Sparse Adversarial Video Attacks with Spatial Transformations

Mu, Ronghui†[1]
ronghui.mu@lancaster.ac.uk
Ruan, Wenjie*[2]
w.ruan@exeter.ac.uk
Soriano Marcolino,Leandro[1]
l.marcolino@lancaster.ac.uk
Ni,Qiang[1]
q.ni@lancaster.ac.uk

[1] Computing and Communications,
Lancaster University
Lancaster, UK

[2] Computer Science,
University of Exeter,
Exeter, UK

Abstract

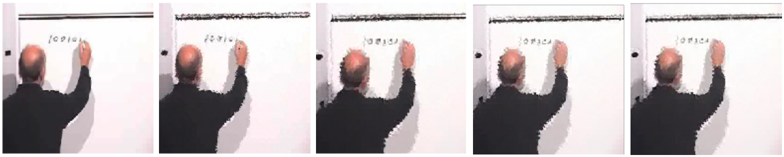
In recent years, a significant amount of research efforts concentrated on adversarial attacks on images, while adversarial video attacks have seldom been explored. We propose an adversarial attack strategy on videos, called DeepSAVA. Our model includes both additive perturbation and spatial transformation 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 [31], text analysis [24], speech recognition [7], and object detection [7]. Despite their huge success in these tasks, recently some researchers have shown that DNNs are surprisingly vulnerable to adversarial attacks [1, 33, 49], 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]. This is raising serious concerns about the readiness of deep learning models, especially on safety-critical applications such as face authentication [25], surveillance systems [29], and medical applications [31]. Hence, it is of vital importance to investigate the performance of DNNs on adversarial examples and evaluate their robustness in an adversary environment.

† This work was done while Ronghui Mu was visiting University of Exeter.

* Corresponding author.



(a) *SSIM=1*, $l_{1,2}$=*0* (b) *SSIM=0.92*, $l_{1,2} = 3.72$ (c) *SSIM=0.92*, $l_{1,2}$=*4.12* (d) *SSIM=0.91*, $l_{1,2}$=*4.01* (e) *SSIM=0.92*, $l_{1,2}$=*3.91*

Figure 1: SSIM and $l_{1,2}$ norm distance for: (a) original image; Perturbed images: (b) noise; (c) scaling + noise; (d) rotation + noise; (e) rotation+scaling + noise.

As such, significant research efforts have emerged to assess the robustness of DNNs under adversarial attacks, notable works include Fast Gradient Sign Methods (FGSM) [49], C&W attack [1], DeepFool [26], and JMSA [43]. 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 [13, 27, 40, 47] 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 *competitive* perturbation upon those frame(s) are of huge importance to the success rate of the attack. Another important consideration is the efficiency. As perturbing each frame of the video is time-consuming, we expect to perform the influential-frame identification and adversarial perturbation simultaneously so we can maintain human imperceptibility and achieve high attacking success rate. In practice, DNNs processing videos are widely applied in real systems such as video surveillance [29], and action recognition [15]. In particular, most of those applications directly relate to the decisions concerning property security or human health and safety. As a result, investigating adversarial samples on videos is urgently needed. However, to achieve a high-performance adversarial attack on videos, the following challenges need a careful treatment:

Maintaining the fidelity of the produced adversarial examples with real-world scenarios: One important criterion for adversarial attacks is that the perturbed example should resemble a real-world instance as close as possible. Current video attack strategies all adopt the l_p-norm metric to measure the fidelity of the perturbed examples. Although l_p-norm is effective to capture noise contamination, it is sensitive to naturally-occurred transformations such as rotation, spatial shift, and scaling [51]. Taking Figure 1 as an example, a slight rotation or scaling of pixels will lead to an obvious difference in l_p-norm distance. 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 [51]. 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 [46], 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 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 [18]. 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.* [16] first proposed to use 2D CNN to classify each frame of the video. Szegedy *et al.* then developed the Inception-v3 [34, 35], 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

[1] https://github.com/TrustAI/DeepSAVA

[2] https://www.youtube.com/channel/UCBDswZC2QhBhTOMUFNLchCg

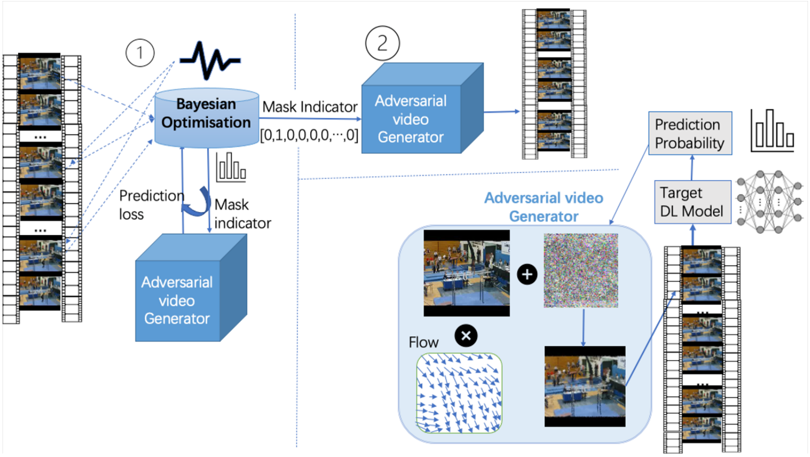


Figure 2: Overview of DeepSAVA: the key frame is alternatively identified by BO; the additive and spatial-transformed perturbations are applied to the selected frame to generated adversarial examples.

	Flickering [27]	RL [47]	Heuristic [41]	Append [3]	BlackBox attack [13]	GAN-based attack [22]	Sparse Attack [40]	Deep SAVA
Similarity metric	l_p	l_p	l_1	l_∞	l_∞	l_p	$l_{2,1}$	SSIM
Spatial-transformed perturbation	✗	✗	✗	✗	✗	✗	✗	✓
Additive Perturbation	✓	✓	✓	✓	✓	✓	✓	✓
Identify Key Frames	✗	✓	✓	✗	✗	✗	✗	✓
Transferability Study	✗	✗	✗	✓	✓	✗	✓	✓
Sparse Attack	✗	✓	✓	✗	✗	✗	✓	✓

Table 1: Comparison with related work in different aspects.

model [6, 28]. As for the 3D CNNs [37], it can learn temporal features from videos by inputting all frames in three dimensions directly. Vadis *et al.* [2] 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.* [33] first proposed to add visually imperceptible noise on the images to mislead pre-trained CNN to give the wrong prediction label. Goodfellow *et al.* [9] proposed to use a gradient-based approach, the fast gradient sign method (FGSM), to generate adversarial examples. DeepFool [26] 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 [1, 23, 36, 45]. These works mentioned above only apply additive perturbation on pixels. Some works [14, 20, 21, 44, 46] 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 [10, 14, 50] also utilize other types of metrics such as SSIM to quantify the human perception, but none of them explored the SSIM-guided spatial transformation.

Adversarial attack on videos: Wei *et al.* [40] 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 gradient-based optimisation and evaluated the performance on the CNN+LSTM model. Li *et al.* [22] used a GAN network to

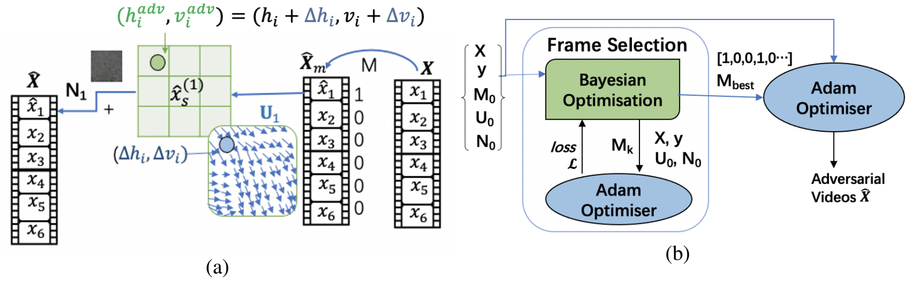


Figure 3: (a) The process to perturb one frame of a clean video **X**, where the first frame is masked to be perturbed by spatial flow vector **U** and noise **N**. (b) The systematic optimisation process by using Bayesian Optimisation and Adam Optimiser.

generate offline universal perturbations for each frame. Chen *et al.* [3] proposed to append a noise frame to the end of videos, which is obtained based on all videos. Naeh *et al.* [27] applied flickering temporal perturbations on each frame to generate universal perturbations for the I3D model. Jiang *et al.* [13] were the first to propose a black-box approach to attack videos. Wei *et al.* [41] proposed to use a heuristic method and Yan *et al.* [47] 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.

3 Methodology

Problem Definition: The video classifier is defined as $J(\cdot;\theta)$ with pretrained weights θ. The input clean video is defined as $\mathbf{X} = (x_1, x_2, ..., x_T) \in \mathbb{R}^{T\times W\times H\times C}$, 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 $\hat{\mathbf{X}}$. In order to obtain the adversarial example, the original video is perturbed by a spatial transformer $\mathcal{S}$, and additive noise $\mathcal{D}$. Given that the ground truth label of input video $\mathbf{X}$ is y, the objective function is:

$$\arg\min \lambda \ell_{similar}(\hat{\mathbf{X}}, \mathbf{X}) - \ell_{adv}\left(\mathbf{1}_y, J(\hat{\mathbf{X}};\theta)\right), \tag{1}$$

where $\mathbf{1}_y$ is the one-hot encoding of y; $\ell_{similar}$ is the similarity loss function to measure the distance between generated adversarial and original video; ℓ_{adv} is the loss function to measure the difference between ground truth and prediction label. The parameter λ is set to balance these two loss terms. Additionally, the cross-entropy is used to calculate the ℓ_{adv}, which is proved to be effective in [40].

3.1 Sparse Spatial Transform Adversarial Attack

Structural Similarity Index Measure (SSIM): The SSIM was first proposed in [38], and is detailed in [39]. 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 [51]. 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.

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 $m_t \in \{\mathbf{0}, \mathbf{1}\}$ indicates whether the t-th frame is masked to be perturbed. The masked video $\mathbf{X}_m$ is formed through the map function $\mathcal{M}(M, \mathbf{X})$, and then fed into the spatial transformer $\mathcal{S}$.

Algorithm 1 Bayesian Optimisation for video frame selection

```
Input: $\mathbf{X}^{T\times W\times H\times C}$; $y$; $G$; $\lambda$; $F$; Number of steps to explore $K$;
Output: Frame selection mask indicator $M$

Initialize flow network parameter $\mathbf{U}_0$ and additive noise $\mathbf{N}_0$;
for $k \leftarrow 1, K$ do
    Find $M$=argmax$_M F\left(M \mid D_{1:k-1}\right)$
    Train $G(\mathbf{X}, y, M, \lambda)$ using Adam to obtain $\mathcal{L}$
    Add $M, \mathcal{L}$ to the dataset $D_{1:k-1}$
Return the best $M$ with lowest $\mathcal{L}$.
```

Algorithm 2 DeepSAVA adversarial generator (G)

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

Initialize flow vector $\mathbf{U}_0$, and additive noise $\mathbf{N}_0$;
for $step \leftarrow 1, maxStep$ do
    $\hat{\mathbf{X}} = \mathbf{N}\cdot M + \mathcal{S}(\mathbf{U}, \mathbf{X}, M)$
    $\mathcal{L} = \lambda(1 - SSIM(\hat{\mathbf{X}}, \mathbf{X})) - \ell_{adv}\left(\mathbf{1}_y, J(\hat{\mathbf{X}}; \theta)\right)$
    Apply Adam to optimise $\mathbf{U}$ and $\mathbf{N}$ to minimize $\mathcal{L}$
```

Spatial Transformed Perturbation: Given the t-th frame $x^t \in \mathbb{R}^{W\times H\times C}$ of input video $\mathbf{X}$, 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 [12] $\mathcal{S}$ is a differentiable model composed by flow displacement vectors $\mathbf{U} = ((\Delta\mathbf{H}^1, \Delta\mathbf{V}^1), (\Delta\mathbf{H}^2, \Delta\mathbf{V}^2), \dots, (\Delta\mathbf{H}^T, \Delta\mathbf{V}^T)) \in \mathbb{R}^{T\times 2\times H\times W}$ (where $\mathbf{H}^t = (h_0^t, h_1^t, \dots, h_n^t)$, $\mathbf{V}^t = (v_0^t, v_1^t, \dots, v_n^t) \in \mathbb{R}^{H\times W}$), which is used to synthesize the 2D coordinate of adversarial videos. Suppose $\hat{x}_n^t$ 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 $\hat{\mathbf{X}}_S = \mathcal{S}(\mathbf{U}, \mathbf{X}, M)$.

Additive Perturbation: The additive perturbation is the most common way to generate adversarial examples [1, 9]. We define the additive model as $\mathcal{D}$ with parameter $\mathbf{N} \in \mathbb{R}^{T\times W\times H\times C}$. We combine spatial transformation and additive perturbation to generate adversarial videos as (illustrated in Figure 3 (a)): $\hat{\mathbf{X}} = \mathcal{D}(\mathbf{N}, \hat{\mathbf{X}}_S, M) = \mathbf{N}\cdot M + \hat{\mathbf{X}}_S$.

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) [8] and genetic algorithms (GA) [42], but both spent about 200s to find the final result which is much longer than about 16s taken by BO. In Appendix D, we also showed the performance of BO and the brute force search (i.e., selecting the most effective frame by perturbing each frames in a video).

The generated adversarial video is formed as $\hat{\mathbf{X}} = \mathbf{N}\cdot M + \mathcal{S}(\mathbf{U}, \mathbf{X}, M)$. In this paper, the similarity loss $\ell_{similar}$ and adversarial loss ℓ_{adv} in problem (1) can be expressed as $\ell_{similar}(\hat{\mathbf{X}}, \mathbf{X}) = 1 - SSIM(\hat{\mathbf{X}}, \mathbf{X}) = \mathcal{L}_s(\mathbf{N}, \mathbf{U}, \mathbf{X}, M)$ and $\ell_{adv}\left(\mathbf{1}_y, J(\hat{\mathbf{X}}; \theta)\right) = \mathcal{L}_a(\mathbf{N}, \mathbf{U}, \mathbf{X}, M)$. Therefore, problem (1) can be simplified as: $\arg\min_{M,\mathbf{N},\mathbf{U}} \lambda\mathcal{L}_s(\mathbf{N}, \mathbf{U}, \mathbf{X}, M) - \mathcal{L}_a(\mathbf{N}, \mathbf{U}, \mathbf{X}, M)$.

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 [17] 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: $\mathbb{E}[\max\left(\mathcal{L}(M) - \mathcal{L}\left(M^+\right), 0\right)]$, where $\mathcal{L}(M)$ is the loss from Adam by fixing M; $\mathcal{L}\left(M^+\right)$ is the best value obtained so far and M^+ 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 $\mathcal{L}_k$ will be computed by the Adam, 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.

	UCF101			HMDB51		
Models	CNN+LSTM	I3D	Inception-v3	CNN+LSTM	I3D	Inception-v3
Accuracy	74%	94.9%	71.2%	43%	80%	47%

Table 2: Training accuracy of the classifiers to be attacked.

In Algorithm 1, the next sampling point M is obtained by maximizing the acquisition function F based on previous sampling data set $D_{1:k-1}$ (Line 3). After adversarial Generator (G) is optimized, the loss $\mathcal{L}$ for M is calculated. Then the M with its corresponding $\mathcal{L}$ 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 $\mathbf{U}$ and additive noise $\mathbf{N}$ are optimised via Adam.

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 [32] and HMDB51 [19]. 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 [35], which is widely used in the image recognition task with high accuracy; *I3D*, a 3D-CNN based model, pre-trained on Kinetics [2]; *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 [40] and Sparse Flickering. For the works shown in Table 1, only [40] is the white-box sparse attack; [41][47] are black-box sparse attack methods. As our work is a white-box sparse attack, we choose the most related one, Sparse [40], as the main baseline. We perform perturbation directly on the frame, while [3] 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 [22], GANs are used to attack real-time video, which is not comparable to our method. We modified Flickering [27], 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 (please see our full experimental results in Appendix B).

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.

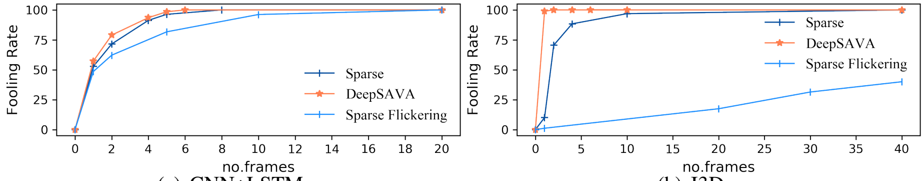
4.1 Comparison with baseline methods

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 the maximum

Models	Attack Method	UCF101		HMDB51	
		FR	ANI	FR	ANI
CNN+LSTM	Sparse	52.77% ± 2.44%	16.45	95.2% ± 1.8%	16.4
	Sparse Flickering	48.48% ± 1.67%	23.55	91.94% ± 2.93%	8.4
	DeepSAVA(without BO)	56.22% ± 1.65%	8.32	99.27% ± 0.34%	8.42
	DeepSAVA(BO)	57.22% ± 1.36%	8.77	100%	6.6
I3D	Sparse	10.12% ± 1.19%	44	5.74% ± 1.25%	25.1
	Sparse Flickering	1.15% ± 0.68%	13	0%	-
	DeepSAVA(without BO)	47.57% ± 2.64%	12.15	46.39% ± 3.86%	12.2
	DeepSAVA(BO)	99.89% ± 0.11%	6.47	99.92% ± 0.08%	5.35
Inception-v3	Sparse	42.25% ± 4.30%	33.70	45.82% ± 1.56%	22.06
	Sparse Flickering	21.73% ± 1.39%	35.4	27.55% ± 0.98%	27.25
	DeepSAVA(without BO)	68.86% ± 1.83%	13.29	68.98% ± 3.19%	11.84
	DeepSAVA(BO)	70.39% ± 2.78%	10.52	74.74% ± 0.82%	9.07

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.



(a) CNN+LSTM

(b) I3D

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

allowed iteration (100), and we also found that even when it reaches the maximum iteration, the lp-norm and *SSIM* distances are still acceptable (in Appendix G). We also show the results of average absolute perturbation distance in Appendix C. Given that, setting a constraint on the maximum search number to 100 will not lead to large distortion.

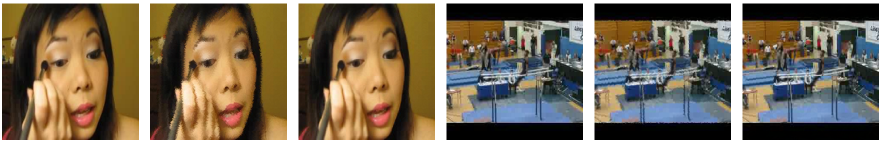
We run the experiments 10 times, and show the average results with a 99% confidence interval. For the methods without frame selection, the first frame is perturbed.As shown in Table 3, BO selection is more efficient than the one without BO. This happens because it is able to select the most critical 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 (for inception-v3 model is presented in Appendix F). It can be seen that the more frames 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

$l_{2,1}$-norm						
Constraint	$l_{2,1}$ budget = 0.08			$l_{2,1}$ budget = 0.09		
Method	Sparse	DeepSAVA(no BO)	DeepSAVA	Sparse	DeepSAVA(no BO)	DeepSAVA
FR	40.51%	48.1%	88.61%	41.77%	54.43%	93.67%
Time (s)	8018.9	2629	1535.8	14001	3729	1573.82
SSIM						
Constraint	*SSIM* budget = 0.98			*SSIM* budget = 0.96		
Method	Sparse	DeepSAVA(no BO)	DeepSAVA	Sparse	DeepSAVA(no 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 4: Attack I3D model on UCF101 dataset under $l_{2,1}$ and *SSIM* constraint separately.



(a) apply eye makeup (b) apply-lips (DeepSAVA) (c) apply-lips (Sparse) (d) ParallelBars (e) Haircut (DeepSAVA) (f) Haircut (Sparse)

Figure 5: Original, and adversarial examples generated by DeepSAVA and Sparse [40] when only one frame in the video is perturbed. The red labels are the wrong predictions.

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 [40] attack. In [6], the SSIM budget for attacking image is set to 0.95, thus we choose the SSIM constraints above 0.95. In [48], 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 $0.1 * 40 = 4$, as we have 40 frames). As we can see in Table 4, under small fixed budgets, DeepSAVA outperforms the Sparse [40] in both cases in terms of FR and total time (more experiment results are presented in Appendix E).

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 [45]. 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.

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 5 shows the FR for three classifiers on the UCF101 dataset. Four approaches are taken to attack the model: 1) only noise ($\mathcal{D}$), 2) only spatial

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

Table 5: Effects of combining noise ($\mathcal{D}$) and spatial transformation ($\mathcal{S}$) by modifying a different number of frames on UCF101; Mask N means that N frames are modified.

Models	CNN+LSTM		CNN+Vanilla RNN		CNN+GRU	
	Sparse	DeepSAVA	Sparse	DeepSAVA	Sparse	DeepSAVA
CNN+LSTM	100%	100%	34.42%	41.38%	64.35%	85.34%
CNN+Vanilla RNN	100%	100%	100%	100%	100%	100%
CNN+GRU	79.34 %	84.75%	40.70%	56.03%	100%	100%

Table 6: Fooling Rate across recurrent models on UCF101.

transformation ($\mathcal{S}$), 3) combination of additive perturbation and spatial transformation ($\mathcal{D}+\mathcal{S}$), 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.

4.2 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 present the transferability across time-related networks here (for the transferability across other CNN models, please refer our experiments in Appendix H). We perform the transferability experiments on the UCF101 dataset for different RNNs. The features of original videos are extracted firstly by CNN (Inception-v3) model and then are fed into vanilla RNN [30], LSTM [11], and GRU [4] networks respectively. The training accuracy for vanilla RNN and GRU are 65.16% and 73.05% respectively.

As Figure 4 shows that the Sparse [40] performs better than the Sparse Flickering in terms of FR, we choose the Sparse [40] as the baseline method. The fooling rates (FR) of the generated videos across models are presented in Table 6. The models in rows are used 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 6. 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.

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] 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.

[2] 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.

[3] Zhikai Chen, Lingxi Xie, Shanmin Pang, Yong He, and Qi Tian. Appending adversarial frames for universal video attack. *arXiv e-prints*, 2019.

[4] 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, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1179. URL https://www.aclweb.org/anthology/D14-1179.

[5] Jeff Donahue, Lisa Anne Hendricks, Marcus Rohrbach, Subhashini Venugopalan, Sergio Guadarrama, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39 (4):677–691, 2017. doi: 10.1109/TPAMI.2016.2599174.

[6] 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. doi: 10.1109/QoMEX.2019.8743213.

[7] 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, April 2017. URL https://hal.archives-ouvertes.fr/hal-01484447.

[8] In Fortran, William Press, Saul Teukolsky, William Vetterling, and Brian Flannery. Numerical recipes. *Cambridge, UK, Cambridge University Press*, 01 1992.

[9] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv e-prints*, 2014.

[10] 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.

[11] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9: 1735–80, 12 1997. doi: 10.1162/neco.1997.9.8.1735.

[12] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015.

[13] 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.

[14] Matt Jordan, Naren Manoj, Surbhi Goel, and Alexandros G. Dimakis. Quantifying perceptual distortion of adversarial examples. *arXiv e-prints*, 2019.

[15] M. Esat Kalfaoglu, Sinan Kalkan, and A. Aydin Alatan. Late Temporal Modeling in 3D CNN Architectures with BERT for Action Recognition. *arXiv e-prints*, 2020.

[16] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014. doi: 10.1109/CVPR.2014.223.

[17] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

[18] Yu Kong and Yun Fu. Human action recognition and prediction: A survey. *arXiv e-prints*, 2018.

[19] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. HMDB: A large video database for human motion recognition. In *2011 International Conference on Computer Vision*, pages 2556–2563. IEEE, 2011.

[20] Cassidy Laidlaw and Soheil Feizi. Functional adversarial attacks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32, pages 10408–10418. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/6e923226e43cd6fac7cfe1e13ad000ac-Paper.pdf.

[21] Cassidy Laidlaw, Sahil Singla, and Soheil Feizi. Perceptual adversarial robustness: Defense against unseen threat models. *arXiv preprint arXiv:2006.12655*, 2020.

[22] Shasha Li, Ajaya Neupane, Sujoy Paul, Chengyu Song, Srikanth V. Krishnamurthy, Amit K. Roy Chowdhury, and Ananthram Swami. Stealthy adversarial perturbations against real-time video classification systems. *In Proceedings of the 2019 Network and Distributed System Security Symposium*, 2019. doi: 10.14722/ndss.2019.23202. URL http://dx.doi.org/10.14722/ndss.2019.23202.

[23] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. *ICLR*, 2017.

[24] Varsha Mittal, Durgaprasad Gangodkar, and Bhaskar Pant. Exploring The Dimension of DNN Techniques For Text Categorization Using NLP. In *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pages 497–501. IEEE, 2020.

[25] Amir Mohammadi, Sushil Bhattacharjee, and Sébastien Marcel. Deeply vulnerable: a study of the robustness of face recognition to presentation attacks. *Iet Biometrics*, 7(1):15–26, 2017.

[26] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. DeepFool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016.

[27] Itay Naeh, Roi Pony, and Shie Mannor. Flickering adverparsearial attacks against video recognition networks. *arXiv e-prints*, 2020.

[28] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[29] Jinqiu Pan, Yue Yin, Jian Xiong, Wang Luo, Guan Gui, and Hikmet Sari. Deep learning-based unmanned surveillance systems for observing water levels. *IEEE Access*, 6:73561–73571, 2018.

[30] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

[31] Dinggang Shen, Guorong Wu, and Heung-Il Suk. Deep learning in medical image analysis. *Annual Review of Biomedical Engineering*, 19(1):221–248, 2017. URL https://doi.org/10.1146/annurev-bioeng-071516-044442. PMID: 28301734.

[32] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv e-prints*, 2012.

[33] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014. URL http://arxiv.org/abs/1312.6199.

[34] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[35] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[36] Thomas Tanay and Lewis Griffin. A boundary tilting persepective on the phenomenon of adversarial examples. *arXiv e-prints*, 2016.

[37] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3D Convolutional Networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015.

[38] Zhou Wang and Alan C Bovik. A universal image quality index. *IEEE signal processing letters*, 9 (3):81–84, 2002.

[39] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4): 600–612, 2004.

[40] Xingxing Wei, Jun Zhu, Sha Yuan, and Hang Su. Sparse adversarial perturbations for videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8973–8980, 2019.

[41] Zhipeng Wei, Jingjing Chen, Xingxing Wei, Linxi Jiang, Tat-Seng Chua, Fengfeng Zhou, and Yu-Gang Jiang. Heuristic black-box adversarial attacks on video recognition models. In *In Proceedings of the AAAI*, pages 12338–12345, 2020.

[42] Darrell Whitley. A genetic algorithm tutorial. *Statistics and Computing*, 4:65–85, 1994.

[43] Rey Wiyatno and Anqi Xu. Maximal jacobian-based saliency map attack. *arXiv preprint arXiv:1808.07945*, 2018.

[44] Eric Wong, Frank R. Schmidt, and J. Zico Kolter. Wasserstein adversarial examples via projected sinkhorn iterations. *arXiv e-prints*, 2020.

[45] Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. Generating adversarial examples with adversarial networks. *IJCAI*, 2018.

[46] Chaowei Xiao, Jun-Yan Zhu, Bo Li, Warren He, Mingyan Liu, and Dawn Song. Spatially Transformed Adversarial Examples. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=HyydRMZC-.

[47] Huanqian Yan, Xingxing Wei, and Bo Li. Sparse black-box video attack with reinforcement learning. *arXiv e-prints*, 2020.

[48] Jianchao Yang, John Wright, Thomas S Huang, and Yi Ma. Image super-resolution via sparse representation. *IEEE transactions on image processing*, 19(11):2861–2873, 2010.

[49] Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li. Adversarial examples: Attacks and defenses for deep learning. *IEEE transactions on neural networks and learning systems*, 30(9):2805–2824, 2019.

[50] Zhengyu Zhao, Zhuoran Liu, and Martha Larson. Towards large yet imperceptible adversarial image perturbations with perceptual color distance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1039–1048, 2020.

[51] Zhou Wang and A. C. Bovik. Mean Squared Error: Love it or leave it? A new look at Signal Fidelity Measures. *IEEE Signal Processing Magazine*, 26(1):98–117, 2009. doi: 10.1109/MSP.2008.930649.