In recent years, a significant amount of research efforts concentrated on adversarial attacks on images, while adversarial video attacks have seldom been explored. We propose an adversarial attack strategy on videos, called DeepSAVA. Our model includes both additive perturbation and spatial transformation by a unified optimisation framework, where the structural similarity index measure is adopted to measure the adversarial distance. We design an effective and novel optimisation scheme which alternatively utilizes Bayesian optimisation to identify the most influential frame in a video and Stochastic gradient descent (SGD) based optimisation to produce both additive and spatial-transformed perturbations. Doing so enables DeepSAVA to perform a very sparse attack on videos for maintaining human imperceptibility while still achieving state-of-the-art performance in terms of both attack success rate and adversarial transferability. Our intensive experiments on various types of deep neural networks and video datasets confirm the superiority of DeepSAVA.

1 Introduction

In the past decade, Deep Neural Networks (DNNs) have demonstrated their outstanding performance in various domains, such as image classification [36, 53], text analysis [26], speech recognition [8], and object detection [8, 61]. Despite their huge success in these tasks, recently some researchers have shown that DNNs are surprisingly vulnerable to adversarial attacks [2, 32, 41, 59], e.g., adding a small human-imperceptible perturbation to an input image can fool DNNs, enabling the model to make an arbitrarily wrong prediction with high confidence [33, 41]. This is raising serious concerns about the readiness of deep learning models, especially on safety-critical applications such as face authentication [27], surveillance systems [31], and medical applications [36]. Hence, it is of vital importance to investigate the performance of DNNs on adversarial examples and evaluate their robustness in an adversary environment [34, 38, 56].

As such, significant research efforts have emerged to assess the robustness of DNNs under adversarial attacks, notable works include Fast Gradient Sign Methods (FGSM) [59], C&W attack [2], DeepFool [28], and JMSA [50]. These attack strategies primarily concentrate on image-related tasks, yet the adversarial robustness of deep learning models on videos has not been comprehensively explored. Recently, a number of works [15, 29, 47, 57] are aware of the values of the adversarial attacks on videos. Theoretically, attacking videos is more challenging compared with images because videos contain temporal information. So video attack not only requires to achieve minimal adversarial distance but also needs to perturb as few frames as possible. As such, identifying the most effective frame(s) and generating...
Achieving a high attacking success rate without compromising the human imperceptibility: Different to static images, one important criterion for maintaining the fidelity of the produced adversarial examples with real-world scenarios: Assessment, Structural Similarity Index Measure (SSIM) is an alternative signal fidelity measure that is superior to $l_p$-norm on some applications where human perceptual criterion matters \[63\]. As Figure 1 shows, SSIM is less sensitive to both noise and spatial transformations such as rotation and scaling. Thus, attacks constrained by $l_p$-norm cannot capture some spatial transformations that naturally happen in a real-world scenario such as the shaking, vibration, or rotation of a camera.

**Achieving a high attacking success rate without compromising the human imperceptibility:** Different to static images, videos contain sequential data structure and change dynamically with the temporal dimension. Hence, the existing attack strategies designed for images are not directly applicable to videos. Perturbing all frames of a video, although could achieve a high fooling rate, is time-consuming and also potentially compromises human imperceptibility. Thus, perturbing as few frames as possible while maintaining a high attacking success rate is highly desired on adversarial video attacks, which can be tackled by sparse attacks.

**Enabling the adversarial video attack to be effective across diverse types of DNNs:** In a real-world scenario, we may not be able to access the parameters, structures, or even datasets of a pre-trained DNN. Thus, similar to adversarial attacks on images, a strong adversarial transferability that can work across diverse unseen models is desirable. However, unlike DNNs for images that are without temporal structure, video models are more complicated and include diverse neural units for recurrent operations. Hence, achieving a satisfying adversarial transferability across unseen models is also a non-trivial challenge.

As a result, we provide a pioneering exploration to deal with the above challenges. We propose an adversarial video attack for DNNs, called DeepSAVA, which can i) capture a wide range of adversarial instances including both noise contamination and various spatial transformations; ii) achieve sparse attack, i.e., only perturbing very few frames of a video while still achieving a state-of-the-art attack success rate; and iii) obtain better adversarial transferability across various recurrent models compared with baseline methods.

In summary, there are three key contributions in DeepSAVA:

**Structural Similarity Index Measure (SSIM):** Instead of $l_p$-norm, we adopt the SSIM metric in the loss function to constrain the distance between adversarial and clean videos. As demonstrated by the community of Image Quality Assessment, SSIM is an alternative signal fidelity measure that is superior to $l_p$-norm on some applications where human perceptual criterion matters \[63\]. As Figure 1 shows, SSIM is less sensitive to both noise and spatial transformations such as rotation and scaling, which is more resemble human perception.

**Combination of additive perturbation and spatial transformation:** We are the first work to combine additive and spatial-transformed perturbation for video attacks. According to the image attack used spatial transformation perturbation \[55, 60\], changing the positions of pixels could improve perceptual realism and make it locally smooth. In DeepSAVA, we introduce a new term in the loss function for optimising both additive and spatial transformation perturbation. With a proper SSIM-based constraint, we could produce strong perturbations combined with additive and
Figure 2: Overview of DeepSAVA: the key frame is alternatively identified by BO; the additive and spatial-transformed perturbations are applied to the selected frame to generated adversarial examples.

spatial transformation. Such combined perturbation enables DeepSAVA to achieve successful attacks by just perturbing one frame.

**Novel alternating optimisation strategy:** we are also the first work that uses Bayesian optimisation (BO) to choose the most critical frames of the video in attacks. To achieve a video attack that can perturb as few frames as possible, we design an alternating optimisation strategy that can effectively identify the key frames via BO and then initiate additive and spatial-transformed perturbations on the selected key frames by stochastic gradient descent (SGD) based optimiser. Such an alternating process happens in each iteration of the optimisation until key frames are found. Combining the above two ingredients, the proposed novel optimisation strategy could achieve a better fooling rate than baselines.

The flow chart of our method is illustrated in Figure 2. We anonymously release the code of DeepSAVA ¹ and generated adversarial videos across multiple models ².

2 Related Work

**Video Action Recognition Models:** The video classification task primarily focused on action recognition ²⁰. The works on video classification using DNNs are developed in two ways: using 2D or 3D-based convolution neural networks (CNN). Since the CNNs have obtained state-of-the-art performance on image classification, Karpathy et al. ¹⁸ first proposed to use 2D CNN to classify each frame of the video. Szegedy et al. then developed the Inception-v3 ³⁹,⁴⁰, which is commonly used as a baseline classification model. As 2D-CNNs use incomplete video information, some works added layers containing temporal information such as LSTM to integer CNN features extracted over time which is referred to as CNN+LSTM model ⁶, ³⁰. As for the 3D CNNs ⁴³, it can learn temporal features from videos by inputting all frames in three dimensions directly. Vadis et al. ³ proposed a two-stream inflated 3D CNN (I3D) to build the 2D kernel first and then merge the pooling layer and kernel into a 3D network. By pre-training the I3D on Kinetics Dataset, it could reach state-of-the-art performance on recognising UCF101 and HMDB51 action video datasets.

**Adversarial attack on images:** The adversarial attack on images has been explored extensively recently. Szegedy et al. ⁴¹ first proposed to add visually imperceptible noise on the images to mislead pre-trained CNN to give the wrong prediction label. Goodfellow et al. ¹⁰ proposed to use a gradient-based approach, the fast gradient sign method (FGSM), to generate adversarial examples. DeepFool ²⁸ is then proposed to find the minimal perturbation by iteratively linearizing the loss function. Other gradient-based optimisation algorithms to generate perturbation were also proposed ², ²⁵, ⁴², ⁵⁴. These works mentioned above only apply additive perturbation on pixels. Some works ¹⁶, ²², ²³, ⁵¹, ⁵⁵ use a functional perturbation which is non-additive-only perturbation like spatial transformation. These perturbation slightly perturb the location of pixels. Some works such as ¹¹, ¹⁶, ⁶² also utilize other types of metrics such as SSIM to quantify the human perception, but none of them explored the SSIM-guided spatial transformation. For more details on adversarial attacks, please refer to our recent survey ¹³ and tutorial ¹⁴⁵⁴.

**Adversarial attack on videos:** Wei et al. ⁴⁷ claimed that they are the first to attack videos. Instead of attacking each frame of a video, they apply additive perturbations on randomly selected frames and use $l_{2,1}$ norm to guide the

¹https://github.com/TrustAI/DeepSAVA
²https://www.youtube.com/channel/UCBDswZC2QhBlsTOMUFLNchCg
Sparse Adversarial Video Attacks with Spatial Transformations

A PREPRINT

3 Methodology

Problem Definition: The video classifier is defined as \( J(\cdot; \theta) \) with pretrained weights \( \theta \). The input clean video is defined as \( X = (x_1, x_2, \ldots, x_T) \in \mathbb{R}^{T \times W \times H \times C} \), where \( T \) is the length of the video (number of frames), and \( W, H, C \) represents the width, height, and the number of channels of each frame; its adversarial video generated is represented as \( \hat{X} \). In order to obtain the adversarial example, the original video is perturbed by a spatial transformer \( S \), and additive noise \( D \). Given that the ground truth label of input video \( X \) is \( y \), the objective function is:

\[
\arg\min \lambda \ell_{\text{similar}}(\hat{X}, X) - \ell_{\text{adv}} \left(1_y, J(\hat{X}; \theta)\right),
\]

where \( 1_y \) is the one-hot encoding of \( y \); \( \ell_{\text{similar}} \) is the similarity loss function to measure the distance between generated adversarial and original video; \( \ell_{\text{adv}} \) is the loss function to measure the difference between ground truth and prediction label. The parameter \( \lambda \) is set to balance these two loss terms. Additionally, the cross-entropy is used to calculate the \( \ell_{\text{adv}} \), which is proved to be effective in \( [47] \).

3.1 Sparse Spatial Transform Adversarial Attack

Structural Similarity Index Measure (SSIM): The SSIM was first proposed in \( [44] \), and is detailed in \( [45] \). It measures the local similarities between the local pixels on three aspects: structures, contrasts, and brightness. As we mentioned before, the SSIM is less sensitive to the combined perturbation and more similar to human perception than \( l_p \)-norms \( [63] \). As the SSIM is differentiable with respect to the input variable (the definition and derivation process of SSIM are shown in Appendix A), we apply SSIM to calculate the similarity loss to constrain the perturbation during the optimisation process. The overall SSIM score for the video is calculated by summing up the SSIM loss over all frames.

| Simularity metric | Flickering | RL | Heuristic | Append | BlackBox attack | GAN-based attack | Sparse Attack | Deep SAVA |
|-------------------|------------|----|-----------|--------|-----------------|-----------------|--------------|----------|
| Spatial-transformed perturbation | \( l_p \) | \( l_p \) | \( l_1 \) | \( l_\infty \) | \( l_p \) | \( l_2 \cdot 1 \) | SSIM |
| Additive Perturbation | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) |
| Identify Key Frames | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) | \( \checkmark \) |
| Transferability Study | \( \times \) | \( \times \) | \( \times \) | \( \times \) | \( \times \) | \( \times \) | \( \times \) |
| Sparse Attack | \( \times \) | \( \times \) | \( \times \) | \( \times \) | \( \times \) | \( \times \) | \( \times \) |

Table 1: Comparison with related work in different aspects.

Figure 3: (a) The process to perturb one frame of a clean video \( X \), where the first frame is masked to be perturbed by spatial flow vector \( U \) and noise \( N \). (b) The systematic optimisation process by using Bayesian Optimisation and Adam Optimiser.

gradient-based optimisation and evaluated the performance on the CNN+LSTM model. Li et al. \( [24] \) used a GAN network to generate offline universal perturbations for each frame. Chen et al. \( [4] \) proposed to append a noise frame to the end of videos, which is obtained based on all videos. Naeh et al. \( [29] \) applied flickering temporal perturbations on each frame to generate universal perturbations for the I3D model. Jiang et al. \( [15] \) were the first to propose a black-box approach to attack videos. Wei et al. \( [48] \) proposed to use a heuristic method and Yan et al. \( [57] \) used a reinforcement learning algorithm to select the key frames to perform black-box attack. However, these works only applied additive perturbation based on \( l_p \)-norm distance. In Table 1, we compare our method with existing related works on video attacks in six aspects. Our work applies the SSIM-guided non-additive perturbation on selected frames to generate adversarial videos efficiently. We also propose a novel alternating optimisation strategy to select the key frames.
Algorithm 1 Bayesian Optimisation for video frame selection

**Input:** $X^T \times W \times H \times C$, $y$; $G, \lambda, F$; Number of steps to explore $K$.

**Output:** Frame selection mask indicator $M$.

1: Initialize flow network parameter $U_0$ and additive noise $N_0$.
2: for $k \leftarrow 1, K$ do
3:    Find $M^{k} = \text{argmax}_M F (M | D_{t=k-1})$
4:    Train $G(X, \lambda, M, \lambda)$ using Adam to obtain $L$
5:    Add $M, L$ to the dataset $D_{t=k-1}$
6: Return the best $M$ with lowest $L$.

**Algorithm 2 DeepSAVA adversarial generator (G)**

**Input:** $X^T \times W \times H \times C$; $M$; $y$; $\hat{\lambda}$

1: Initialize flow vector $U_0$, and additive noise $N_0$.
2: for step $\leftarrow 1, \max \text{Step}$ do
3:    $\hat{X} = N \cdot M + S(U, X, M)$
4:    $\mathcal{L} = \lambda (1 - \text{SSIM} (\hat{X}, X)) - \text{adv} \left( 1, J(\hat{X}; \theta) \right)$
5: Apply Adam to optimise $U$ and $N$ to minimize $\mathcal{L}$

**Sparse Attack:** The mask indicator $M = (m_1, m_2, ..., m_T) \in \mathbb{R}^T$ is used to choose the key frames in the video, where $m_t \in \{0, 1\}$ indicates whether the $t$-th frame is masked to be perturbed. The masked video $X_m$ is formed through the map function $M(X, X)$, and then fed into the spatial transformer $S$.

**Spatial Transformed Perturbation:** Given the $t$-th frame $x^t \in \mathbb{R}^W \times H \times C$ of input video $X$, $x^n_t$ denotes the $n$-th pixel of $x^t$ and its location in the frame can be represented by a 2D coordinate $(h^n_t, v^n_t)$. The spatial transformer $S$ is a differentiable model composed by flow displacement vectors $U = ((\Delta H^1, \Delta V^1), (\Delta H^2, \Delta V^2), ..., (\Delta H^T, \Delta V^T)) \in \mathbb{R}^{T \times 2 \times H \times W}$ (where $H^t = (h^t_0, h^t_1, ..., h^t_m), V^t = (v^t_0, v^t_1, ..., v^t_m) \in \mathbb{R}^{m \times W}$), which is used to synthesize the 2D coordinate of adversarial videos. Suppose $\hat{x}^t_n$ with location $(\hat{h}_n^t, \hat{v}_n^t)$ is the adversarial example transformed from $x^n_t$, given its corresponding spatial displacement vector $(\Delta h^n_t, \Delta v^n_t)$, the new location of original pixel $x^n_t$ can be represented as $(h^n_t, v^n_t) = (\hat{h}_n^t + \Delta h^n_t, \hat{v}_n^t + \Delta v^n_t)$. Considering the sparse attack mask indicator $M$, we can represent the transformed adversarial video as $\hat{X}_S = S(U, X, M)$.

**Additive Perturbation:** The additive perturbation is the most common way to generate adversarial examples [2][10]. We define the additive model as $D$ with parameter $N \in \mathbb{R}^{T \times W \times H \times C}$. We combine spatial transformation and additive perturbation to generate adversarial videos as (illustrated in Figure 3(a)): $\hat{X} = D(N, \hat{X}_S, M) = N \cdot M + \hat{X}_S$.

### 3.2 Novel Alternating Optimisation Strategy

In this paper, we utilize the Bayesian Optimisation (BO) to select the most critical frames. As the frame selection is a discrete variable optimisation problem, we also tried other discrete optimisation techniques such as simulated annealing (SA) [9] and genetic algorithms (GA) [49], but both spent about 200s to find the final result which is much longer than about 16s taken by BO.

The generated adversarial video is formed as $\hat{X} = N \cdot M + S(U, X, M)$. In this paper, the similarity loss $\ell_{\text{sim}}$ and adversarial loss $\ell_{\text{adv}}$ in problem (1) can be expressed as $\ell_{\text{sim}}(\hat{X}, X) = 1 - \text{SSIM} (\hat{X}, X) = L_a (N, U, X, M)$ and $\ell_{\text{adv}} \left( 1, J(\hat{X}; \theta) \right)$

As $M$ is a discrete binary vector, which makes problem (4) non-differentiable, we solve it systematically by a novel alternating optimisation strategy: at each iteration, we optimise $M$ by BO first; and then by fixing $M$, the problem becomes differentiable, which can be solved by Stochastic Gradient Descent (SGD) based optimisation. We choose the Adam optimiser [19] because of its robust and fast convergence performance. This process repeats for a fixed number of iterations, continuously improving the solution via both techniques alternatively.

BO proposes sampling points from the search space through acquisition functions to obtain the reward of previous points. Expected Improvement (EI) is applied as acquisition function $F$, which is widely employed in BO: $\mathbb{E}[\max \{L(M) - L(M^t), 0\}]$, where $L(M)$ is the loss from Adam by fixing $M$. $L(M^t)$ is the best value obtained so far and $M^t$ is its location.

During the BO process, we will find the best mask indicator through several iterations. In the $k$-th iteration of BO, we first sample a candidate $M^k$ according to the acquisition function $F$.

Then, the corresponding loss $L_{\text{adv}}$ will be computed by $\hat{X}$, which will then affect the next sampled point $M^{k+1}$ for the next iteration. When the BO reaches the maximum exploration number, the best $M$ with minimum loss will be fed into the Adam optimiser to generate the final adversarial video. The process is illustrated in Figure 3(b).

Algorithm 1 and 2 detail the BO selection and adversarial videos generation algorithms respectively. In Algorithm 1, the next sampling point $M$ is obtained by maximizing the acquisition function $F$ based on previous sampling data set $D_{t=k-1}$ (Line 3). After adversarial Generator (G) is optimized, the loss $L$ for $M$ is calculated. Then the $M$ with its corresponding $L$ are appended to the sampling pool $D$ to propose the next sampling point. In Algorithm 2, according to the optimised mask indicator $M$, the final flow vector $U$ and additive noise $N$ are optimised via Adam.
Table 2: Training accuracy of the classifiers to be attacked.

| Models     | Attack Method                  | UCF101   | HMDB51   |
|------------|--------------------------------|----------|----------|
| CNN+LSTM   | Sparse                         | 52.77% ±2.44% | 16.45% ±1.8% |
|            | Sparse Flickering               | 48.38% ±1.67% | 23.55% ±2.93% |
|            | DeepSAVA(without BO)            | 56.22% ±1.65% | 8.32% ±0.34% |
|            | DeepSAVA(BO)                    | 57.22% ±1.36% | 8.77% ±0.00% |
| I3D        | Sparse                         | 10.12% ±1.19% | 44% ±1.26%  |
|            | Sparse Flickering               | 1.15% ±0.68%  | 0% ±0.00%   |
|            | DeepSAVA(without BO)            | 47.57% ±2.64% | 12.15% ±3.86% |
|            | DeepSAVA(BO)                    | 99.89% ±0.11% | 6.47% ±0.08% |
| Inception-v3| Sparse                        | 42.25% ±4.30% | 33.70% ±1.56% |
|            | Sparse Flickering               | 21.73% ±1.39% | 35.4% ±0.98% |
|            | DeepSAVA(without BO)            | 68.86% ±1.83% | 13.29% ±3.19% |
|            | DeepSAVA(BO)                    | 70.39% ±2.78% | 10.52% ±0.82% |

Table 3: Comparison with baselines, DeepSAVA without BO and with BO on different models by only perturbing one frame. ‘-’ means that there is no successful attack. Gray cell shows the best results.

4 Experiments

Dataset: As action recognition video datasets are widely used in adversarial video attack studies, we choose two popular benchmark action recognition datasets to evaluate the performance of our method: UCF101 [37] and HMDB51 [21]. Both datasets are realistic action recognition datasets. The UCF101 contains 13,320 videos with 101 categories such as playing instruments, body movements, human-object interaction. Similarly, the HMDB51 has around 7,000 videos within 51 categories related to body-motion and facial actions.

Action Recognition Models: We evaluate DeepSAVA on three classifiers: Inception-v3, a 2D-CNN based model [40], which is widely used in the image recognition task with high accuracy; I3D, a 3D-CNN based model, pre-trained on Kinetics [5]; CNN with LSTM, which is pre-trained on ImageNet to extract features from videos and then input these features to train the LSTM network. The training accuracy of all classifiers is shown in Table 2.

Baseline methods: Two baseline methods are used for comparison, the Sparse [47] and Sparse Flickering. For the works shown in Table 1 only [47] is the white-box sparse attack; [48] [57] are black-box sparse attack methods. As our work is a white-box sparse attack, we choose the most related one, Sparse [47], as the main baseline. We perform perturbation directly on the frame, while [4] appended additional frame in the end of video, which is more visible to human. So we did not include it as a baseline due to its compromise on the similarity of human perception. In [24], GANs are used to attack real-time video, which is not comparable to our method. We modified Flickering [29], which perturbs all frames, into a sparse one as the Sparse Flickering baseline, but we still show the performance of perturbing all frames.

Experiments Setting: The length of all input videos is crafted to be the same (40 frames). We randomly select 200 videos from different categories in the test dataset. For those experiments without saying the specific constraint, the maximum allowed search iteration (100 iterations) is applied; all experiments use Adam optimiser with 0.01 learning rate. The parameter λ is set to 1.5 for the CNN+LSTM model, and 1.0 for the I3D and Inception-v3 models. For λ, values that can balance the fooling rate and perturbation strength are used.

Metrics: Fooling Rate (FR): the percentage of generated adversarial videos that are misclassified successfully. Average Number of Iterations (ANI): the average number of iterations taken to generate adversarial examples successfully based on the same original videos, which is used to measure the efficiency when we set a constraint on the maximum allowed iteration.

4.1 Comparison With Baseline Methods

In this section, we will show the comparison results between DeepSAVA and baselines. Since running BO will add extra time to choose the frame, to make the comparison more complete, we also take the DeepSAVA without BO selection into account.

Limited iterations: Since each method uses a different metric, in order to control the maximum allowed perturbation we limit the number of search iterations for all methods. Each iteration only allows a small amount of perturbation (controlled by the learning rate of Adam optimiser), following the same setup used by the baselines. The results in Table 3 show that the ANIs are much below...
Sparse Adversarial Video Attacks with Spatial Transformations

Sparse Adversarial Video Attacks with Spatial Transformations

A Preprint

Figure 4: Fooling Rate of attacking different number of frames across different models.

| Constraint | $l_{2,1}$ budget = 0.08 | $l_{2,1}$ budget = 0.09 |
|------------|--------------------------|--------------------------|
| Method     | Sparse       | DeepSAVAno BO | DeepSAVANo BO | Sparse       | DeepSAVAno BO | DeepSAVA     |
| FR         | 40.51%       | 48.1%         | 88.61%       | 41.77%       | 54.43%       | 93.67%       |
| Time (s)   | 8018.9       | 2629          | 1335.8       | 14001        | 3729         | 1573.8       |

| Constraint | SSIM budget = 0.98 | SSIM budget = 0.96 |
|------------|-------------------|-------------------|
| Method     | Sparse       | DeepSAVAno BO | DeepSAVANo BO | Sparse       | DeepSAVAno BO | DeepSAVA     |
| FR         | 8.06%        | 16.56%     | 35.44%       | 10.1%        | 51.9%        | 96.20%       |
| Time (s)   | 5842.32      | 1285.1     | 1424.4       | 13789.23     | 5633.28      | 1545.5       |

Table 5: Attack I3D model on UCF101 dataset under $l_{2,1}$ and SSIM constraint separately.

We run the experiments 10 times, and show the average results with a 99% confidence interval. For the methods without frame selection, the first frame is perturbed. As shown in Table 3, BO selection is more efficient than the one without BO. This happens because it is able to select the most critical frame, which can improve the efficiency on most of the cases. For the CNN+LSTM model, DeepSAVA increases the FR slightly compared with the baselines; while for the I3D model, we can see that the FR grows significantly. The BO selection process is also essential for I3D. Without BO, only about half of the test videos can be attacked successfully; after applying BO, the FR increases to nearly 100%. As for the Inception-v3 model, the FR increases when applying DeepSAVA. It can be concluded that the CNN+LSTM is the most robust classification model among the three classifiers. Although the I3D has the highest classification accuracy, it is more vulnerable to attacks, even when only one frame is modified. That might happen because the I3D model relies heavily on the integral structure of the video itself and some frames may be more important.

We find that the position of key frames is related to the classifiers evaluated: for CNN+LSTM, the frames in the front are more often selected, and for other CNN networks, the position is variant. Thus, it is reasonable that the BO cannot improve the FR for CNN+LSTM model as much as the I3D, as we attacked the first frame when not selecting. We also show the results in Figure 4 for attacking a different number of frames across I3D and CNN+LSTM models. It can be seen that the more frames are attacked, the higher the fooling rate obtained.

Fixed $l_{2,1}$ norm and SSIM: For the purpose of a fair comparison, we also present the results under fixed $l_{2,1}$ and SSIM budgets for perturbing only one frame. The maximum allowed iteration is set to be 500 to limit the time. As the baseline methods are based on $l_p$-norm and our method is on SSIM, we take experiments under the same $l_p$-norm constraint and SSIM constraint, respectively. Based on the results of fixed iterations, we randomly select 200 videos from different categories to attack the I3D model on the UCF101 dataset. During the experiments, the Sparse Flickering spent days to achieve the constraint, thus we will only compare with the Sparse [47] attack. In [7], the SSIM budget for attacking image is set to 0.95, thus we choose the SSIM constraints above 0.95. In [58], it states that the difference between the images is imperceptible when the $l_{2,1}$ score is 4, given that, we also set the $l_{2,1}$-norm budget to below 0.1 (since $0.1 \cdot 40 = 4$, as we have 40 frames). As we can see in Table 13 under small fixed budgets, DeepSAVA outperforms the Sparse [47] in both cases in terms of FR and total time.
Sparse Adversarial Video Attacks with Spatial Transformations

A PREPRINT

Sparse Adversarial Video Attacks with Spatial Transformations

4.2 Average Absolute Perturbation

Average Absolute Perturbation (AAP) is introduced to measure the perturbation level for each method. As mentioned previously, the sparse Flickering adds a small perturbation per frame, but cannot obtain comparable results to ours. Thus, we choose the pure sparse attack (Sparse) as the main baseline to show the average absolute perturbation. As the baseline is guided by $l_{1,2}$ norm and ours is based on SSIM loss, we will record the average perturbation of $l_{1,2}$ and SSIM separately. To achieve a fair comparison, we set the maximum $l_{1,2}$ norm ball constraint as 0.1 and maximum SSIM constraint as 0.92. Suppose the fooling rate is $f$, and distant matrix is $D$, which can be set to (1-SSIM) or $l_{1,2}$ norm, thus the average absolute perturbation(AAP) can be represented as:

$$AAP(D) = \frac{\sum N D(V_{adv} - V_{original})}{N} * f + D_{max} * (1 - f),$$

where $V_{adv}$ denotes the generated adversarial video that could successfully mislead the classifier and $D_{max}$ is the maximum constraint; $N$ is the number of adversarial samples achieving successful attack. We run experiments on 200 random selected videos of UCF101 dataset and record the results of FR, ANI, AAP($l_{1,2}$) and AAP(SSIM) in Table 6. From the results, we can see that for the model I3D and Inception-v3, our method could achieve better performance in terms of efficiency, fooling rate and AAP(SSIM). For the CNN+LSTM model, our method engages higher fooling rate and spends less time, and the AAP($l_{1,2}$) and AAP(SSIM) are also acceptable compared with the baseline model.

4.3 Visualization of Results

The generated adversarial frames by DeepSAVA are presented in Figure 5. Because of the spatial transformation, the frame looks a little bit shaky but not obvious in human eyes. In fact, in the real world, it is normal to see that there are a few frames with instabilities during video shooting and transmitting. That’s why we apply the spatial transformation in video attacks to improve the efficiency and fooling rate. In practice, a distortion in one frame of a video is less noticeable than a static image since this specific frame only appears for 0.047 seconds in human eyes [54]. We could also see that it does not lead to a noticeable perturbation as shown by our video demos.

When transmitting the videos in the real world, the generated frames need to be compressed into videos first and then decompressed to frames. We found that for the additive-only perturbed frames, they may not remain adversarial examples after such transmission. Our experiments demonstrate that DeepSAVA can be immune to short video compression due to the fact that perturbation based on spatial transformation can be well preserved during compression while additive perturbation may disappear.
Sparse Adversarial Video Attacks with Spatial Transformations

| Approach                        | CNN+LSTM Mask1 | CNN+LSTM Mask4 | Inception-v3 Mask1 | Inception-v3 Mask4 | I3D Mask1 | I3D Mask4 |
|---------------------------------|----------------|----------------|-------------------|-------------------|-----------|-----------|
| Fixed the frame (first n-th)    | 52.77% ± 2.24% | 91.28% ± 1.95% | 42.45% ± 4.30%    | 10.12% ± 1.19%   |
| \(D\)                           |                |                |                   |                   |           |           |
| \(S\)                           | 55.27% ± 1.82% | 91.89% ± 1.45% | 63.91% ± 5.61%    | 29.99% ± 2.36%   |
| \(D + S\)                       | 56.22% ± 1.65% | 92.99% ± 1.85% | 68.86% ± 1.83%    | 47.57% ± 2.64%   |
| Using BO to choose frame        |                |                |                   |                   |           |           |
| \(D + S\)                       | 57.22% ± 1.36% | 93.51% ± 1.33% | 70.39% ± 2.78%    | 99.89% ± 0.11%   |

Table 7: Effects of combining noise (\(D\)) and spatial transformation (\(S\)) by modifying a different number of frames on UCF101; Mask \(N\) means that \(N\) frames are modified.

4.4 Ablation Study

We perform ablation experiments to study the effects of combined perturbation for a different number of attacked frames by comparing with additive noise only and spatial transform only perturbations, and the effects of BO selection. Table 7 shows the FR for three classifiers on the UCF101 dataset. Four approaches are taken to attack the model: 1) only noise (\(D\)), 2) only spatial transformation (\(S\)), 3) combination of additive perturbation and spatial transformation (\(D + S\)), and 4) combined perturbation with BO selection. To make more comprehensive evaluations on the superiority of combination, we attack a different number of frames for the CNN+LSTM model as it has the lowest FR when only perturbing one frame. All experiments showed the combination power to increase the FR; using BO selection is also useful, especially for the I3D model.

4.5 The Accuracy of Bayesian Optimisation Selection

To justify whether the Bayesian Optimisation could select the most critical frames, we take the brute force search experiments to obtain the upper bound of the performance: when the selection frame is 1, we select the key frame manually one by one of the video, and then record the maximum loss found by the search. We randomly select 100 videos from UCF101 in different categories. The fooling rates, average maximum loss and average time spent for one video on three models are shown in Table 8. From the results, we can obtain the same results as the brute force search, but spending much less time, which confirms the effectiveness of BO optimisation.

| Approach            | CNN+LSTM | Inception-v3 | I3D |
|---------------------|----------|--------------|-----|
|                     | FR       | average loss | FR  | average loss | FR  | average loss | time (s) |
| BO Selection        | 55.81%   | 0.21         | 72.22% | 3.39     | 100% | 1.35         | 16.1     |
| brute force search   | 55.81%   | 0.27         | 72.22% | 3.39     | 100% | 1.35         | 70.4     |

Table 8: Fooling Rate, average selected maximum loss and average time spent for one video of BO Selection and brute force search.

4.6 Transferability across recurrent models

The transferability across models is an important evaluation of adversarial attacks, which can be treated as a black-box problem without accessing parameters of the target model. In our work, the I3D and Inception-v3 only contain CNN, while the recurrent neural networks (RNN) like CNN+LSTM contains the time-related network. Due to the unique time-related structure of videos, we mainly focus the transferability across time-related networks. We perform the transferability experiments on the UCF101 dataset for different RNNs, Inception-v3 and I3D models. The features of original videos are extracted firstly by CNN (Inception-v3) model and then are fed into vanilla RNN [35], LSTM [12], and GRU [5] networks respectively. The training accuracy for vanilla RNN and GRU are 65.16% and 73.05% respectively.

As Figure 4 shows that the Sparse [47] performs better than the Sparse Flickering in terms of FR, we choose the Sparse [47] as the baseline method. The fooling rates (FR) of the generated videos across models are presented in Table 9. The models in rows are used

| Models | LSTM | Vanilla RNN | GRU | Inception-v3 | I3D |
|--------|------|-------------|-----|--------------|-----|
|        | Sparse | DeepSAVA | Sparse | DeepSAVA | Sparse | DeepSAVA | Sparse | DeepSAVA | Sparse | DeepSAVA |
| LSTM   | 100%   | 100%       | 34.42% | 41.38%       | 64.35% | 85.34% | 50.0%   | 52.17%   | 53.48% | 54.62%   |
| Vanilla RNN | 79.34% | 84.75% | 40.70% | 56.03%       | 71.74% | 82.40% | 60.50% | 64.02%   |
| GRU    | 79.34% | 84.75% | 40.70% | 56.03%       | 71.74% | 82.40% | 60.50% | 64.02%   |
| Inception-v3 | 22.35% | 24.36% | 22.35% | 26.72% | 22.35% | 31.03% | 100%   | 100%     | 33.51% | 37.80% |
| I3D    | 6.56%  | 10.08%     | 7.01%  | 9.48%        | 7.64%  | 8.62%  | 13.04% | 14.13%   | 100%   | 100%     |

Table 9: Fooling Rate across recurrent models on UCF101.
to generate adversarial videos and the models in columns are the target attack classifiers. Here we disturb seven frames of a video to enlarge the attacking success rate. We use the adversarial examples generated from the white-box attack for the transferability, which leads to the FR in the diagonal being 100%. These adversarial examples are then used to attack other models (like a black-box attack) as detailed in Table [9]. Comparing with the baseline, our approach has a higher FR which indicates a better performance in terms of transferability. The difference between vanilla RNN and the other models is that vanilla RNN has no memory component, so it shows a weak performance on the video classification task. As we observed, adversarial videos generated from LSTM and GRU models can fool the vanilla RNN successfully. Additionally, the FR across GRU and LSTM are around 85%, which shows good transferability between the recurrent models with memory. However, from the fooling rates shown in Table [9], we could see that the transferability from RNNs to CNNs is not as good as that from CNNs to RNNs. The fooling rates to attack RNN models are higher than those to attack the I3D and Inception-v3 models. While that happens maybe because the I3D model has the highest training accuracy, it engages the lowest fooling rate when performing the black-box attack on it. As we can also conclude that due to the lowest training accuracy the Vanilla RNN model obtains, it achieves the highest fooling rate on attacking unseen Vanilla RNN model. Overall, compared with the Sparse baseline, our method could achieve better transferability.

5 Conclusion

In this paper, we apply spatial transformed perturbation and additive noise to attack as few frames as possible to obtain the sparse adversarial videos. We run experiments on the UCF101 and HMDB51 action dataset and three models. The most influential frames to be attacked are selected by a joint optimisation strategy with Bayesian optimisation (BO) and SGD-based optimisation. Additionally, the quality of generated adversarial examples is measured by SSIM instead of $l_p$-norm. We obtain better results than state-of-the-art sparse baselines in terms of both fooling rate and transferability. Our most significant results are for the I3D model, by only attacking one frame of the video to obtain 99.5% to 100% attack success rate.

Acknowledgements This work was supported by Partnership Resource Fund (PRF) Towards the Accountable and Explainable Learning-enabled Autonomous Robotic Systems (AELARS), funded via the UK EPSRC projects on Offshore Robotics for Certification of Assets (ORCA) [EP/R026173/1]. This work was also funded by the Faculty of Science and Technology of Lancaster University. We also thank the High End Computing facility at Lancaster University for the computing resources, and Abdulrahman Kerim and Washington L. S. Ramos for the thorough reviews.

References

[1] Nicolas Berthier, Youcheng Sun, Wei Huang, Yanghao Zhang, Wenjie Ruan, and Xiaowei Huang. Tutorials on testing neural networks. arXiv preprint arXiv:2108.01734, 2021.
[2] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In 2017 ieee symposium on security and privacy (sp), pages 39–57. IEEE, 2017.
[3] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. In proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 6299–6308, 2017.
[4] Zhikai Chen, Lingxi Xie, Shanmin Pang, Yong He, and Qi Tian. Appending adversarial frames for universal video attack. arXiv e-prints, 2019.
[5] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1724–1734, Doha, Qatar, Oct. 2014. Association for Computational Linguistics.
[6] Jeff Donahue, Lisa Anne Hendricks, Marcus Rohrbach, Subhanshi Venugopalan, Sergio Guadarrama, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(4):677–691, 2017.
[7] S. A. Fezza, Y. Bakhti, W. Hamidouche, and O. Déforges. Perceptual Evaluation of Adversarial Attacks for CNN-based Image Classification. In 2019 Eleventh International Conference on Quality of Multimedia Experience (QoMEX), pages 1–6, 2019.
[8] Dominique Fohr, Odile Mella, and Irina Illina. New Paradigm in Speech Recognition: Deep Neural Networks. In IEEE International Conference on Information Systems and Economic Intelligence, Marrakech, Morocco, 2017.
[9] In Fortran, William Press, Saul Teukolsky, William Vetterling, and Brian Flannery. Numerical recipes. Cambridge, UK, Cambridge University Press, 01 1992.
[10] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. arXiv e-prints, 2014.
[11] Diego Gragnaniello, Francesco Marra, Luisa Verdoliva, and Giovanni Poggi. Perceptual quality-preserving black-box attack against deep learning image classifiers. Pattern Recognition Letters, 147:142–149, 2021.
[12] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. Neural computation, 9:1735–80, 12 1997.
[13] Xiaowei Huang, Daniel Kroening, Wenjie Ruan, et al. A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability. Computer Science Review, 37:100270, 2020.
[14] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In Advances in neural information processing systems, pages 2017–2025, 2015.
[15] Linxi Jiang, Xingjun Ma, Shaoxiang Chen, James Bailey, and Yu-Gang Jiang. Black-box adversarial attacks on video recognition models. In Proceedings of the 27th ACM International Conference on Multimedia, pages 864–872, 2019.
[16] Matt Jordan, Naren Manoj, Surbhi Goel, and Alexandros G. Dimakis. Quantifying perceptual distortion of adversarial examples. arXiv e-prints, 2019.
where $\mu (x)$, the variance of $x$, and co-variance of $x$ and $\hat{x}$ can be represented as $\mu_x$, $\sigma_x^2$ and $\sigma_{x,\hat{x}}$. They can be calculated respectively:

$$\mu_x = \frac{1}{N_p} \left( 1^T \cdot x \right)$$
$$\sigma_x^2 = \frac{1}{N_p - 1} \left( x - \mu_x \right)^T (x - \mu_x)$$

The mean of $x$, the variance of $x$, and co-variance of $x$ and $\hat{x}$ can be represented as $\mu_x$, $\sigma_x^2$ and $\sigma_{x,\hat{x}}$. They can be calculated respectively:

$$\mu_x = \frac{1}{N_p} \left( 1^T \cdot x \right)$$

$$\sigma_x^2 = \frac{1}{N_p - 1} \left( x - \mu_x \right)^T (x - \mu_x)$$

The structural similarity index (SSIM) measure in Equation (2) can be expressed as:

$$SSIM(x, \hat{x}) = \frac{(2\mu_x \mu_{\hat{x}} + C_1)(2\sigma_{x,\hat{x}} + C_2)}{\mu_x^2 + \mu_{\hat{x}}^2 + \sigma_{x,\hat{x}}^2 + C_2}$$

The local SSIM is formed by these terms:

$$S(x, \hat{x}) = s(x, \hat{x}) \cdot c(x, \hat{x}) \cdot b(x, \hat{x}) =$$

$$\frac{\sigma_{x,\hat{x}} + D_1}{\sigma_x \sigma_{\hat{x}} + D_1} \cdot \frac{2\sigma_x \sigma_{\hat{x}} + D_2}{\sigma_x^2 + \sigma_{\hat{x}}^2 + D_2} \cdot \frac{2\mu_x \mu_{\hat{x}} + D_3}{\mu_x^2 + \mu_{\hat{x}}^2 + D_3}$$

The local SSIM is formed by these terms:

$$S(x, \hat{x}) = s(x, \hat{x}) \cdot c(x, \hat{x}) \cdot b(x, \hat{x}) =$$

$$\frac{\sigma_{x,\hat{x}} + D_1}{\sigma_x \sigma_{\hat{x}} + D_1} \cdot \frac{2\sigma_x \sigma_{\hat{x}} + D_2}{\sigma_x^2 + \sigma_{\hat{x}}^2 + D_2} \cdot \frac{2\mu_x \mu_{\hat{x}} + D_3}{\mu_x^2 + \mu_{\hat{x}}^2 + D_3}$$

where $\mu_x$ and $\mu_{\hat{x}}$ denote means, $\sigma_x$ and $\sigma_{\hat{x}}$ are standard deviations of $x$ and $\hat{x}$, respectively; $\sigma_{x,\hat{x}}$ represents the cross correlation of $x$ and $\hat{x}$ after deleting means; $D_1, D_2,$ and $D_3$ are weight parameters. For SSIM metric, a value of 1 means that the two images compared are the same. As the SSIM is calculated based on pixel level, it uses a sliding window method, which moves pixel by pixel by the window across the whole image. As we use uniform pooling to combine the total SSIM for the whole videos, suppose we
Sparse Adversarial Video Attacks with Spatial Transformations

Table 10: The results of DeepSAV(A(without BO) on UCF101 dataset for different λ values.

\[ \text{SSIM}(X, \hat{X}) = \frac{1}{N} \sum_{i=1}^{N} \text{SSIM}(x_i, \hat{x}_i) \]  

where \( x_i \) and \( \hat{x}_i \) are the \( i \)-th pixel of each frame in the video. To apply the gradient decent optimisation method described in Section 3, we have to compute the gradient of SSIM with respect to the adversarial video example \( \hat{X} \). As equation (9) shows, to compute \( \nabla_{\hat{X}} \text{SSIM}(X, \hat{X}) \), we only need to calculate the gradient \( \nabla_{\hat{x}_i} \text{SSIM}(x_i, \hat{x}_i) \). The process is represented as follows. [46] We define four parameters to deduce the derivative of local SSIM:

\[ M_1 = 2\mu_x\mu_\hat{x} + C_1, \quad M_2 = 2\sigma_x\sigma_\hat{x} + C_2 \]
\[ P_1 = \mu_x^2 + \mu_\hat{x}^2 + C_1, \quad P_2 = \sigma_x^2 + \sigma_\hat{x}^2 + C_2 \]

Therefore, the gradient can be expressed as:

\[ \nabla_{\hat{x}_i} \text{SSIM}(x_i, \hat{x}_i) = \frac{2}{N P_1 P_2} [M_1 P_1 (M_2 \mu_x - P_2 \hat{x}_i) + P_2 P_2 (M_2 - M_1) \mu_x + M_1 M_2 (P_1 - P_2) \mu_{\hat{x}_i}] \]

B Effects of λ

To decide the value of λ, we applied the DeepSAV without BO selection on 200 random selected videos of UCF101 dataset to evaluate the effect of λ. The average success perturbation (ASP) is the average of the SSIM score of perturbation for the adversarial examples that could mislead the model successfully:

\[ \text{ASP(SSID)} = \text{avg}(SSID(V_{adv} - V_{original})) \]

where \( V_{adv} \) denotes the generated adversarial video that could successfully mislead the classifier and \( V_{original} \) is the original video. The results of applying \( \lambda = 0.8, 1.0, 1.5 \) on three models are presented in Table 10. We can see that the bigger the \( \lambda \), the lower the FR while the lower the perturbation. While, for the CNN+LSTM model, the fooling rate remains the same across all tested \( \lambda \) values, but the perturbation level is the lowest at \( \lambda = 1.5 \). Thus, we choose \( \lambda = 1.5 \) for the CNN+LSTM model and \( \lambda = 1.0 \) for I3D and Inception-v3 model to trade off the performance in terms of the fooling rate and average success perturbation.

C Comparison Experiments with \( l_p \)-norm and SSIM Constraints

C.1 I3D model

| Constraint | \( l_{2,1} \) budget = 0.1 | SSIM budget = 0.94 |
|------------|-----------------------------|---------------------|
| Methods    | Sparse | DeepSAV(no BO) | DeepSAV | Sparse | DeepSAV(no BO) | DeepSAV |
| FR         | 60%    | 58.22% | 91.25%    | 12.9%  | 66.2% | 97.46% |
| Time (s)   | 24092.8 | 4109.78 | 1483.96   | 30803.2 | 19774.4 | 20866.4 |

Table 11: Comparison with Sparse baseline method

C.2 CNN+LSTM model

| Constraint | \( l_{2,1} \) budget = 0.08 | \( l_{2,1} \) budget = 0.1 |
|------------|-----------------------------|-----------------------------|
| Methods    | Sparse | DeepSAV(no BO) | DeepSAV | Sparse | DeepSAV(no BO) | DeepSAV |
| FR         | 55.71% | 48.57% | 50% | 58.57% | 58.57% | 57.14% |
| Time (s)   | 22800.5 | 13777.6 | 15010 | 23336.8 | 19774.4 | 20866.4 |

Table 12: Attack CNN+LSTM model on UCF101
For the CNN+LSTM model, we can see that although the Sparse baseline could obtain higher fooling rate, but it will spent much more time to generate the adversarial examples. Our method using combined perturbation will spent less time and obtain comparable fooling rate. Because here we select the first frame to attack, the BO cannot improve the performance as I3D model.