

# Robustness Guarantees for Deep Neural Networks on Videos

Min Wu and Marta Kwiatkowska

Department of Computer Science, University of Oxford, UK

{min.wu, marta.kwiatkowska@cs.ox.ac.uk}

## Abstract

*The widespread adoption of deep learning models places demands on their robustness. In this paper, we consider the robustness of deep neural networks on videos, which comprise both the spatial features of individual frames extracted by a convolutional neural network and the temporal dynamics between adjacent frames captured by a recurrent neural network. To measure robustness, we study the maximum safe radius problem, which computes the minimum distance from the optical flow sequence obtained from a given input to that of an adversarial example in the neighbourhood of the input. We demonstrate that, under the assumption of Lipschitz continuity, the problem can be approximated using finite optimisation via discretising the optical flow space, and the approximation has provable guarantees. We then show that the finite optimisation problem can be solved by utilising a two-player turn-based game in a cooperative setting, where the first player selects the optical flows and the second player determines the dimensions to be manipulated in the chosen flow. We employ an anytime approach to solve the game, in the sense of approximating the value of the game by monotonically improving its upper and lower bounds. We exploit a gradient-based search algorithm to compute the upper bounds, and the admissible A\* algorithm to update the lower bounds. Finally, we evaluate our framework on the UCF101 video dataset.*

## 1. Introduction

Deep neural networks (DNNs) have been developed for a variety of tasks, including self-driving cars, malicious software classification, and abnormal network activity detection. While the accuracy of neural networks has significantly improved, matching human cognitive perception, they are susceptible to adversarial examples. An *adversarial example* is an input which, whilst initially classified correctly, is misclassified with a slight, often imperceptible, perturbation. *Robustness* of neural networks has been an active topic of investigation, and a number of approaches have been proposed (see Related Work below.) However, most

existing works focus on robustness of neural networks on image classification problems, where convolutional neural networks (CNNs) are sufficient. One assumption that CNNs rely on is that inputs are independent of each other, and they are unable to accept a *sequence* of input data when the final output is dependent on intermediate outputs. In reality, though, tasks often contain sequential data as inputs, for instance, in machine translation [23], speech/handwriting recognition [9, 10, 6], and protein homology detection [12]. To this end, *recurrent neural networks (RNNs)* come into play. For RNNs, the connections between neurons form a directed graph along a temporal sequence, which captures temporal dynamic behaviours. Unlike CNNs, RNNs can use the internal state (memory) to process sequential inputs.

In this work, we study robustness guarantees for neural networks, including CNNs and RNNs, on *videos*. Video classification is challenging because it comprises both the spatial features on each individual frames, which can be extracted by CNNs, as well as the temporal dynamics between neighbouring frames, which can be captured by RNNs. Specifically, we develop a methodology for evaluating robustness of videos based on the notion of *maximum safe radius* (or dually, minimum adversarial distortion [28]), which captures the maximum allowed magnitude of a perturbation. Our method is based on a game-based approach of [31] and provides guarantees against perturbations up to a given magnitude. We compute upper and lower bounds of the maximum safe radius, and demonstrate their convergence trend on the UCF101 video dataset.

**Related work** Robustness of neural networks has been mainly studied in the context of image classification, and, to the best of our knowledge, very few works address robustness guarantees for videos. We review only approaches most relevant to our approach. Early methods evaluate the robustness of networks by checking whether they are vulnerable to *adversarial attacks*, which attempt to craft a perturbation imposed on the input so that it is misclassified by a well-trained network. In the computer vision and security communities, various attack techniques have been developed, such as the limited-memory Broyden-Fletcher-Goldfarb-Shanno algorithm [24], the fast gradient

sign method [8] and its extended Jacobian saliency map based attack [19], as well as the optimisation-based C&W attack [2]. Nevertheless, adversarial attacks cannot provide guarantees. To this end, various *verification* approaches for neural networks emerged in the formal methods community. These include constraint solving tools Reluplex [16] and Planet [4]; mixed integer linear programming method Sherlock [3]; the AI<sup>2</sup> approach [7] involving abstract interpretation; algorithms utilising linear approximation such as FastLin/FastLip [29] and Crown [32]; and also search-based DLV [14] and game-based tool DeepGame [31].

The above approaches evaluate the robustness of feedforward networks, but we are interested in providing guarantees for both convolutional and recurrent layers with time-series inputs. Although there are some adversarial attacks against RNNs [20] and videos [15, 27], verification approaches are rare. For example, POPQORN [17] employs the linear approximation method, bounding the network output by two linear functions in terms of input; and Wang et al. [25] extract deterministic finite automata from RNNs as the oracle and use them to evaluate adversarial accuracy. Our approach instead draws on DeepGame [31], where a game-based verification framework is proposed for computing the maximum safe radius for feedforward networks through discretising the input space via Lipschitz continuity [11], but in this work we are able to handle CNN + RNN architectures and video inputs.

## 2. Preliminaries

### 2.1. Deep neural networks

Let  $\mathcal{N}$  be a neural network with a set of classes  $C$ . Given an input  $\mathbf{v}$  and a class  $c \in C$ , we use  $\mathcal{N}(\mathbf{v}, c)$  to denote the confidence of  $\mathcal{N}$  believing that  $\mathbf{v}$  is in class  $c$ . We work with the Softmax logit value of the last layer, but the methods can be adapted to the probability value after normalisation. Thus,  $\mathcal{N}(\mathbf{v}) = \arg \max_{c \in C} \mathcal{N}(\mathbf{v}, c)$  is the class into which  $\mathcal{N}$  classifies  $\mathbf{v}$ . Moreover, as  $\mathcal{N}$  in this work can have convolutional and recurrent layers, we let  $\mathcal{N}_C$  denote the convolutional part and  $\mathcal{N}_R$  the recurrent part. Specifically, since the inputs we consider are videos, we let the input domain  $D$  be  $\mathbb{R}^{l \times w \times h \times ch}$ , where  $l$  is the length of  $\mathbf{v}$ , i.e., the number of frames, and  $w, h, ch$  are the width, height, and channels of each frame, respectively.

### 2.2. Optical flow

In order to capture the dynamic characteristics of the moving objects in a video, we utilise *optical flow* [1, 26], which is a pattern of the apparent motion of the image objects between two consecutive frames caused by the movement of the objects or the camera. Several methods exist in the computer vision community to compute flows, e.g., the Lucas-Kanade [18] and Gunnar Farneback [5] algorithms.

**Definition 1 (Optical Flow).** Consider a pixel  $\mathcal{P}(x, y, t)$  in a frame, where  $x, y$  denote the horizontal and vertical positions respectively, and  $t$  denotes time dimension. If after  $dt$  time, the pixel moves by distance  $(dx, dy)$  in the next frame, then  $\mathcal{P}(x, y, t) = \mathcal{P}(x + dx, y + dy, t + dt)$  holds. After taking Taylor series approximation, removing common terms, and dividing by  $dt$ , the Optical Flow Equation is  $f_x u + f_y v + f_t = 0$ , such that  $f_x = \frac{\partial f}{\partial x}$ ,  $f_y = \frac{\partial f}{\partial y}$ ,  $u = \frac{\partial x}{\partial t}$ ,  $v = \frac{\partial y}{\partial t}$ , where  $f_x, f_y$  are the image gradients,  $f_t$  is the gradient along time, and the motion  $(u, v)$  is unknown.

### 2.3. Distance metrics and Lipschitz continuity

In robustness evaluation,  $L^p$  distance metrics are often used to measure the discrepancy between inputs, denoted as  $\|\alpha - \alpha'\|_p$ , where  $p \in \{1, 2, \infty\}$  indicates Manhattan ( $L^1$ ), Euclidean ( $L^2$ ), and Chebyshev ( $L^\infty$ ) distances. Because our inputs are videos, i.e., sequences of frames, we will need a suitable metric. In this work, we work directly with  $L^p$  metrics on *optical flows*, as described in the next section. Moreover, we consider neural networks that satisfy *Lipschitz continuity*, and note that all networks with bounded inputs are Lipschitz continuous, such as the common fully-connected, convolutional, ReLU, and softmax layers. We denote by  $Lip_c$  the *Lipschitz constant* for class  $c$ .

## 3. Robustness: formulation and approximation

In this section, we formulate the robustness problem and provide an approximation with provable guarantees.

### 3.1. Robustness and maximum safe radius

In this work, we focus on *pointwise robustness*, which is defined as the invariance of a network’s classification over a small neighbourhood of a given input. Following this, the *robustness* of a classification decision for a specific input can be understood as the non-existence of adversarial examples in the neighbourhood of the input. Here, we work with the norm ball as a neighbourhood of an input, that is, given an input  $\mathbf{v}$ , a distance metric  $L^p$ , and a distance  $d$ ,  $B(\mathbf{v}, L^p, d) = \{\mathbf{v}' \mid \|\mathbf{v} - \mathbf{v}'\|_p \leq d\}$  is the set of inputs whose distance to  $\mathbf{v}$  is no greater than  $d$  based on the  $L^p$  norm. Intuitively, the norm ball  $B$  with centre  $\mathbf{v}$  and radius  $d$  limits perturbations to at most  $d$  with respect to  $L^p$ . Then (pointwise) robustness is defined as follows.

**Definition 2 (Robustness).** Given a network  $\mathcal{N}$ , an input  $\mathbf{v}$ , a distance metric  $L^p$ , and a distance  $d$ , an adversarial example  $\mathbf{v}'$  is such that  $\mathbf{v}' \in B(\mathbf{v}, L^p, d)$  and  $\mathcal{N}(\mathbf{v}') \neq \mathcal{N}(\mathbf{v})$ . Define the robustness of  $\mathbf{v}$  by  $\text{ROBUST}(\mathcal{N}, \mathbf{v}, L^p, d) \models \nexists \mathbf{v}' \in B(\mathbf{v}, L^p, d) \text{ s.t. } \mathcal{N}(\mathbf{v}') \neq \mathcal{N}(\mathbf{v})$ . If this holds, we say  $\mathcal{N}$  is safe w.r.t.  $\mathbf{v}$  within  $d$  based on the  $L^p$  norm.

While the above definition returns only True or False, we take a step further to quantify the measurement of robustness. That is, we compute the distance to the original

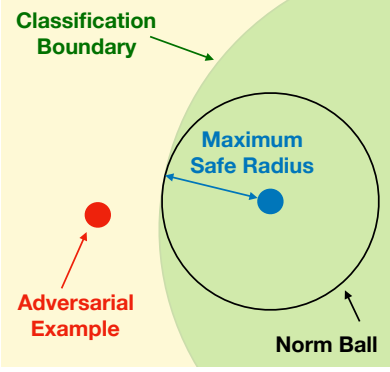


Figure 1. Maximum safe radius.

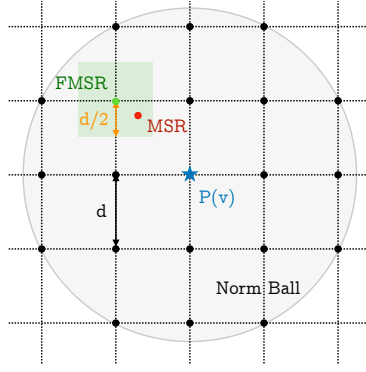


Figure 2. Error bounds.

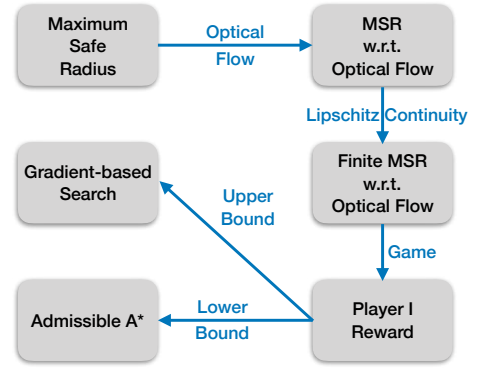


Figure 3. Robustness verification workflow.

input in the sense that, if exceeding the distance, there definitely exists an adversarial example, whereas, within the distance, all the points are safe, see Figure 1. We formally define this distance as the *maximum safe radius* as follows.

**Definition 3** (Maximum Safe Radius). *Given a network  $\mathcal{N}$ , an input  $\mathbf{v}$ , a distance metric  $L^p$ , and a distance  $d$ , the maximum safe radius (MSR) problem is to compute the minimum distance from input  $\mathbf{v}$  to an adversarial example  $\mathbf{v}'$ , i.e.,*

$$\text{MSR}(\mathcal{N}, \mathbf{v}, L^p, d) = \min_{\mathbf{v}' \in \mathcal{B}} \{ \|\mathbf{v} - \mathbf{v}'\|_p \mid \mathbf{v}' \in \mathcal{B}(\mathbf{v}, L^p, d) \text{ such that } \mathcal{N}(\mathbf{v}') \neq \mathcal{N}(\mathbf{v}) \}. \quad (1)$$

If  $\mathbf{v}'$  does not exist in  $\mathcal{B}$ , we let  $\text{MSR} = d + \epsilon$ , i.e.,  $d$  plus a small value  $\epsilon$ .

### 3.2. Maximum safe radius w.r.t. optical flow

In existing works that evaluate a network’s robustness over images, it is common to manipulate each image at pixel- or channel-level, and then compute the distance between the perturbed and original inputs. However, as we deal with time-series inputs, i.e., *videos*, instead of manipulating directly on each individual frame, we impose perturbation on each *optical flow* that is extracted from every pair of adjacent frames, so that both spatial features on frames and temporal dynamics between frames can be captured.

**Definition 4** (Optical Flow). *Given an input video  $\mathbf{v}$  with number  $l$  of frames, i.e.,  $\mathbf{v} = \{\mathcal{F}_1, \dots, \mathcal{F}_t, \dots, \mathcal{F}_l\}, t \in [1, l], t \in \mathbb{N}^+$ , the optical flow extraction function  $f : \mathcal{F}_t, \mathcal{F}_{t+1} \mapsto \mathbf{p}_t$  maps every two adjacent frames  $\mathcal{F}_t, \mathcal{F}_{t+1}$  into an optical flow  $\mathbf{p}_t$ . Then, for the sequential video  $\mathbf{v}$ , a sequence of optical flows can be extracted, i.e.,  $\mathbf{P}(\mathbf{v}) = \{\mathbf{p}_1, \dots, \mathbf{p}_t, \dots, \mathbf{p}_{l-1}\}, t \in [1, l-1], t \in \mathbb{N}^+$ .*

We remark that the distance between the flow sequences of two videos, denoted as  $\|\mathbf{P}(\mathbf{v}) - \mathbf{P}(\mathbf{v}')\|_p$ , can be measured similarly to that of two videos  $\|\mathbf{v} - \mathbf{v}'\|_p$  by the  $L^p$  norms in the standard way, as they are essentially tensors.

Then, to study the crafting of adversarial examples, we construct *manipulations* on the optical flow to obtain perturbed inputs. Note that if the input values are bounded,

e.g.,  $[0, 255]$  or  $[0, 1]$ , then the perturbed inputs need to be restricted to be within the bounds.

**Definition 5** (Optical Flow Manipulation). *Given an input  $\mathbf{v}$  with optical flow sequence  $\mathbf{P}(\mathbf{v})$ , an instruction function  $\Theta : \mathbb{N}^+ \rightarrow \mathbb{Z}$ , and a manipulation magnitude  $\tau \in \mathbb{R}$ , we define the optical flow manipulation  $\mathcal{M}_{\Theta, \tau}(\mathbf{p}_t)(i) =$*

$$\begin{cases} \mathbf{p}_t[i] + \Theta(i) \cdot \tau, & \text{if } i \in [1, w \times h], i \in \mathbb{N}^+ \\ \mathbf{p}_t[i], & \text{otherwise} \end{cases} \quad (2)$$

where  $w, h$  denote the width and height, i.e.,  $\Theta$  acts on flow dimensions. In particular, when  $\theta : \mathbb{N}^+ \rightarrow \{+1, -1\}$ , we say the manipulation is atomic, denoted as  $\mathcal{M}_{\theta, \tau}$ .

Instead of manipulating input  $\mathbf{v}$  directly, we manipulate the optical flow using  $\mathcal{M}_{\Theta, \tau}(\mathbf{P}(\mathbf{v}))$ , and then impose the manipulated flow on the original frames to obtain the corresponding perturbed input  $\mathbf{v}'$  such that  $\mathbf{P}(\mathbf{v}') = \mathcal{M}_{\Theta, \tau}(\mathbf{P}(\mathbf{v}))$ . To this end, we compute the distance from  $\mathcal{M}_{\Theta, \tau}(\mathbf{P}(\mathbf{v}))$  to  $\mathbf{P}(\mathbf{v})$  instead of that from  $\mathbf{v}'$  to  $\mathbf{v}$  because the former reflects both spatial and temporal manipulations. That is, we compute the *maximum safe radius w.r.t. optical flow*  $\text{MSR}(\mathcal{N}, \mathbf{P}(\mathbf{v}), L^p, d)$  rather than  $\text{MSR}(\mathcal{N}, \mathbf{v}, L^p, d)$ .

### 3.3. Approximation based on Lipschitz continuity

Here, we utilise the fact that the networks studied in this work are *Lipschitz continuous*, which allows us to discretise the neighbourhood space of an optical flow sequence and consider a finite number of points on a grid instead of infinitely many. First, based on the definitions of optical flow and input manipulation, we transform the MSR problem into the following *finite maximum safe radius* problem.

**Definition 6** (Finite Maximum Safe Radius). *Given an input  $\mathbf{v}$  with the optical flow sequence  $\mathbf{P}(\mathbf{v})$  and a manipulation operation  $\mathcal{M}_{\Theta, \tau}$ , then the finite maximum safe radius (FMSR) w.r.t. optical flow is*

$$\begin{aligned} \text{FMSR}(\mathcal{N}, \mathbf{P}(\mathbf{v}), L^p, d, \tau) &= \min_{\mathbf{p}_t \in \mathbf{P}(\mathbf{v})} \min_{\theta \in \Theta} \{ \\ &\|\mathbf{P}(\mathbf{v}) - \mathcal{M}_{\theta, \tau}(\mathbf{P}(\mathbf{v}))\|_p \mid \mathcal{M}_{\theta, \tau}(\mathbf{P}(\mathbf{v})) \in \mathcal{B}(\mathbf{P}(\mathbf{v}), L^p, d) \\ &\text{s.t. } \mathcal{N}(\mathbf{v}') \neq \mathcal{N}(\mathbf{v}) \text{ and } \mathbf{P}(\mathbf{v}') = \mathcal{M}_{\theta, \tau}(\mathbf{P}(\mathbf{v})) \}. \end{aligned} \quad (3)$$

If  $\mathbf{v}'$  does not exist in  $B$ , we let  $\text{FMSR} = d + \epsilon$ .

Intuitively, we aim to find a set of atomic manipulations  $\theta \in \Theta$  to impose on a sequence of optical flows  $\mathbf{p}_t \in P(\mathbf{v})$ , such that the distance between the flow sequences is minimal, and the corresponding perturbed input  $\mathbf{v}'$  is an adversarial example. Considering that, within a norm ball  $B(P(\mathbf{v}), L^p, d)$ , the set of manipulations is finite for a fixed magnitude  $\tau$ , the FMSR problem only needs to explore a finite number of the ‘grid’ points. To achieve this, we let  $\bar{g}$  be a  $\tau$ -grid point such that  $|\bar{g} - P(\mathbf{v})| = n \times \tau$ , and  $\Gamma(P(\mathbf{v}), L^p, d)$  be the set of  $\tau$ -grid points whose corresponding optical flow sequences are in  $B$ . Note that all the  $\tau$ -grid points are reachable from each other via manipulation. By selecting an appropriate  $\tau$ , we ensure that the optical flow space can be covered by small sub-spaces. That is,  $B(P(\mathbf{v}), L^p, d) \subseteq \bigcup_{\bar{g} \in \Gamma} B(\bar{g}, L^p, \frac{1}{2}\tilde{d}(L^p, \tau))$ , where the grid width  $\tilde{d}(L^p, \tau)$  is  $|\mathcal{D}|\tau$  for  $L^1$ ,  $\sqrt{|\mathcal{D}|\tau^2}$  for  $L^2$ , and  $\tau$  for  $L^\infty$ . Now, we can use FMSR to estimate MSR within the error bounds, as shown in Figure 2. We remark that approximation is only involved in the estimation of MSR from FMSR, which does not compromise the guarantees provided by FMSR.

**Theorem 1** (Error Bounds). *Given a manipulation magnitude  $\tau$ , the optical flow space can be discretised into a set of  $\tau$ -grid points, and MSR can be approximated as*

$$\begin{aligned} & \text{FMSR}(\mathcal{N}, P(\mathbf{v}), L^p, d, \tau) - \frac{1}{2}\tilde{d}(L^p, \tau) \\ & \leq \text{MSR}(\mathcal{N}, P(\mathbf{v}), L^p, d) \leq \text{FMSR}(\mathcal{N}, P(\mathbf{v}), L^p, d, \tau). \end{aligned} \quad (4)$$

We next show how to determine  $\tau$ . Note that, in order to make sure each  $\tau$ -grid point  $\bar{g}$  covers all the possible manipulation points in its neighbourhood, we compute the largest  $\tau$ , which can be obtained via *Lipschitz continuity*. For a network  $\mathcal{N}$  that is Lipschitz continuous at input  $\mathbf{v}$ , given Lipschitz constant  $\text{Lip}_c, c \in C$ , we have

$$\tilde{d}'(L^p, \tau) \leq \frac{\min_{c \in C, c \neq \mathcal{N}(\mathbf{v})} \{\mathcal{N}(\mathbf{v}, \mathcal{N}(\mathbf{v})) - \mathcal{N}(\mathbf{v}, c)\}}{\max_{c \in C, c \neq \mathcal{N}(\mathbf{v})} (\text{Lip}_{\mathcal{N}(\mathbf{v})} + \text{Lip}_c)}. \quad (5)$$

The proof is given in Appendix A.1 of an extended version of the paper [30]. Here we remark that, while  $\tilde{d}'(L^p, \tau)$  is with respect to input  $\mathbf{v}$  and  $\tilde{d}(L^p, \tau)$  is with respect to the flow sequence  $P(\mathbf{v})$ , the relation between them is dependent on the flow extraction method used. As this is not the main focus of this work, we do not expand on this topic.

#### 4. A game-based verification approach

In this section, we show that the finite optimisation problem FMSR of Definition 6 can be reduced to the computation of a player’s reward when taking an optimal strategy in a game-based setting. The verification workflow is in Figure 3. To this end, we adapt the game-based approach proposed in [31] for robustness evaluation of CNNs on images.

#### 4.1. Problem solving as a two-player game

We define a two-player turn-based game, where Player I chooses which optical flow to perturb, and Player II then imposes atomic manipulations within the selected flow.

**Definition 7** (Game). *Given a network  $\mathcal{N}$ , an input  $\mathbf{v}$  and its extracted optical flow sequence  $P(\mathbf{v})$ , we let  $\mathcal{G}(\mathcal{N}, \mathbf{v}, L^p, d) = (S \cup (S \times P(\mathbf{v})), s_0, \{T_I, T_{II}\}, L)$  be a game model, where*

- $S \cup (S \times P(\mathbf{v}))$  denotes the set of game states, in which  $S$  is the set of Player I’s states whereas  $S \times P(\mathbf{v})$  is the set of Player II’s states. Each  $s \in S$  corresponds to an optical flow sequence  $P(s)$  in the norm ball  $B(P(\mathbf{v}), L^p, d)$ .
- $s_0 \in S$  is the initial state such that  $P(s_0)$  corresponds to the original optical flow sequence  $P(\mathbf{v})$ .
- $T_I : S \times P(\mathbf{v}) \rightarrow S \times P(\mathbf{v})$  is Player I’s transition relation defined as  $T_I(s, \mathbf{p}_t) = (s, \mathbf{p}_t)$ , and  $T_{II} : (S \times P(\mathbf{v})) \times \Theta \rightarrow S$  is Player II’s transition relation defined as  $T_{II}((s, \mathbf{p}_t), \theta) = \mathcal{M}_{\theta, \tau}(\mathbf{p}_t)$ , where  $\mathcal{M}_{\theta, \tau}$  is the atomic manipulation of Definition 5. Intuitively, in a game state  $s$ , Player I selects a flow  $\mathbf{p}_t$  of  $P(s)$  and enters into a Player II’s state  $(s, \mathbf{p}_t)$ , where Player II then chooses an atomic manipulation  $\mathcal{M}_{\theta, \tau}$  on  $\mathbf{p}_t$ .
- $L : S \cup (S \times P(\mathbf{v})) \rightarrow C$  is the labelling function that assigns to each game state the corresponding class.

To compute FMSR of Definition 6, we let the game  $\mathcal{G}$  be cooperative. When it proceeds, two players take turns – Player I employs a strategy  $\sigma_I$  to select optical flow, then Player II employs a strategy  $\sigma_{II}$  to determine atomic manipulations – thus forming a path  $\rho$ , which is a sequence of states and actions  $s_0 \mathbf{p}_0 s_1 \theta_1 s_2 \mathbf{p}_2 s_3 \dots$ . Formally, we define the strategy of the game as follows. Let  $\text{Path}_I^F$  be a set of finite paths ending in Player I’s state, and  $\text{Path}_{II}^F$  be a set of finite paths ending in Player II’s state, then we define a *strategy profile*  $\sigma = (\sigma_I, \sigma_{II})$ , such that  $\sigma_I : \text{Path}_I^F \rightarrow \mathcal{D}(P(\mathbf{v}))$  of Player I maps a finite path to a distribution over next actions, and similarly  $\sigma_{II} : \text{Path}_{II}^F \rightarrow \mathcal{D}(\Theta)$  for Player II.

Intuitively, by imposing atomic manipulations in each round, the game searches for potential adversarial examples with increasing distance to the original optical flow. Given  $\rho$ , let  $\mathbf{v}'_\rho$  denote the input corresponding to the last state of  $\rho$ , and  $P(\mathbf{v}'_\rho)$  denote its optical flow sequence, we write the *termination condition*  $tc(\rho) \equiv (\mathcal{N}(\mathbf{v}'_\rho) \neq \mathcal{N}(\mathbf{v})) \vee (\|P(\mathbf{v}'_\rho) - P(\mathbf{v})\|_p > d)$ , which means that the game is in a state whose corresponding input is either classified differently, or the associated optical flow sequence is outside the norm ball. To quantify the distance accumulated along a path, we define a reward function as follows. Intuitively, the reward is the distance to the original optical flow if an adversarial example is found, and otherwise it is the weighted summation of the rewards of its children on the game tree.

**Definition 8 (Reward).** Give a finite path  $\rho$  and a strategy profile  $\sigma = (\sigma_I, \sigma_{II})$ , we define a reward function  $R(\sigma, \rho)$

$$= \begin{cases} \|\mathbf{P}(\mathbf{v}'_\rho) - \mathbf{P}(\mathbf{v})\|_p, & \text{if } tc(\rho) \text{ and } \rho \in \text{Path}_I^F \\ \sum_{\mathbf{p}_t \in \mathcal{P}(\mathbf{v})} \sigma_I(\rho)(\mathbf{p}_t) \cdot R(\sigma, \rho T_I(\text{last}(\rho), \mathbf{p}_t)), & \text{if } -tc(\rho) \text{ and } \rho \in \text{Path}_I^F \\ \sum_{\theta \in \Theta} \sigma_{II}(\rho)(\theta) \cdot R(\sigma, \rho T_{II}(\text{last}(\rho), \theta)), & \text{if } \rho \in \text{Path}_{II}^F \end{cases} \quad (6)$$

where  $\sigma_I(\rho)(\mathbf{p}_t)$  is the probability of Player I choosing flow  $\mathbf{p}_t$  along  $\rho$ , and  $\sigma_{II}(\rho)(\theta)$  is that of Player II determining atomic manipulation  $\mathcal{M}_{\theta, \tau}$  along  $\rho$ . Also,  $\rho T_I(\text{last}(\rho), \mathbf{p}_t)$  and  $\rho T_{II}(\text{last}(\rho), \theta)$  are the resulting paths of Player I, Player II applying  $\sigma_I, \sigma_{II}$ , respectively, i.e., extending  $\rho$  by adding a new state after transition.

## 4.2. Robustness guarantees

We now confirm that the game can return the optical value of the reward function as the solution to the FMSR problem. Proof of Theorem 2 is in Appendix A.2 of [30].

**Theorem 2 (Guarantees).** Given an input  $\mathbf{v}$ , a game model  $\mathcal{G}(\mathcal{N}, \mathbf{v}, L^p, d)$ , and an optimal strategy profile  $\sigma = (\sigma_I, \sigma_{II})$ , the finite maximum safe radius problem is to minimise the reward of initial state  $s_0$  based on  $\sigma$ , i.e.,

$$\text{FMSR}(\mathcal{N}, \mathcal{P}(\mathbf{v}), L^p, d, \tau) = \min R(\sigma, s_0). \quad (7)$$

## 5. Computation of the converging upper and lower bounds

Since computing FMSR directly is difficult, we instead exploit algorithms to compute its upper and lower bounds. We emphasise that there is no approximation in the guarantees provided by the bounds, e.g., any adversarial perturbation of magnitude less than the lower bound will definitely not cause a misclassification. This includes all the possible inputs within the lower bound, not just a discrete subset.

### 5.1. Upper bound: gradient-based search

We propose a gradient-based search algorithm to compute an upper bound of FMSR. Here, we utilise the spatial features extracted from individual frames.

**Definition 9 (Spatial Features).** Given a network  $\mathcal{N}$ , let  $\mathcal{N}_C$  denote the convolutional part, then  $\mathcal{N}_C : \mathbf{v} \mapsto \eta \in \mathbb{R}^{l \times m}$  maps from input  $\mathbf{v}$  to its extracted spatial features  $\eta$ , which has consistent length  $l$  of  $\mathbf{v}$  and feature dimension  $m$  of a frame. Then, we pass  $\eta$  into the recurrent part  $\mathcal{N}_R$  and obtain the classification, i.e.,  $\mathcal{N}_R : \eta \mapsto \mathcal{N}(\mathbf{v}, c), c \in C$ .

The objective is to manipulate optical flow as imperceptible as possible while altering the final classification. We

write the objective function as follows:

$$\forall t \in [1, l-1], t \in \mathbb{N}^+, \min \mathbf{p}_t + \epsilon \cdot \nabla_{\mathbf{p}_t}(\mathcal{N}, \mathbf{v}) \quad \text{s.t.} \quad \nabla_{\mathbf{p}_t}(\mathcal{N}, \mathbf{v}) = \frac{\partial \mathbf{1}_{\mathbf{v}}^{\mathcal{N}}}{\partial \eta} \odot \frac{\partial \eta}{\partial \mathbf{p}_t} \quad (8)$$

where  $\epsilon$  is a constant, and  $\nabla_{\mathbf{p}_t}(\mathcal{N}, \mathbf{v})$  is the perturbation imposed on  $\mathbf{p}_t$ . The key point is to minimise  $\nabla_{\mathbf{p}_t}(\mathcal{N}, \mathbf{v})$  so that the perturbation is not noticeable while simultaneously changing  $\mathcal{N}(\mathbf{v})$ . Here, we utilise the loss of  $\mathcal{N}$  on  $\mathbf{v}$ , denoted as  $\mathbf{1}_{\mathbf{v}}^{\mathcal{N}}$ , to quantify the classification change. Intuitively, if  $\mathbf{1}_{\mathbf{v}}^{\mathcal{N}}$  increases,  $\mathcal{N}(\mathbf{v})$  is more likely to change. By utilising the concept of spatial features  $\eta$ , we rewrite  $\nabla_{\mathbf{p}_t}(\mathcal{N}, \mathbf{v})$  as  $\partial \mathbf{1}_{\mathbf{v}}^{\mathcal{N}} / \partial \eta \odot \partial \eta / \partial \mathbf{p}_t$ , where  $\partial \mathbf{1}_{\mathbf{v}}^{\mathcal{N}} / \partial \eta$  denotes the gradient of the network's loss w.r.t. the spatial features,  $\partial \eta / \partial \mathbf{p}_t$  denotes the gradient of the spatial features w.r.t. the optical flow, and  $\odot$  denotes the Hadamard product.

We introduce the computation of the two parts below. On one hand,  $\partial \eta / \partial \mathbf{p}_t$  essentially exhibits the relation between spatial features and optical flow. Here we reuse input manipulation (Definition 5) to compute  $\partial \eta / \partial \mathbf{p}_t$ , though instead of manipulating the flow we impose perturbation directly on the frame. Intuitively, we manipulate the pixels of each frame to see how the subtle optical flow between the original and the manipulated frames will influence the spatial features. Each time we manipulate a single pixel of a frame, we get a new frame which is slightly different. If we perform  $\mathcal{M}_{\theta, \tau}$  on pixel  $\mathcal{F}[m, n]$ , and denote the manipulated frame as  $\mathcal{F}_{m, n}$ , its spatial features as  $\eta_{m, n}$ , the subtle optical flow between  $\mathcal{F}_{m, n}$  and  $\mathcal{F}$  as  $\delta \mathbf{p}_{m, n}$ , then  $\partial \eta / \partial \mathbf{p}_t$  can be computed as in Equation (9) below.

$$\frac{\partial \eta}{\partial \mathbf{p}_t} = \begin{pmatrix} \|\eta_{1,1} - \eta\|_p & \dots & \|\eta_{1,w} - \eta\|_p \\ \|\delta \mathbf{p}_{1,1}\|_p & \dots & \|\delta \mathbf{p}_{1,w}\|_p \\ \vdots & \ddots & \vdots \\ \|\eta_{h,1} - \eta\|_p & \dots & \|\eta_{w,h} - \eta\|_p \\ \|\delta \mathbf{p}_{h,1}\|_p & \dots & \|\delta \mathbf{p}_{w,h}\|_p \end{pmatrix} \quad (9)$$

On the other hand,  $\partial \mathbf{1}_{\mathbf{v}}^{\mathcal{N}} / \partial \eta$  shows how the spatial features will influence the classification, which can be reflected by the loss of the network. After getting  $\eta$  from  $\mathcal{N}_C$ , we can obtain  $\mathbf{1}_{\mathbf{v}}^{\mathcal{N}}$  from  $\mathcal{N}_R$ . If we perform pixel manipulation  $\mathcal{M}_{\theta, \tau}(\mathcal{F}[m, n])$  on frame  $\mathcal{F}$ , and obtain a new input, denoted as  $\mathbf{v}_{\mathcal{F}[m, n]}$ , then for this frame we have the gradient in Equation (10) below.

$$\frac{\partial \mathbf{1}_{\mathbf{v}}^{\mathcal{N}}}{\partial \eta} = \begin{pmatrix} \mathbf{1}_{\mathbf{v}_{\mathcal{F}[1,1]}}^{\mathcal{N}} - \mathbf{1}_{\mathbf{v}}^{\mathcal{N}} & \dots & \mathbf{1}_{\mathbf{v}_{\mathcal{F}[1,w]}}^{\mathcal{N}} - \mathbf{1}_{\mathbf{v}}^{\mathcal{N}} \\ \|\eta_{1,1} - \eta\|_p & \dots & \|\eta_{1,w} - \eta\|_p \\ \vdots & \ddots & \vdots \\ \mathbf{1}_{\mathbf{v}_{\mathcal{F}[h,1]}}^{\mathcal{N}} - \mathbf{1}_{\mathbf{v}}^{\mathcal{N}} & \dots & \mathbf{1}_{\mathbf{v}_{\mathcal{F}[w,h]}}^{\mathcal{N}} - \mathbf{1}_{\mathbf{v}}^{\mathcal{N}} \\ \|\eta_{h,1} - \eta\|_p & \dots & \|\eta_{w,h} - \eta\|_p \end{pmatrix} \quad (10)$$

**Remark.** From Definition 9, we know that the spatial features  $\eta = \mathcal{N}_C(\mathbf{v})$  only depend on each individual frame  $\mathcal{F}$  of  $\mathbf{v}$  and do not capture the temporal information be-

---

**Algorithm 1:** Admissible A\* for DNN Verification

---

**Input :** Game  $\mathcal{G}(\mathcal{N}, v, L^p, d)$ , terminating condition  $tc$   
**Output:** Lower bound of FMSR

```
1 procedure ADMISSIBLEA*( $\mathcal{G}(\mathcal{N}, v, L^p, d), tc$ ):  
2    $root \leftarrow s_0$ ;  
3   while ( $\neg tc$ ) do  
4      $P(root) \leftarrow \text{Player I}(root, \text{Farneback})$ ;  
5     for  $p_t$  in  $P(root)$  do  
6        $p_t[i] \leftarrow \text{Player II}(p_t)$ ;  
7        $newnodes \leftarrow \mathcal{M}_{\theta, \tau}(p_t)(i)$ ;  
8       for  $node$  in  $newnodes$  do  
9          $dist \leftarrow \text{DistanceEstimation}(node)$ ;  
10       $root \leftarrow \text{MaximumSafeRadius}(distances)$ ;  
11  return  $\|P(root) - P(s_0)\|_p$ 
```

---

tween frames. That is, when  $\mathcal{N}_C$  remains unchanged,  $\eta$  and  $\mathcal{F}$  have a direct relation, which indicates that the gradient of the latter can reflect that of the former. Therefore, during implementation, instead of the distance between  $\eta_{m,n}$  and  $\eta$ , we calculate that between  $\mathcal{F}_{m,n}$  and  $\mathcal{F}$ , i.e.,  $\|\mathcal{F}_{m,n} - \mathcal{F}\|_p$ .

## 5.2. Lower bound: admissible A\*

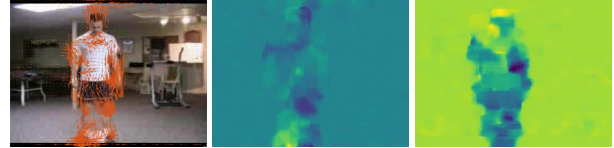
We exploit admissible A\* to compute a lower bound of Player I’s reward, i.e., FMSR. An A\* algorithm gradually unfolds the game model into a tree, in the sense that it maintains a set of children nodes of the expanded partial tree, and computes an estimate for each node. The key point is that in each iteration it selects the node with the *least* estimated value to expand. The estimation comprises two components: (1) the exact reward up to the current node, and (2) the estimated reward to reach the goal node. To guarantee the lower bound, we need to make sure that the estimated reward is minimal. For this part, we let the A\* algorithm be *admissible*, which means that, given a current node, it never overestimates the reward to the terminal goal state. For each state  $s$  in the game model  $\mathcal{G}$ , we assign an estimated distance value  $\text{DistanceEstimation}(s) = \|P(s) - P(s_0)\|_p + \text{heuristic}(P(s))$ , where  $\|P(s) - P(s_0)\|_p$  is the distance from the original state  $s_0$  to the current state  $s$  based on the  $L^p$  norm, and  $\text{heuristic}(P(s))$  is the admissible heuristic function that estimates the distance from the current state  $s$  to the terminal state. Here, we use  $\tilde{d}(L^p, \tau)$  in Equation (4). We present the admissible A\* algorithm in Algorithm 1.

## 6. Experimental results

This section presents the evaluation results of our framework to approximate the *maximum safe radius w.r.t. optical flow* on a video dataset. We perform the experiments on a Linux server with NVIDIA GeForce GTX Titan Black GPUs, and the operating system is Ubuntu 14.04.3 LTS. The results are obtained from a VGG16 [21] + LSTM [13] network on the UCF101 [22] video dataset. Details about

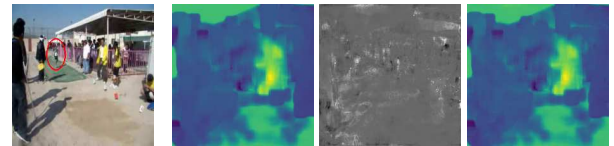


(a) SoccerJuggling at 0 s and 1 s.

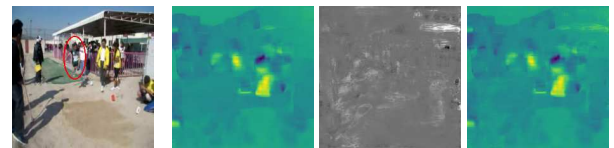


(b) Optical flow and its magnitude (left) and direction (right).

Figure 4. Illustration of how optical flow is able to capture the dynamics of the moving objects. (a) Two sampled frames from SoccerJuggling with original size  $320 \times 240 \times 3$ . (b) The optical flow (red arrows) extracted between the frames, and its two characteristics: magnitude and direction.



(a) Frame at 1 s. (b) magnitude: original, difference, and perturbed.



(c) Frame at 2 s. (d) direction: original, difference, and perturbed.

Figure 5. Imperceptible perturbations on optical flow, in terms of magnitude and direction, leading to misclassification from LongJump (100.00%) to FloorGymnastics (86.10%). (a)(c) Sampled frames at 1 s and 2 s with size  $224 \times 224 \times 3$ . (b) Original and perturbed magnitude. (d) Original and perturbed direction.

the dataset, the network structure, and training/testing parameters can be found in Appendix A.3 of [30].

### 6.1. Adversarial examples via manipulating flows

We illustrate how *optical flow* can capture the temporal dynamics of the moving objects in neighbouring frames. In this case, we exploit the Gunnar Farneback algorithm [5] as it computes the optical flow for all the pixels in a frame, i.e., *dense* optical flow, instead of a sparse feature set. Figure 4 presents an optical flow generated from two adjacent frames of a video labelled as SoccerJuggling: (a) shows two frames sampled at 0 s and 1 s; and (b) exhibits the characteristics of the flow: magnitude and direction. We observe that, when the indoor background essentially remains unchanged, the motion of playing football is clearly captured by the flow. More examples are included in Appendix A.4 of [30].

We now demonstrate how a very slight *perturbation on the flow*, almost imperceptible to human eyes, can lead to a

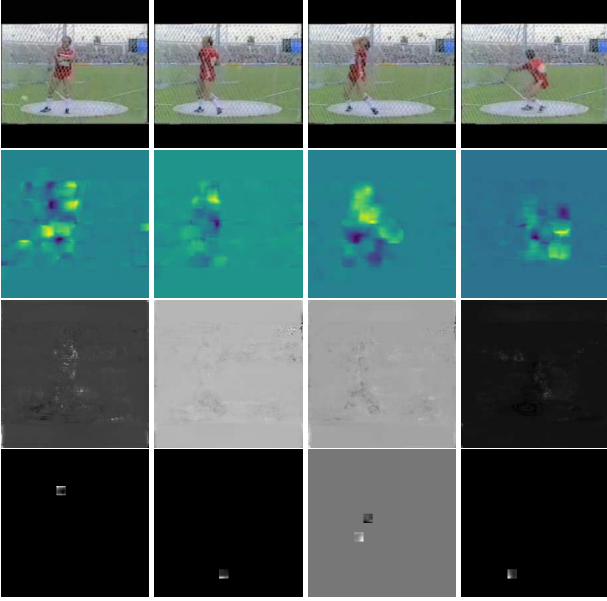


Figure 6. Examples of *unsafe* and *safe* perturbations on the optical flows of a HammerThrow video. Top row: sampled frames from 0 s to 3 s. 2nd row: optical flows of the frames from 0 s to 4 s. 3rd row: *unsafe* perturbations on the flows corresponding to the upper bound. Bottom row: *safe* perturbations reflecting the lower bound.

misclassification of the whole video. Figure 5 exhibits that a video originally classified as LongJump with confidence 100.00% is manipulated into FloorGymnastics with confidence 86.10%. Two sampled frames at 1 s and 2 s are shown in the 1st column. If we compare the original optical flow of magnitude and direction (2nd column) generated from the frames with the perturbed ones (4th column), we can hardly notice the difference (3rd column). However, the classification of the video has changed.

## 6.2. Converging upper and lower bounds

We illustrate the convergence of the bound computation for the *maximum safe radius* with respect to manipulations on the optical flows extracted from the consecutive frames of a video. Take a HammerThrow video as an example. Figure 6 exhibits four sampled frames (top row) and the optical flows extracted between them (2nd row). By utilising our framework, we present an approximation of MSR in Figure 7, where the red line indicates the descending trend of the *upper* bound, whereas the blue line denotes the ascending trend of the *lower* bound. Intuitively, after 20 iterations of the gradient-based algorithm, the upper bound, i.e., minimum distance to an adversarial example, is 5670.31 based on the  $L^2$  distance metric. That is, manipulations imposed on the flows exceeding this upper bound may be *unsafe*. Figure 6 (3rd row) shows some of such unsafe perturbations on each optical flow, which result in the misclassification of the video into FrontCrawl with confidence 99.86%. As

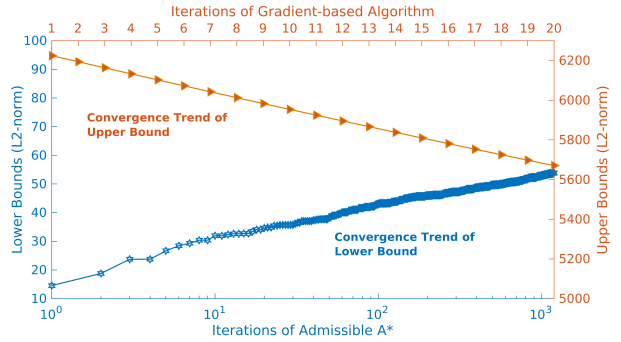
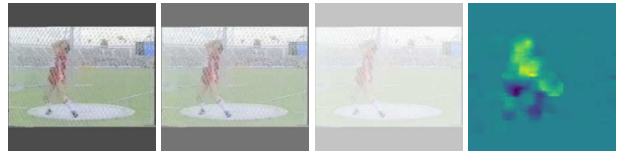
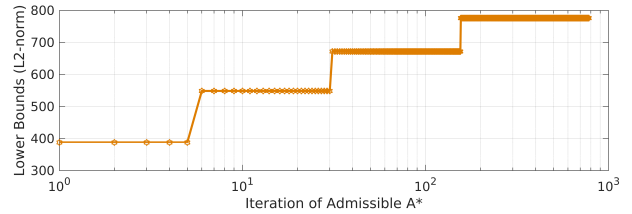


Figure 7. *Converging bounds* of the maximum safe radius of the HammerThrow video with respect to manipulations on extracted flows. Decreasing *upper* bound from the gradient-based algorithm shown in red, and increasing *lower* bound from admissible  $A^*$  in blue. Note that these two vertical axes have different scales.



(a) Brightness increase and the corresponding unaffected optical flow.



(b) Lower bounds of the maximum safe radius.

Figure 8. *Safe brightness* changes to the HammerThrow video and the corresponding lower bounds of the maximum safe radius. (a) The frame at 2 s of HammerThrow with increasing brightness, and the optical flow extracted from the same frame taking into account the brightness change. (c) The ascending lower bounds of the maximum safe radius reflecting the brightness change.

for the lower bound, we observe that, after 1000 iterations of the admissible  $A^*$  algorithm, the lower bound reaches 52.95. That is, manipulations within this  $L^2$  norm ball are absolutely *safe*. Some of such safe perturbations can be found in the bottom row of Figure 6. Due to space limit, we include another example in Appendix A.5 of [30].

## 6.3. Extension to naturally plausible distortions

Our framework can be extended to practical applications where distortions are more *natural and physically plausible* to the modality of the data itself, because all the perturbations preserving the semantic content of a video are essentially compositions of various atomic manipulations, and thus can be easily incorporated.

Take the “*brightness change*” perturbation as an example. As illustrated in Figure 8, we increase the brightness of

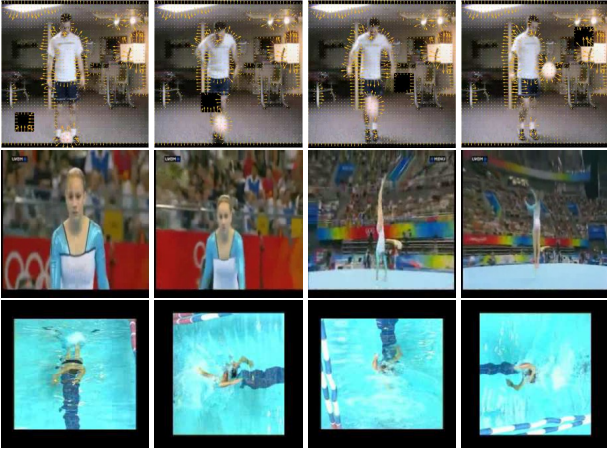


Figure 9. Some possible extensions of the adversarial perturbations to more *naturally plausible distortions*. Top: “camera occlusion” to the SoccerJuggling video with the *Horn-Schunck* optical flow method. Middle: “horizontal flip” to the FloorGymnastics video. Bottom: “angular rotation” to the FrontCrawl video.

the HammerThrow video on the frame level. That is, each pixel in the same frame is simultaneously brightened by the atomic manipulation  $\tau$ , thus resulting in the overall distance to the original video increasing by  $d'(L^p, \tau \cdot w \cdot h)$ , where  $w$  denotes the width of the frame and  $h$  height. The corresponding lower bounds of MSR are computed in Figure 8(b). Intuitively, this means that any degree of brightness alteration is definitely *safe* as long as the distance to the original video is less than the computed lower bound. For instance, after 10 iterations, the lower bound is 548.68 based on the  $L^2$  norm, then any frame-level brightness increase less than 548.68 in the Euclidean distance will not change the classification of this video. One interesting phenomenon observed is that, as exhibited in Figure 8(a), when the brightness of a frame increases, the extracted optical flow on the same frame is *not* significantly affected, due to the fact that the motion is relatively unchanged. In other words, optical flow can naturally discard some perturbations that do not alter the underlying temporal dynamics.

Apart from the “brightness change”, we include some other possible natural distortions of the adversarial perturbations in Figure 9. We observe that the “camera occlusion” here is very similar to the safe perturbations of the HammerThrow video in Figure 6 (bottom row), and thus can be handled using similar methods. The “horizontal flip” and the “angular rotation” involve manipulations that form a group, and to deal with those our approach would need to be extended, for example by incorporating network invariances. Finally, regarding these various adversarial perturbations, we remark that whether the perturbations are visible for a human largely depends on the *manipulation* type – with the same distance to the original input, manipulations such as these physically plausible distortions are cer-

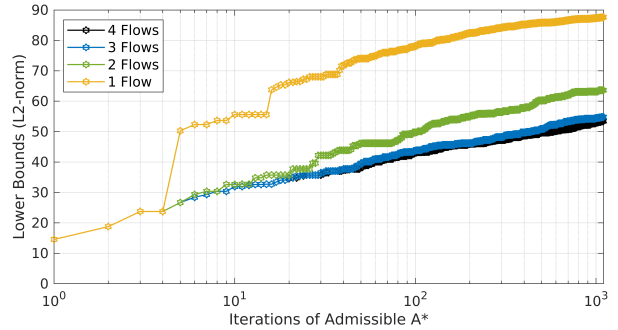


Figure 10. Different lower bounds of the maximum safe radius of a HammerThrow video with varying number of manipulated flows.

tainly more visible than unsafe perturbations produced by the gradient-based search algorithm and the safe perturbations created from the admissible A\* algorithm.

#### 6.4. Efficiency and scalability

As for the *computation time*, the upper bound requires the gradient of optical flow with respect to the frame, and because we extract dense optical flow, the algorithm needs to traverse each pixel of a frame to impose atomic manipulations; thus it takes around 30 minutes to retrieve the gradient of each frame. Once the gradient of the whole video is obtained, and the framework enters into the cooperative game, i.e., the expansion of the tree, each iteration takes minutes. Meanwhile, for the lower bound, the admissible A\* algorithm expands the game tree in each iteration which takes minutes, and updates the lower bound wherever applicable. Note that initially the lower bound may be updated in each iteration, but when the size of the game tree increases, it can take hours to update. Moreover, we analyse the *scalability* of our framework via an example of a HammerThrow video in Figure 10, which shows the lower bounds obtained with respect to varying dimensions of the manipulated flows. We observe that, within the same number of iterations, manipulating fewer flows leads to faster convergence, especially when the flows contain more spatial and temporal features.

### 7. Conclusion

In this work, we study the *maximum safe radius* problem for neural networks, including CNNs and RNNs, with respect to the optical flow extracted from sequential videos. Possible future work includes extending our framework to provide robustness guarantees for application domains where time-series input data and recurrent units are also involved, for example natural language processing.

**Acknowledgements:** This project has received funding from the European Research 20 Council (ERC) under the European Unions Horizon 2020 research and innovation programme (grant agreement No. 834115) and the EPSRC Programme Grant on Mobile Autonomy (EP/M019918/1).



## References

- [1] Andrew Burton and John Radford. *Thinking in perspective: critical essays in the study of thought processes*, volume 646. Routledge, 1978.
- [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] Souradeep Dutta, Susmit Jha, Sriram Sankaranarayanan, and Ashish Tiwari. Output range analysis for deep feedforward neural networks. In *NASA Formal Methods*, pages 121–138, 2018.
- [4] Rüdiger Ehlers. Formal verification of piece-wise linear feed-forward neural networks. In *Automated Technology for Verification and Analysis*, pages 269–286, 2017.
- [5] Gunnar Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Scandinavian conference on Image analysis*, pages 363–370. Springer, 2003.
- [6] Santiago Fernández, Alex Graves, and Jürgen Schmidhuber. An application of recurrent neural networks to discriminative keyword spotting. In *International Conference on Artificial Neural Networks*, pages 220–229. Springer, 2007.
- [7] Timon Gehr, Matthew Mirman, Dana Drachler-Cohen, Petar Tsankov, Swarat Chaudhuri, and Martin Vechev. AI2: Safety and robustness certification of neural networks with abstract interpretation. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 3–18, May 2018.
- [8] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.
- [9] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE, 2013.
- [10] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5-6):602–610, 2005.
- [11] Matthias Hein and Maksym Andriushchenko. Formal guarantees on the robustness of a classifier against adversarial manipulation. In *Advances in Neural Information Processing Systems 30*, pages 2266–2276. 2017.
- [12] Sepp Hochreiter, Martin Heusel, and Klaus Obermayer. Fast model-based protein homology detection without alignment. *Bioinformatics*, 23(14):1728–1736, 2007.
- [13] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [14] Xiaowei Huang, Marta Kwiatkowska, Sen Wang, and Min Wu. Safety verification of deep neural networks. In *International Conference on Computer Aided Verification*, pages 3–29. Springer, 2017.
- [15] Nathan Inkawhich, Matthew Inkawhich, Yiran Chen, and Hai Li. Adversarial attacks for optical flow-based action recognition classifiers. *arXiv preprint arXiv:1811.11875*, 2018.
- [16] Guy Katz, Clark Barrett, David L Dill, Kyle Julian, and Mykel J Kochenderfer. Reluplex: An efficient SMT solver for verifying deep neural networks. In *International Conference on Computer Aided Verification*, pages 97–117. Springer, 2017.
- [17] Ching-Yun Ko, Zhaoyang Lyu, Lily Weng, Luca Daniel, Ngai Wong, and Dahua Lin. POPQORN: Quantifying robustness of recurrent neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3468–3477, 2019.
- [18] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2, IJCAI’81*, pages 674–679, 1981.
- [19] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 372–387. IEEE, 2016.
- [20] Nicolas Papernot, Patrick McDaniel, Ananthram Swami, and Richard Harang. Crafting adversarial input sequences for recurrent neural networks. In *MILCOM 2016-2016 IEEE Military Communications Conference*, pages 49–54, 2016.
- [21] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.
- [22] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. In *CRCV-TR-12-01*, 2012.
- [23] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.
- [24] 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.
- [25] Qinglong Wang, Kaixuan Zhang, Xue Liu, and C Lee Giles. Verification of recurrent neural networks through rule extraction. *arXiv preprint arXiv:1811.06029*, 2018.
- [26] David H Warren and Edward R Strelow. *Electronic spatial sensing for the blind: contributions from perception, rehabilitation, and computer vision*, volume 99. Springer Science & Business Media, 2013.
- [27] Xingxing Wei, Jun Zhu, Sha Yuan, and Hang Su. Sparse adversarial perturbations for videos. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, pages 8973–8980, 2019.
- [28] Tsui-Wei Weng, Huan Zhang, Pin-Yu Chen, Jinfeng Yi, Dong Su, Yupeng Gao, Cho-Jui Hsieh, and Luca Daniel. Evaluating the robustness of neural networks: An extreme value theory approach. In *6th International Conference on Learning Representations*, 2018.
- [29] Tsui-Wei Weng, Huan Zhang, Hongge Chen, Zhao Song, Cho-Jui Hsieh, Luca Daniel, Duane Boning, and Inderjit Dhillon. Towards fast computation of certified robustness for ReLU networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5276–5285, 2018.

- [30] Min Wu and Marta Kwiatkowska. Robustness guarantees for deep neural networks on videos. *arXiv preprint arXiv:1907.00098*, 2019.
- [31] Min Wu, Matthew Wicker, Wenjie Ruan, Xiaowei Huang, and Marta Kwiatkowska. A game-based approximate verification of deep neural networks with provable guarantees. *Theoretical Computer Science*, 807:298 – 329, 2020.
- [32] Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. Efficient neural network robustness certification with general activation functions. In *Advances in Neural Information Processing Systems 31*, pages 4939–4948. 2018.