Black-box Adversarial Attacks on Video Recognition Models

Linxi Jiang*1, Xingjun Ma*2, Shaoxiang Chen1, James Bailey2, Yu-Gang Jiang†1

Abstract—Deep neural networks (DNNs) are known for their vulnerability to adversarial examples. These are examples that have undergone a small, carefully crafted perturbation, and which can easily fool a DNN into making misclassifications at test time. Thus far, the field of adversarial research has mainly focused on image models, under either a white-box setting, where an adversary has full access to model parameters, or a black-box setting where an adversary can only query the target model for probabilities or labels. Whilst several white-box attacks have been proposed for video models, black-box video attacks are still unexplored. To close this gap, we propose the first black-box video attack framework, called V-BAD. V-BAD is a general framework for adversarial gradient estimation and rectification, based on Natural Evolution Strategies (NES). In particular, V-BAD utilizes *tentative perturbations* transferred from image models, and *partition-based rectifications* found by the NES on partitions (patches) of tentative perturbations, to obtain good adversarial gradient estimates with fewer queries to the target model. V-BAD is equivalent to estimating the projection of an adversarial gradient on a selected subspace. Using three benchmark video datasets, we demonstrate that V-BAD can craft both untargeted and targeted attacks to fool two state-of-the-art deep video recognition models. For the targeted attack, it achieves >93% success rate using only an average of $3.4 \sim 8.4 \times 10^4$ queries, a similar number of queries to state-of-the-art black-box image attacks. This is despite the fact that videos often have two orders of magnitude higher dimensionality than static images. We believe that V-BAD is a promising new tool to evaluate and improve the robustness of video recognition models to black-box adversarial attacks.

Index Terms—Adversarial examples, video recognition, black-box attack, model security.

1 Introduction

Deep Neural Networks (DNNs) are a family of powerful models that have demonstrated superior performance in a wide range of visual understanding tasks that has been extensively studied in both the multimedia and computer vision communities such as video recognition[1], [2], [3], [4], image classification[5], [6] and video captioning[7], [8]. Despite their current success, DNNs have been found to be extremely vulnerable to adversarial examples (or attacks) [9], [10]. For classification DNNs, adversarial examples can be easily generated by applying adversarial perturbations to clean (normal) samples, that maximize the classification error [10], [11], [12]. For images, the perturbations are often small and visually imperceptible to human observers, but they can fool DNNs into making misclassifications with high confidence. The vulnerability of DNNs to adversarial examples has raised serious security concerns for their deployment in security-critical applications, such as face recognition [13] and self-driving cars[14]. Hence, the study of adversarial examples for DNNs has become a crucial task for secure deep learning.

Adversarial examples can be generated by an attack method (also called an adversary) following either a white-box setting (white-box attacks) or a black-box setting (black-box attacks). In the white-box setting, an adversary has full access to the target model (the model to attack), including model training settings and parameters. In the black-box setting, an adversary only has partial information about the target model, such as the labels or probabilities output by the model. White-box methods generate an adversarial example by applying one step or multiple steps of perturbations on a clean test sample, following the direction of the adversarial gradient [10], [12]. The adversarial gradient is the gradient of an adversarial loss, which is typically defined to maximize (rather than minimize) classification error. However, in the black-box setting, adversarial gradients are not accessible to an adversary. In this case, the adversary can first attack a local surrogate model and then transfer these attacks to the target model [15], [16], [17]. Alternatively they may use a black-box optimization method such as Finite Differences (FD) or Natural Evolution Strategies (NES), to estimate the gradient [18], [19], [20].

A number of attack methods have been proposed [11], [12], [21], however, most of them focus on either image models, or video models but in a white-box setting [22], [23]. Different from these works, in this paper, we propose a framework for the generation of adversarial attacks against video recognition models, specifically in a black-box setting. Significant progress has been achieved for black-box image attacks, but not for black-box video attacks. A key reason is that videos typically have much higher dimensionality (often two magnitudes higher) than static images. On static images, for most attacks to succeed, existing black-box methods must use $\sim 10^4$ queries [18], [20] on CIFAR-10 [24] images, and $\sim 10^5$ queries [19] on ImageNet [25] images. Due to their massive input dimensions, black-box attacks on videos generally require two orders of magnitude more queries for gradient estimation than are needed for images.

* *indicates equal contributions.*
† *indicates corresponding author.*

- *Linxi Jiang, Shaoxiang Chen, and Yu-Gang Jiang are with the School of Computer Science, Fudan University, Shanghai, China. E-mail: {lxjiang18, sxchen13, ygj}@fudan.edu.cn.*
- *Xingjun Ma, James Bailey are with The University of Melbourne. Email: {xingjun.ma, baileyj}@unimelb.edu.au.*

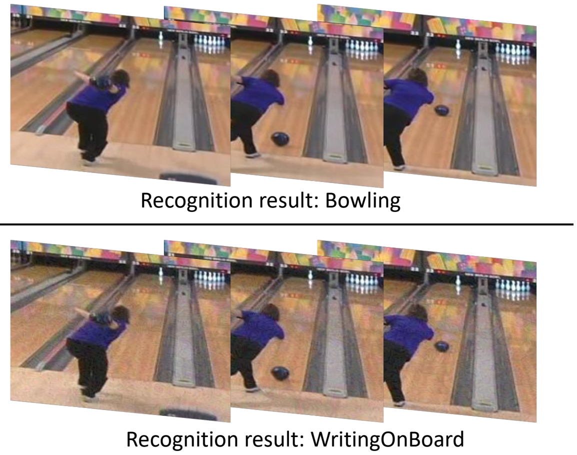


Fig. 1: An example of black-box video adversarial attacks (targeted). The original video (top) can be correctly recognized while the adversarial one (bottom) generated by our proposed method is misclassified by the same video model.

This makes black-box video attacks impractical, taking into account time and budget constraints. To better evaluate the robustness of video models, it is therefore important to explore efficient black-box methods that can generate attacks using fewer queries.

In this paper, we propose a simple and efficient framework for the generation of black-box adversarial attacks on video recognition models. Intuitively, we exploit the advantages of transferability of adversarial gradient and black-box gradient estimation methods. In particular, we first generate tentative perturbations as a rough estimate of the true adversarial gradient using ImageNet-pretrained DNNs. We then rectify these tentative perturbations in partitions using NES, by querying the target model. Our proposed framework only needs to estimate a small number of directional derivatives (of perturbation partitions), rather than estimating pixel-wise derivatives, making it an efficient framework for black-box video attacks. Figure 1 shows an example of video adversarial attacks generated by our proposed method. In summary, our main contributions are:

- We study the problem of black-box attacks on video recognition models and propose a general framework called V-BAD, to generate black-box video adversarial examples. To the best of our knowledge, our proposed framework V-BAD is the first black-box adversarial attack framework for videos.
- Our proposed framework V-BAD exploits both the transferability of adversarial gradients via the use of tentative perturbations, and the advantages of gradient estimation via NES on partitions (patches) of tentative perturbations. We show that V-BAD is equivalent to estimating the projection of the adversarial gradient on a selected subspace.
- We conduct an empirical evaluation using three benchmark video datasets and two state-of-the-art video recognition models. We show that V-BAD can achieve high attack success rates with few queries to target models, making it a useful tool for the robustness evaluation of video models.

2 Related Work

White-box Image Attack. The fast gradient sign method(FGSM) crafts an adversarial example by perturbing a normal sample along the gradient direction towards maximizing the classification error [10]. FGSM is a fast one-step attack, and can be applied iteratively to improve adversarial strength [13]. Projected Gradient Descent (PGD) [21] is another iterative method that is regarded as the strongest first-order attack. PGD projects the perturbation back onto the ϵ-ball of a sample $\mathbf{x}$ when perturbation goes beyond the ϵ-ball. The C&W attack solves the attack problem via an optimization framework [12], and is arguably the state-of-the-art white-box attack. There also exists other types of white-box methods, *e.g.*, Jacobian-based Saliency Map Attack (JSMA) [26], DeepFool [27] and elastic-net attack (EAD) [28].

Black-box Image Attack. In the black-box setting, an adversarial gradient is not directly accessible by an adversary. As such, black-box image attacks either exploit the transferability of adversarial examples or make use of gradient estimation techniques. It was first observed in [9] that adversarial examples are transferable across models, even if they have different architectures or were trained separately. [17] trains a surrogate model locally on synthesized data, with labels obtained by querying the target model. It then generates adversarial examples from the surrogate model using white-box methods to attack the target model. However, training a surrogate model on synthesized data often incurs a huge number of queries, and the transferability of generated adversarial examples is often limited. [20] proposes using Finite Differences (FD), a black-box gradient estimation method, to estimate the adversarial gradient. [18] accelerates FD-based gradient estimation with dimensionality reduction techniques such as PCA. Compared to FD, [19] demonstrates improved performance with fewer queries by the use of Natural Evolutionary Strategies (NES).

White-box Video Attack. In contrast to image adversarial examples, much less work has been done for video adversarial examples. White-box video attacks were first investigated in [29], which discussed the sparsity and propagation of adversarial perturbations across video frames. [22] leverages Generative Adversarial Networks (GANs) to perturb each frame in real-time video classification. In this paper, we explore black-box attacking methods against state-of-the-art video recognition models, which to the best of our knowledge, is the first work on black-box video attacks.

Video Recognition Models. Encouraged by the great success of Convolutional Neural Networks (CNNs) on image recognition tasks, many works propose to adapt image (2D) CNNs to video recognition. [1] explores various approaches for extending 2D CNNs to video recognition based on features extracted from individual frames. CNN+LSTM based models [30], [31] exploit the temporal information contained in successive frames, with recurrent layers capturing long term dependencies on top of CNNs. C3D[2], [32], [33] extends the 2D spatio-only filters in traditional CNNs to 3D spatio-temporal filters for videos, and learn hierarchical spatio-temporal representations directly from videos. However, C3D models often have a huge number

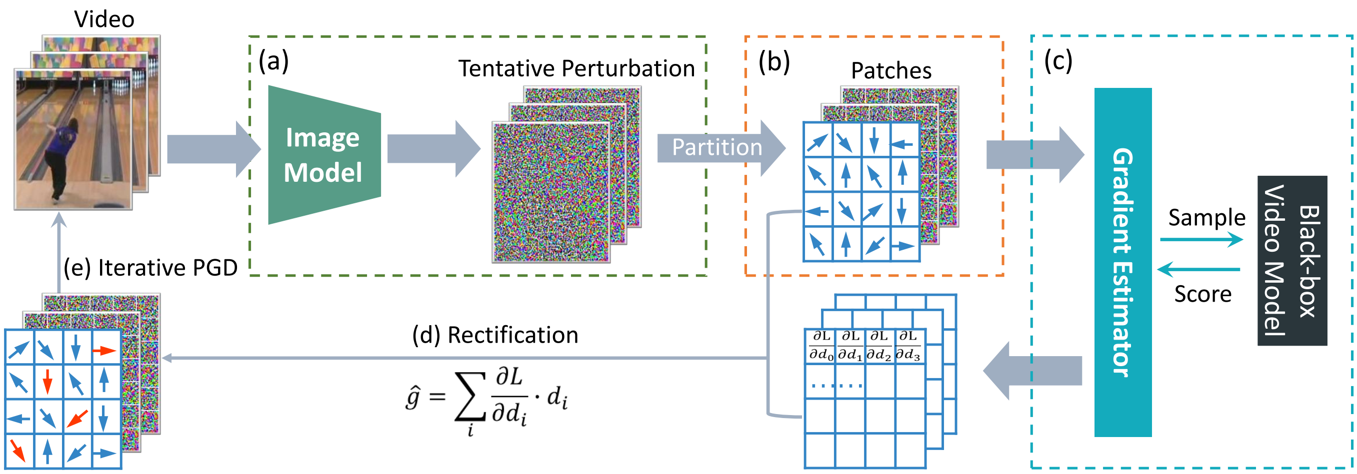


Fig. 2: Overview of the proposed V-BAD framework for black-box video attacks.

of parameters, which makes training difficult. To address this, [3] propose the Inflated 3D ConvNet(I3D) with Inflated 2D filters and pooling kernels of traditional 2D CNNs. In this paper, we use two representative state-of-the-art video recognition models, CNN+LSTM and I3D, as our target models to attack.

3 Proposed Framework V-BAD

3.1 Preliminaries

We denote a video sample by $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^{T \times H \times W \times C}$ with T, H, W, C denoting the number of frames, frame height, frame width, and the number of channels respectively, and its associated true class by $y \in \mathcal{Y} = \{1, \cdots, K\}$. Video recognition is to learn a classification DNN $f(\mathbf{x};\theta) : \mathcal{X} \to \mathcal{Y}$ by minimizing the classification loss $\ell(f(\mathbf{x};\theta), y)$, and θ denotes the parameters of the network. When the context is clear, we abbreviate $f(\mathbf{x};\theta)$ as $f(\mathbf{x})$, and $\ell(f(\mathbf{x};\theta), y)$ as $\ell(\mathbf{x}, y)$. The goal of adversarial attack is to find an adversarial example $\mathbf{x}_{adv}$ that maximizes the classification error, while remaining in the ϵ-ball centered at $\mathbf{x}$ ($\|\mathbf{x}_{adv} - \mathbf{x}\|_p \leq \epsilon$):

$$\arg\max_{\mathbf{x}_{adv}} \ell(\mathbf{x}_{adv}, y) \quad s.t. \quad \|\mathbf{x}_{adv} - \mathbf{x}\|_p \leq \epsilon, \tag{1}$$

where $\|\cdot\|_p$ is the L^p-norm. The maximum perturbation magnitude ϵ is often bounded so as to be small and imperceptible to human observers. Here, we only focus on the L^∞-norm, that is, $\|\mathbf{x}_{adv} - \mathbf{x}\|_\infty \leq \epsilon$, but our framework also applies to other norms. We also denote the adversarial loss by ℓ_{adv}, the ground truth (or white-box) adversarial gradient by $g = \nabla_{\mathbf{x}} \ell_{adv}(\mathbf{x}, y)$.

Threat Model. Our threat model follows the query-limited black-box setting as follows. The adversary takes the video classifier f as a black-box and only has access to its output of the top 1 score. More specifically, during the attack process, given an arbitrary clean sample $\mathbf{x}$, the adversary can query the target model f to obtain the top 1 label $\hat{y}$ and its probability $P(\hat{y}|\mathbf{x})$. The adversarial are asked to generate attacks within Q queries. We consider both untargeted and targeted attacks. For an untargeted attack, the goal is to generate an adversarial example $\mathbf{x}_{adv}$ such that $f(\mathbf{x}_{adv}) \neq y$, while for a targeted attack with target class y_{adv}, the goal is to find an adversarial example $\mathbf{x}_{adv}$ such that $f(\mathbf{x}_{adv}) = y_{adv}$.

3.2 Framework Overview

The structure of the proposed framework, namely V-BAD, for black-box video attacks is illustrated in Figure 2. Following steps (a)-(e) highlighted in the figure, V-BAD perturbs an input video iteratively, using PGD as follows: a) It first passes video frames into a public image model (ImageNet pretrained), to obtain pixel-wise *tentative perturbations*; b) It then applies a partition operation to split up tentative perturbations into patches; c) A black-box gradient estimator estimate the best amount of rectification required to make tentative patches as close as possible to the true adversarial gradient; d) Patch-based Rectification is performed on the tentative perturbation patches, to obtain a rectified perturbation (or gradient estimate); e) A one step PGD update on the input video is performed, according to the rectified perturbation. This process iterates until an attack succeeds, *i.e.*, an untargeted/targeted adversarial example has been found, or the query limit is reached. Steps a) - d) are for gradient estimation, while step e) is for perturbing the input video. We begin by a brief introduction of step e) (*e.g.* PGD), similar to[19]. We then describe in detail the proposed gradient estimation framework of steps a) - d).

Untargeted V-BAD Attack. For an untargeted attack, we perform PGD iteratively, substituting the adversarial gradient with the estimated gradient:

$$\mathbf{x}_{adv}^{(i)} = \Pi_{[\mathbf{x}^{(0)}-\epsilon, \mathbf{x}^{(0)}+\epsilon]}(\mathbf{x}_{adv}^{(i-1)} - \alpha \cdot \hat{g}_{i-1})) \tag{2}$$

where Π is a projection operation [21], $\mathbf{x}^{(0)}$ is the original video sample, α is the step size, ϵ is the perturbation bound, and $\hat{g}_{i-1}$ is the gradient estimate with respect to $\mathbf{x}_{adv}^{(i-1)}$.

Targeted V-BAD Attack. For a targeted attack, we need to ensure the target class is in the top-1 classes, because the score of the target class is required for gradient estimation. Thus, instead of the original video sample $\mathbf{x}^{(0)}$, we begin with a sample from the target class, then gradually (step by step) move it into the ϵ-ball of $\mathbf{x}^{(0)}$ while maintaining the

Algorithm 1 Targeted V-BAD attack

Input: Top-1 probability $P(y|\mathbf{x})$ with respect to classifier f, target class y and video $\mathbf{x}$
Output: Adversarial video $\mathbf{x}_{adv}$ with $||\mathbf{x}_{adv} - \mathbf{x}||_\infty \leq \epsilon$
Parameters: Perturbation bound ϵ_{adv}, starting perturbation ϵ_0, epsilon decay δ_ϵ, PGD step size α

$\epsilon \leftarrow \epsilon_0$
$\mathbf{x}_{adv} \leftarrow$ video of target class y_{adv}
while $\epsilon > \epsilon_{adv}$ **do**
  $h \leftarrow \textsc{TentDirectGen}(\mathbf{x}_{adv})$
  $d \leftarrow \textsc{Partition}(h)$
  $\hat{g} \leftarrow \textsc{Recttify}(P(y_{adv}|\mathbf{x}_{adv}), d)$
  $\hat{\epsilon} \leftarrow \epsilon - \delta_\epsilon$
  $\hat{\mathbf{x}}_{adv} \leftarrow \textsc{Clip}(\mathbf{x}_{adv} - \alpha \cdot \hat{g}, \mathbf{x} - \hat{\epsilon}, \mathbf{x} + \hat{\epsilon})$
  if $y_{adv} = \textsc{Top-1}(P(\cdot|\hat{\mathbf{x}}_{adv}))$ **then**
    $\mathbf{x}_{adv} \leftarrow \hat{\mathbf{x}}_{adv}$
    $\epsilon \leftarrow \hat{\epsilon}$
  else
    $\hat{\mathbf{x}}_{adv} \leftarrow \textsc{Clip}(\mathbf{x}_{adv} - \alpha \cdot \hat{g}, \mathbf{x} - \epsilon, \mathbf{x} + \epsilon)$
    if $y_{adv} = \textsc{Top-1}(P(\cdot|\hat{\mathbf{x}}_{adv}))$ **then**
      $\mathbf{x}_{adv} \leftarrow \hat{\mathbf{x}}_{adv}$
    end if
  end if
end while
return $\mathbf{x}_{adv}$

targeted class the top-1 class. The targeted V-BAD attack is described in Algorithm 1.

3.3 Tentative Perturbation

Tentative perturbation refers to initialized pixel-wise adversarial perturbation, generated each time before rectification (see Section 3.4) from image models. As PGD only needs the sign of adversarial gradient to perturb a sample, we define tentative perturbation as the sign values (1, 0 or -1) of the perturbation, rather than its raw values. In each perturbation step t, tentative perturbation is computed for the perturbed sample at the previous step (*e.g.* $\mathbf{x}_{adv}^{(t-1)}$) or the original sample for the first step (*e.g.* $\mathbf{x}^0$).

Here, we discuss three types of tentative perturbation. 1) Random: the perturbation for each input dimension is generated randomly with 50% probability of being either 1 or -1. Random perturbation only provides a stochastic exploration of the adversarial space which can be extremely inefficient, given the massive input dimensions of videos. 2) Static: the perturbation for each input dimension is fixed to 1. Fixed perturbation imposes a strong constraint on the exploration space, with the same tentative perturbation used for each gradient estimation. 3) Transferred: tentative perturbation can alternatively be transferred from pre-trained image models that are often obtainable off-the-shelf. Since natural images share certain similar patterns, image adversarial examples often transfer across models or domains, though the transferability can be limited and will vary depending on the content of the images. To better exploit such transferability for videos, we propose to white-box attack a pre-trained image model such as an ImageNet pre-trained DNN, in order to extract the tentative perturbation for each frame. The resulting transferred perturbation can provide useful guidance for the exploration space and help reduce the number of queries.

The transferred tentative perturbation is defined as:

$$h(\mathbf{x}) = \text{sign}\Big(\nabla_{\mathbf{x}} \frac{1}{T} \sum_{t=1}^{T} \|M_t \circ f_I^l(\mathbf{x}_t) - M_t \circ R_t\|_2^2\Big), \quad (3)$$

where sign$(\cdot)$ is the sign function, $\circ$ is the element-wise product, $\|\cdot\|_2^2$ is the squared L^2-norm, T is the total number of frames, f_I^l is the l-th layer (*e.g.* feature layer) of an image model f_I, M_t is a random mask on the t-th frame, R_t is a target feature map used to form a feature-based "adversarial loss", and has the same dimension as $f_I^l(\mathbf{x}_t)$. For untargeted attack, R_t is a feature map of Gaussian random noise, while for targeted attack, R_t is the feature map $f_I^l(\mathbf{x}')$ of the t-th frame of a video $\mathbf{x}'$ from the target class y_{adv}. The random mask M_t is a pixel-wise mask with each element is randomly set to either 1 or 0. The use of random mask can help avoid getting stuck in situations where the frame-wise tentative perturbation is extremely inaccurate (but partially useful) to allow any effective rectification.

3.4 Partition-based Rectification

As tentative perturbation is just a transferred, rough estimate of the true adversarial gradient, it should be further rectified to better approximate the true adversarial gradient. For this, we propose gradient estimation methods to estimate the amount of rectification required to improve the tentative perturbation *in partitions*. That is, we first partition a tentative perturbation into separate parts, which we call perturbation patches, and then estimate the patch-level rectifications for these partition patches. Rectifying perturbation patches rather than individual pixels, allows us to significantly reduce the number of queries required for each estimation, compared to existing pixel-wise gradient estimation methods that have been proposed for black-box image attacks. For rectification, each perturbation patch is adjusted by multiplying all its dimensions by a rectification factor found by a gradient estimator such as FD or NES. Let h denote the tentative perturbation, let $\hat{g}$ be the rectified perturbation, $G^{(i)}$ the set of dimensions in the i-th patch, and v_i the rectification factor for patch i. The rectification applied to each dimension in patch i is then:

$$\hat{g}_j = v_i h_j, \ \forall j \in G^{(i)}. \quad (4)$$

Note that we omit a normalization term in the above equation, which does not affect the effectiveness of the rectified perturbation and will be discussed in detail in Section3.5.

Partitioning Methods. We consider three types of partitioning strategies. 1) Random: dividing input dimensions randomly into a certain number of partitions. Note that the input dimensions of one random partition patch can be non-adjacent, although we refer to it as a patch. Random does not consider the local correlations between input dimensions but can be used as a baseline to assess the effectiveness of carefully designed partitioning methods. 2) Uniform: splitting a frame uniformly into some patches. This will produce frame patches that preserve local dimensional correlations. 3) Semantic: partitioning the video input according to its semantic content. It is promising to explore the correlation

Algorithm 2 NES Estimation of Partition Rectification (Targeted)

Input: Top-1 probability $P(y_{adv}|\mathbf{x})$ with respect to classifier f, target class y_{adv}, video $\mathbf{x}$, and tentative perturbation partition d
Output: Estimate of $\nabla_v P(y_{adv}|\mathbf{x} + \sum_j d^{(j)} \cdot v_j)$
Parameters: Search variance σ, number of samples n, number of perturbation units N, rectification vector v
$\eta \leftarrow \mathbf{0}_n$
for $i = 1$ **to** n **do**
  $\delta^{(i)} \leftarrow \mathcal{N}(\mathbf{0}_N, \boldsymbol{I}_{N \cdot N})$
  $z_{2i-1} \leftarrow P(y_{adv}|\mathbf{x} + \sum_j d^{(j)} \cdot (v_j + \sigma\delta_j^{(i)}))$
  $\eta \leftarrow \eta + \delta^{(i)} \cdot \text{FITNESS}(z_{2i-1})$
  $z_{2i} \leftarrow P(y_{adv}|\mathbf{x} - \sum_j d^{(j)} \cdot (v_j + \sigma\delta_j^{(i)}))$
  $\eta \leftarrow \eta - \delta^{(i)} \cdot \text{FITNESS}(z_{2i})$
end for
return $\frac{1}{2n\sigma}\eta$

between the semantic content of the current input and its adversarial gradient. In this paper, we empirically investigate the first two partitioning strategies, *i.e.*, Random and Uniform, and leave semantic partitioning for future work.

Partitioning and Rectification. The rectification factors v are directional derivatives specified to each patch that can be estimated by any derivative-free optimization methods. Each patch can constitute a direction vector with the same size as the input sample by zero padding: assigning zero values to dimensions that do not belong to the patch, while keeping its values at dimensions that belong to the patch. That is, for each tentative patch $G^{(i)}$, we have a direction vector $d^{(i)}$:

$$d_j^{(i)} = \begin{cases} h_j & j \in G^{(i)} \\ 0 & j \notin G^{(i)} \end{cases} \tag{5}$$

For the estimation of directional derivatives, we propose to use the NES estimator which has been shown to be more efficient than FD for black-box image attacks [19]. We will discuss the efficiency of FD compared to NES in Section 4. Instead of maximizing the adversarial objective directly, NES maximizes the expected value of the objective under a search distribution. Different from [19] where the gradient of adversarial loss with respect to the input is estimated, we estimate the directional derivative of the adversarial loss with respect to direction vectors d for tentative patches. For a loss function $\ell(\cdot)$, current parameters v, $\mathbf{x}$ and a search distribution $\pi(\hat{v}|v)$, we have:

$$\nabla_v \mathbb{E}_{\pi(\hat{v}|v)} \ell(\hat{\mathbf{x}}) = \mathbb{E}_{\pi(\hat{v}|v)} \left[\ell(\hat{\mathbf{x}}) \nabla_v \log\left(\pi(\hat{v}|v)\right) \right], \tag{6}$$

where $\hat{\mathbf{x}} = \mathbf{x} + \sum d^{(i)} \hat{v}_i$. Following [19], [34], [35], we use the normal distribution as the search distribution, that is, $\hat{v} = v + \sigma\delta$ where σ is the search variance and $\delta \sim \mathcal{N}(0, I)$. We use antithetic sampling to generate a population of n values δ_i: sample Gaussian noise for $i \in \{1, \cdots, \frac{n}{2}\}$ and set $\delta^{(j)} = -\delta^{(n-j+1)}$ for $j \in \{(\frac{n}{2}+1), \cdots, n\}$. Evaluating the gradient with a population of n points sampled under this scheme yields the following gradient estimate:

$$\nabla_v \mathbb{E}[\ell(\hat{\mathbf{x}})] \approx \frac{1}{\sigma n} \sum_{i=1}^{n} \delta_i \ell(\mathbf{x} + \sum d^{(j)} \cdot (v_j + \sigma\delta_j^{(i)})). \tag{7}$$

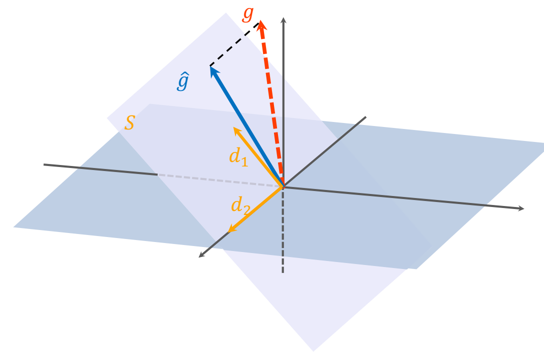


Fig. 3: Projection ($\hat{g}$) of gradient g on subspace ß.

Note that the search is over the rectification v not the input $\mathbf{x}$, that is, δ_i has the same dimension as v which also equals to the number of patches. v is initialized with all zero values in every estimation so as to guarantee that $\hat{\mathbf{x}}$ is centered at $\mathbf{x}$.

The complete algorithm with respect to targeted attack can be found in Algorithm 2, where FITNESS is a ranking-based nonlinear fitness function [35]. FITNESS computes a fitness value between 0 and 1 for each $\delta^{(i)}$, with a value close to 1 "awards" good $\delta^{(i)}$ that leads to higher probability on adversarial class y_{adv}, and a value close to 0 "punishes" bad $\delta^{(i)}$ that has low probability on y_{adv} (but the top-1 class is still y_{adv}) or the top-1 class is not y_{adv} any more. With this estimated gradient for v, we can update the rectification as $\hat{v} = v + \nabla_v \mathbb{E}[\ell(\hat{\mathbf{x}})]$. We then use $\hat{v}$ to rectify tentative perturbation h to $\hat{g}$ following Equation (4). This allows us to apply one step of PGD update on sample $\mathbf{x}$ to get an intermediate adversarial example $\hat{\mathbf{x}} = \mathbf{x} + \alpha \cdot \text{sign}(\hat{g})$ (α is the step size). The framework iteratively generates new tentative perturbations and performs new partition-based rectifications, based on the intermediate adversarial examples, until the adversarial target is achieved (see Algorithm 1). Note that, for untargeted attack, the top-1 probability should focus on the true class y, and the goal is to minimize $P(y|\mathbf{x})$, rather than maximizing $P(y_{adv}|\mathbf{x})$ as for targeted attack.

3.5 Analysis of V-BAD

We prove that partition-rectified perturbation is an estimation of the projection of the adversarial gradient on a selected subspace. Let $\mathcal{S}$ be the input space with respect to $\mathcal{X}$, direction vectors (d) of the tentative patches define a vector subspace ß $\subseteq \mathcal{S}$. In view of Equation (4), instead of using the raw factor v_i, let's now consider the rectification of partition d_i by a normalized factor $v_i / \|d_i\|_2$. Suppose we have a perfect estimation of directional derivative v: $v_i = \frac{\partial \ell}{\partial d_i}$, then the rectified perturbation becomes:

$$\hat{g} = \sum_i d_i \cdot (\frac{\partial \ell}{\partial d_i} / \|d_i\|_2), \tag{8}$$

where $\|\cdot\|$ is the L^2-norm. As d_i is the direction of an input patch, the directional derivative $\frac{\partial \ell}{\partial d_i}$ is the magnitude of the projection of adversarial gradient g on direction d_i if the model is differentiable. Thus, $\hat{g}$ is the vector sum of the gradient projection on each patch direction d_i. Therefore, $\hat{g}$ can be regarded as the projection of adversarial gradient g on subspace ß, as the projection of a vector on a subspace is the vector sum of the vector's projections on the orthogonal basis vectors of the subspace. Figure 3 illustrates a toy example of the projection, where g is the adversarial gradient, $\hat{g}$ is the projection of g on a subspace ß that is defined by the direction vectors of two patches: $d = [d_1, d_2]$.

The main difference between our proposed partition-based rectification and the direct estimation of adversarial gradient is that partition rectification decouples the adversarial gradient estimation into two parts: 1) directional basis of subspace ß (d) and 2) directional derivative (v). In an extreme case where projection $\Pi_{ß}(g) = g$, that is, each input dimension (or pixel) is a partition, the rectified perturbation $\hat{g}$ becomes exactly the adversarial gradient g, given perfect estimation of v. In a typical case, partition rectification is equivalent to estimating the projection of the adversarial gradient on a subspace ß. A beneficial property of the projection $\hat{g}$ is that it is the closest vector to the gradient in subspace ß having $\hat{g} = \arg\min_u(\|g - u\|_2), \forall u \in ß \subseteq \mathcal{S}$. This enables us to only consider the subspace ß to find a good estimation of the adversarial gradient g.

TABLE 1: Test Accuracy (%) of the video models.

Model	UCF-101	HMDB-51	Kinetics-400
I3D	91.30	63.73	64.71
CNN+LSTM	76.29	44.38	53.20

TABLE 2: Results for V-BAD with different: 1) tentative perturbation, 2) partitioning methods, and 3) estimation methods. Symbol "+" indicates fixed methods for untested components of V-BAD. Best results are highlighted in bold. ANQ: average number of queries; SR: success rate.

Components of V-BAD	Method	UCF-101 ANQ	UCF-101 SR (%)
Tentative Perturbation + partition: Uniform + estimation: NES	Static	51786	70
	Random	107499	95
	Single	57361	100
	Ensemble	**49797**	**100**
Partitioning Method + tentative: Ensemble + estimation: NES	Random	77881	100
	Uniform	**49797**	**100**
Estimation Method + tentative: Ensemble + partition: Uniform	FD	61585	70
	NES	**49797**	**100**

4 Experiments

In this section, we provide a comprehensive evaluation of our proposed V-BAD framework and its variants, for both untargeted and targeted video attacks on three benchmark video datasets, against two state-of-the-art video recognition models. We also investigate different choices of tentative perturbation, partitioning method and estimation method in an ablation study.

4.1 Experimental Setting

Datasets. We consider three benchmark datasets for video recognition: UCF-101 [36], HMDB-51 [37], and Kinetics-400 [38]. UCF-101 is an action recognition data set of $13{,}320$ realistic action videos, collected from YouTube, It consists of 101 action categories ranging from human-object/human-human interaction, body-motion, playing musical instruments to sports. HMDB-51 is a dataset for human motion recognition, which contains 6849 clips from 51 action categories including facial actions and body movements, with each category containing a minimum of 101 clips. Kinetics-400 is also a dataset for human action recognition, which consists of approximately $240{,}000$ video clips from 400 human action classes with about 400 video clips (10 seconds) for each class. At test time, we use 32-frame snippets for UCF-101 and HMDB-51, and 64-frame snippets for Kinetics-400, as it has longer videos.

Video Recognition Models. We consider two state-of-the-art video recognition models I3D and CNN+LSTM, as our target models to attack. I3D is an inflated 3D convolutional network. We use a Kinetics-400 pretrained I3D and finetune it on other two datasets. We sample frames at 25 frames per second. CNN+LSTM is a combination of the conventional 2D convolutional network and the LSTM network. For CNN+LSTM we use a ImageNet pretrained ResNet101 as frame feature extractor, then finetune a LSTM built on it. Input video frames are subsampled by keeping one out of every 5 for CNN+LSTM Model. Note we only consider the RGB part for both video models. The test accuracy of the two models can be found in Table 1. The accuracy gap between ours and the one reported in [3] is mainly caused by the availability of fewer input frames at test time.

Image Models. We use ImageNet [25] pretrained deep networks as our image model for the generation of the tentative perturbation. ImageNet is an image dataset that contains more than 10 million natural images from more than 1000 classes. Given the difference between our video datasets and ImageNet, rather than using one model, we chose an ensemble of ImageNet models as our image model: ResNet50 [39], DenseNet121 [40], and DenseNet169 [40].

Attack Setting. For each dataset, we randomly select one test video, from each category, that is correctly classified by the target model. We then apply an attack method to attack the target model by perturbing the videos towards misclassification for an untargeted attack, and a randomly chosen target class for a targeted attack. For all datasets, we set the maximum adversarial perturbation magnitude to $\epsilon = 0.05$ per frame. The query limit, *i.e.*, the maximum number of queries to the target model is set to $Q = 3 \times 10^5$, which is similar to the number of queries required for most black-box image attacks to succeed. We run the attack until an adversarial example is found (attack succeeds) or we reach the query limit. We evaluate different attack strategies in terms of 1) success rate (SR), the ratio of successful generation of adversarial examples under the perturbation bound within the limit of number of queries; and 2) average number of queries (ANQ), required for a successful attack (excluding failed attacks).

V-BAD Setting. For tentative perturbation, we average the perturbation extracted from the three image models

TABLE 3: Targeted attacks on UCF-101/HMDB-51/Kinetics-400 against I3D/CNN+LSTM models. Best results are in bold.

Target Model	Attack	UCF-101 ANQ	UCF-101 SR (%)	HMDB-51 ANQ	HMDB-51 SR (%)	Kinetics-400 ANQ	Kinetics-400 SR (%)
I3D	SR-BAD	67909	96.0	40824	96.1	63761	98.0
	P-BAD	104986	96.0	62744	96.8	84380	97.0
	V-BAD	**60687**	**98.0**	**34260**	**96.8**	**54528**	**100.0**
CNN + LSTM	SR-BAD	147322	45.5	67037	82.4	109314	73.0
	P-BAD	159723	60.4	72697	90.2	117368	85.0
	V-BAD	**84294**	**93.1**	**44944**	**98.0**	**70897**	**97.0**

TABLE 4: Untargeted attacks on UCF-101/HMDB-51/Kinetics-400 against I3D/CNN+LSTM models. Best results are in bold.

Target Model	Attack	UCF-101 ANQ	UCF-101 SR (%)	HMDB-51 ANQ	HMDB-51 SR (%)	Kinetics-400 ANQ	Kinetics-400 SR (%)
I3D	SR-BAD	5143	98.0	1863	100.0	1496	100.0
	P-BAD	11571	98.0	4162	100.0	3167	100.0
	V-BAD	**3642**	**100.0**	**1152**	**100.0**	**1012**	**100.0**
CNN + LSTM	SR-BAD	8674	100.0	684	100.0	1181	100.0
	P-BAD	12628	100.0	1013	100.0	1480	100.0
	V-BAD	**784**	**100.0**	**197**	**100.0**	**293**	**100.0**

to obtain the final perturbation. For partitioning, we use uniform partitioning to get 8×8 patches per frame and set the NES population size of each estimation as 48, which works consistently well across different datasets in terms of both success rate and number of queries. For search variance σ in NES, we set it to 10^{-6} for the targeted attack setting and 10^{-3} for the untargeted attack setting. This is because a targeted attack needs to keep the target class in the top-1 class list to get the score of the targeted class, while the aim of an untargeted attack is to remove the current class from top-1 class position, which enables a large search step. We use PGD attack for step-wise perturbation with dynamically chosen step size. For the targeted attack, we adjust the step size α and epsilon decay δ_ϵ dynamically. If the ratio of failing to maintain the adversarial class is higher than a threshold of 50%, we halve the step size α. If we fail to reduce the perturbation size ϵ after 100 times, we halve the epsilon decay δ_ϵ.

4.2 Ablation Study of V-BAD

In this section, we evaluate variants of V-BAD, with different types of 1) tentative perturbation, 2) partitioning methods and 3) estimation methods. Experiments were conducted on a subset of 20 randomly selected categories from UCF-101 dataset.

Tentative Perturbation. We first evaluate V-BAD with the three different types of tentative perturbation discussed in Section 3.3: 1) Random, 2) Static, and 3) Transferred. For transferred, we test two different strategies, with either a single image model ResNet-50 (denoted as "Single") or an ensemble of the three considered image models (denoted as "Ensemble"). The partitioning and estimation methods were set to Uniform and NES respectively. The results in terms of success rate (SR) and average number of queries (ANQ) can be found in Table 2. A clear improvement can be observed for the use of transferred tentative perturbation compared to static or random perturbation. The number of queries required for successful attacks was dramatically reduced from more than 10^5 to less than 6×10^4 by using only a single image model. This confirms that an image model alone can provide effective guidance for attacking video models. The number was further reduced to around 5×10^4 via the use of an ensemble of three image models. Compared to random perturbation, static perturbation requires fewer queries to succeed, with 50% less queries due to the small exploration space. However, static perturbation has a much lower success rate (70%) than random perturbation (95%). This indicates that fixed directions can generate adversarial examples faster with restricted exploration space, but at the same time, such restrictions may render the algorithm to be stuck in some local minima, without exploring other directions that could lead to more potentially more powerful attacks.

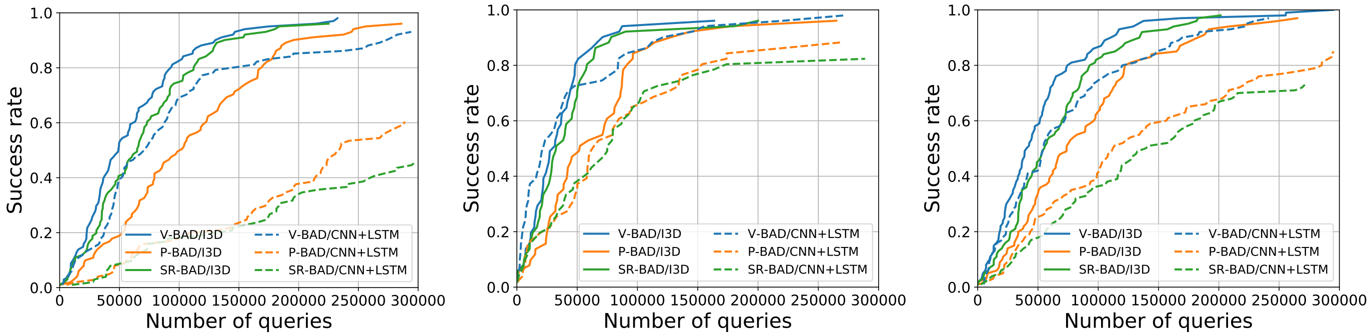
Partitioning Methods. In this experiment, we investigate the two types of partitioning methods introduced in Section 3.4, namely, Random and Uniform. For tentative perturbation, we use the best method found in the previous experiments - Ensemble. Results are reported in Table 2. As can be seen, both partitioning methods can find successful attacks, but the use of uniform partitioning significantly reduces the number of queries by 36%, to around 5×10^4 from 8×10^4 of random partitioning. This is because a tentative perturbation generated from image models often shares certain local patterns, but random partitioning tends to destroy such locality. Recall that partitioning is applied in every step of perturbation, and as such, uniform partitioning can help to maintain stable and consistent patches across different perturbation steps. This allows the rectification to make continuous corrections to the same local patches.

Rectification Estimators. As we mentioned in Section 3.4, any derivative-free (or black-box) optimization methods can be used to estimate the rectification factors. Here, we compare two of the methods that have been used for black-box image attacks: FD (Finite Difference) [18] and NES (Natural Evolution Strategies) [19]. For a fair comparison, we made some adjustments to the number of patches used by the FD estimator, so that FD and NES noth require a similar number of queries per update. The results are reported in Table 2. NES demonstrates a clear advantage over FD: FD only achieves 70% success rate within 3×10^5 queries, while NES has a 100% success rate with a lower average number of queries.

Based on the ablation results, in the following experiments, we set the V-BAD framework to: ensemble tentative directions, uniform partitioning and NES rectification estimator.

4.3 Comparison to Existing Attacks

In this section, we compare our V-BAD framework with two existing state-of-the-art black-box image attack methods [18], [19]. Instead of directly applying the two image attack methods to videos, we instead incorporate their logic into our V-BAD framework to obtain two variants of V-BAD.



(a) Results on UCF-101 (b) Results on HMDB-51 (c) Results on Kinetics-400

Fig. 4: Comparative results for targeted attack.

Baselines. The first baseline method is the pixel-wise adversarial gradient estimation using NES, proposed in [19]. This can be easily achieved by setting the patches in V-BAD to pixels, *i.e.*, each pixel is a patch. We denote this baseline variant of V-BAD by P-BAD. P-BAD uses static (*e.g.* all positive) tentative perturbation, following the setting in [19], where no transferred gradients were used. The NES population size for P-BAD is set as 96, since there are many more parameters to estimate. The second baseline method is grouping-based adversarial gradient estimation using FD, proposed by [18]. This method explores random partitioning to reduce the large number of queries required by FD, and does not use tentative directions. Accordingly, we use the variant of V-BAD that utilizes static tentative directions and random partioning, and denote it by SR-BAD. Different from its original setting with FD, here we use NES for SR-BAD which was found more efficient in our previous experiments. It is worth mentioning that a dimensionality reduction technique (*e.g.* PCA) was also explored in [18] to project inputs to low dimensional features. Although it increased attack success rate on simple black-white images (*e.g.* MNIST [41]), it did not help attacks on natural images (*e.g.* CIFAR10 [24]), due to the information loss caused by the projection.

Targeted Black-box Video Attack. Comparison results for targeted attacks are reported in Table 3. Among the three methods, V-BAD achieved the best success rates, consistently using least number of queries across the three datasets and two recognition models. Specifically, V-BAD only takes $(3.4 \sim 8.4) \times 10^4$ queries to achieve a success rate of above 93%. Note that this is comparable to state-of-the-art black-box image attacks [19]. Comparing P-BAD and V-BAD, pixel-wise estimation by P-BAD does not seem to yield more effective attacks, whereas on the contrary, uniform partitioning by V-BAD not only reduces $\sim 50\%$ queries, but also leads to more successful attacks. Compare the performance on different target models, an obvious degradation of performance can be observed on CNN+LSTM model. This is because CNN+SLTM has a lower accuracy than I3D, making it relatively robust to targeted attacks (not for untargeted attacks), an observation that is consistent with findings in [42]. However, this impact is much smaller on V-BAD where the accuracy decreases less than 5%, while the accuracy of P-BAD and SR-BAD has a huge drop, especially on UCF-101(from 96.0% to 45.5%). This is further illustrated in Figure 4, showing change in success rate with the number of queries. This can probably be explained by the better transferability of transferred tentative perturbation on CNN+LSTM than I3D due to the similar 2D CNN used in CNN+LSTM video model. As in Figure 4, the advantage of better transferability even overcomes the low accuracy drawback of CNN+LSTM on HMDB-51 dataset: V-BAD/CNN+LSTM is above V-BAD/I3D for the first 4×10^4 queries. A targeted video adversarial examples generated by V-BAD is illustrated in Figure 1, where video on the top is the original video with the correct class and video at the bottom is the video adversarial misclassified as the adversarial class.

TABLE 5: Cosine similarity between varies tentative or rectified gradients and the actual gradient.

Tentative	Static	Random	Transferred
Cosine	7.177×10^{-5}	-1.821×10^{-5}	-2.743×10^{-4}
Rectified	SR-BAD	P-BAD	V-BAD
Cosine	3.480×10^{-3}	3.029×10^{-3}	4.661×10^{-3}

Untargeted Black-box Video Attack. Results for untargeted attacks are reported in Table 4. Compared to targeted attacks, untargeted attacks are much easier to achieve, requiring only $\sim 10\%$ queries of targeted attacks. Compared to other baselines, V-BAD is the most effective and efficient attack across all datasets and recognition models. It only takes a few hundred queries for V-BAD to completely break the CNN+LSTM models. This indicates that video models are as vulnerable as image models to black-box adversarial attacks. This has serious implications for the video recognition community to consider.

4.4 Gradient Estimate Quality

We further explore the quality of various tentative gradients (perturbations) and rectified gradients generated by different variants of V-BAD. We measure the gradient quality by calculating the cosine similarity between the ground-truth adversarial gradient and the tentative/rectified gradients. The results are based on 20 random runs of the attacks on 50 videos randomly chosen from UCF-101, and are reported in Table 5. Consistent with the comparison experiments, V-BAD generates the best gradient estimates and P-BAD has the worst estimation quality. All the rectified gradients are much better than the tentative gradients. This verifies that tentative perturbations can be significantly improved by proper rectification. One interesting observation is that the transferred tentative perturbation (from an ensemble of ImageNet models) has a large negative cosine similarity, which is opposite to our design. One explanation could be that there is a huge gap between the image model and the video model. However, note that while the transferred perturbation is opposite to the gradient, it serves as a good initialization and yields better gradient estimation after rectification. It is noteworthy that there is still a considerable gap between the gradient estimate and the actual gradient. From one angle, it reflects that we do not need very accurate gradient estimation to generate adversarial examples. From another angle, it suggests that black-box attack based on gradient estimation has great scope for further improvement.

5 Conclusion

In this paper, we investigated the problem of black-box adversarial attack against video recognition models, and proposed the first framework, V-BAD, for the generation of video adversarial examples through only black-box queries to a video model. To address efficiency issues caused by the high dimensionality of videos, we decoupled adversarial gradient estimation into a two-step process: tentative perturbation transfer followed by partition-based rectification with NES estimation method. We demonstrated the effectiveness and efficiency of V-BAD by attacking two state-of-the-art video recognition models on three benchmark video datasets. Compared to existing black-box methods, V-BAD achieved high success rate using significantly less queries for both targeted and untargeted attacks. Our results suggest that video models are also highly vulnerable to black-box adversarial attacks, and that effective defenses for video models will need to be developed for secure video recognition.

References

[1] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *CVPR*, 2014.

[2] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221–231, 2013.

[3] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *CVPR*, 2017.

[4] Z. Wu, Y.-G. Jiang, X. Wang, H. Ye, and X. Xue, "Multi-stream multi-class fusion of deep networks for video classification," in *ACM MM*, 2016.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.

[6] R.-W. Zhao, Z. Wu, J. Li, and Y.-G. Jiang, "Learning semantic feature map for visual content recognition," in *ACM MM*, 2017.

[7] Z. Yang, Y. Han, and Z. Wang, "Catching the temporal regions-of-interest for video captioning," in *ACM MM*, 2017.

[8] S. Liu, Z. Ren, and J. Yuan, "Sibnet: Sibling convolutional encoder for video captioning," in *ACM MM*, 2018.

[9] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *ICLR*, 2014.

[10] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *ICLR*, 2015.

[11] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *CVPR*, 2015.

[12] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *S&P*, 2017.

[13] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *arXiv preprint arXiv:1607.02533*, 2016.

[14] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust physical-world attacks on deep learning visual classification," in *CVPR*, 2018.

[15] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *CVPR*, 2017.

[16] N. Papernot, P. McDaniel, and I. Goodfellow, "Transferability in machine learning: from phenomena to black-box attacks using adversarial samples," *arXiv preprint arXiv:1605.07277*, 2016.

[17] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *ASIACCS*, 2017.

[18] A. N. Bhagoji, W. He, B. Li, and D. Song, "Practical black-box attacks on deep neural networks using efficient query mechanisms," in *ECCV*, 2018.

[19] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin, "Black-box adversarial attacks with limited queries and information," 2018.

[20] 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 *AISec*, 2017.

[21] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *ICLR*, 2018.

[22] S. Li, A. Neupane, S. Paul, C. Song, S. V. Krishnamurthy, A. K. R. Chowdhury, and A. Swami, "Adversarial perturbations against real-time video classification systems," *arXiv preprint arXiv:1807.00458*, 2018.

[23] R. Rey-de Castro and H. Rabitz, "Targeted nonlinear adversarial perturbations in images and videos," *arXiv preprint arXiv:1809.00958*, 2018.

[24] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Citeseer, Tech. Rep., 2009.

[25] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009.

[26] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *EuroS&P*, 2016.

[27] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *CVPR*, 2016.

[28] P.-Y. Chen, Y. Sharma, H. Zhang, J. Yi, and C.-J. Hsieh, "Ead: elastic-net attacks to deep neural networks via adversarial examples," in *AAAI*, 2018.

[29] X. Wei, J. Zhu, and H. Su, "Sparse adversarial perturbations for videos," *arXiv preprint arXiv:1803.02536*, 2018.

[30] Y.-G. Jiang, Z. Wu, J. Wang, X. Xue, and S.-F. Chang, "Exploiting feature and class relationships in video categorization with regularized deep neural networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 2, pp. 352–364, 2018.

[31] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: Deep networks for video classification," in *CVPR*, 2015.

[32] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *ICCV*, 2015.

[33] G. Varol, I. Laptev, and C. Schmid, "Long-term temporal convolutions for action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1510–1517, 2018.

[34] T. Salimans, J. Ho, X. Chen, S. Sidor, and I. Sutskever, "Evolution strategies as a scalable alternative to reinforcement learning," *arXiv preprint arXiv:1703.03864*, 2017.

[35] D. Wierstra, T. Schaul, T. Glasmachers, Y. Sun, J. Peters, and J. Schmidhuber, "Natural evolution strategies," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 949–980, 2014.

[36] K. Soomro, A. R. Zamir, and M. Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, 2012.

[37] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "Hmdb: a large video database for human motion recognition," in *ICCV*, 2011.

[38] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev *et al.*, "The kinetics human action video dataset," *arXiv preprint arXiv:1705.06950*, 2017.

[39] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[40] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *CVPR*, 2017.

[41] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel, "Handwritten digit recognition with a back-propagation network," in *NIPS*, 1990.

[42] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry, "Robustness may be at odds with accuracy," in *ICLR*, 2019.