Appending Adversarial Frames for Universal Video Attack

Zhikai Chen
Xi'an Jiaotong University
zhikai_chen@outlook.com

Lingxi Xie
Huawei Noah's Ark Lab
198808xc@gmail.com

Shanmin Pang✉
Xi'an Jiaotong University
pangsm@xjtu.edu.cn

Yong He
Xi'an Jiaotong University
hy0275@stu.xjtu.edu.cn

Qi Tian
Huawei Noah's Ark Lab
tian.qi1@huawei.com

Abstract

This paper investigates the problem of generating adversarial examples for video classification. We project all videos onto a semantic space and a perception space, and point out that adversarial attack is to find a counterpart which is close to the target in the perception space but far from the target in the semantic space. Based on this formulation, we notice that conventional attacking methods mostly used Euclidean distance to measure the perception space, but we propose to make full use of the property of videos and assume a modified video with a few consecutive frames replaced by dummy contents (e.g., a black frame with texts of 'thank you for watching' on it) to be close to the original video in the perception space though they have a large Euclidean gap. This leads to a new attack approach which only adds perturbations on the newly-added frames. We show its high success rates in attacking six state-of-the-art video classification networks, as well as its universality, i.e., transferring well across videos and models.

1. Introduction

Deep neural networks, while being powerful in learning from complicated visual data, are vulnerable to small noise known as adversarial perturbations. Researchers designed a lot of attacking algorithms to add imperceptible perturbations onto well-trained neural networks so that the prediction is dramatically destroyed. Successful scenarios include image classification [17, 22, 20], object detection and semantic segmentation [34], super-resolution [8], visual question answering [35], image captioning [8], *etc*. Researchers conjectured that adversaries are closely related to the working mechanism as well as explainability of deep neural networks [30, 11], and both adversarial attack and defense have been attracting attentions in both academia and industry.

Compared to adversarial attacks for visual recognition on still images, the same topic on video data has been fewer investigated. There exist a few video-based attack methods [33, 15, 18], but they mostly added perturbations to individual video frames, as if video is a kind of high-dimensional data with no internal structures. Having ignored the property that video is a sequence of images and neighboring frames are closely correlated, these attack methods suffer either high perceptibility or limited transferability across videos or models.

This paper studies this problem from a new perspective. We first define two spaces named semantic space and perception space, respectively, where the former is determin-

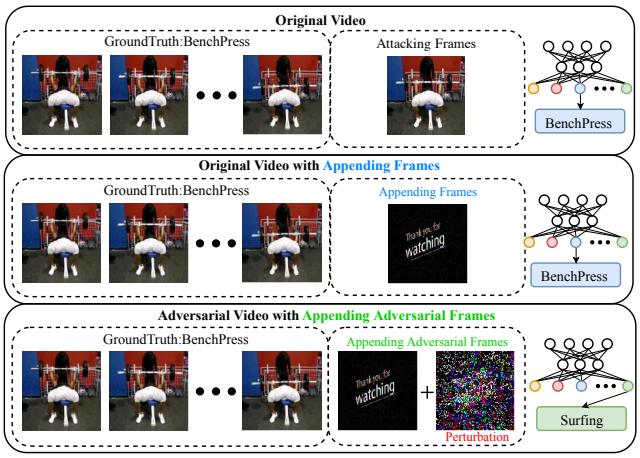

Figure 1: We generate adversarial examples for video classification by replacing the ending part of the input clip with a few dummy frames, *i.e.*, black images with texts of 'thank you for watching'. The video clip was originally (correctly) recognized as 'BenchPress'. While adding dummy frames alone does not alter the classification result, adding slight perturbations on them changes the prediction to 'Surfing' (a success attack).

istic provided the target model (*e.g.*, for video classification) but the latter involves human perception which is quite subjective and difficult to depict, *i.e.*, defining the distance between two elements in the space. For image-based adversarial attack, people often use the l_2 distance to approximate the perception space, but we notice that video-based attack can benefit from an intriguing property: if some consecutive frames of a video are replaced by some dummy contents (*e.g.*, a black frame with texts of 'thank you for watching' on it), people may not perceive that it is a attack. In other words, the perception space has an intriguing property that lies in its subspaces (*e.g.*, a few consecutive frames): the points that correspond to the original and dummy clips have nearly a zero distance in the subspace.

Making full use of this observation, we present a novel video attack method named **appending adversarial frames** (A^2F) and demonstrate its effectiveness on the video classification task. As shown in Fig. 1, the idea is very simple: given a target video, we first replace a few consecutive frames (*e.g.*, in the end of the video) by dummy frames, and then add adversarial perturbations **only** to these dummy frames. Note that the first step often does not fool the deep network, but pushes the target video towards the classification border, so that the second step can achieve the attack with very small perturbations. From the conventional perspective, A^2F has large perceptibility (measured in the Euclidean space), but unless educated beforehand, people may not notice that the black frames come from manual editing.

Compared to a regular attack, our approach enjoys threefold benefits. **First**, a high success rate. Note that the dummy frames do not contain any semantic information, *i.e.*, lying close to the semantic border, so that pushing it towards any class requires much fewer efforts and thus easier to accomplish. **Second**, a low perceptibility, *e.g.*, in terms of pixel-level difference or the number of iterations, which is for the same reason. **Third** and most importantly, a strong ability in transferring across different scenarios. Since the perturbations are added beyond these common, class-agnostic frames, we shall expect that the noise is highly associated to the target class, and thus it produces the same classification result when moved to a new video or even a new classification network. This paves the way of **universal** adversarial attack which is believe to be more challenging yet threatening for real-world scenarios.

We evaluate our approach on two popular video classification datasets, namely, UCF101 [27] and HMDB51 [16]. We select six victim models from a wide range of video classification methods, including CNN+LSTM [10], C3D [31], ResNet3D [12], P3D [25], and I3D [4] on both ResNet [13] and Inception [29]. We start with white-box attacks, in which the parameters of the victim models are known to the attacker, and then transfer the perturbations computed on each video to other unseen models, *i.e.*, black-box attacks. Our approach enjoys superior success rate while the perturbations added to the dummy frames are much smaller than that added by the baseline approaches to the original video frames.

The main contributions of this paper are as follows:

- We provide a new insight that understands adversarial attacks in the semantic and perception spaces, from which we point out that video attack should be considered differently from image attack, since the perception space can be decomposed according to individual frames.
- We propose to append adversarial frames to videos and demonstrate that the idea is effective in different scenarios, including having a high success rate and being less perceptible to observers.
- We further discuss different application scenarios of A^2F, with the most important one being black-box attack, *i.e.*, transferring the attack across different target videos and victim models, which we show the promise of A^2F being a universal attack method.

2. Related Work

Video Recognition. The power of deep learning architectures is not only shown in image classification (ImageNet [9]), but also shown in video recognition. There are many successful video recognition models. For instance, Donahue *et al.* [10] proposed a class of recurrent long-term models that can be jointly trained to learn temporal dynamics and convolutional perceptual representations, and demonstrated superior performance on recognition and description of images and videos. Tran *et al.* [31] addressed the problem of learning spatio tempora features for videos using 3D ConvNets. Noting that the training of 3D ConvNets is computationally extensive, Carreira *et al.* [4] introduced a two-stream Inflated 3D (I3D) ConvNets, which built upon on 2D kernels but inflated filters and pooling kernels into 3D. Another representative strategy to reduce training cost was proposed in [25], which simulated 3D convolutions with 2D convolutions on spatial domain plus 1D convolutions on adjacent feature maps in time. Recently, Hara *et al.* [12] examined the architectures of various 3D CNNs from relatively shallow networks to very deep ones on current video datasets.

Adversarial Attacks on Images. There are fruitful attack methods in the literature. Among the first to introduce adversarial examples against deep neural networks was [30]. After that, Goodfellow *et al.* [11] used the sign of the gradient to propose a fast attack method called Fast Gradient Sign Method (FGSM). FGSM seeks the direction that can maximize the classification errors to update each pixel. The subsequent method I-FGSM [17] extends FGSM by more iterations, and can generate adversarial examples

in the physical world. In [20], an iterative method called Projection Gradient Descent (PGD) was proposed. PGD makes the perturbations project back to the ϵ-ball which center is the original data when perturbations over the ϵ-ball. Moosavi *et al.* [22] proposed *DeepFool* to find the closest distance from the original input to the decision boundary of adversarial examples. Moosavi *et al.* [21] analyzed universal perturbations and its relationship between different classification regions of decision boundary. Liu *et al.* [19] studied the transferability of both non-targeted and targeted adversarial examples, and proposed an ensemble-based approaches to generate adversarial examples with stronger transferability. Sabour *et al.* [26] performed a targeted attack by minimizing the distance of the representation of intermediate neural network layers instead of the output layer. There also exists other methods for white-box attack, *e.g.* Jacobian-based Saliency Map Attack (JSMA) [24] and Elastic net attack (EAD) [6]. Except the white-box image attack, some black-box attack methods *e.g.* Zeroth-order optimization attack (ZOO) [7] used a black-box method to estimate the adversarial gradient. HopSkipJumpAttack [5], which estimated the gradient direction using binary information at the decision boundary, is also a family of black-box algorithms. Naseer *et al.* [23] indicate that we can enhance adversarial transferability by maximize distortions in the network feature space.

Adversarial Attacks on Videos. There are several attack methods proposed for generating video adversarial examples. Wei *et al.* [33] claimed that they were the first to explore the adversarial examples in videos. In their paper, they mainly investigated the sparsity and propagation of adversarial perturbations across video frames. Recently,Li Sabour *et al.* [18] showed that we can use a Generative Adversarial Networks (GANs) like architecture to generate perturbations in real-time video classifier. Jiang *et al.* [15] was the first work on black-box video attacks against video recognition models. In this paper, we use an optimization based method to find adversarial perturbations. Being different from previous works that add perturbations to the original videos, we append the adversarial frames to the original videos. To our knowledge, we are the first to propose to append adversarial frames to videos. Furthermore, to make the attack comparison more reasonable, we modify the attacking method [33] to be *adaptive attacks* [3, 2, 14] when evaluating on the effectiveness of our method.

3. Our Approach

3.1. Adversarial Attack, Semantic and Perception Spaces

We use $\mathbf{J}(\cdot;\boldsymbol{\theta})$ to denote the threat model with $\boldsymbol{\theta}$ indicating network parameters. We always consider the threat model as a deep neural network, which means the classifier $\mathbf{J}(\cdot;\boldsymbol{\theta})$ is a complicated yet differentiable function. Let $\mathbf{X}$ be the input video. Without loss of generality, we assume $\mathbf{X} \in \mathbb{R}^{T\times W\times H\times D}$, where T denotes the number of frames, and W, H, and D denote the width, height, and the number of channels of each frame, respectively. From the conventional perspective, the goal of adversarial attack is to find an adversarial example, $\hat{\mathbf{X}}$, which shares the same dimensionality with $\mathbf{X}$, looks identical to $\mathbf{X}$, but $\mathbf{J}(\mathbf{X};\boldsymbol{\theta})$ and $\mathbf{J}(\hat{\mathbf{X}};\boldsymbol{\theta})$ are very different, *e.g.*, after hard quantization, classified into different classes.

Here, we provide an alternative formulation of this problem. We assume that the threat model, $\mathbf{J}(\mathbf{X};\boldsymbol{\theta})$, defines a C-dimensional **semantic space** so that each video can be projected onto the space, *i.e.*, being assigned with a class distribution. In addition, we also project each video onto a perception space, in which the distance between two elements, say, $\mathbf{X}$ and $\hat{\mathbf{X}}$, indicates how people can perceive that $\hat{\mathbf{X}}$ is a perturbed counterpart of $\mathbf{X}$. Given a fixed threat model, the semantic space is mostly deterministic, where we use $\ell(\mathbf{1}_y, \mathbf{J}(\hat{\mathbf{X}};\boldsymbol{\theta}))$ ($\mathbf{1}_y$ is a one-hot vector with the y-th class being 1, and y is the dominant class of $\mathbf{J}(\hat{\mathbf{X}};\boldsymbol{\theta})$) to measure the distance between $\mathbf{X}$ and $\hat{\mathbf{X}}$, where $\ell(\cdot,\cdot)$ is the cross-entropy loss function. However, the perception space is subjective and difficult to depict – one cannot easily come up with a standard of perceptibility, and researchers mostly used the Euclidean space to approximate the perception space, *i.e.*, the distance between $\mathbf{X}$ and $\hat{\mathbf{X}}$ is measured by $\|\hat{\mathbf{X}} - \mathbf{X}\|_2$.

This formulation leads to the so-called basic attack (BA) method, which was introduced in [33] and follow-up approaches later [15, 18] added frame-level constraints to it. These methods all adding their perturbations to original frames, which is difficult in some situations as manipulating the original videos needs a high authority. As for the attack transferability, these methods almost perform not well when attacking across models. From the semantic understanding perspective, this is because the orientations of perturbations in different models are always random, while a primary across models settings will not be able to constrain the perturbations to a specific orientation.

On the other hand, we argue that the internal structure of the video should be considered in generating effective adversarial attacks, yet the BA method does not make full use of it, but simply regard the video as an arbitrary type of high-dimensional data. This motivates us to find a better solution.

3.2. An Intriguing Property of the Video Perception Space

The key is to notice that for video attack, the perception space has an intriguing property which has not been exploited. Note that video allows the scenario to be switched sparsely, which means that one can replace a consecutive

part of a video with another, yet this operation may not be perceived by the observer if the semantics of the original clip and the replacement do not heavily conflict with each other. Note that this property does not hold for still images, as one often expects an image to have pixel-level continuity, *e.g.*, most observers are sensitive to the change that an image patch is replaced by another.

This assumption changes the conventional definition of the perception space. We no longer measure it solely by the Euclidean distance, but allow some 'shortcut connections' in the space. Specifically, let a set of consecutive frames define a subspace, then we assume that all video clips in the subspace, without heavy semantic conflict, have a zero distance in the perception space. Note that this is a complementary depiction of the perception space. Though not complete, it is sufficient in generating video-based adversarial examples efficiently, as shown in later parts.

We make a side comment here. The essence is to make use of the internal structure of video data (each frame can be a relatively individual semantic unit) and thus investigate the perception space from its subspaces. Such decomposition also affects the semantic space, which we shall see in Section 3.6 that if the replaced frames do not contain semantic information, the video, in the semantic space, will be pushed towards the border of classification. This indeed creates convenience for generating universal perturbations.

3.3. Video Attack by Appending Adversarial Frames

Inspired by the above analyses, we propose a novel way of attacking videos. It is named the **Appending Adversarial Frames ($\mathbf{A^2F}$)** method, which replaces a few consecutive frames of the target video with dummy frames, and add perturbations **only** on these new frames. Here, a dummy frame should satisfy two conditions, namely, it does not contain any semantic information, yet it is not likely to cause the observer doubt that the video has been manually edited. We investigate a simple choice, that the frame is an all-black image with texts of 'thank you for watching' displayed on it. Such a frame is commonly seen at the end of a video, so most people may not notice it is an attacked video[1].

Mathematically, let $\hat{\mathbf{X}} = \{f_1, f_2, \ldots, f_{T-\Delta T}, \Delta\}$, where $\Delta \in \mathbb{R}^{\Delta T \times W \times H \times C}$ be the appending dummy frames without perturbations, and $\hat{\Delta}$ be its adversarial frames with perturbations. Thus, $\mathbf{E} = \hat{\Delta} - \Delta$ is the added adversarial perturbations.

We use $\mathbf{A^2F}$ to find $\mathbf{E}$ to maximize the difference between the output from video classification models which take $\hat{\mathbf{X}}$ and $\mathbf{X}$ as input:

$$\arg\min_{\mathbf{E}} \lambda||\mathbf{E}||_p - \ell(\mathbf{1}_y, \mathbf{J}(\hat{\mathbf{X}}; \boldsymbol{\theta})) \tag{1}$$

where $||\mathbf{E}||_p$ is the ℓ_p norm (we select $p = \infty$ in this paper) of $\mathbf{E}$, which is used to measure the magnitude of the adversarial perturbations. The parameter λ is the weight applied to balance different items in the objective function.

If the goal is to misclassify the generated adversarial video $\hat{\mathbf{X}}$ to a predefined label (*i.e.*, the target label). then we can modify the problem to minimize the difference between the target label and the prediction.

$$\arg\min_{\mathbf{E}} \lambda||\mathbf{E}||_p + \ell(\mathbf{1}_{y^*}, \mathbf{J}(\hat{\mathbf{X}}; \boldsymbol{\theta})) \tag{2}$$

where y^* is the targeted label.

With $\mathrm{A^2F}$, we develop a series of variants to overcome different problems in various attack scenarios.

3.4. Spatial Mask Attack

It is important for us to control the perturbation to make it more imperceptive. Su *et al.* [28] find that only modify one pixel can also generate adversarial examples and Brown *et al.* [1] generate an adversarial patch which can be added to a part of the original image and make it being misclassified, *etc*. Those works all suggested that we can do a part attack in our attack conditions. To the end, we apply the idea of the spatial mask to decorate perturbations called **$\mathbf{A^2F}$-SM**. In particular,

$$\arg\min_{\mathbf{E}} \lambda||\mathbf{M} \circ \mathbf{E}||_p - \ell(\mathbf{1}_y, \mathbf{J}(\hat{\mathbf{X}})) \tag{3}$$

where $\mathbf{M} \in \{0, 1\}^{\Delta T \times W \times H \times C}$ is the binary spatial mask of the perturbation $\mathbf{E}$, and only the region to be attacked is set to 1. Note that the operator $\circ$ denotes the element-wise multiplication.

3.5. Transfer Attack

There are a lot of works that show the adversarial examples have the attack transferability[21, 19], [33] also show the ability between 3DConv based models and [18] shows they can transfer in CNN+LSTM based models. In this section, we proposed the adaptable method focus on generating the adversarial frames can transfer across videos and models.

Although we can compute different perturbations for different videos to generate a series of specific adversarial frames with Eq. (1), it is also possible to find a *video-agnostic* adversarial perturbation that can apply to any input video for a certain video-classification method **$\mathbf{A^2F}$-AV**.

$$\arg\min_{\mathbf{E}} \lambda||\mathbf{E}||_p - \sum_{n=1}^{N} \alpha_n \ell(\mathbf{1}_{y_n}, \mathbf{J}(\hat{\mathbf{X}}_n; \boldsymbol{\theta})) \tag{4}$$

[1] Of course, we can insert frames at the beginning or in the middle of the video, possibly with different texts (*e.g.*, 'welcome to watch our video' in the beginning), and these options have the same ability of attack. Throughout this paper, we only perform experiments with adversarial frames added in the end of each video.

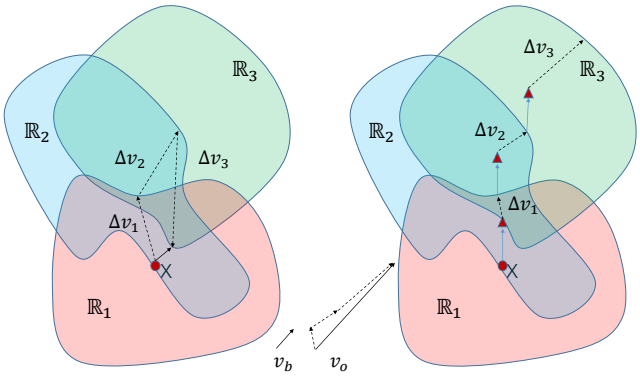

Figure 2: An explanation of the mechanism of existing methods and our attacking method. The left figure shows the schematic of existing methods, while the right figure shows the schema represented by our method, where v_b and v_o denote the universal perturbation generated by the two methods, respectively. X with red dot denotes the batch of video data and the red triangle denotes the video data after appending the adversarial frames. As for the classification region $\mathbb{R}_i$, they are showed in different shape with different colors. The blue arrow line in the classification region denotes the shift that is caused by the adversarial frames appended to the video, the black dotted arrow line represents the minimal perturbation Δv_i at each iteration, and the black arrow line denotes the final universal perturbation we computed.

where N is the total number of videos for finding the universal adversarial perturbation. The parameter α_n is the contribution of the n-th video to generate the adversarial perturbation, and $\hat{\mathbf{X}}_n$ is the n-th adversarial video.

We can also develop a *model-agnostic* attack method to generate a universal adversarial perturbation across models. Specifically, we use an ensemble-based method **A^{2}F-AM** to solve the problem:

$$\arg\min_{\mathbf{E}} \lambda||\mathbf{E}||_p - \sum_{k=1}^{K} \beta_k \ell(\mathbf{1}_{y_k}, \mathbf{J}(\hat{\mathbf{X}}; \boldsymbol{\theta}_k)) \qquad (5)$$

where K is the total number of models, and $\mathbf{J}(\cdot; \boldsymbol{\theta}_k)$ is the k-th model. Similar to α_n, β_k is the weight of model $\mathbf{J}(\cdot; \boldsymbol{\theta}_k)$.

3.6. Why is Appending-Frame Attack Effective?

In order to make the adversarial video can across the classification boundary to a wrong decision area, it is expected that the representation in semantic space is greatly changed while the representation in perception space is kept. Although we empirically find that the operation of appending frames will keep the video in the original semantic space, it provides an easy way to generate adversarial perturbations.

Table 1: Basic accuracy (%) of different video models.

Models	UCF-101	HMDB-51
I3D-ResNet	57.7	97.2
I3D-Inception	94.9	96.8
CNN+LSTM	34.5	92.4
C3D	50.9	99.9
ResNet3D	83.7	91.6
P3D	58.8	95.2

As illustrated in Fig. 2, we show how perturbations are generated gradually for both BA (left figure) and our method (right figure) when the iteration goes on. At each iteration, the minimal perturbation Δv_i is solved to reach the classification boundary. We can see the difference between two methods. BA can not constrain the attack orientation as it chooses uncertain gradient ascent direction. With the stochastic attack orientation, there comes out the possibility of canceling each attack orientation out, which of course makes the perturbations small. As the figure shown, small perturbations likely can not fight against or stay far away decision boundary, and therefore consequently result in a weak universal perturbation. However, with the setting of appending adversarial frames, we can make the attack orientation more uniform and similar, which helps a lot in choosing a more reasonable attack orientation. As the figure shown, our method constrains the attack orientation certainly and reduces the chance of counteracting the attack contribution, therefore our method is more likely to find robust universal perturbations and consequently has strong transfer ability. In what follows, we verify our hypothesis with extensive experiments in Sec. 4.5.

As the assumption we proposed above, the key point of the effectiveness is that we can generate the perturbations with an constrained orientation. The adversarial perturbations can be simply regarded as the sum of Δv_i. While a powerful perturbation is not only comes from the length of Δv_i, but also the angular similarity between Δv_i because higher angular similarity means we can reduce the conflicting between Δv_i. So we use cosine similarity as a measure to calculate the angular similarity trying to find out whether our methods can maximize the similarity for generating more constrained orientation perturbations.

$$\begin{aligned} \mathbf{V_i} &= \sum_{i=1}^{n} \Delta v_i \\ \mathbf{D} &= \sum_{i=1}^{n-1} \frac{|\mathbf{V_i} \cdot \Delta v_{i+1}|}{||\mathbf{V_i}||_2 ||\Delta v_{i+1}||_2} \end{aligned} \qquad (6)$$

$\mathbf{V_i}$ is the perturbations in i-th step in across videos or models settings, $\mathbf{D}$ denotes the total angular distance during the perturbation generating periods.

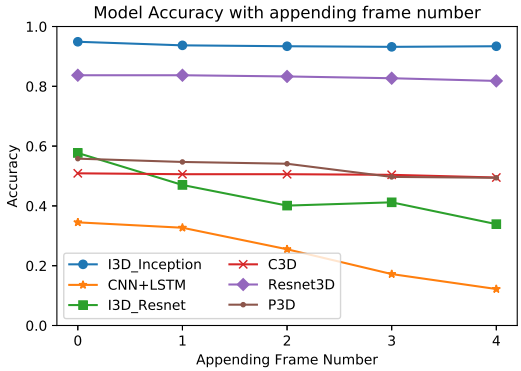

Figure 3: Model accuracy w.r.t the number of appending frames in UCF-101.

4. Experiments

In this section, we evaluate our proposed method comprehensively and compare our method with the state-of-the-art white-box video attack methods.

4.1. Experimental Setting

Datasets. We use two popular benchmark datasets for video classification: UCF-101 [27] and HMDB-51 [16]. UCF-101 is an action recognition dataset collected from YouTube, which contains 13320 realistic action videos within 101 action categories (*e.g.* body-motion, sports, playing instruments, human-interaction). Similarly, HMDB-51 is comprised of 6849 video clips distributed in 51 action categories including facial actions and body movements. Each category contains at least 101 video clips.

Video Recognition Models. We consider up to six state-of-the-art video classification models, namely, I3D-Inception, I3D-ResNet, CNN+LSTM, C3D, ResNet3D and P3D, as our target models to attack. I3D is an inflated 3D convolutional network. The difference between I3D-ResNet and I3D-Inception lies at the basic 2D model inflated to 3D. More specifically, the I3D-Inception model utilizes the UCF-101 dataset to fine-tune the pre-trained model [4], while I3D-ResNet is based on the ResNet101 model trained on ImageNet. For CNN+LSTM, we use ResNet101 pretained on ImageNet as feature extractor, and then train LSTM with features output from ResNet101. P3D and Resnet3D are all implemented officially. It should be noted that we only consider the RGB part for these models. The accuracy of the six models on UCF-101 and HMDB-51 can be found in Tab. 1. The accuracy gap between ours and those reported in [10, 31, 4, 25, 32] is mainly caused by the availability of different input frames at test time.

Attack Settings. We ensure every single video to be attacked is classified successfully and set to have $T = 28$ frames. The number of replaced frames Δ set to 2 and 4 in the basic attack phase (attack a specific video under a specific model) and transferable attack phase, respectively. The basic attack algorithm keeps the same settings with A^{2}F. We select 500 videos from different categories in the test dataset to evaluate the attack performance for the different attack strategies in different attack scenarios. The parameters λ, α and β are tuned in the training phase. All of the experiments are stopped when we find the adversarial perturbations or we reach the number of maximum iterations.

Table 2: Comparison of BA and A^{2}F with different video classification models.

Target Model	Methods	UCF-101		HMDB-51	
		FR (%)	AAP	FR (%)	AAP
I3D-ResNet	BA	**100**	0.22	**100**	0.31
	A^{2}F	**100**	**0.05**	**100**	**0.06**
I3D-Inception	BA	**99.5**	0.20	**100**	0.28
	A^{2}F	**99.5**	**0.08**	**100**	**0.07**
CNN+LSTM	BA	**100**	0.20	**100**	0.28
	A^{2}F	**100**	**0.02**	**100**	**0.02**
C3D	BA	**99.5**	0.24	**100**	0.30
	A^{2}F	97.3	**0.14**	96.8	**0.16**
ResNet3D	BA	**97.8**	0.25	**100**	0.30
	A^{2}F	95.1	**0.09**	**100**	**0.07**
P3D	BA	**100**	0.20	**100**	0.28
	A^{2}F	**100**	**0.02**	**100**	**0.02**

To quantitative evaluate attack models, we use the following performance measures. (i) Fooling Rate (FR): the ratio of the generated adversarial videos that are successfully misclassified. (ii) Average Absolute Perturbation (AAP): $AAP = \frac{1}{N \times S} \sum_{n=1}^{N} \frac{\sum |\mathbf{E}_n|}{\Delta T_n}$, where N is the number of test videos, and S is the spatial size of perturbations ($S = 224 \times 224$). $\mathbf{E}_n$ is the changed magnitude of perturbation for the n-th video, and ΔT_n is the number of adversarial frames for the n-th video. Note that all of the experiments are based on the video classification models classify successfully. (iii) Difference between Intermediate Layer (DIFF): denotes the Euclidean distance between the adversarial frames and the original frames at the l-th intermediate layer.

4.2. Attack Performance

Based on the property described in Sec 3.2, the appending operation will have little or no impact on the representation in the perception space. Furthermore, it also hardly changes the semantics of the original videos. As shown in Fig. 3, there is only a slight drop in performance of recognition models with the number of appending frames from 1 to 4.

While appending frames does not fool deep networks,

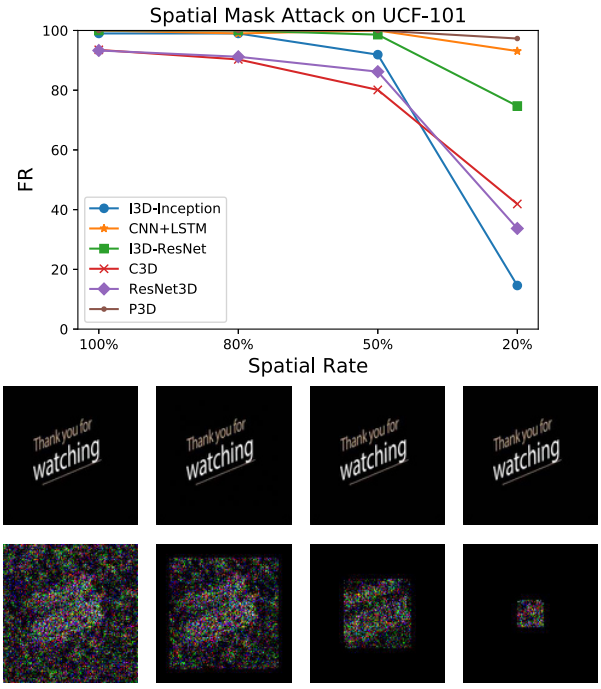

Figure 4: Illustration the (**FR**) values with different spatial rates. We here report the results when spatial rates are 100%, 80%, 50%, 20%, respectively. To be simple, the spatial masks are constrained to be squares. In the bottom of the figure, the first row is example adversarial frames of attacking ResNet3D and the second row is the corresponding perturbations.

the operation, however, makes video attack become easier. Tab. 2 lists the performance of A^{2}F. on UCF-101 and HMDB-51. It indicates that A^{2}F outperforms BA by a large gap on AAP, and meanwhile remains almost the same high FR as BA.

For example, with recognition model P3D, both A^{2}F and BA achieve perfect FR on UCF-101. However, the AAP of A^{2}F is just 0.02, while the AAP of perturbation generated by BA [33] is up to 0.2, which is 10 times compared with A^{2}F. This means the proposed A^{2}F generates high quality adversarial videos with more imperceptible perturbations.

4.3. Attack with Spatial Mask

With aforementioned methods, each pixel of the adversarial frames is perturbed as the size of the perturbations is the same as the adversarial frames. This may increase the risk to perceive adversarial frames. To increase the concealing chance and reduce the distance between the projection original data in perception space, one can restrict the perturbation in a much smaller size. It is quite easy for us to construct arbitrary shape perturbations by changing shapes of spatial masks. However, to be simple, we only show the fooling rates for square perturbations. As the Fig. 4 shown, the FR values decrease slowly when the spatial rate drops from 100% to 50%. This means spatial mask attack is indeed effective for generating more imperceptible adversarial frames while slightly or not hurting FR with a suitable spatial rate. Furthermore, it is easy to understand that when the spatial rate is too small, the generated perturbations can not have enough ability to attack classification models. For instance, as shown, the FR drops a lot when the spatial rate equals to 20%.

Table 3: Comparison of BA-AV and A^{2}F-AV in transferability across different videos.

Target Model	Methods	UCF-101		HMDB-51	
		FR (%)	AAP	FR (%)	AAP
I3D-ResNet	BA-AV	95.4	0.62	93.0	0.70
	A^{2}F-AV	**98.1**	**0.52**	**97.8**	**0.60**
I3D-Inception	BA-AV	2.6	**0.34**	2.0	**0.25**
	A^{2}F-AV	**69.3**	1.25	**2.3**	0.84
CNN+LSTM	BA-AV	18.1	**0.09**	**69.6**	**0.13**
	A^{2}F-AV	**47.1**	0.16	45.7	0.21
C3D	BA-AV	97.9	**0.75**	**98.0**	**0.68**
	A^{2}F-AV	**98.1**	1.21	96.9	1.75
ResNet3D	BA-AV	45.2	**0.65**	58.6	**0.49**
	A^{2}F-AV	**96.6**	1.21	**94.1**	0.79
P3D	BA-AV	20.7	**0.11**	46.9	0.16
	A^{2}F-AV	**98.4**	0.25	**97.4**	**0.15**

4.4. Attack Transferability

Cross-Video Transferability. The experimental results in Tab. 2 show that the A^{2}F can increase the effectiveness of adversarial perturbations for specific videos. However, we argue that a successful adversarial perturbation should not only perform well for a specific video, but also should hold the capability in transferring across different videos. We evaluate the transferability of adversarial perturbations across videos on UCF-101 and HMDB-51 by A^{2}F-AV (Eq. (4)). The performance of universal adversarial perturbations across videos is listed in Tab. 3. One can clearly see that, compared with the baseline BA, A^{2}F-AV has much superior performance in FR. For example, with P3D as the treat model, the FRs of BA and A^{2}F-AV on UCF-101 are 20.7% and 98.4%, respectively. The result suggests that our method can significantly enhance the attack ability in transferring across different videos. Our method has larger AAP than BA in this scenario. This is mainly because our attack orientation is more uniform and less random, which helps the perturbation accumulate gradually in the semantic space without breaking the decision boundary of threat models.

Cross-Model Transferability. We also evaluate the transferability of perturbations across models with A^{2}F-AM (Eq. (5)). To the end, we use the evaluated six models to

Table 4: Comparison of BA-AM and A^2F-AM in transferability across models on UCF-101 dataset. The first column indicates we use the Leave-One-Out ensemble method that excludes one model to produce perturbations. For instance,'−I3D-ResNet' means the corresponding ensemble model excludes I3D-ResNet. The numbers in the 3-8 columns are the fooling rates (%) for each attacked model.

Models	Method	I3D-ResNet	ResNet3D	P3D	I3D-Inception	C3D	CNN+LSTM
−I3D-ResNet	BA-AM	0.0	78.7	84.6	87.8	70.8	56.2
	A^2F-AM	**39.5**	68.1	97.4	42.9	85.4	81.6
−ResNet3D	BA-AM	100	0.0	84.6	87.8	70.8	38.9
	A^2F-AM	89.5	**6.4**	97.4	52.2	85.4	71.4
−P3D	BA-AM	100	80.9	15.4	87.8	72.9	58.8
	A^2F-AM	86.8	74.5	**59.0**	50.0	85.4	83.7
−I3D-Inception	BA-AM	100	83.0	97.4	0.0	73.0	61.1
	A^2F-AM	86.8	78.7	100	**2.0**	85.4	50.0
−C3D	BA-AM	100	83.0	100	90.0	0.0	64.7
	A^2F-AM	92.1	80.9	100	60.0	**20.8**	79.6
−CNN+LSTM	BA-AM	100	80.9	97.4	97.8	72.9	35.7
	A^2F-AM	89.5	74.5	100	55.6	85.4	**77.6**

explore the across models perturbations. Specifically, we use the Leave-One-Out ensemble method that excludes one model to produce perturbations, and then attack each model with the generated perturbations. The corresponding results are shown in Tab. 4. The result shows that our method have a better performance in fooling those models. For example, with perturbations generated by ensemble models without the P3D model, the fooling rates of attacking P3D for BA and A^2F-AM are 15.4% and 59.0%, respectively. This means the gain of our method over the baseline is over 40% for the unseen threat model. We think the improvement may be caused by the constrained and unified attacking orientation of our method, which makes the perturbations stay far away from the classification boundary.

4.5. Angular Similarity

According to the equation in Eq. (6), we calculate the angular similarity under across videos and models scene. The total attacked video number of across videos settings is 1000 and the ensemble strategy of across models is still Leave-One-Out. The experimental results are shown in Tab. 5 that the angular similarity of A^2F is higher than BA in all models under all situations. The higher similarity represents a smaller angular between each vector in the feature subspace which can alleviate the gradient conflict and help them towards the same direction. Refer to the results in Tab. 3 and Tab. 4, it seems to have a positive correlation between FR and angular similarity.

5. Conclusions

In this paper, we present an interesting idea of attacking videos. We take advantage of the temporal property of videos, *i.e.*, changing a few ending frames of it, though introducing a large perturbation in terms of the pixel-wise metrics, can still be easily concealed, *i.e.*, most people would not notice that the video has been attacked. On the other hand, by adding adversarial perturbations *only* on these new frames, the perceptibility of the added noise becomes smaller yet the attack is verified easier to transfer across videos and even different networks. In other words, our method, though simple, provides an effective pipeline of universal video attack.

Table 5: Comparison of BA, and A^2F with the Angular similarity between perturbations.

Target Model	Methods	Across Videos	Across Models
I3D-ResNet	BA	0.2968	0.0319
	A^2F	**2.1971**	**0.0997**
I3D-Inception	BA	0.0015	0.0269
	A^2F	**0.8986**	**0.0620**
CNN+LSTM	BA	0.0234	0.0184
	A^2F	**1.3571**	**0.0808**
C3D	BA	0.0982	0.0255
	A^2F	**0.4389**	**0.0838**
ResNet3D	BA	0.0325	0.0088
	A^2F	**0.6019**	**0.0406**
P3D	BA	0.0018	0.0182
	A^2F	**0.9837**	**0.0648**

Acknowledgement

This work was supported in part by NSFC under Grant 61972312, in part by the Key Research and Development Program of Shaanxi under Grant 2020GY-002, and in part by the China Postdoctoral Science Foundation under Grant 2019M653335.

References

[1] Tom B. Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. *CoRR*, abs/1712.09665, 2017. 4

[2] Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, and Alexey Kurakin. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*, 2019. 3

[3] Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *ACM Workshop on Artificial Intelligence and Security (AISec)*, 2017. 3

[4] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, pages 6299–6308, 2017. 2, 6

[5] Jianbo Chen, Michael I. Jordan, and Martin J. Wainwright. Hopskipjumpattack: A query-efficient decision-based attack. *arXiv: 1904.02144*, 2019. 3

[6] Pin-Yu Chen, Yash Sharma, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. EAD: elastic-net attacks to deep neural networks via adversarial examples. *arXiv:1709.04114*, 2017. 3

[7] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. *arXiv:1708.03999*, 2017. 3

[8] Jun-Ho Choi, Huan Zhang, Jun-Hyuk Kim, Cho-Jui Hsieh, and Jong-Seok Lee. Evaluating robustness of deep image super-resolution against adversarial attacks. In *ICCV*, 2019. 1

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 2

[10] Jeff Donahue, Lisa Anne Hendricks, Marcus Rohrbach, Subhashini Venugopalan, Sergio Guadarrama, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. *arXiv:1411.4389*, 2014. 2, 6

[11] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv:1412.6572*, 2014. 1, 2

[12] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and ImageNet? In *CVPR*, 2018. 2

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2

[14] Cormac Herley and Paul C Van Oorschot. Sok: Science, security and the elusive goal of security as a scientific pursuit. In *2017 IEEE Symposium on Security and Privacy (S&P)*, pages 99–120. IEEE, 2017. 3

[15] Linxi Jiang, Xingjun Ma, Shaoxiang Chen, James Bailey, and Yu-Gang Jiang. Black-box adversarial attacks on video recognition models. *arXiv:1904.05181*, 2019. 1, 3

[16] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. HMDB: a large video database for human motion recognition. In *ICCV*, pages 2556–2563, 2011. 2, 6

[17] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv:1607.02533*, 2016. 1, 2

[18] Shasha Li, Ajaya Neupane, Sujoy Paul, Chengyu Song, Srikanth V. Krishnamurthy, Amit K. Roy-Chowdhury, and Ananthram Swami. Stealthy adversarial perturbations against real-time video classification systems. *arXiv:1807.00458*, 2018. 1, 3, 4

[19] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. In *Proceedings of 5th International Conference on Learning Representations*, 2017. 3, 4

[20] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv:1706.06083*, 2017. 1, 3

[21] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *CVPR*, pages 1765–1773, 2017. 3, 4

[22] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In *CVPR*, 2016. 1, 3

[23] Muzammal Naseer, Salman H. Khan, Shafin Rahman, and Fatih Porikli. Task-generalizable adversarial attack based on perceptual metric, 2018. 3

[24] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *EuroS&P*, 2016. 3

[25] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *ICCV*, pages 5533–5541, 2017. 2, 6

[26] Sara Sabour, Yanshuai Cao, Fartash Faghri, and David J Fleet. Adversarial manipulation of deep representations. *arXiv:1511.05122*, 2015. 3

[27] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv:1212.0402*, 2012. 2, 6

[28] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *CoRR*, abs/1710.08864, 2017. 4

[29] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016. 2

[30] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv:1312.6199*, 2013. 1, 2

[31] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatio-temporal features with 3D convolutional networks. In *ICCV*, 2015. 2, 6

[32] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, pages 6450–6459, 2018. 6

[33] Xingxing Wei, Jun Zhu, and Hang Su. Sparse adversarial perturbations for videos. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33, 03 2018. 1, 3, 4, 7

[34] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan Yuille. Adversarial examples for semantic segmentation and object detection. In *ICCV*, 2017. 1

[35] Xiaojun Xu, Xinyun Chen, Chang Liu, Anna Rohrbach, Trevor Darrell, and Dawn Song. Fooling vision and language models despite localization and attention mechanism. *arXiv:1709.08693*, 2017. 1