Stealthy 3D Poisoning Attack on Video Recognition Models

Shangyu Xie, Yan Yan, and Yuan Hong, *Senior Member, IEEE*

Abstract—Deep Neural Networks (DNNs) have been proven to be vulnerable to poisoning attacks that poison the training data with a trigger pattern and thus manipulate the trained model to misclassify data instances. In this paper, we study the poisoning attacks on video recognition models. We reveal the major limitations of the state-of-the-art poisoning attacks on *stealthiness* and *attack effectiveness*: (i) the frame-by-frame poisoning trigger may cause temporal inconsistency among the video frames which can be leveraged to easily detect the attack; (ii) the feature collision-based method for crafting poisoned videos could lack both generalization and transferability. To address these limitations, we propose a novel stealthy and efficient poisoning attack framework which has the following advantages: (i) we design a 3D poisoning trigger as natural-like textures, which can maintain temporal consistency and human-imperceptibility; (ii) we formulate an ensemble attack oracle as the optimization objective to craft poisoned videos, which could construct convex polytope-like adversarial subspaces in the feature space and thus gain more generalization; (iii) our poisoning attack can be readily extended to the black-box setting with good transferability. We have experimentally validated the effectiveness of our attack (e.g., up to 95% success rates with only less than $\sim 0.5\%$ poisoned dataset).

Index Terms—Poisoning Attack, Video Recognition, Machine Learning Security

1 Introduction

Deep neural networks (DNNs) have been extensively studied in different domains, especially video recognition, such as self-driving [1], action recognition [2] and anomaly detection [3]. However, DNNs have been proven to be vulnerable to adversarial attacks. Evasion attacks [4] were first proposed to craft adversarial examples to deviate the learning models [5], [6], [7], [8], [9], [10].

Different from the aforementioned evasion attacks during inference phase, data poisoning attacks [11], [12], [13], [14], [15], [16] target the training phase of machine learning models, where the adversaries aim to inject poisoned data instances into the training dataset and thus degrade the performance of the model trained with such poisoned dataset. For example, a classic form of data poisoning attacks [12], [15] aims to enforce the trained model to misclassify a particular set of inputs. Recently, another form of poisoning attacks [11], [16], [17], [18], [19] can pose a more sophisticated threat to the DNN models, i.e., the attacker can inject the poisoned data generated with a small trigger pattern and then set up a link between the trigger with a target label (installing backdoor). Thus, the trained DNN model on the poisoned data will consistently misclassify the data involving the trigger to the target label while still making correct classification on the clean data. For example, a sticker on the road sign can effectively change the classification result from "stop sign" to "speed limit" [17]. However, the sticker can be easily detected since it is highly human-perceptible. Turner et al. [16] proposes a clean-label poisoning attack with Generative Adversarial Network (GAN) without changing the label of poisoned data, which improves the stealthiness of the attack. That is, such clean-label poisoning attacks will not degrade the test accuracy given the normal data, which can be harder to be detected by evaluating the overall performance of the trained model on a clean holdout test dataset.

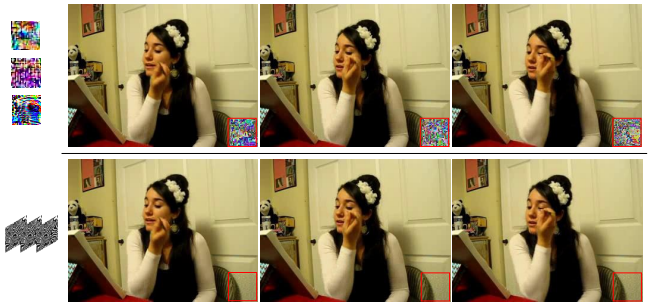


Fig. 1. A poisoned video example "Apply EyeMakeup" with the poisoning trigger (the squares of pixels leftside): recent work [19] (top) vs. ours (bottom). Our 3D poisoning trigger is more human-imperceptible as nature-like textures compared to [19]'s trigger of highly-deviated pixels.

While most existing data poisoning attacks focus on images [16], [17], [18], there are very limited works on DNN-based video models. It is worth noting that Zhao et al. [19] first explores the poisoning attack on the video models by extending a conventional image attack [16] to achieve high performance, which still has the major flaws on poisoning video models as following: (i) the patched frame-by-frame poisoning triggers [19] could jeopardize the temporal consistency in videos such that the poisoning attack might be easily detected, which can degrade the attack *stealthiness* and thus cause attack failure in the testing. We have experimentally shown such poisoned instances with trigger can be accurately detected by the state-of-the-art detection scheme on temporal consistency, AdvIT [20]; (ii) most poisoning

• *Shangyu Xie, Yan Yan and Yuan Hong are with the Department of Computer Science, Illinois Institute of Technology, Chicago, IL, 60616. E-mail: sxie14@hawk.iit.edu, yyan34@iit.edu, yuan.hong@iit.edu*

Manuscript received April 19, 2005; revised August 26, 2015.

attacks rely on feature collision [18], [19] with input-specific data samples by one-to-one mapping (similar to the targeted evasion attack [7]), which could lack *generalization* to the unseen data even if injecting multiple poisoned data [19]; (iii) under the black-box setting, poisoning attacks may not work well by attacking the substitute model since the target model can have very different classification boundaries (low *transferability*). These could greatly degrade the attack performance.

To address such limitations, we propose a novel stealthy and effective poisoning attack framework against the video recognition DNN models. Specifically, we first design a 3-dimensional (3D) poisoning trigger with temporal consistency based on a computer graphic primitive for *stealthiness*, which obtains good human-imperceptibility as natural-like textures. Figure 1 demonstrates an example of our 3D poisoning trigger compared with the state-of-the-art [19] on poisoning videos. Second, we craft the poisoned videos with the integration of an ensemble attack oracle (as the attack optimization objective), which formulates a convex polytope to cover the targeted videos in feature representation space (to provide more attack generalization and flexibility). Third, our proposed attack can craft more transferable poisoned videos by explicitly optimizing the attack in the intermediate layer feature representation of a video DNN model, which works in the black-box setting. Therefore, our main contributions are summarized as below:

- To our best knowledge, we are the first to reveal the limitations of state-of-the-art video poisoning attack in both stealthiness and attack performance.
- We design novel 3D poisoning triggers with a classic computer graphic primitive to ensure the attack stealthiness, which can be easily generated (by a few parameters) and human-imperceptible (nature-like patterns or textures, see Figure 1).
- Based on the 3D poisoning trigger, we propose a general attack framework, which can efficiently craft poisoned videos by formulating an ensemble attack oracle as objective. We further optimize the attack in aspect of attack generalizability and transferability.
- We conduct extensive experiments to validate the attack effectiveness and stealthiness with the benchmark of the previous attack methods. Besides, we have experimentally shown that the proposed attack can bypass various state-of-the-art defense schemes. We also show that our 3D poisoning attack can be readily downgraded to image domain.

The remainder of this paper is organized as follows. Section 2 introduces some background and related work. Section 3 and Section 4 present the preliminaries and design goal of our attack framework, respectively. Then, we illustrate the detailed design of the proposed attack in Section 5. Section 6 demonstrates the experimental results. Section 7 discusses potential methods to mitigate or advance our attack and Section 8 concludes the paper.

2 Background and Related Work

In this section, we first review the related literature of poisoning attack, which also includes the existing defense mechanisms. We also briefly present the taxonomy for DNN-based video recognition models.

2.1 Poisoning Attacks and Defenses

Poisoning Attacks. Poisoning attack injects poisoned instances (generated with some specific triggers) into the training dataset [12], [15], [16], [17], [18], [19], which can install the particular trigger as backdoor into the DNN. Thus, at the inference phase, the trained DNN model will misclassify the test instances with the presence of such trigger pattern. There are mainly two types of poisoning attacks: (i) poison-label attack can change both the training instances and their corresponding labels; (ii) clean-label attack changes the training instances without changing the labels. Poison-label attack can be mitigated by data filter since the poisoned data (mislabeled data) visually look different from the clean data but belonging to the same label. For example, Gu et al. [17] first proposes the poisoning attack on the deep learning application, BadNets, which injects patterns (e.g., stickers) into the poisoned data and also changes the corresponding label to the target label (as the poison-label attack). However, the patched triggers (e.g., stickers) can be easily detected via filtering or humans.

To improve the stealthiness of poisoning attack, Turner et al. [16] proposes clean-label attacks without changing the poisoned labels, by utilizing GAN to craft the poisoned instances for feature collision [12]. Saha et al. [18] presents a universal optimization method to generate multiple poisoned instances for one specific source instance, which could achieve relatively high success rates but still lack generalization. Zhu et al. [15] studies the transferability of poisoning attack and generates more transferable poisoned data based on convex polytope. It is limited to attacking the images without the backdoor trigger. Besides, such attack cannot be directly applied to videos since it would be computationally impractical to directly craft the poisoned videos due to the two-fold optimization with additional constraints. Zhao et al. [19] first studies the poisoning attack in the video domain, which aims to jointly craft universal triggers and poisoned videos with adversarial perturbations. Although this method has shown to be effective, it has some major flaws, e.g., temporal inconsistency aroused by the trigger, and low transferability as depicted before.

To address the above limitations, we propose a novel attack scheme for attacking video recognition models. Table 1 summarizes the main difference between our proposed attack and the state-of-the-art attacks. Our attack outperforms them on both stealthiness and attack effectiveness while attacking video DNN models (see the design goal in Section 4 and experimental results in Section 6).

Defenses against Poisoning Attacks. There have been several works which defend against the data poisoning attacks. For instance, Steinhardt et al. [21] proposed a certified defense scheme by constructing approximate upper bounds on the loss across the poisoning attacks. Tran et al. [22] proposed a spectral signature detection method for detecting poisoned instances in the training dataset. They observed that the poisoned data could be different from the clean data in the latent DNN space, which can be used for removing the poisoned data as outliers from the training data. Liu

TABLE 1
Comparison of our work and existing clean-label poisoning attacks.

	Turner et al. [16]	Poison Frog [12]	Zhu et al. [15]	Hidden [18]	Zhao et. al [19]	Ours
Trigger	Adv. Perturbation	-	-	Rand. Pixels	Deviated Pixels	3D Procedural
Backdoor	✓	✗	✗	✓	✓	✓
Video Dom.	✗	✗	✗	✗	✓	✓
Stealthiness	⋆⋆	⋆	⋆	⋆⋆	⋆	⋆⋆⋆
Attack Effec.	⋆	⋆	⋆⋆	⋆⋆	⋆⋆	⋆⋆⋆

et al. [23] proposed a fine-pruning method to prune the abnormal units to prevent the poisoning attack. Another approach is the neural network cleanse [24], which checks if the trained model is poisoned via reverse engineering the poisoning triggers with the gradient information. Then, neural cleanse uses an input filter to filter the poisoned data using a simple technique called median absolute deviation. We have experimentally evaluated the resistance of our proposed attack against such defense schemes. The experimental results show that our attack can bypass these defense schemes.

2.2 DNN-based Video Recognition

Well-designed DNN models, e.g., C3D [2], I3D [25], TSM [26] and X3D [27] have been widely adopted for efficient and accurate video recognition, such as action classification [28] and anomaly detection [3] in surveillance systems. Starting from the C3D model, the 3D convolutional networks on learning spatio-temporal features have significantly improved the performance of video recognition. Moreover, I3D improves C3D via inflating the 2D convolution filters (in conventional image networks) into the 3D convolution. We will evaluate our attack on such two most representative video DNN models on two large-scale video datasets, UCF101 [29] and HMDB51 [30] for video classification (see details in Section 6.1).

3 Attack Preliminaries

In this section, we first introduce the threat model, including the attack scenario, the adversary's knowledge/capability. We then formulate 3D poisoning attacks with video models.

3.1 Threat Model

Attack Scenario. We consider the *clean-label* poisoning attack [16], [18], [19] in the video domain. That is, the attacker will generate the poisoned videos, which visually look as the original clean data (thus keeping the clean label to bypass the detection of the data filtering/humans). It should be noted the attacker cannot control the labeling process (different from the poison-label setting). To improve the attack performance, we inject a set of poisoned videos [18], [19] (still a very small portion of training dataset, e.g., 0.5%). Besides, since the generation of the poisoned video is offline, we do not consider the extra computational costs of generating poisoning videos (as pre-attack phase).

For the victim's model, we consider the transfer learning setting [15], [19], [31], i.e., given a pre-trained DNN models as feature extractor (yet kept *frozen*), we can finetune a linear classifier based on to the specific applications/datasets. Such transfer learning-based approach have been shown to be practical and effective considering the relatively small computation costs in various domains. For example, we can utilize a pre-trained I3D model on kinetics-400 dataset to extract the video features and train (fine-tune) a SVM classifier on UCF101 dataset for action classification [25].

Attacker's Knowledge. We consider both white-box and black-box setting. For white-box setting, the attacker only knows the victim's model architecture (white-box) [18]. For the black-box, the attacker will not have access to model's architecture and parameters as the black-box evasion attacks [6]. Then, the attacker can utilize a substitute model to craft poisoned videos to attack the victim's model (via transferability). In both white-box and black-box setting, we assume that the attacker knows the training dataset for training victim's model (thus can generate poisoned data).

Attacker's Capability. As depicted above, we assume that the attacker can successfully inject a small number of well-crafted poisoned data into the victim's training dataset, which follows the setting of previous works [16], [18]. This is reasonable since the victim could obtain the training dataset by crawling from the online resources with web crawler. That is, the attacker only needs to put the generated poisoned data on the internet as public resources, which could be very likely collected by the victim. The attacker cannot control the training process of victim's model.

3.2 Attack Formulation

Denote the target video by v_{tar}, and the source video by v_{src}. Given a poisoning trigger $\mathcal{P}_n$ and a binary mask M (the location of patch is 1 while non-patched locations are 0), the attacker can generate a patched source video v'_{src} by patching the poisoning trigger $\mathcal{P}_n$ to the source video v_{src}:

$$v'_{src} = v_{src} \odot (1 - M) + \mathcal{P}_n \odot M \tag{1}$$

where $\odot$ denotes the Hadamard multiplication. We assume that the patch location on all the frames in one video are fixed, and can also be changed by modifying the binary mask M. It should be noted that we have verified the location of trigger will not arouse the attack results significantly. We can always adjust the patch location accordingly to obtain more visual imperceptibility (e.g., in the background).

To enable a successful poisoning attack, we will generate a poisoned video v_{poi} which visually looks like the target video v_{tar} such that it can be labeled with the target label. Meanwhile, the poisoned video v_{poi} should be similar to the patched source video v'_{src} in the feature representation of a DNN model (to cause feature collision) [16], [18]. Thus, a video instance (belonging to the source class) patched

with the trigger can be misclassified into the target class. Formally, the attacker can craft the poisoned video as the following objective function:

$$v_{poi} = \arg\min_{v} ||\mathcal{F}(v) - \mathcal{F}(v'_{src})||_2^2 + \lambda D(v, v_{tar}) \quad (2)$$

where $\mathcal{F}(\cdot)$ outputs the video features extracted by a DNN model as feature extractor (e.g., in C3D model [2], the output of the fc7 layer is a 4096-dimensional feature vector). $D(\cdot)$ is a distance function (e.g., ∞-norm distance) to quantify the distance between v_{poi} and v_{tar} (the maximum pixel change). The attack optimization consists of two terms:

1) The first term makes the feature representation of the poisoned video $\mathcal{F}(v)$ close to the patched source video $\mathcal{F}(v'_{src})$.
2) The second term ensures that v_{poi} looks like the target video v_{tar}, which is upper bounded by ϵ.

We utilize the hyperparameter $\lambda > 0$ to weigh the two terms in the optimization. Conventionally, given one specific pair of source and target videos, the attacker can generate the poisoned video by solving Eq. 2 using the projected gradient descent (PGD) algorithm [8]. Also, multiple poisoned videos will be crafted to increase the success rate [18], [19].

4 Attack Design Goals & Insights

In this section, we will illustrate the major limitations of current poisoning attacks and then briefly introduce our design idea to address such limitations, respectively. Our attack design aims to improve from the following two aspects: 1) stealthiness; 2) attack performance.

G1: Stealthiness. In general, the stealthiness issues of poisoning attack with trigger mainly consist two aspects: 1) the poisoned videos to be injected into the training set (training phase); 2) the patched video with poisoning trigger at the inference phase. Recall that current state-of-the-art poisoning attacks are in clean-label setting, i.e., keeping the original labels of poisoned instances [16], [18]. This can be achieved by bounding $D(v_{poi}, v_{tar})$ (in Eq. 2). However, such clean-label setting can only solve the former stealthiness issue with poisoned videos, which aims to bypass the data filtering or humans. In
stealthiness issue of patched vide
still exists, especially in video don

More specifically, the generate
ally consist of irregular pixels in th
improve attack effectiveness to so
also result in temporal inconsister
frames, which can be accurately
the-art detection scheme, e.g., Ac
consistency. Besides, as shown
deviated pixels in the triggers co
humans. Both of these could direc
poisoning attack during the infere

To address this, we construct a
ger based on a computer graphic
[32], [33], which obtains no notic
and thus potentially fit for stealth
tack (detailed in Section 5.1). We ha
that our 3D poisoning trigger c

scheme, e.g., AdvIT while comparing with the state-of-the-art attacks [18], [19]. Also, we validate that our poisoning attack obtains good human imperceptibility by both quantitative measurement and human survey.

G2: High Attack Performance. As depicted earlier, we need to improve poisoning attack on both *generalization* and *transferability*. On the one hand, conventional poisoning attacks rely on the feature collision with the specific source/target instances (one-to-one mapping) [18], [19], where minimizing the distance in the feature space could cause source instances to be trapped into the boundary belonging to target label (successful attack). However, such one-to-one mapping for feature collision can be restrictive like the targeted evasion attack [7], which could be still hard to attack unseen data instances even they usually inject multiple poisoned instances (*lacking generalization*). On the other hand, sometimes the attacker may not know the victim's model (in black-box setting), then the feature collision attack may not work on the substitute model since the models can be very different, e.g., feature extractor. That is, for feature collision mapping, the small distance for one pair of source/target instances on one model's feature extractor may change to larger in case of another model (*low transferability*).

Instead, we define an attack primitive, namely, *Ensemble Attack Oracle* (Definition 1) with an ensemble of a set of crafted poisoned videos, which aims to construct some adversarial subspaces as *convex polytope* [34], [35] in the feature space to entrap the source video (for a successful attack) [15], [36], [37]. Different from one-to-one mapping in feature collision (with some isolated adversarial points), we can formulate a convex polytope with a set of poisoned videos, which can tolerate more generalization errors and also loose the generation of the poisoned videos. Therefore, such adversarial subspaces by ensemble attack oracle can lead to more transferable attack [15], [35]. Figure 2 demonstrates the comparison of our ensemble attack oracle with feature collision.

Furthermore, we improve the ensemble attack optimization with two empirical yet effective calibrations. We first leverage Empirical Risk Minimization (ERM) to obtain more generalization. Then for the transferability, we utilize the intermediate layer's features instead of the final output feature

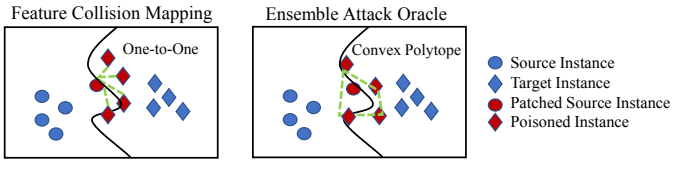


Fig. 2. Feature Collision Mapping vs. Ensemble Attack Oracle

5 Attack Framework Design

In this section, we elaborate the attack design for **G1** and **G2** (Section 4), respectively. We overview the main steps of the proposed attack framework (shown in Figure 3).

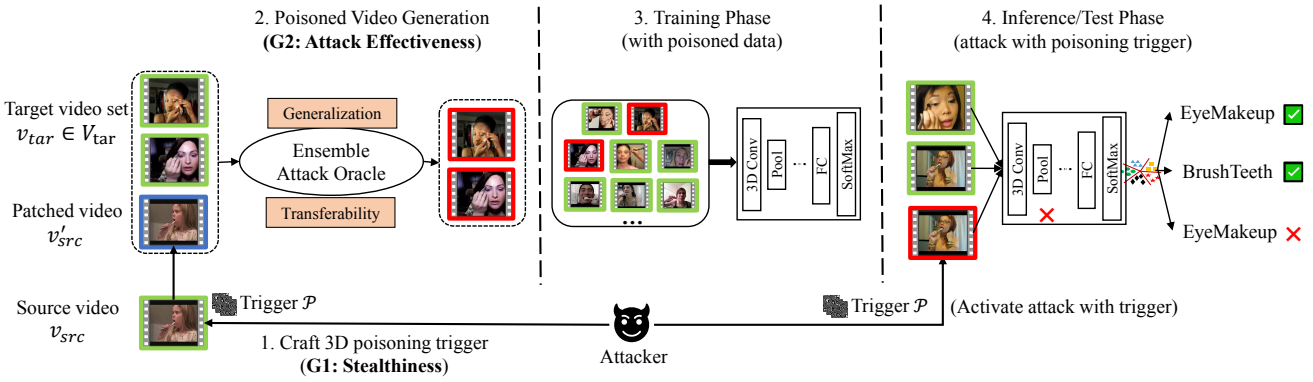


Fig. 3. Overview of 3D poisoning attack framework. There are four main steps: 1. the attacker crafts 3D poisoning trigger (for stealthiness); 2. with the optimization of both generalization and transferability, the attacker formulates an ensemble attack oracle to generate a set of poisoned videos (for attack effectiveness); 3. after the attacker injects the poisoned data to the training dataset, the victim will train the DNN model with the poisoned dataset; 4. during the test phase, the attacker can patch the 3D trigger on the test video (to activate the poisoing attack), which can be misclassified to the target label, e.g., "BrushTeeth" to "EyeMakeup".

5.1 3D Poisoning Trigger Generation

Procedural noise [32], [33] refers to the algorithmically generated visual patterns by some predefined functions, which have been widely used in film production and video games to enrich the visual details, e.g., texture and shading. It is inherently continuous and parameterized to compute [33]. Also, the noise has no noticeable directional artifacts. All these attributes enable the procedural noise (as computer graphic primitive) to be potentially fit for generating human-imperceptible poisoning trigger (*stealthiness*).

To craft the 3D poisoning trigger, we utilize one common type of procedural noise, i.e., Perlin noise [32] due to its ease of generation and popularity. Perlin noise was first proposed by Perlin as an image modeling primitive to produce the natural-like textures. As a lattice gradient noise, the noise value is determined by computing a set of 12 pseudorandom gradient vectors at the midpoints of 12 edges of a lattice cube, and then utilizing a quintic polynomial equation, e.g., $q(t) = 6t^5 - 15t^4 + 10t^3$ to interpolate the pre-defined vectors [32]. It can be computed efficiently with a few parameters. Thus, the Perlin noise can be readily extended to construct the 3D poisoning trigger.

More formally, we denote every pixel of 3D poisoning trigger by its 3D coordinates (x, y, t), where $(x, y), x, y \in [0, d-1]$ are the coordinates in frame t (the trigger is a square of $d \times d$). Denote the Perlin noise value of each pixel (x, y, t) by $s(x, y, t)$. To enrich the visual details (e.g., natural-looking texture for *stealthiness*), we can aggregate a set of octaves (the number of octaves denoted as Λ). Besides, we define two new parameters of wavelength λ_s and λ_t to determine the attribute of octaves along the spatial (location) and temporal (frame), respectively. Then the noise value at 3D coordinates (x, y, t) can be updated as:

$$\mathcal{P}(x, y, t) = \sum_{\ell=0}^{\Lambda} s(x \cdot \frac{2^\ell}{\lambda_s}, y \cdot \frac{2^\ell}{\lambda_s}, t \cdot \frac{2^\ell}{\lambda_t}) \tag{3}$$

To further improve the stealthiness by enabling various visual perturbations with different color spaces in the video, we extend Eq. 3 with a color mapping function [40]. Then, the noise value of (x, y, t) for the 3D poisoning trigger can be generated as:

$$\mathcal{P}_n(x, y, t) = K * cmap(\mathcal{P}(x, y, t), \phi) \tag{4}$$

where $cmap(b, \phi) = \sin(b \cdot 2\pi\phi)$ is a sine color map function, which bounds the noise with the circular property. K is the upper bound of the 3D trigger (in ℓ_∞-norm).

With such function, our attack can craft the poisoning trigger for patching the video on-the-fly (*video-agnostic*), i.e., we can manipulate the visual texture of the trigger pattern by adjusting the function parameters. For instance, we can first determine the location of the poisoning trigger, e.g., bottom right with trigger size $d = 30$. Since the video classification usually analyzes each video clip with 16 consecutive video frames, we can compute 3D poisoning trigger referring to Eq. 4, $t \in [0, 15]$. Also, we can control the style of trigger pattern by adjusting the parameter of the color map function. Once we obtain the poisoning trigger, we can craft the poisoned videos as depicted below. We have experimentally validated that our 3D poisoning trigger ensures good *stealthiness* and human-imperceptibility with both quantitative and human survey study (Section 6.4).

5.2 Poisoned Video Generation

Following **G2**, we construct an *Ensemble Attack Oracle* [15], [36], [37] to improve attack effectiveness as the following.

Definition 1 (Ensemble Attack Oracle). *Given a patched source video v'_{src} and a set of N poisoned videos to be crafted $V_{poi} = \{v^i_{poi}, i \in [1, N]\}$, then an ensemble attack oracle, denoted as $\mathcal{A}(V_{poi}, v'_{src}, \mathcal{F}(\cdot))$, is to compute the feature representation distance between the linear combination of the poisoned videos set V_{poi} and v'_{src}:*

$$\mathcal{A}(V_{poi}, v'_{src}, \mathcal{F}(\cdot)) = || \sum_i^N w_i \mathcal{F}(v^i_{poi}) - \mathcal{F}(v'_{src})||_2^2$$
$$s.t. \quad \sum_i^N w_i = 1, w_i > 0, v^i_{poi} \in V_{poi}$$

With the ensemble attack oracle, we build a relaxed connection from the poisoned videos to patched source

video in the feature space. That is, the convex polytope space constructed by a set of poisoned videos can obtain more generalization than the one-to-one mapping for feature collision [18], [19]. Take Figure 2 as an example, for feature collision-based attack, we craft the poisoned video one by one to approach the source videos at the boundary, which could change the classification boundary and thus cause misclassifcation. We can observe that there are four poisoning videos approaching the patched source video on the lefthand side. In general, we can inject more poisoned videos to arouse more change of the boundary (and thus increase the attack success rate). On the righthand side, the four poisoned videos would formalize a convex polytope space with the ensemble attack oracle, where the source videos can be easier to be entrapped for more attack effectiveness.

More formally, we have the following proposition to show attack correctness of such attack oracle:

Proposition 1. *If $\mathcal{A}(V_{poi}, v'_{src}) = 0$ holds, and given $\forall v^i_{poi} \in V_{poi}$ to be labeled with the target class c and successfully injected into the training set, then v'_{src} will be misclassified into the target class c by victim's model (as successful attack).*

Proof. We denote the video linear classifier (after feature extractor $\mathcal{F}(\cdot)$) as $g(\cdot)$. Since all the poisoned videos are labeled to class c, then $\forall v^i_{poi} \in V_{poi}$, we have

$$Pr[g(\mathcal{F}(v^i_{poi})) = c] > Pr[g(\mathcal{F}(v^i_{poi})) = c'] \tag{5}$$

where $c' \neq c$ is other labels. Given $\mathcal{A}(V_{poi}, v'_{src}) = 0$, i.e.,

$$\mathcal{F}(v'_{src}) = \sum_i^N w_i \mathcal{F}(v^i_{poi}) \tag{6}$$

we thus get:

$$\begin{aligned} Pr[g(\mathcal{F}(v'_{src})) = c] &= Pr[g(\sum_i^N w_i \mathcal{F}(v^i_{poi})) = c] \\ = \sum_i^N w_i Pr[g(\mathcal{F}(v^i_{poi})) = c] &> \sum_i^N w_i Pr[g(\mathcal{F}(v^i_{poi})) = c'] \\ &= Pr[g(\mathcal{F}(v'_{src})) = c'] \end{aligned}$$

□

According to Proposition 1, we can craft a set of poisoned videos to enable source videos to be entrapped in the convex polytope in feature space. Note that such ensemble oracle can also provide more transferable attack due to the larger adversarial subspaces (convex polytope). Then we reformulate the attack optimization function by minimizing $\mathcal{A}$ (enable the source video to be covered by convex polytope):

$$\begin{aligned} &\min_{V_{poi}} \ \mathcal{A}(V_{poi}, v'_{src}, \mathcal{F}(\cdot)) \\ &s.t. \ \forall v^i_{poi} \in V_{poi}, D(v^i_{poi}, v_{tar}) \leq \epsilon \end{aligned} \tag{7}$$

Moreover, we can further improve our poisoning attack with the following calibration:

(i) Attack Generalization. A simple approach to improve the attack generalization is to attack a set of sampled data instances (aka. universal attack [41]). Thus, to further improve the attack generalization on unseen source videos (not in the training set), we update Eq. 7 with the expectation on a pre-selected patched source video set V'_{src} by normalizing the distance (to avoid bias).

$$\begin{aligned} &\min_{V_{poi}} \mathop{\mathbb{E}}_{v'_{src} \sim V'_{src}} \frac{\mathcal{A}(V_{poi}, v'_{src}, \mathcal{F}(\cdot))}{||\mathcal{F}(v'_{src})||_2^2} \\ &s.t. \ \ \forall i \in N, D(v^i_{poi}, v_{tar}) \leq \epsilon \end{aligned} \tag{8}$$

(ii) Transferability in Intermediate Layers. As depicted before, we utilize the feature representations of intermediate layers to improve attack transferability. Then, we update Eq. 8 with all the feature representations of the intermediate layers across the entire model as below:

$$\begin{aligned} &\min_{V_{poi}} \mathop{\mathbb{E}}_{v'_{src} \sim V'_{src}} \sum_{k=1}^{L} [\frac{\mathcal{A}(V_{poi}, v'_{src}, \mathcal{F}_k)}{||\mathcal{F}_k(v'_{src})||_2^2}] \\ &s.t. \ \ \forall i \in N, D(v^i_{poi}, v_{tar}) \leq \epsilon \end{aligned} \tag{9}$$

where L is the total number of layers and $\mathcal{F}_k, k \in [1, L]$ is the k-th layer function to output feature representations.

Since the above objective function (Eq. 9) includes one ensemble attack oracle (the linear combination of the poisoned videos' features), we utilize an efficient optimization method to iteratively update both linear coefficients $W = \{w_i\}, i \in [1, N]$ and poisoned videos $V_{poi} = \{v^i_{poi}\}, i \in [1, N]$. Specifically, we will fix one as the constraint while optimizing the other one. Given the set of poisoned videos V_{poi}, we use forward-backward splitting [42] (which is more efficient than back-propagation with neural model) to compute the optimal coefficients $W = \{w_i\}, i \in [1, N]$; then fixing coefficients W, we update the poisoned videos for one gradient step (due to computational efficiency). Note that we choose Adam optimizer [43] to update the poisoned videos since it converges more reliably. To find the optimal poisoned videos and coefficients, we will repeat the two sub-steps until convergence. Algorithm 1 depicts the details.

Algorithm 1: Poisoned Video Generation

Input: Target video set V_{tar}, Source video set V_{src}, Feature Layer function $\mathcal{F}_k(\cdot), k \in [1, L]$, 3D poisoning trigger $\mathcal{P}_n$
Output: N poisoned videos $V_{poi} = \{v^i_{poi}, i \in [1, N]\}$

Initialize N target videos $v^i_{tar} \in V_{tar}$ to be poisoned $V_{poi} = \{v^i_{poi} \leftarrow v^i_{tar}, i \in [1, N]\}$
Initialize $w_i \leftarrow \frac{1}{N}, i \in [1, N]$
while *not converged* **do**
  Randomly sample $v_{src} \leftarrow^s V_{src}$
  $v'_{src} = v_{src} \odot (1 - M) + \mathcal{P}_n \odot M$
  `// Given` V_{poi}`, update` w_i
  for $k \rightarrow 1$ *to* L **do**
    $C \leftarrow [\mathcal{F}_k(v^1_{poi}), \mathcal{F}_k(v^2_{poi}), \cdots, \mathcal{F}_k(v^N_{poi})]$
    $\tau \leftarrow \frac{1}{||C^\top C||_2}$
    update $w_i \leftarrow w_i - \tau C^\top (C w_i - \mathcal{F}_k(v'_{src}))$
  `// Given` w_i`, update` V_{poi}
  for $i \rightarrow 1$ *to* N **do**
    Graident step on v^i_{poi}
    Clip v^i_{poi} to be bounded via $||v^i_{poi} - v^i_{tar}||_\infty \leq \epsilon$
return N poisoned videos

6 EXPERIMENTS

In this section, we evaluate our 3D poisoning attack on different video datasets and DNN models with various base-

lines. We first introduce the experimental setup, including the datasets, models, baselines and the attack methodology. Then, we demonstrate the experimental results in aspects of attack performance and stealthiness (corresponding to our design goal **G1**/**G2**). Besides, we conduct the extensive ablation studies to study the effect of 3D poisoning trigger on the whole attack. We also experimentally shows that the proposed attack can resist defense schemes. Finally, we demonstrate that our 3D poisoning attack can be extended to the image domain (2D).

6.1 Experimental Setup

Datasets. We evaluate the attack on two commonly used real datasets for video recognition:

- The UCF101 [29] dataset has 13,320 video clips in 101 different action categories, e.g., archery, fencing, and punch.
- The HMDB51 [30] dataset contains 6,766 video clips which are categorized into 51 different actions, e.g., fencing, hit, gun shooting, and sword exercises.

For each dataset, we choose 80% of each category as the training dataset, from which we choose the target category to generate poisoned videos. Then, the remaining 20% videos are used for the test dataset. Note that we keep the test videos clean to evaluate model accuracy under different model setting (poisoned or clean). For *stealthiness*, a successful poisoning attack is also expected to maintain the original model accuracy (inference) after training on clean/poisoned training dataset, besides obtaining human-imperceptible perturbations.

Target Models. We mainly evaluate our attack on two state-of-the-art DNNs for video recognition, C3D [2] and I3D [25]. For both C3D and I3D, we first train the models on kinetic-400 dataset [44] as pre-trained models (working as feature extractor). Then we jointly fine-tune the last layer of models and a SVM classifier on UCF101 and HMDB51 datasets for video classification, respectively. Note that our target models only consider the RGB inputs (modifying the RGB values at the pixel level).

Baselines. Recall that there are very few works on the poisoning attack in the video domain, we utilize the most recent clean-label video attack [19] (denoted as "Baseline1"). We also extend a recent state-of-the-art image poisoning attack [18] to the video domain as the baseline (denoted as "Baseline2"). In addition, we downgrade our proposed 3D poisoning attack to 2D image and compare with Baseline2 [18]. The experimental results (Section 6.6) show that our attack can also effectively attack in image domain.

Attack Methodology. For both UCF101 [29] and HMDB51 dataset [30], we split the dataset into 80% training set and 20% test dataset (remain intact to evaluate the model accuracy). Take the first group of experiments (attack effectiveness) as an example, we randomly choose 50 pairs of source and target categories from the UCF101 dataset. For every source/target pair, we randomly select 20% videos of source category as the source video set V_{src}, to which we aim to attack, i.e., the source video patched with the 3D trigger during the test phase will be misclassified into the target class. We also randomly select 20% (as poisoning percentage, $\sim$ 0.2% out of the entire training set) videos from target category as target video set V_{tar}. Then, we generate the poisoned videos following Algorithm 1 (unless explicitly specified, the parameters will keep the same). The poisoning trigger size is 30×30 out of the 320×240 video frame. we set the upper bound ϵ is 8. We use Adam [43] with a relatively large learning rate of 0.05, and perform at most 3000 iterations on crafting poisoned videos for each experiment.

6.2 Attack Performance

To fully evaluate attack performance of the poisoning attack, our evaluation include the following three aspects:

1) The impact on model performance with clean data.
2) The effectiveness (attack success rate) with various models/datasets/attack parameters.
3) The comparison with baselines on attack success rate/transferability.

1) Impact on model performance. As depicted above, the poisoning attack should not impact the normal performance of victim's model (with poisoned training data) too much to keep stealthy. We evaluate the accuracy of the retrained model training with poisoned video dataset and normal training model with the clean dataset. we report both accuracy on the clean test dataset (excluded from the training videos), with UCF101 and HMDB51 dataset, respectively.

TABLE 2
Test accuracy of the clean and poisoned models.

Model / Dataset	C3D		I3D	
	Clean	Poisoned	Clean	Poisoned
UCF101	82.7%	81.5%	87.5%	86.3%
HMDB51	52.3%	51.1%	63.7%	62.4%

Table 2 summarizes the results for both C3D and I3D. We can observe that the poisoned video can maintain almost same accuracy compared with the original model, which shows that our 3D poisoning attack will not arouse too much change (slight drop) on the model prediction (only fool the model while presenting with poisoning trigger).

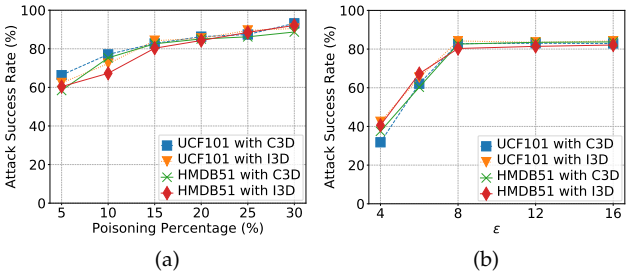


Fig. 4. Attack success rate vs. poisoning percentage (a) and perturbation bound ϵ (b) on the UCF101 and HMDB51.

2) Attack effectiveness. We first evaluate the attack performance for specific pairs of source and target video categories with the fixed poisoning trigger size and poisoning percentage. The poisoning trigger size is 20×20 out of the 320×240 video frame. We set trigger's magnitude $K = 10$.

TABLE 3
Attack performance of our 3D poisoning attack on the 9 randomly selected pairs of source/target video categories from the UCF101 dataset against the C3D and I3D models. $\epsilon = 8$ and poisoning percentage: 20%.

Source/ Target	BrushTeeth/ ApplyEyeMake	Biking/ HairCut	CliffDiving/ Rowing	Fencing/ JumpingJack	Hammering/ Archery	LongJump/ Skiing	Knitting/ Punch	Punch/ Typing	Skiing/ Tachi
C3D	90.7%	86.2%	89.2%	89.9%	96.1%	93.4%	87.0%	94.7%	96.2%
I3D	89.1%	88.3%	93.1%	92.5%	88.3%	86.7%	93.1%	88.5%	92.0%

TABLE 4
Attack performance of our attack vs. the baseline attacks [19] and [18], denoted as "Baseline1" and "Baseline2". Target category: "Apply EyeMakeup". $\epsilon = 8$ and poisoning percentage: 30%.

Model/ Dataset	Method	Brush Teeth	Biking	CleanAnd Jerk	Frisbee Catch	Horse Race	Long Jump	Playing Dhol	Punch	Skiing	Taichi
I3D/ UCF101	Baseline1	71.0%	76.2%	87.5%	88.0%	70.2%	74.9%	91.3%	82.5%	81.7%	86.0%
	Baseline2	80.5%	83.0%	86.2%	85.0%	76.2%	78.5%	84.2%	86.0%	87.4%	88.0%
	Ours	95.0%	90.4%	93.6%	91.7%	89.5%	94.0%	92.3%	96.5%	94.4%	93.8%

We randomly select 20% videos from the source category as the source video set V_{src}. We also randomly select 20% (as poisoning percentage, $\sim 0.2\%$ out of the entire training set) videos from target category as target video set V_{tar}. The upper bound $\epsilon = 8$. Table 3 summarizes the results of our 3D poisoning attack applied to 9 randomly selected pairs of source/target video categories in the UCF101 dataset against the C3D/I3D models. We can observe that our attack achieves high success rates on both C3D and I3D models, even with a small poisoning percentage, which shows both effectiveness and efficiency of our attack (note that small poisoning percentage reflects high efficiency).

We also evaluate the attack performance with the varying poisoning percentage and perturbation bound. As shown in Figure 4(a), the attack success rate also increases as the poisoning percentage grows. Our poisoning attack still achieves high success rates (>80%) even though the poisoning percentage is only 15%. This is consistent with the former results. From Figure 4(b), we observe that the attack success rate at first increases and then does not change as perturbation bound increases from 8 to 16. This is because the craft poisoned video will be easier with a high perturbation bound. Note the perturbations with poisoned video are still small (8 out of 255).

3) Comparison with Baselines. Table 4 demonstrates the results of our 3D poisoning attack applied to the UCF101 dataset (against the I3D model) comparing with the two baselines [18], [19], denoted as "Baseline1" and "Baseline2". We set "Apply EyeMakeup" as the target category, and the source categories (e.g., "biking") as [19]. The trigger size is 20×20 and the poisoning percentage is 30%.

As shown in Table 4, our attack achieves high success rates (>89%). For example, our attack can achieve 95.0% success rate on the source category of "Brush Teeth" and 90.4% on the "Biking" while Baseline1 only achieves 71.0%(<95.0%) and 76.2%(<90.4%) on such two categories, respectively (the third and forth columns). Moreover, comparing the remaining results, we can observe that our 3D poisoning attack can perform much better than both baselines. Such results are reasonable since our attack obtains good attack generalization for crafting poisoned videos.

Attack Transferability. For poisoning attack, we refer to the tranferability of poisoned data to be applied to another model. Then we evaluate the transferability of our attack compared with baselines (the same notations as above). Specifically, we choose one model (e.g., C3D) as substitute to generate poisoned videos, and we evaluate attack success rate on another model (e.g., I3D) trained with the poisoned videos, and vice versa. Figure 4(b) summarizes the overall results. The results show that our poisoning attack obtains high transferability across different models while the baselines lack such transferability (no more than 12% success rate). For example, our attack can achieve 50.5% success rate while Baseline1 only 8.6% on UCF101 dataset. Such results are reasonable. Considering the conventional poisoning attacks focus on the feature collision [18], [19] with fixed feature extractor function, the poisoning attack only obtain less transferability (the feature extractor of different models can be different, i.e., one successful crafted poisoned video for one feature extractor may not work for another). On the contrary, our attack can craft the poisoned videos (ensured by Eq. 9) which obtain good generalization and transferability. This also conforms with the previous results.

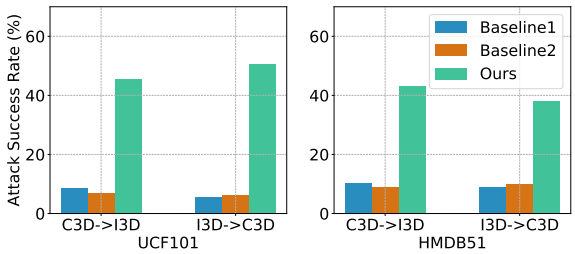


Fig. 5. Attack transferability of our attack vs. baselines.

Computational Overheads. Table 5 demonstrates the average running time for crafting 10 poisoned videos for 8 randomly selected pairs of source/target category in the UCF101 dataset. From the table, we can see that the average running time for one videos is around 1 minute at most. Considering our poisoning attack only injects very small number of poisoned videos, the computation overhead for crafting poisoned video is tolerable.

TABLE 5
Average runtime for crafting poisoned videos (sec).

Biking/ HairCut	CliffDiving/ Rowin	Fencing/ JumpingJack	Hammering/ Archery
35	39	28	62
LongJump/ SKiing	Knitting/ Punch	Punch/ Typing	Skiing/ Tachi
47	35	40	32

6.3 Attack Stealthiness

Recall that we reveal stealthiness issue of current poisoning attacks at inference phase can be caused by the highly-deviated poisoning trigger. That is, the videos patched with poisoned trigger can be easily identified by human (visual impact) or detection schemes. Thus we evaluate the stealthiness of our attack on the following aspects. For visual impact, we conduct both quantitative and human study. We adopt the state-of-the-art detection scheme for detecting the poisoning trigger.

1) Quantitative perceptual metric, i.e., SSIM [45].
2) Human-imperceptibility survey study.
3) Video poisoning detection, i.e., AdvIT [20].

1) SSIM. Structural Similarity (SSIM) is a perceptual metric to quantify the visibility of errors between a distorted image and the original image based on the degradation of structural information [45]. The range of SSIM is $(0, 1]$. A higher SSIM value indicates a better quality of the distorted image. Then we can utilize SSIM to quantify the visual impact of poisoning trigger. We choose the average SSIM for all the frames of one video as the SSIM of the video. Recall that our poisoning trigger is upper bounded by K (Equation 4). We set K as $\{5, 8, 10, 12\}$. Next, we randomly choose 100 poisoned video with one 3D trigger from each category, and average the SSIM as the final result.

TABLE 6
Average SSIM of 100 poisoned videos with varying K.

K	5	8	10	12
SSIM	0.997	0.994	0.986	0.984

In Table 6, the SSIM of the videos is very close to 1, which shows that the 3D poisoning trigger rarely affects the visual information. Thus, the attacker can simply adjust the parameters of the poisoning trigger function (e.g., K) with no significant visual changes in the poisoning attack. Note we also conduct extensive ablation study of 3D poisoning trigger in Section 6.4.

2) Human study. We conducted a human survey study to evaluate whether our poisoning attack could cause visual effect to humans (with the IRB exempt approval).

For the setup of study, we first generate the videos (including original videos, patched videos with trigger) by our attack. Specifically, we randomly pick 500 videos from the UCF101 and HMDB51 datasets. To avoid bias on the distribution of data samples, we first randomly choose 250 videos to generate 250 pairs of videos (the poisoned videos and original clean videos), and the remaining for 250 pairs of clean videos and their duplicates. Then we distribute an online survey to 50 anonymous students (not record any personal information, e.g., major, age or gender), which ask each participant to annotate 10 pairs of videos as ("visual difference" or "no visual difference"). Finally, we received 490 valid annotations of video pairs (49 students have submitted their results), including 244 poisoned pairs. Figure 6 demonstrates the results (left-side). We found that 97.5% (238) out of such 244 poisoned videos are annotated as "no visual difference", while only 2.5% are identified (as "visual difference"). There also exist 8 annotations identified as "visual difference" in the remaining 246 pairs of original videos and their duplicates.

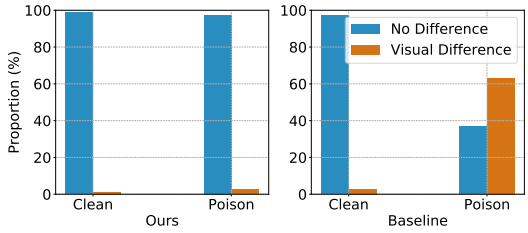


Fig. 6. Results of human survey on our attack vs. baseline [19].

We also repeat the same group of study for baseline attack [19] but selecting *different* videos from the dataset, which aims to avoid the connection with the previous study for our attack. That is, the previous annotation of our attack will enable the participants to have prior knowledge and then make biased annotation on the same pair of videos for baseline attack (vice versa). From Figure 6, we observe that 63.8% (157 out of 246 valid poisoned videos) are identified by the same group of participants. All the above results have indicated that our attack achieve high human-imperceptibility (significantly better than the baseline [19]).

Figure 10 gives two example pairs of source and target videos in categories "PlayingDhol" and "Apply Eye-Makeup". Due to strictly bound perturbations, the poisoned target video is visually similar to the target video. The patched source video with 3D poisoning trigger is also very similar to the clean source video.

3) Poisoning detection. Recall that we observe the poisoning triggers can directly cause the stealthiness issue by temporal video frame. We adopt a state-of-the-art detection scheme AdvIT [20] to detect the video patched with poisoning trigger (thus validate the limitation of previous attack [19]). AdvIT is originally designed to identify adversarial perturbation in the videos. Based on the assumption that perturbations can destroy the video frame consistency, AdvIT can find the temporal inconsistency among video frames by the optical flow information.

We identify the poisoning triggers of highly-deviated pixels [19] could be potentially destroy temporal inconsistency of video frames as adversarial perturbations. Then we adopt AdvIT to detect the poisoning trigger in the videos. Specifically, AdvIT first utilizes a DNN-based optical flow estimator, i.e., FlowNet [46], which can compute the optical flow information of suspicious video (usually a few video frames since the poisoning trigger is patched on the whole video). Then such optical flow information can be used to reconstruct some pseudo frames. We can output a inconsistency score between the suspicious video frames

and pseudo video frames. Since the perturbations/triggers usually consists of deviated pixels, the optical flow information along with video frame will be destroyed. That is, the higher inconsistency score, the higher possibility poisoning trigger's existence. Note that the trigger is usually fixed, e.g., bottom right. We can always separate the video with different regions for more precise detection.

TABLE 7
SSIM and Detection AUC of AdvIT.

Dataset / Trigger	UCF101		HMDB51	
	SSIM	AUC	SSIM	AUC
Baseline1	0.804	98.5%	0.822	99.3%
Baseline2	0.841	99.2%	0.865	98.4%
Ours	0.956	61.3%	0.973	58.6%

In the experiments, we choose other two types of poisoning trigger from the baselines: i) randomly generated static trigger [18]; ii) universal adversarial trigger from video poisoning attack [19] for comparison. We fix the trigger size as 20×20 and patch location is bottom right (as fixed in [19]). we randomly select 400 clean videos from the UCF101 and HMDB51 (200 each dataset), and apply both our 3D poisoning trigger and two baselines' trigger to generate patched videos. We set the upper bound of poisoning trigger to be $K = 8$. We report the Area Under Curve (AUC) values of AdvIT for detecting trigger and the average SSIM values of the corresponding videos for detection in Table 7 .

From the table, we can observe that the SSIM of our poisoned videos is close to 1, which shows that our 3D poisoning trigger rarely affects the visual information. Besides, the AUC values of ours are close to random guess (e.g., 61.3% for UCF101 dataset) while all other two baselines can be almost fully detected by AdvIT (the AUC values are close to 1). This is reasonable since the temporal consistency could be destroyed with the baseline's (highly deviated pixels).

6.4 Understandings of 3D Poisoning Trigger

We also perform ablation studies with 3D poisoning trigger in aspects of the stealthiness and attack performance with various trigger size, upper bound and patched location. Specifically, for every experiment, we will vary one parameter independently while fixing others and report the corresponding results. Figure 7 first visualizes the 3D poisoning trigger patched on 16 consecutive frames of the video.

Table 8 shows the attack performance of various trigger parameters on UCF101. We observe that both upper bound K and trigger location do not influence our attack performance much. Then we can adjust the trigger location to match with the background/objects (to improve stealthiness). Moreover, we see that the increase of trigger size can slightly improve the attack performance (as a larger trigger patch can help construct adversarial subspaces and attack easier to some extent), which also degrades stealthiness.

To evaluate the effect of patched trigger location on the attack performance, we choose 5 different locations, i.e., top/bottom + left/right and center on the video frames. We perform the same attack evaluation as previous experiments and report attack success rate. Trigger size is 20. Poisoning percentage is 20% and upper bound is 8. Table 9 shows

TABLE 8
Attack performance *vs.* varying trigger parameters.

Trigger Size			K		
10	20	30	8	10	12
82.3%	82.7%	83.0%	82.7%	82.7%	82.6%

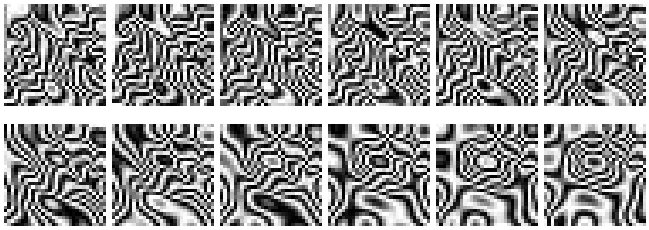


Fig. 7. One example visualization of 3D poisoning trigger (the first 12 consecutive frames). Trigger size: 30×30.

that trigger location cannot impact the attack performance too much. This is reasonable since the poisoning trigger will not directly be used for crafting poisoned videos to cause feature collision (as a backdoor in the victim's model). Besides the temporal consistency, we can further utilize the natural-like texture of our proposed trigger to increase the stealthiness, i.e., match the trigger with the background or the objects. The SSIM values of our poisoned videos also validate this point.

TABLE 9
Attack rate (AR) *vs.* varying trigger locations.

Location	Top Left	Top Right	Bottom Left	Bottom Right	Center
AR	82.6%	83.1%	83.0%	82.7%	82.9%

6.5 Resistance of Attack against Defenses

Besides adopting video detection scheme (Section 6.3), we also conduct extensive experiments to evaluate the resistance of our attack by adopting several state-of-the-art defense schemes: i) Fine-Pruning [23]; ii) Neural Cleanse [24]; iii) Spectral Signature [22], respectively. Additionally, we also design an adaptive defense scheme to fully evaluate the proposed poisoning attack.

Fine Pruning. We evaluate the resistance of all three attacks against the state-of-the-art Fine-Pruning [23]. We set the poisoning percentage to be 30%. The trigger size is 20 and upper bound is 8. We train C3D with the poisoned UCF101 dataset compared with other two baselines. For pruning, we prune the last convolutional layer of C3D model (i.e., Conv5b 512) to evaluate the corresponding accuracy and attack success rate. As shown in Figure 8(a), the attack success rates of both baselines drop drastically when 30% neuron are removed, e.g., Baseline1 from 84.2% to 30.4%. While our poisoning attack can still maintain 80% attack rate, which shows that our attack is more resistant to the neural pruning.

Neural Cleanse. Neural Cleanse [24] can detect whether a trained model is poisoned or not, where it assumes the training instance would require minor modifications by the

attacker. The tested model by Neural Cleanse will output an anomaly index (score) and a score higher than 2 indicates the poisoned model with backdoor trigger. We set the poisoning percentage to be 30%. The trigger size is 20 and upper bound is 8. We train both C3D and I3D model with the UCF101 dataset, respectively. Then we apply Neural Cleanse to detect both C3D and I3D model trained with the UCF101 dataset (by our attack). From Figure 8(b), we can observe that Neural Cleanse fails to detect the poisoned model for both cases, i.e., anomaly index <2 (>2 indicates detected poisoned model) [24].

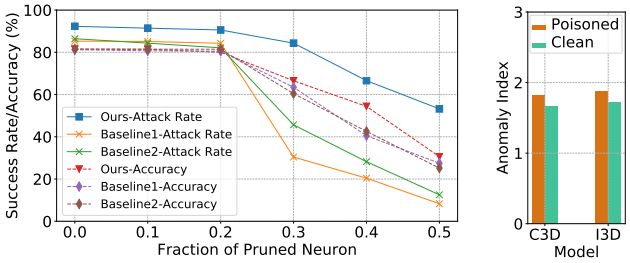


Fig. 8. Attack results against defenses. Left: Fine-pruning [23]. Right: Neural Cleanse [24]

Spectral Signature. We also apply one state-of-the-art detection scheme Spectral Signature [22], to detect the poisoned data in the training dataset, of which the intuition is that the poisoned data can be outliers in some latent spaces (thus can be removed). It uses statistical methods, e.g., SVD to detect the posioned samples as outliers. For experimental setup, we evaluate this scheme with the C3D model on the UCF101 and HMDB51 dataset, respectively. We set the poisoning percentage of the training dataset as 30% as a higher ratio. The trigger size is 20 and upper bound is 8. Then, we apply the detection on the 1000 videos in UCF101 dataset (consisting of 800 clean target videos and 200 generated poisoned videos) and 500 videos in HMDB51 dataset (400 clean target videos and 100 poisoned videos). Figure 9 demonstrates the detection results. From the figure (lefthand), taking UCF101 as an example, we observe that the detection method can only identify a small percentage (∼11%) of poisoned videos while also reporting false positive rate (∼9%) from the clean videos. The result of HMDB51 shows similar results. The above results indicate that such detection cannot mitigate our attack. Also, the attack success rate only downgrades about 4% even we remove the poisoned data as experiments and retrain the model. Note that the two baselines also report the similar results for this detection.

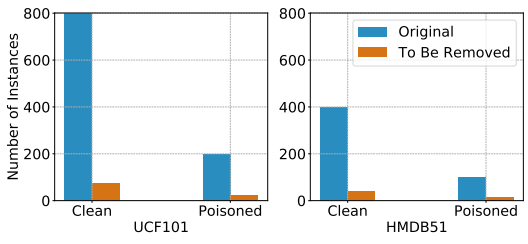


Fig. 9. Detection results of Spectral Detection [22].

Adaptive Defense. To fully evaluate our proposed attack, we also adopt current defense method as adaptive defense to tailor with the attack properties [47]. That is, we facilitate the defender with the knowledge for the 3D poisoning attack, e.g., the computer graphic primitive procedural noise is utilized for constructing poisoning triggers for our 3D poisoning attack. For the defense method, we improve Spectral Signature [22] by applying procedural noise-based poisoning triggers to its learning scheme.

Specifically, the defender will generate poisoned video samples with the procedural noise as a part of the training set for the detector. Thus it would potentially increase the detection performance considering the detector could achieve more generalization with the newly added poisoned videos. Since the defender does not necessarily know the poisoning trigger parameter, we assume that the poisoning triggers are randomly generated and patched on the videos. We follow the same setting as the detection experiments above. We report the final detection results in Table 10 for both UCF101 (first row) and HMDB51 (second row) dataset, respectively. From the table, we can observe that adaptive defense achieves a higher detection rate and also a lower false positive rate on both datasets, e.g., 27% > 11% and 5.6%>9%. Such results show that the adaptive defense can defend our attack to some extent. However, our proposed attack can also change its attack strategy, such as adjusting trigger generation function with another procedural noise to bypass the detection, which would require more robust and adaptive defense schemes.

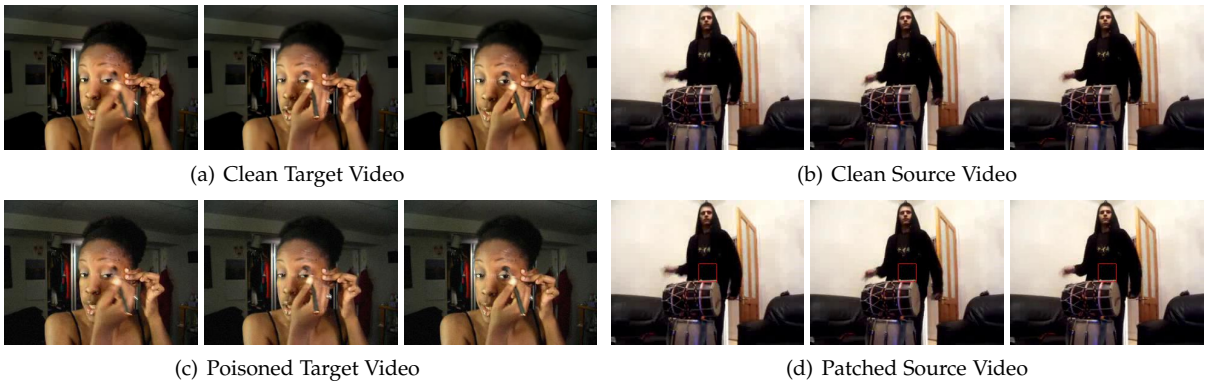
TABLE 10
Detection results of adaptive defense on UCF101 and HMDB51

Clean	Poisoned	Removed Clean	Removed Poisoned
800	200	47/5.6%	53 /27%
400	100	15/3.8%	32 /32%

6.6 Application on Image Poisoning Attack

Considering that the image can be viewed as a one-frame video, we can simply extend our 3D poisoning attack to images, i.e., downgrading the 3D poisoning generation to the 2-dimension by setting the time dimension to be 1. Then, we implement our poisoning attack on the CIFAR10 dataset [48] by benchmarking with the recent image poisoning work, "Baseline2" [18]. Under same experimental setting of the baseline attack (see details in [18]), we choose 10 randomly selected pairs of image categories, such as bird-dog, dog-plane and cat-truck (specific information of image categories pairs refers to Table 7 in [18]). The model is a simplified AlexNet which has four convolutional layers (64, 192, 384, and 256) kernels and two fully connected layers (512 and 10) neurons. The size of poisoning trigger 8×8 and the bound of trigger is 16. The size of images evaluation dataset for each category is 1000. We average all the success rates of 10 randomly selected pairs via our attack compared with the baseline attack.

We present the attack results for four representative pairs of image categories in Table 11. We observe that our downgraded 3D poisoning attack can still achieve high success rate on the image compared with the baseline. Such results



(a) Clean Target Video (b) Clean Source Video

(c) Poisoned Target Video (d) Patched Source Video

Fig. 10. Visualization of selected frames of clean target video (a), clean source (b), poisoned target video (c), and patched source video (d) of one specific pair, i.e., "Apply EyeMakeup" and "PlayingDhol". With strictly bounded perturbation, the poisoned target video (c) is visually no difference compared with the clean target video (a), but close (in feature space) to the patched source video (d) with 3D poisoning trigger, where the trigger (in "red frame") is human-imperceptible.

have shown the flexibility and effectiveness of our attack. Again, our poisoning attack can also ensure the stealthiness of poisoning trigger in the inference phase, whereas the baseline only focuses on hiding the poisoning trigger prior to training and still reveals the trigger pattern for testing.

TABLE 11
Comparison of attack results on the CIFAR10. Baseline attack [18].

Source/Target	bird/dog	dog/ship	frog/plane	cat/truck
Baseline	94.3%	87.6%	90.1%	93.0%
Ours	92.7%	90.4%	90.8%	94.4%

7 Discussion

We will discuss the potential mitigation of our 3D poisoning attacks and advanced attacks to motivate more robust defense schemes as the following.

7.1 Potential Mitigation

Considering the poisoning attack is a data-intensive attack, The potential defense schemes can be studied in the following aspects: 1) the detection of input videos with poisoning trigger during the inference phase, e.g., utilizing the property of trigger in the video domain; 2) the data filtering/detection of the poisoned training data (training phase), e.g., using the adversarial outliers of poisoned training data; 3) the certified robustness [36], [49], [50] against poisoned training data. The first two aspects aim to detect or mitigate the poisoning attack empirically with state-of-the-art schemes, while the last one is theoretical defense scheme against norm-bounded adversarial attack. Considering that the poisoning attack could depend on some intrinsic attributes, e.g., attack by the feature collision of feature representations, we may extend such certified robust scheme to defend against poisoning attacks.

Detection. Recall that we have designed a detection scheme adopting from AdvIT [20], it could effectively detect the poisoned instances with poisoning trigger of the baselines. Thus, to mitigate the risks of the proposed attack, we may utilize the knowledge of the procedural noise as the main defense primitive. That is, we could utilize the procedural noise as the defender's knowledge to revise/adopt the current poisoning or adversarial detection schemes adaptively, such as Spectral Signature [22], AdvIT [20]. We have shown an adaptive defense method based on spectral signature, which can defend against our attack to some extent. Alternatively, we can revise the AdvIT to train a detector by adding procedural noise to increase the detection accuracy, e.g., to enable the detector to memorize the change of optical flow aroused by the procedural-based trigger. We can also leverage ensemble-based [51], [52] method to improve the performance of the detector. For instance, we can choose multiple video models as base models to train multiple detectors and then get an average score for detecting the poisoned videos. Finally, we could leverage a reference database to classify the poisoned videos by k-NN. However, it could bring extra both storage and computational overheads.

Certified scheme. Certified robustness [36], [50] schemes have been shown to defend against adversarial attacks with additive ℓ_p bounded perturbations theoretically. More specifically, the certified scheme, e.g., random smoothing [50], can provide consistent predictions with guarantee for some norm-bounded input sets around one data instance, i.e., ℓ_p ball. That is, such ℓ_p ball could provide a "safe" space to resist such adversarial perturbed inputs. Similarly, we can enforce the trained classifier to form a "anti"-convex polytope [36] against such convex polytope-based poisoning attacks. That is, we can utilize the randomized smoothing method provided by certified schemes to trap the poisoned training data with a larger convex polytope. Thus after training, the model can still classify the poisoned video into the correct label instead of wrong label with high confidence. However, it should be noted the curse of high-dimensionality [53] still exists for certified robustness scheme, e.g., randomized smoothing, especially in video domain. We will work in this direction.

7.2 Advanced Attacks

We propose a *general* attack framework based on 3D poisoning trigger, which can improve the stealthiness of poisoning attack in the video domain (new modeling of poisoning trigger). Besides, our framework also integrates new attack

ensemble to improve attack performance in both generalization and transferability. This can bring more flexibility. For example, there will be some new or unknown video models (black-box), we can attack such models with the transferability. Also, our 3D trigger attack framework can also readily integrate other new attack optimizations or powerful attacks from adversarial attack domain in the future to obtain more attack performance. Note our poisoning attack can achieve good human-imperceptibility (according to the quantitative or human servery results), we can also further integrate our attack into the physical-world attacks [17], [54] based on the natural-like texture or style of poisoning trigger. For example, we can utilize visual light technology, such as smart LED [55], which could help to realize 3D poisoning trigger by programmable building blocks. This can pose a practical threat in the physical world.

8 Conclusion

In this paper, we propose a novel 3D poisoning attack on the video recognition models, which improves both attack stealthiness and effectiveness. Specifically, we utilize a computer graphic primitive to construct the 3D poisoning trigger, which results in significantly less visual changes for stealthiness. Furthermore, we design an optimization method on an ensemble attack objective to craft more effective poisoned videos. We also experimentally validate the performance of our attack, which outperforms the state-of-the-art methods. Since the defenses against the video poisoning attacks are rather limited, our 3D poisoning attack can advance the development of the defense schemes for sake of robustness in the video domain.

Acknowledgments

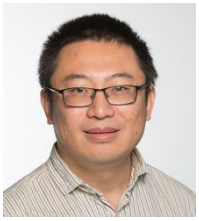
This work is partially supported by the National Science Foundation (NSF) under the Grants No. CNS-2046335 and CNS-2034870. We are also grateful to the anonymous reviewers for their very constructive comments.

References

[1] U. Muller, J. Ben, E. Cosatto, B. Flepp, and Y. L. Cun, "Off-road obstacle avoidance through end-to-end learning," in *NIPS*, 2006.
[2] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *IEEE ICCV*, 2015, pp. 4489–4497.
[3] W. Sultani, C. Chen, and M. Shah, "Real-world anomaly detection in surveillance videos," in *IEEE CVPR*, 2018.
[4] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli, "Evasion attacks against machine learning at test time," in *ECML-PKDD*, 2013.
[5] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *ICLR*, 2015.
[6] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *ACM CCS*, 2017.
[7] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *IEEE SP*, 2017, pp. 39–57.
[8] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.
[9] K. T. Co, L. Muñoz-González, S. de Maupeou, and E. C. Lupu, "Procedural noise adversarial examples for black-box attacks on deep convolutional networks," in *CCS*, 2019, pp. 275–289.
[10] Y. Mirsky, T. Mahler, I. Shelef, and Y. Elovici, "Ct-gan: Malicious tampering of 3d medical imagery using deep learning," in *USENIX Security*, 2019, pp. 461–478.
[11] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, "Targeted backdoor attacks on deep learning systems using data poisoning," *arXiv preprint arXiv:1712.05526*, 2017.
[12] A. Shafahi, W. R. Huang, M. Najibi, O. Suciu, C. Studer, T. Dumitras, and T. Goldstein, "Poison frogs! targeted clean-label poisoning attacks on neural networks," in *NIPS*, 2018.
[13] B. Biggio, B. Nelson, and P. Laskov, "Poisoning attacks against support vector machines," *arXiv preprint arXiv:1206.6389*, 2012.
[14] O. Suciu, R. Marginean, Y. Kaya, H. D. III, and T. Dumitras, "When does machine learning FAIL? generalized transferability for evasion and poisoning attacks," in *USENIX Security*, 2018, pp. 1299–1316.
[15] C. Zhu, W. R. Huang, H. Li, G. Taylor, C. Studer, and T. Goldstein, "Transferable clean-label poisoning attacks on deep neural nets," in *ICML*, 2019, pp. 7614–7623.
[16] A. Turner, D. Tsipras, and A. Madry, "Clean-label backdoor attacks," 2018.
[17] T. Gu, B. Dolan-Gavitt, and S. Garg, "Badnets: Identifying vulnerabilities in the machine learning model supply chain," *arXiv preprint arXiv:1708.06733*, 2017.
[18] A. Saha, A. Subramanya, and H. Pirsiavash, "Hidden trigger backdoor attacks," in *AAAI*, vol. 34, no. 07, 2020, pp. 11 957–11 965.
[19] S. Zhao, X. Ma, X. Zheng, J. Bailey, J. Chen, and Y.-G. Jiang, "Clean-label backdoor attacks on video recognition models," in *IEEE CVPR*, 2020, pp. 14 443–14 452.
[20] C. Xiao, R. Deng, B. Li, T. Lee, B. Edwards, J. Yi, D. Song, M. Liu, and I. Molloy, "Advit: Adversarial frames identifier based on temporal consistency in videos," in *IEEE CVPR*, 2019, pp. 3968–3977.
[21] J. Steinhardt, P. W. W. Koh, and P. S. Liang, "Certified defenses for data poisoning attacks," in *NIPS*, 2017.
[22] B. Tran, J. Li, and A. Madry, "Spectral signatures in backdoor attacks," in *NIPS*, 2018, pp. 8000–8010.
[23] K. Liu, B. Dolan-Gavitt, and S. Garg, "Fine-pruning: Defending against backdooring attacks on deep neural networks," in *RAID*, 2018, pp. 273–294.
[24] B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng, and B. Y. Zhao, "Neural cleanse: Identifying and mitigating backdoor attacks in neural networks," in *IEEE SP*, 2019, pp. 707–723.
[25] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," in *CVPR*, 2017.
[26] J. Lin, C. Gan, and S. Han, "Tsm: Temporal shift module for efficient video understanding," in *ICCV*, 2019.
[27] C. Feichtenhofer, "X3d: Expanding architectures for efficient video recognition," in *IEEE CVPR*, 2020, pp. 203–213.
[28] D. Tran, H. Wang, L. Torresani, and M. Feiszli, "Video classification with channel-separated convolutional networks," in *IEEE ICCV*, 2019, pp. 5552–5561.
[29] K. Soomro, A. R. Zamir, and M. Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, 2012.
[30] H. Kuehne, H. Jhuang, E. Garrote, T. A. Poggio, and T. Serre, "HMDB: A large video database for human motion recognition," in *IEEE ICCV 2011*, 2011, pp. 2556–2563.
[31] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," in *International conference on artificial neural networks*. Springer, 2018, pp. 270–279.
[32] K. Perlin, "Improving noise," ser. SIGGRAPH'02, 2002.
[33] A. Lagae, S. Lefebvre, R. Cook, T. DeRose, G. Drettakis, D. S. Ebert, J. P. Lewis, K. Perlin, and M. Zwicker, "A survey of procedural noise functions," in *Computer Graphics Forum*, no. 8., 2010.
[34] G. Takács, "Convex polyhedron learning and its applications," 2009.
[35] J. Bao, Y.-H. He, E. Hirst, J. Hofscheier, A. Kasprzyk, and S. Majumder, "Polytopes and machine learning," *arXiv preprint arXiv:2109.09602*, 2021.
[36] E. Wong and Z. Kolter, "Provable defenses against adversarial examples via the convex outer adversarial polytope," in *ICML*, 2018, pp. 5286–5295.
[37] V. Sehwag, A. N. Bhagoji, L. Song, C. Sitawarin, D. Cullina, M. Chiang, and P. Mittal, "Analyzing the robustness of open-world machine learning," in *AISec*, 2019, p. 105–116.

[38] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" *arXiv preprint arXiv:1411.1792*, 2014.
[39] N. Inkawhich, W. Wen, H. H. Li, and Y. Chen, "Feature space perturbations yield more transferable adversarial examples," in *IEEE CVPR*, 2019, pp. 7066–7074.
[40] D. A. Szafir, "Modeling color difference for visualization design," *IEEE TVCG*, vol. 24, no. 1, pp. 392–401, 2017.
[41] S. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *IEEE CVPR*, 2017, pp. 86–94.
[42] T. Goldstein, C. Studer, and R. Baraniuk, "A field guide to forward-backward splitting with a fasta implementation," *arXiv preprint arXiv:1411.3406*, 2014.
[43] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[44] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev *et al.*, "The kinetics human action video dataset," *arXiv preprint arXiv:1705.06950*, 2017.
[45] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.
[46] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *IEEE CVPR*, 2017, pp. 2462–2470.
[47] F. Tramer, N. Carlini, W. Brendel, and A. Madry, "On adaptive attacks to adversarial example defenses," in *NIPS*, 2020, pp. 1633–1645.
[48] A. Krizhevsky *et al.*, "Learning multiple layers of features from tiny images," 2009.
[49] A. Raghunathan, J. Steinhardt, and P. Liang, "Certified defenses against adversarial examples," *arXiv preprint arXiv:1801.09344*, 2018.
[50] J. Cohen, E. Rosenfeld, and Z. Kolter, "Certified adversarial robustness via randomized smoothing," in *ICML*, 2019, pp. 1310–1320.
[51] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "Ensemble adversarial training: Attacks and defenses," 2017.
[52] C. Zhang and Y. Ma, "Ensemble machine learning: methods and applications," 2012.
[53] A. Kumar, A. Levine, T. Goldstein, and S. Feizi, "Curse of dimensionality on randomized smoothing for certifiable robustness," in *ICML*, 2020, pp. 5458–5467.
[54] E. Wenger, J. Passananti, A. N. Bhagoji, Y. Yao, H. Zheng, and B. Y. Zhao, "Backdoor attacks against deep learning systems in the physical world," in *IEEE CVPR*, 2021, pp. 6206–6215.
[55] M. Shen, Z. Liao, L. Zhu, K. Xu, and X. Du, "Vla: A practical visible light-based attack on face recognition systems in physical world," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 3, no. 3, sep 2019.

Shangyu Xie received the B.Sc degree from Shanghai Jiao Tong University with dual major in Electrical Engineering and Information Engineering, affiliated with the IEEE Honor Class. He is currently a Ph.D student in the Department of Computer Science at Illinois Institute of Technology, USA. His research focuses on machine learning security and privacy.



Yan Yan is currently a Gladwin Development Chair Assistant Professor in the Department of Computer Science at Illinois Institute of Technology. He was an assistant professor at the Texas State University, a research fellow at the University of Michigan and the University of Trento. He received his Ph.D. in Computer Science at the University of Trento. His research interests include computer vision, machine learning and multimedia.

Yuan Hong (SM'18) received his Ph.D. degree in Information Technology from Rutgers University. He is currently an Assistant Professor in the Department of Computer Science at Illinois Institute of Technology. His research interests focus on data privacy, AI security, mechanism design and optimization. He is a recipient of the National Science Foundation (NSF) CAREER Award, and a Senior Member of the IEEE.