Merry Go Round: Rotate a Frame and Fool a DNN

Daksh Thapar, Aditya Nigam
Indian Institute of Technology Mandi
dakshthapar.github.io,faculty.iitmandi.ac.in/-aditya

Chetan Arora
Indian Institute of Technology Delhi
www.cse.iitd.ac.in/-chetan

Abstract

A large proportion of videos captured today are first person videos shot from wearable cameras. Similar to other computer vision tasks, Deep Neural Networks (DNNs) are the workhorse for most state-of-the-art (SOTA) egocentric vision techniques. On the other hand DNNs are known to be susceptible to Adversarial Attacks (AA) which add imperceptible noise to the input. Both black-box, as well as white-box attacks on image as well as video analysis tasks have been shown. We observe that most AA techniques basically add intensity perturbation to an image. Even for videos, the same process is essentially repeated for each frame independently. We note that definition of imperceptibility used for images may not be applicable for videos, where a small intensity change happening randomly in two consecutive frames may still be perceptible. In this paper we make a key novel suggestion to use perturbation in optical flow to carry out AAs on a video analysis system. Such perturbation is especially useful for egocentric videos, because there is lot of shake in the egocentric videos anyways, and adding a little more, keeps it highly imperceptible. In general our idea can be seen as adding structured, parametric noise as the adversarial perturbation. Our implementation of the idea by adding 3D rotations to the frames, reveal that using our technique, one can mount a black-box AA on an egocentric activity detection system in one-third of the queries compared to the SOTA AA technique.

1. Introduction

Despite achieving superior performance on a variety of computer vision tasks [3,11,12,33], Deep Neural Networks (DNNs) remain remarkably susceptible to imperceptible adversarial perturbations [37]. The goal of an adversarial attack (AA) is, given a clean image, \( I \), create an adversarial perturbation (\( P \)), which when added to the clean image, generates an adversarial sample \( I_{\text{adv}} = I + P \), which tricks a DNN model into producing an incorrect prediction. Since the purpose is to attack a system, the perturbation should be imperceptible to the humans.

The simplest setting to mount such an AA is when the adversary gets full access to the model (\( M \)), including input(\( X \))/output(\( Y \)), and the exact gradients (\( G \)). One can, then, simply backpropagate the loss corresponding to the desired (incorrect) output, and use it guide the perturbation in the input \([16,25,37]\). The setting is called white box attacks, but is usually impractical in real life, due to unavailability of the full access to the model. The alternate setting is the black box setting when an adversary has access to \( X \), and \( Y \), but not \( G \). In this formulation the primary challenge becomes estimating the gradient at the input without having access to \( G \) \([6,16,17]\). The quality of an AA technique is usually determined by how imperceptible the \( P \) is, and additionally in case of black box attacks, how many \((X,Y)\) pairs a technique needs to find a \( P \) corresponding to a particular \( I \).
Researchers have shown both white box and black box attacks for a variety of DNN models across range of tasks [37]. Further, relevant to our context, the attacks have been shown when the input to the model is an image [16,17], or a video [20,49]. Our focus in this paper is on mounting black box adversarial attacks on video analysis (VA) systems.

We note that most of the techniques for AA on a VA system trivially extends the black box pipeline from images to videos. The videos are broken down into frames, and adversarial examples are created by adding random perturbations in the pixel intensities [20, 49]. For a successful attack, these methods require a large number of queries on the target model. For example [20] requires 23K queries on an average for generating a single adversarial sample. We would like to emphasize that a frame-wise attack, using intensity-based noise, do not coordinate the adversarial perturbations between consecutive frames. While a change in intensity level of a few individual pixels may be imperceptible in an individual frame, when played as a video, such random flashes are easily detected by a human being.

One of the key ideas of this paper is to parameterize the perturbation. The parameterization has two advantages, (1) it is easier to regularize within, and across the frames, and (2) one can perturb a large number of pixels, by estimating only a few parameters, thus reducing the query budget, an important consideration in a black box attack. While, the idea of parametric perturbation is generic and can be used in variety of settings, given our focus to videos, we consider it for attack on VA systems, and even more specifically, on egocentric VA systems.

We observe that one of the simplest ways to perform coordinated change in intensity levels of large number of pixels, across frames of a video, is by geometrically transforming each frame. The transformation will cause change in the optical flow, which is an important cue for many VA tasks. At the same time, performing frame-wise geometric transformation maintains semantic integrity of frame contents, keeping it imperceptible to human beings.

Contributions: The key contributions of this work are:

1. We propose to add novel parametric perturbations to mount an AA attack against a computer vision system.
2. For a VA system, we suggest use of geometric transformations to implement such parametric perturbations.
3. We propose a novel DNN architecture for predicting a mix of intensity, and geometric perturbations which can successfully fool a VA system to carry out black box AA attack.
4. Our exhaustive experiments on multitude of benchmark datasets, and VA tasks for egocentric, and third person videos show that our proposed architecture outperforms SOTA techniques, managing to fool a DNN in one-third of the queries as needed by the SOTA.

2. Related Work

Adversarial Attacks: Szegedy et al. [37] have shown that by computing a small noise on the original image, one can create an adversarial example. Papernot et al. [25] have shown that a black box attack can be carried out on a target model by transferring the adversarial examples of a local trained network. However, such a technique still requires knowledge of the dataset and training procedure of the target model. Natural Evolutionary Strategies have been extended in [16] to perform gradient estimation. Ilyas et al. [17] have shown that time and data-dependent priors can reduce the number of queries in black box attacks. The meta-based method has been proposed by Du et al. [6] for black box attacks on image analysis models. However, little work has been done on attacking DNNs for VA. Further, to the best of our knowledge, there is no AA proposed for egocentric VA models.

Adversarial Attacks on Video Analysis Models: For third-person videos, Wei et al. [44] have investigated the sparsity and propagation of adversarial perturbations across videos for creating a white-box attack. Li et al. [22] have proposed Generative Adversarial Networks to synthesize adversarial examples for a video classification DNN. Inkawhich et al. [18] have proposed an FGSM [10] style of attacks for attacking a two-stream video classifier. Chen et al. [4] added a few fake frames to attack video classification DNNs. The first black-box video attack is proposed by Jiang et al. [20], where they have used an ImageNet pre-trained model to create a gradient for each video frame and refined them by using natural-evolution-strategies [16]. More recently, [45, 47] perturb only a few selected frames rather than the whole video. In [49] a motion based sampler for perturbing every frame in the video has been proposed.

Third-person Video Analysis: Recent methods for third-person video action recognition utilize 3D CNNs [2, 7, 19, 40, 43, 50]. 3D CNNs extend 2D filters in temporal dimensions to extract spatio-temporal features from videos. Since early 3D models [19, 40] are hard to train, many follow-up works have been proposed [2, 7, 31, 41]. Two-stream methods proposed in [34] combine a spatial network using RGB images and a temporal network taking optical flow input. Optical flow information has also been found beneficial in few-shot video classification [51].

First-person Video Analysis: Some notable works in general egocentric video analysis include camera wearer’s activity and action recognition [1, 21, 28–30, 35, 36, 42], wearer gaze estimation [15], temporal segmentation [24], and video summarization [32, 46]. Another uniquely egocentric video task is recognizing the wearer capturing the video. The task has attracted lot of attention in recent years [7, 8, 13, 14, 23, 26, 27, 30, 38, 39].
3. Proposed Methodology

3.1. Gradient Estimation

We consider a DNN model $f$, which has been pre-trained for some VA task. The model takes as input a video $V \in \mathbb{R}^{T \times H \times W \times C}$, where $T$, $H$, $W$, and $C$ represent video length, height, width, and number of channels (in each frame) respectively. Assuming a video classification model, the output of $f$ is a label $y \in \{1, \ldots, K\}$, where $K$ is the number of classes.

The goal of an adversarial attack is, given an input video $V$, generate an adversarial video $V_{\text{adv}}$ which minimises the loss function:

$$\mathcal{L} = \max(1_y - \max_{k \neq y}(1_k), 0).$$  \hspace{1cm} (1)

Here $1_i$ is the logit vector corresponding to input $V_{\text{adv}}$, and $1_i$ is the value of $i$th element (corresponding to class $i$) of the vector. Minimizing $\mathcal{L}$ confuses the model with the second most confident class prediction for the untargeted adversarial attack. For the targeted attack $\max_{k \neq y}(1_k)$ can be replaced by the logit of the corresponding class. To simplify the notation, in the rest of the paper we simply use $\mathcal{L}(V, y)$ instead of $\mathcal{L}(f(V), y)$. The adversarial video $V_{\text{adv}}$ is chosen as:

$$\arg \min_{V_{\text{adv}}} \mathcal{L}(\text{Pert}(V_{\text{adv}}), y)$$

s.t.

$$\text{dist}(V_{\text{adv}}, V) \leq \text{max}\_\text{dist},$$

and

$$\#\text{queries} \leq Q.$$  \hspace{1cm} (2)

We can model $V_{\text{adv}}$ using any perturbation parameterized by $\theta \in \mathbb{R}^{T \times d}$, where $d$ is the dimension of $\theta$, s.t. $V_{\text{adv}} = \text{Pert}(V, \theta)$. Here, the function $\text{Pert}(V, \theta)$ applies the perturbation parameterized by $\theta$ on the video $V$. The function $\text{Pert}$ will be dependent upon the type of perturbation and is defined in detail in Sec. 3.2. To generate an adversarial video $V_{\text{adv}}$, we need to find an optimal perturbation $\theta^*$ s.t.:

$$\theta^* = \arg \min_{\theta} \mathcal{L}(\text{Pert}(V, \theta), y)$$

s.t.

$$\|\theta\|_2 \leq k,$$

and

$$\#\text{queries} \leq Q.$$  \hspace{1cm} (3)

Here $k$ is the maximum perturbation allowed. We have used $\ell_2$ norm for constraining the $\theta$, but any other suitable con-
strait on the θ could have been used. The above perturbation framework allows us to generalize the adversarial attacks additive, multiplicative, or even some complex non-differentiable perturbations. Moreover, it allows us to design a parametric perturbation of a very low dimension d which is easier to compute in limited query budget.

The key challenge in black-box adversarial attacks is to estimate the gradient of a model. It is because for this setting, the model is not accessible (beyond input, output), and the gradient ∇gL(Pert(V, θ), y), required for generating Vadv, cannot be directly computed. Hence, we adopt an iterative optimization strategy suggested in [49] for estimating ∇gL(Pert(V, θ), y).

It is important to note that for an iterative optimization, we are only interested in the direction of ∇gL(Pert(V, θ), y) rather than its exact value which also includes the magnitude. Hence, we learn a vector g ∈ ℝT×d whose direction (g∥g∥) aligns with ∇gL(Pert(V, θ), y). In order to estimate such a g, we use the following loss function [17]:

\[
l(g) = -\langle \nabla_g L(Pert(V, \theta), y), \frac{g}{\|g\|} \rangle,
\]

which is the inverse of directional derivative of L, in the direction of the vector g. The inverse direction of directional derivative provides the direction of g’s movement to optimize l(g) and get closer to the desired gradient ∇gL(Pert(V, θ), y) as:

\[
g^* = \arg \min_g l(g).
\]

In order to compute g*, we compute the gradient ∇g l(g), denoted as Δ. We perform a two-query estimation to the expectation and apply the authentic sampling [17] to get:

\[
\Delta = \left[ \frac{l(g + \delta r) - l(g - \delta r)}{\delta} \right] r,
\]

where r ∈ ℝT×d ∈ N(0, 1/2(I)) is the Gaussian noise, and δ is a small number scaling the magnitude of loss variation. In two-query estimation, r vector acts as a directional candidate for the update of g. We query in the direction of r and its opposite direction. This gives us a scalar indicating of how good the candidate r is. We scale r accordingly to form our update of g.

Finally, Eq. (4) can be approximated as [17]:

\[
l(g) \approx -\langle \nabla_g L(Pert(V, \theta), y), \frac{g}{\|g\|} \rangle \\
\approx -\frac{L(Pert(V, \theta + \epsilon g^+), y) - L(Pert(V, \theta), y)}{\epsilon},
\]

where ε is a small approximation constant. Substituting Eq. (7) into Eq. (6), we get GE(V, y, θ, g) as:

\[
\Delta = GE(V, y, \theta, g) \\
= \left[ \frac{L(Pert(V, \theta + \epsilon g^+), y) - L(Pert(V, \theta + \epsilon g^-), y)}{\epsilon} \right] r,
\]

where g^+ = g + δr and g^- = g - δr.

### 3.2. Parametric Noise

It can be observed from Eq. (8), that in order to estimate the gradient, we have utilized a random noise (r). For intensity-based noise, r_{in} ∈ ℝT×H×W×C is used for estimating the gradient g_{in} ∈ ℝT×H×W×C [49]. This requires one to estimate T × H × W × C parameters for the adversarial attack, which may lead to a high number of queries [49], making such attacks unrealistic in practice.

To overcome these limitations, we have proposed a parametric noise (camera rotational noise r_{cr}) which can suitably alter the geometrical properties of a video for an attack. Since, rotation of the camera can be represented as a 3D vector in Euler space, the proposed noise r_{cr} ∈ ℝT×3 requires only T × 3 parameters to be predicted for an adversarial attack. This significantly reduces the number of queries required to predict it in comparison to an intensity-based noise.

We estimate the camera rotational gradient g_{cr} ∈ ℝT×3 from r_{cr} using gradient estimation, as discussed in the previous section. This allows us to find a new perturbation vector θ, with θ_i ∈ ℝ^3 for each frame. Recall, that θ_i corresponds to a 3D rotation for the frame. We compute an Homography using the 3D rotation as \mathcal{H}_i = K · θ_i K^{-1}, where K is the camera internal matrix (assumed identity in our case). The perturbation can be applied on the video as:

\[
Pert_{cr}(V, \theta) = \forall_i (\mathcal{H}_i \ast V_i),
\]

where, V_i is the i^{th} frame in the video V and \ast denotes the geometric transformation of each frame using the Homography \mathcal{H}_i. To ensure that the perturbations are small, we have clipped the magnitude of r_{cr} to 0.18 radians.

We observe that in our experiments the number of queries required to render a successful black-box attack gets substantially reduced by using parametric noise, but at the expense of success rate (refer Sec. 4.2). Hence, we propose to mix it with intensity based perturbation, using a learnable composition parameter, as described in the next section.

### 3.3. Gradient Composition

In order to address the issue of low success rate using parametric noise, we propose a novel learnable gradient composition framework which suitably combines intensity-based, and parametric perturbations. Such fusion exploits spatio-temporal properties of a particular segment in a video
to dynamically adjust the weights of two kinds of perturbation, and achieve lower queries. For example, if there is very small motion between two frames, intensity based noise can be more effective. However, in the case of large temporal movements of objects or camera, the rotational noise can be useful. We propose a Siamese network based architecture, named DifferenceNet, to predict the weight of each perturbation for a frame.

**DifferenceNet**: The proposed DifferenceNet model is a 3D CNN model (with I3D [2] pipeline) trained to calculate semantic difference between input video \( V \) and adversarial video \( V_{adv} \). The task of DifferenceNet is to provide a low difference score to videos which are semantically similar otherwise a high score. This is achieved by training the network with a dual margin contrastive loss function [48]. The network is trained over positive pairs which have the camera rotations between the frames corresponding to actual videos and negative pairs having abrupt rotations between the frames. To create positive and negative pairs, real Homographies \( \mathcal{H}_{real} \), between the frames from the given dataset \( D \) and random/fake Homographies have been generated. Application of \( \mathcal{H}_{real} \), \( \mathcal{H}_{rand} \) on a video segment \( V \), gives us \( (V^p, V^n) \) constituting a positive and negative pair as \( ((V, V^p)), ((V, V^n)) \) respectively. Finally, the trained network is utilized for gradient composition as described below.

**Gradient Composition**: For a given input \( V \), intensity based perturbation, and camera based perturbations are combined as:

\[
\hat{V} = \text{Pert}_{in}(V, \alpha \gamma \theta_{in}) \\
V_{adv} = \text{Pert}_{cr}(\hat{V}, \alpha (1 - \gamma) \theta_{cr}),
\]

(10)

where, \( \gamma \in [0, 1]^{T \times 1} \) is the composition parameter and \( \alpha \) is a small constant. Since \( \gamma \) depends on semantic difference between \( (V, V_{adv}) \), we have utilized DifferenceNet to predict its value:

\[
d = \text{DifferenceNet}(V, V_{adv}) \\
\gamma = \gamma - \sigma \left( \frac{\delta d}{\delta \gamma} \right),
\]

(11)

where \( \sigma \) is a small constant.

3.4. Projected Gradient Descent

Finally, projection gradient descent (PGD) has been utilized to translate gradient estimation and its combination into an efficient Adversarial Example Optimization (AEO). We update intensity based perturbation (Pert\( (V, \theta_{in}) \)), camera rotational perturbation (Pert\( (V, \theta_{cr}) \)), and composition parameter \( \gamma \) in every iteration of PGD. The complete procedure is shown in Algorithm 1.

**Algorithm 1**: Adversarial Example Optimization (AEO)

\[
\begin{align*}
\text{Input:} & \quad \text{Original video } V, \text{ its label } y, \text{ learning rate } \alpha \\
\text{Output:} & \quad V_{adv} \\
\end{align*}
\]

1. Initialise \( g_{in} = 0, g_{cr} = 0, \theta_{in} = 0, \theta_{cr} = 0 \) and \( \gamma = 0.5 \)

2. while arg max \( |f(V)| = y \) do

3. \[ \Delta_{in} = GE(V, y, \theta_{in}, g_{in}) \] // Eq 8

4. \[ \Delta_{cr} = GE(V, y, \theta_{cr}, g_{cr}) \] // Eq 8

5. \( g_{in} = g_{in} - \eta \Delta_{in} \) // Grad. Update

6. \( g_{cr} = g_{cr} - \eta \Delta_{cr} \) // Grad. Update

7. \( \theta_{in} = \theta_{in} - g_{in} \) // Param. Update

8. \( \theta_{cr} = \theta_{cr} - g_{cr} \) // Param. Update

9. \( V = \text{Pert}_{in}(V, \theta_{in} \gamma \alpha) \) // Grad. Composition

10. \( V_{adv} = \text{Pert}_{cr}(V, \theta_{cr}(1 - \gamma) \alpha) \) // Grad. Composition

11. \( d = \text{DifferenceNet}(V, V_{adv}) \)

12. \( \gamma = \gamma - \alpha \times \frac{\delta d}{\delta \gamma} \)

13. \( V = V_{adv} \)

4. Experiments and Results

In this section, we provide the details of the experimental analysis performed to validate the efficacy of the proposed method. We start with the details of the experimental setup, including details about the datasets used, target DNN models attacked, attack setting, and evaluation metrics. Finally, we show the comparative analysis and ablation study using both quantitative and qualitative experiments.

4.1. Dataset and Evaluation

**Datasets**: We perform video attacks on three video tasks: third-person action recognition using Kinetics-400 [2] dataset, first-person activity recognition via Epic-Kitchens [5] dataset, and first-person wearer recognition using IITMD-WFP [38] dataset. Kinetics-400 is a large-scale dataset that has around 300K videos in 400 classes. Epic-Kitchens is a first-person activity recognition dataset that consists of 55 hours of egocentric videos from 32 subjects and contains 125 labeled activities performed by the subjects. IITMD-WFP dataset [38] consists of 3.1 hours of videos captured from 31 different subjects. The dataset has been captured under indoor and outdoor scenarios.

**DNN Video Analysis Models Used for Experiments**: For third-person video action recognition, we follow the experimental setup of [49]. We choose video action recognition model I3D [2] as our black-box model. For I3D training on Kinetics-400, we train it from ImageNet initialized weights. For first-person activity recognition, we choose
Table 1. Untargeted attacks on Kinetics-400, Epic-Kitchens, and IITMD-FPR. The attacked models are I3D, Rolling-Unrolling LSTM, and EgoGaitNet respectively.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>ANQ</th>
<th>SR%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kinetics-400</td>
<td>V-Bad [20]</td>
<td>4,047</td>
<td>99.75</td>
</tr>
<tr>
<td></td>
<td>ME-Sampler [49]</td>
<td>2,717</td>
<td>99.00</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>1,257</td>
<td>99.33</td>
</tr>
<tr>
<td>Epic-Kitchens</td>
<td>V-Bad [20]</td>
<td>8,483</td>
<td>99.71</td>
</tr>
<tr>
<td></td>
<td>ME-Sampler [49]</td>
<td>7,326</td>
<td>100.00</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>3,564</td>
<td>100.00</td>
</tr>
<tr>
<td>IITMD-FPR</td>
<td>V-Bad [20]</td>
<td>5,480</td>
<td>94.67</td>
</tr>
<tr>
<td></td>
<td>ME-Sampler [49]</td>
<td>6,025</td>
<td>92.62</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>3,487</td>
<td>96.33</td>
</tr>
</tbody>
</table>

Table 2. Targeted attacks on Kinetics-400, Epic-Kitchens, and IITMD-FPR. The attacked models are I3D, Rolling-Unrolling LSTM, and EgoGaitNet respectively.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>ANQ</th>
<th>SR%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kinetics-400</td>
<td>V-Bad [20]</td>
<td>23,182</td>
<td>92.95</td>
</tr>
<tr>
<td></td>
<td>ME-Sampler [49]</td>
<td>11,120</td>
<td>94.67</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>6,234</td>
<td>95.82</td>
</tr>
<tr>
<td>Epic-Kitchens</td>
<td>V-Bad [20]</td>
<td>44,326</td>
<td>84.23</td>
</tr>
<tr>
<td></td>
<td>ME-Sampler [49]</td>
<td>22,541</td>
<td>89.12</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>15,283</td>
<td>91.56</td>
</tr>
<tr>
<td>IITMD-FPR</td>
<td>V-Bad [20]</td>
<td>34,382</td>
<td>82.19</td>
</tr>
<tr>
<td></td>
<td>ME-Sampler [49]</td>
<td>18,759</td>
<td>86.67</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>9,910</td>
<td>87.33</td>
</tr>
</tbody>
</table>

Table 3. Ablation study on Kinetics-400, Epic-Kitchens, and IITMD-FPR. The attacked models are I3D, Rolling-Unrolling LSTM, and EgoGaitNet respectively.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>ANQ</th>
<th>SR%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kinetics-400</td>
<td>Only Intensity</td>
<td>3,569</td>
<td>99.0</td>
</tr>
<tr>
<td></td>
<td>Only Rotation</td>
<td>1,067</td>
<td>38.19</td>
</tr>
<tr>
<td></td>
<td>Manual Composition</td>
<td>1,884</td>
<td>62.50</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>1,257</td>
<td>99.33</td>
</tr>
<tr>
<td>Epic-Kitchens</td>
<td>Only Intensity</td>
<td>8,238</td>
<td>100.00</td>
</tr>
<tr>
<td></td>
<td>Only Rotation</td>
<td>3,286</td>
<td>62.81</td>
</tr>
<tr>
<td></td>
<td>Manual Composition</td>
<td>4,467</td>
<td>79.67</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>3,564</td>
<td>100.00</td>
</tr>
<tr>
<td>IITMD-FPR</td>
<td>Only Intensity</td>
<td>6,356</td>
<td>95.23</td>
</tr>
<tr>
<td></td>
<td>Only Rotation</td>
<td>3,286</td>
<td>58.42</td>
</tr>
<tr>
<td></td>
<td>Manual Composition</td>
<td>4,019</td>
<td>72.48</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>3,487</td>
<td>96.33</td>
</tr>
</tbody>
</table>

Rolling-Unrolling LSTM [9] as our black-box model. The pre-trained weights of the model have been provided by the authors. For first-person wearer recognition, we choose EgoGaitNet [38] model. We perform the training procedure as suggested by the authors, and using the code provided.

**Attack Setting [49]:** We perform both untargeted and targeted attacks under limited queries. An untargeted attack requires the given video to be mis-classified to any wrong label, whereas a targeted attack requires classifying it to a specific label. We randomly select one video from each category for each dataset following the setting in [49]. The target model correctly classifies all selected original videos. We normalize the pixels between 0-1. We constrain the maximum intensity perturbation to 0.03, maximum camera rotational perturbation to 0.18 radians, and maximum queries to Q = 60,000 for untargeted attack. For targeted attack we choose maximum intensity perturbation to 0.05, maximum camera rotational perturbation to 0.18 radians, and maximal queries to Q = 200,000. If a technique is not able to find adversarial perturbation within these constraints, we record it as having consumed Q queries.

**Evaluation Metric [49]:** We use the average number of queries (ANQ) required in generating adversarial examples and the attack success rate (SR) as the metrics for comparison. ANQ measures the average number of queries made in attacking across all videos, and SR gives the overall success rate in attacking within a query budget Q. Thus, a smaller ANQ and higher SR are desirable.

### 4.2. Quantitative Comparison

**Untargeted Attacks:** We report the effectiveness of our proposed method compared to SOTA in Tab. 1. We compare with V-BAD [20], and ME-Sampler [49]. To the best of our knowledge these are the only two video based adversarial attack models with the source code available. We see that our technique achieves comparable SR as the SOTA, while taking a fraction of query budget in comparison. We also report the comparative performance on top-5 performing classes of each of the attacked model in Tab. 4.

**Targeted Attack:** We report the results of the targeted attacks in Tab. 2. We also report the results of top-5 performing classes of each attacked model in Tab. 5. Similar to untargeted attacks, here also we observe similar SR performance and a large improvement in query budget. For example, on Epic-Kitchens, our method consumes only 15,283 queries, in comparison to 44,326 by V-BAD and 22,541 by ME-Sampler, an improvement of almost 3×. Even for Kinetics dataset, we outperform V-BAD and ME-Sampler by saving 16,948 and 4,886 queries, respectively, and achieve a comparable success rate.
### Dataset Method

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>Class 1 ANQ</th>
<th>Class 2 ANQ</th>
<th>Class 3 ANQ</th>
<th>Class 4 ANQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kinetics-400</td>
<td>V-Bad [20]</td>
<td>4,618</td>
<td>4,975</td>
<td>4,857</td>
<td>4,573</td>
</tr>
<tr>
<td></td>
<td>ME-Sampler [49]</td>
<td>2,246</td>
<td>2,554</td>
<td>2,794</td>
<td>2,825</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>1,851</td>
<td>1,719</td>
<td>1,548</td>
<td>1,881</td>
</tr>
<tr>
<td>Epic-Kitchens</td>
<td>V-Bad [20]</td>
<td>8,421</td>
<td>8,156</td>
<td>8,195</td>
<td>8,711</td>
</tr>
<tr>
<td></td>
<td>ME-Sampler [49]</td>
<td>7,672</td>
<td>7,914</td>
<td>7,574</td>
<td>7,057</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>6,496</td>
<td>6,944</td>
<td>6,700</td>
<td>6,994</td>
</tr>
<tr>
<td>IITMD-FPR</td>
<td>V-Bad [20]</td>
<td>5,836</td>
<td>5,706</td>
<td>5,517</td>
<td>5,225</td>
</tr>
<tr>
<td></td>
<td>ME-Sampler [49]</td>
<td>5,720</td>
<td>5,661</td>
<td>6,566</td>
<td>5,970</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>3,531</td>
<td>3,718</td>
<td>3,304</td>
<td>3,087</td>
</tr>
</tbody>
</table>

Table 4. Untargeted attacks on top-4 performing classes of Kinetics-400, Epic-Kitchens, and IITMD-FPR. The attacked models are I3D, Rolling-Unrolling LSTM, and EgoGaitNet respectively.

### 4.3. Qualitative Analysis

The comparative qualitative analysis of the proposed framework with ME-Sampler [49] is shown in Fig. 3. We have shown the analysis for three video segments, choosing the middle frame from each video. For detailed analysis, please refer to the supplementary material. The first column shows the original frame, the second column shows the attacked frame using ME-Sampler [49], and the third column shows the attacked frame using our proposed technique. We have also mentioned the number of queries required for the successful attack for each frame. It is evident from the figure that our proposed framework, similar to ME-Sampler, produces imperceptible perturbation to the video frame. However, our proposed framework requires substantially smaller number of queries for successful attack.

### 4.4. Ablation Study

**Intensity based Vs Geometric Perturbation:** We have conducted ablation study to understand importance of various components of the proposed architecture. Our method introduces a mix of intensity based and geometric noise. In Tab. 3 we show the results, when only one of the noise type is used for perturbation. We see that only intensity based attack causes much more query to generate the perturbation, whereas rotation based attacks require much lesser queries but also a much lower success rate. Combining the both as in the proposed framework, achieves high success rate at a lower query budget.

**Manual Vs Learnt γ:** The composition factor to combine the intensity based and geometric perturbation is automatically learnt by our model using DifferenceNet. In Tab. 3 we also show the results after setting composition weight manually. One can see that similar to geometric perturbation, the configuration achieves low success rate, at a low query budget. Automated learning of composition weight gives best results, thus validating the need of DifferenceNet.

**Distribution of γ:** One of the key components of our model
Figure 3. Comparative Qualitative Analysis of the proposed system. The detailed analysis is in the supplementary material. The first column shows the original frame, the second column shows the attacked frame using ME-Sampler [49], and the third column shows the attacked frame using our proposed technique.

Figure 4. Histogram of the learned composition parameter on Epic-Kitchens dataset. The minimum and maximum values of $\gamma$ are 0.07 and 0.96 respectively. Given such a variability of $\gamma$, learnable gradient composition is required for successful attacks.

is the learnable gradient composition framework, where the composition parameter $\gamma$ is learned using DifferenceNet. Fig. 4 shows the histogram of the learned composition parameters on Epic-Kitchens dataset. We see that the distribution of $\gamma$ parameter is similar to the Gaussian distribution. We report the mean of the Gaussian as 0.55 and standard deviation as 0.11. The minimum and maximum values of $\gamma$ are 0.07 and 0.96 respectively. Given such a variability of $\gamma$ (for successful attacks), it is no surprise that manual gradient composition fails completely as also shown in our ablation study (see Tab. 3).

Relationship between $\gamma$ and Video Content: To understand the relationship between $\gamma$ value and the corresponding video, we chose few videos having low, middle and high $\gamma$ values. A few representation frames of these videos are shown in Fig. 5. We observe that the videos having small spatio-temporal variation, results in higher $\gamma$. Conversely, large variations results in smaller $\gamma$. This is expected, since in the videos where spatio-temporal variation is small, intensity-based noise has more affect rather than geometric noise. Hence, the proposed framework favors intensity noise by learning a high $\gamma$ value.

5. Conclusion

Black-Box adversarial attacks on DNNs for videos analysis have utilized intensity-based noise for adversarial perturbation. However, such frameworks, require a large number of queries for estimating the perturbation. To overcome that, we propose a parametric noise based adversarial attack. It utilizes both intensity-based noise and camera rotational noise for generating the adversarial video. Gradient estimation has been done over both noises and are merged using a learnable novel gradient composition framework. We have shown the efficacy of the proposed framework on both first-person and third-person video analysis tasks.

6. Acknowledgement

This work was supported in part by the DST, Government of India, under project id T-138.
References


[27] Hyun Soo Park, Eakta Jain, and Yaser Sheikh. Predicting primary gaze behavior using social saliency fields. In Proc-


