," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 12338–12345.

[21] Z. Wang, C. Sha, and S. Yang, "Reinforcement learning based sparse black-box adversarial attack on video recognition models," 2021, *arXiv:2108.13872*.

[22] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin, "Black-box adversarial attacks with limited queries and information," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 2142–2151.

[23] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6379–6390.
[24] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," 2017, *arXiv:1706.06083*.
[25] D. Wierstra, T. Schaul, J. Peters, and J. Schmidhuber, "Natural evolution strategies," in *Proc. IEEE Congr. Evol. Comput.*, 2008, pp. 3381–3387.
[26] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017.
[27] H. Yan and X. Wei, "Efficient sparse attacks on videos using reinforcement learning," in *Proc. ACM Int. Conf. Multimedia*, 2021, pp. 2326–2334.
[28] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial examples: Attacks and defenses for deep learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 9, pp. 2805–2824, Sep. 2019.
[29] S. Li et al., "Adversarial perturbations against real-time video classification systems," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2019, pp. 1–15.
[30] H. Zhang, L. Zhu, Y. Zhu, and Y. Yang, "Motion-excited sampler: Video adversarial attack with sparked prior," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 240–256.
[31] S. Li et al., "Adversarial attacks on black box video classifiers: Leveraging the power of geometric transformations," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 2085–2096.
[32] Z. Wei, J. Chen, Z. Wu, and Y. Jiang, "Cross-modal transferable adversarial attacks from images to videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 15 064–15 073.
[33] Z. Wu, X. Wang, Y.-G. Jiang, H. Ye, and X. Xue, "Modeling spatial-temporal clues in a hybrid deep learning framework for video classification," in *Proc. ACM Int. Conf. Multimedia*, 2015, pp. 461–470.
[34] L. Yuan et al., "Tokens-to-token ViT: Training vision transformers from scratch on ImageNet," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 538–547.
[35] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 391–405.
[36] K. Zhou, Y. Qiao, and T. Xiang, "Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 7582–7589.
[37] V. Konda and J. Tsitsiklis, "Actor-critic algorithms," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 1999, pp. 1008–1014.
[38] M. T. Spaan, "Partially observable Markov decision processes," in *Reinforcement Learning*. Berlin, Germany: Springer, 2012, pp. 387–414.
[39] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.
[40] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," 2012, *arXiv:1212.0402*.
[41] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: A large video database for human motion recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 2556–2563.
[42] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4724–4733.
[43] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "SlowFast networks for video recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 6202–6211.
[44] M. Contributors, "OpenMMLab's next generation video understanding toolbox and benchmark," 2020. [Online]. Available: https://github.com/open-mmlab/mmaction2
[45] P. Marziliano, F. Dufaux, S. Winkler, and T. Ebrahimi, "A no-reference perceptual blur metric," in *Proc. Int. Conf. Image Process.*, 2002, pp. III–III.
[46] A. Hore and D. Ziou, "Image quality metrics: PSNR vs. SSIM," in *Proc. 20th Int. Conf. Pattern Recognit.*, 2010, pp. 2366–2369.
[47] S.-Y. Lo, J. M. J. Valanarasu, and V. M. Patel, "Overcomplete representations against adversarial videos," in *Proc. IEEE Int. Conf. Image Process.*, 2021, pp. 1939–1943.
[48] C. Xiao et al., "AdvIT: Adversarial frames identifier based on temporal consistency in videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3968–3977.
[49] A. Ilyas, L. Engstrom, and A. Madry, "Prior convictions: Black-box adversarial attacks with bandits and priors," 2018, *arXiv:1807.07978*.
[50] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, "ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models," in *Proc. 10th ACM Workshop Artif. Intell. Secur.*, 2017, pp. 15–26.



Xingxing Wei (Member, IEEE) received the BS degree in automation from Beihang University, China, and the PhD degree in computer science from Tianjin University. He is now an associate professor with Beihang University (BUAA). His research interests include computer vision, adversarial machine learning and its applications to multimedia content analysis. He is the author of referred journals and conferences in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *IEEE Transactions on Multimedia*, *IEEE Transactions on Cybernetics*, *IEEE Transactions on Geoscience and Remote Sensing*, *International Journal of Computer Vision*, *Pattern Recognition*, *Computer Vision and Image Understanding*, CVPR, ICCV, ECCV, ACMMM, AAAI, IJCAI etc.

Songping Wang is currently working toward the master degree with the School of Software, Beihang University (BUAA). His research interests include deep learning and adversarial robustness in machine learning.

Huanqian Yan received the bachelor's degree in the field of computer science and technology from the Changchun University of Science and Technology, in 2015, and the master's degree in computer application and technology from Lanzhou University, in 2018. He is currently working toward the PhD degree with the School of Computer Science and Engineering, Beihang University, Beijing, China. His current research interests include object detection, adversarial examples, and clustering analysis, etc.