RoVISQ: Reduction of Video Service Quality via Adversarial Attacks on Deep Learning-based Video Compression

Jung-Woo Chang* Mojan Javaheiri† Seira Hidano† Farinaz Koushanfar*
*University of California San Diego †KDDI Research, Inc.
juc023, mojan, farinaz}@ucsd.edu, se-hidano@kddi.com

Abstract—Video compression plays a crucial role in video streaming and classification systems by maximizing the end-user quality of experience (QoE) at a given bandwidth budget. In this paper, we conduct the first systematic study for adversarial attacks on deep learning-based video compression and downstream classification systems. Our attack framework, dubbed RoVISQ, manipulates the Rate-Distortion (R-D) relationship of a video compression model to achieve one or both of the following goals: (1) increasing the network bandwidth, (2) degrading the video quality for end-users. We further devise new objectives for targeted and untargeted attacks to a downstream video classification service. Finally, we design an input-invariant perturbation that universally disrupts video compression and classification system in real time. Unlike previously proposed attacks on video classification, our adversarial perturbations are the first to withstand compression. We empirically show the resilience of RoVISQ attacks against various defenses, i.e., adversarial training, video denoising, and JPEG compression. Our extensive experimental results on various video datasets show RoVISQ attacks deteriorate peak signal-to-noise ratio by up to 5.6dB and the bit-rate by up to $2.4\times$ while achieving over 90% attack success rate on a downstream classifier. Our user study further demonstrates the effect of RoVISQ attacks on users’ QoE. We provide several example attacked videos used in our survey on https://sites.google.com/view/demo-of-rovisq/home.

I. INTRODUCTION

Video content accounts for more than 80% of the internet traffic [63]. Live video traffic has experienced an even higher growth with the advent of streaming services, e.g., remote vehicle control [45], remote surgery [231], and UHD streaming [3] [4]. In addition, recent developments in deep learning have given rise to live video analysis services such as activity recognition for health care diagnosis [24], video surveillance [62], and autonomous driving [31]. Video compression technologies are a key enabler for the aforesaid media streaming and content analysis applications. As shown in Figure 1(a), live video streaming and classification systems typically consist of four main components, i.e., front-end video sources (cameras), video encoder, video decoder, and back-end video subscribers. The video encoder and decoder compress the video for efficient real-time streaming to back-end subscribers. Recent literature proposes Deep Neural Network (DNN) based encoder and decoder for video compression [26], [44], [74]. Due to their enhanced performance over conventional codecs [60], [170], DNNs are now adopted in the next generation of video compression technologies by the Moving Picture Experts Group (MPEG) standardization group [1] and industry [2].

Signal transformation techniques, e.g., compression, have been trusted to remove the adversarial effect in image recognition [6].

Network and Distributed System Security (NDSS) Symposium 2023
27 February - 3 March 2023, San Diego, CA, USA
ISBN 1-891562-83-5
https://dx.doi.org/10.14722/ndss.2023.23165
www.ndss-symposium.org

Fig. 1: High-level view of (a) a live video streaming and classification system, and (b) RoVISQ attacks performed by injecting adversarial perturbations to the front-end video sources. To demonstrate the adversarial effect, we present the peak signal-to-noise ratio (PSNR), bit-rate, and an example outcome of video activity recognition.

We empirically show that the state-of-the-art adversarial instances created for video classification [36], [37], [69] can also be removed after video compression. Nevertheless, the robustness of the DNN-based video compression pipeline itself against adversarial perturbations remains largely unknown to date.

In this work, we present RoVISQ, Reduction of Video Service Quality, as a new threat to video compression. We show, for the first time, that DNN-based video compression systems can be exploited by adversaries to not only reduce the users’ quality of experience (QoE), but also attack downstream video recognition services. The success of RoVISQ attacks is indebted to solving a novel unified optimization problem. The solution crafts adversarial perturbations that simultaneously attack the video compression and the downstream video classification system as shown in Figure 1(b).

Proposed Attacks. The goal of video compression is to simultaneously minimize the number of bits used for encoding while reducing the distortion. Contemporary DNN-based video compression frameworks [26], [44], [74] typically employ a $R$-$D$ optimization [49], [61] that minimizes the distortion ($D$) at a given bit-rate ($R \leq R_T$) where $R_T$ is the available bit-rate budget. In this paper, we aim to break the $R$-$D$ model of a pre-trained video compression model.

We propose three novel attacks on video compression shown in Figure 2, namely a video quality attack, bandwidth attack, and RD attack. The video quality attack increases $D$ while keeping $R$ constant, whereas the bandwidth attack reduces compression rate ($R$) while maintaining $D$. Our RD attack simultaneously increases $R$ and $D$. The attacker can choose between aforesaid strategies based on the adversarial goal and characteristics of the video streaming service. We also propose a compression-robust classifier attack that causes a downstream video recognition service to misclassify all
actions in the input video. Notably, ours is the first attack on video classification that remains successful when the perturbed video goes through various signal transformations, i.e., video compression, denoising, and frame-by-frame JPEG compression.

RoVISQ attacks are applicable to both online and offline settings. In the offline scenario, videos are locally stored in a camera or smartphone and the attack latency is not constrained. For the challenging online scenario, we preprocess universal perturbations that can attack any video stream in real time. We assume the attacker has access to open-source video compression/classification models that can be used as a surrogate to train the universal perturbation.

**Design Challenges and Solutions.** We are faced with several new design challenges not present in prior works that attack only video classification [56], [57], [53], [69], [71]. Firstly, video compression systems, by design, are extremely robust to noise. State-of-the-art video compression schemes [26], [44], [74] group a series of frames into sequences called Group of Pictures (GOP) to allow back-end users to access live video streams at any time. The GOP structure inherently improves robustness against error propagation and sudden scene changes during compression, thereby making the adversarial attack more challenging. We address this problem by modeling the temporal coding structure of the GOP, which is then incorporated in our attack formulation. We validate the effectiveness of our attacks on three types of GOP structures [26], [44], [74].

Secondly, previous attacks [30], [37], [53], [69], [71] on video recognition do not consider video compression in their threat models. Recent work shows that conventional compression algorithms can be used as a defense against adversarial attacks to classification [6], [29], [42], [54]. Specifically, lossy compression with quantization reduces the spatiotemporal data, thereby easily removing the adversarial perturbations. To provide a compression-resistant perturbation for video recognition, we consider the video compression in our adversarial attack. In the video compression pipeline, the reconstruction error is propagated to the next frame during encoding. Consequently, we design a new model to train the perturbations by using the information from different time steps and combining all information to break the original $R$-$D$ optimization.

Finally, the adversary does not have prior knowledge about the content of the streaming video. While launching the online attacks, the perturbation should be effectively injected at any time in the video encoder. Additionally, different video compression frameworks support various encoding parameters, i.e., compression ratio and GOP structure [26], [44], [74]. However, the real time constraints of the online attack prohibit the adversary from customizing the injected noise for the underlying encoding parameters. To tackle the above challenges, we generate a universal adversarial perturbation that is trained via a novel offline process. Specifically, our training adjusts the universal perturbation to be applicable to various input videos, encoding parameters, and video compression/classification models. We further utilize a temporal transformation function during training which enables us to perform the attack in any order of the video sequence. Notably, our universal perturbations are not only effective for the surrogate video compression/classification models that they are trained on (white-box attack), but they can also degrade the performance of unseen video compression/classification models (black-box attack).

In summary, our key contributions are as follows:

- We perform the first systematic study of adversarial attacks on deep learning-based video compression. We further maliciously exploit the video compression pipeline to direct our attacks towards downstream video recognition systems.
- We propose four new adversarial attacks that result in high-impact security and QoE consequences. We formalize RoVISQ attacks as well-defined optimization problems that can be solved to obtain perturbations affecting the $R$, $D$, and downstream classification.
- We construct a well-designed universal perturbation that is invariant to the underlying DNN model, encoding parameters, and input videos. Our universal perturbation is reusable for various video encoders and can be injected at any time during live video streaming and classification. This enables us to conduct an online attack in real time under strict latency constraints.
- Our adversarial attacks are the first to withstand the inherent denoising performed during video compression. Additionally, we show the resiliency of RoVISQ attacks against various defenses namely, adversarial training, video denoising, and JPEG compression.

**II. Preliminaries**

Figure 3 illustrates the commonly used modules in state-of-the-art DNN-based video compression and classification frameworks. Video compression employs temporal prediction to minimize the difference between the previously decoded frame and the current frame. The encoder sends bitstreams that conform to a specific channel standard to the decoder. The compressed video is then reconstructed from the bitstreams by the decoder. State-of-the-art video compression models can be grouped into two classes, namely, pixel-space and feature-space frameworks. Pixel-space video compression [44], [74] resembles traditional video codecs [60], [70]. More recently, feature-space video compression [26] has been proposed which includes additional modules for conversion to the feature-space (shown with dashed border in Figure 3). Our adversarial perturbations are applicable to both classes of video compression models. Note that our perturbations are applied to the input of the video compression system and are generated while targeting all internal modules shown in Figure 3, e.g., motion estimation.

We further include an optional video classifier as the target downstream task after compression. This module predicts the activities performed in the video and is commonly deployed in video analysis services. The downstream video classification service receives as input the decoded frames from the video compression pipeline. In what follows, we explain the internal mechanisms of the video compression modules shown in Figure 3. Throughout the paper, we use “video compression” and “video coding” interchangeably. All DNN-based modules in Figure 3 are commonly used in SOTA video-compression.

**Temporal Coding Structure.** Since back-end viewers can join the live video stream at random points, the bitstreams are transmitted in...
The quality of the decoded frame can be further improved through entropy coding [7] and sent to the decoder. View of the GOP structures in our paper. A GOP contains different post-processing methods proposed in [26], [74].

The reconstructed residual frame are added to create the decoded frames and the reconstructed motion map. The predicted frame and the reconstructed residual amount is compressed using a non-linear residual encoder. Finally, the quantized latent representations from the motion network is employed to compress the motion vectors with a high compression rate. The encoder also includes a DNN-based motion compensation module that efficiently quantizes the redundancy in the latent space. Specifically, the P frame references most of the GOP, are constructed using video compression to remove temporal redundancy. Specifically, the P frame references the previous picture for the prediction, whereas the B frame uses both previous and forward frames to achieve the highest compression rate.

**Video Encoder** uses a DNN-based motion estimation algorithm to measure the movement between the current frame and the previous reconstructed frame, dubbed the motion map. An auto-encoder network is employed to compress the motion vectors with a high compression rate. The encoder also includes a DNN-based motion compensation module which predicts the current frame using the reconstructed motion map and previous frames. The residual frame is then computed using the current frame and the prediction. The residual amount is compressed using a non-linear residual encoder network that efficiently quantizes the redundancy in the latent space. Finally, the quantized latent representations from the motion compression and residual compression are converted to bitstreams via entropy coding [7] and sent to the decoder.

**Video Decoder.** After receiving the bitstreams from the video encoder, the video decoder recovers the latent representations using entropy decoding. The decoder then applies motion and residual decoders to the latent representations and obtains the reconstructed motion map and residual frame. Similar to the encoder side, the decoder includes a DNN-based motion compensation module that predicts the current frame using information from the previous frames and the reconstructed motion map. The predicted frame and the reconstructed residual frame are added to create the decoded frame, which is then stored in a buffer. The decoder is thus able to reconstruct each frame with a same visual quality as the encoder. The quality of the decoded frame can be further improved through post-processing methods proposed in [26], [74].

### III. Threat Model

**A. Attack Scenarios**

RoVISQ attacks are targeted towards video data generated by front-end sources (e.g., smartphones and surveillance cameras) that is sent to back-end user(s) through a video compression pipeline. Since prior works on adversarial video [37], [53], [69], [71] do not consider the role of video compression, their perturbations are mostly eliminated once the video is compressed. As such, the effect of previously proposed attacks on video classification will in fact not reach the target back-end user. To address this problem, our proposed attacks modify the video data at the first stage in the entire video compression and classification systems as shown in Figure 1-(b).

RoVISQ attacks are applicable to autonomous vehicles [45], remote surgery [5], UHD streaming [68], and virtual reality [75], where the adversary invokes low-quality or denial-of-service (DoS), e.g., as a competitor or malicious entity. Ideally, the attacker’s goal is to subvert the video streaming system over a long period of time. Therefore, our formulation ensures that RoVISQ perturbations applied to the input video are imperceptible to naked eye as visualized in Appendix Figure 12. Therefore, our attacks to the input video remain stealthy, making it hard for the streaming service to identify the source of distortion for the users. Our threat model follows the state-of-the-art in image/video adversarial attack literature [10], [11], [14], [36], [37], [69], [71]; we assume the attacker uses malware or man-in-the-middle schemes to carry out the attack. We consider both online and offline scenarios.

The online attack directly perturbs the front-end sources as the video is generated in real time. For this scenario, we design well-crafted universal perturbations that can be used to attack any given video sequence at any time step. Since our universal perturbations are agnostic to the content of the video, the attacker does not require any knowledge about the streaming data. They can therefore carry out the attack by injecting RoVISQ’s pre-trained perturbations using man-in-the-middle methods [71], e.g., by running Ettercap with ARP poisoning [51]. Specifically, as noted in [71], a large number of video surveillance systems include unencrypted camera-server communications. Thus, the attacker can capture the stream from the camera network and modify it using our online perturbations before sending it to the media server and subsequently the back-end users. Alternatively, an attacker may alter the streaming video at the application level using malicious software installed during
We consider the following two offline attack scenarios: 1) the adversary accesses the original raw data and perturbs the video, or denial-of-service. If the media server lowers the video resolution to reduce network traffic, the RD attack is further exacerbated. This is due to the observation made by [50], which show that distortion is more perceptually noticeable at a reduced video resolution.

Compression-Robust Classifier Attack. Our final attack manipulates the classification result of the decoded video in scenarios where the downstream task uses a DNN-based video recognition. This attack is particularly challenging as the video coding framework inherently invalidates most adversarial examples using DNN-based temporal coding. As such, we carefully craft an optimization problem that generates perturbations that are robust to video coding. We propose two variants of the classifier attack, i.e., targeted and untargeted. Targeted attacks misguide the video classifier to a particular class while untargeted attacks subvert the models to predict any of the incorrect classes.

C. Adversary’s Capability and Knowledge

Video coding standards are jointly determined by various standardization organizations, e.g., the ISO/IEC MPEG, ITU-T Video Coding Experts Group (VCEG), and industry partners (Google, Cisco, Netflix, Qualcomm, etc.) [1], [2], [15], [50]. To continue the development of the latest video coding standards, the reference software algorithms are made publicly available and are accompanied by several detailed specification documents [55] to ensure ease of use by clients. These open-source software for video compression are directly integrated into research and commercial products. Thus, we assume the attacker has white-box access to an open-source video compression model, i.e., the adversary knows the DNN architecture and weights. We also assume white-box access to an open-source video classifier as explored in prior works [37], [53], [69]. These white-box systems are leveraged by the attacker as a surrogate model to create the universal perturbations. The universal perturbations can then be applied to unseen models in a black-box attack scenario, i.e., when the adversary does not know the internals of the underlying DNN models in a video compression/classification system.

To construct our online attack via universal perturbations, we assume the attacker also has access to a set of public (benign) videos. As shown in our experiments, even though the training and test datasets for the universal perturbations have different distributions, our attacks are still applicable to the test data. Therefore, the attacker can use any open-source video dataset which is easily accessible.

IV. ATTACK CONSTRUCTION

This section explains RoVISQ attacks on video compression and classification for both online and offline scenarios. We first formulate our attacks as well-defined optimization problems following the threat model. We then adapt our formulations to enable offline and online attacks.
We denote the resulting adversarial video by 

\[ \bar{x}_t = x_t + \delta_t \]

where \( \delta_t \) represents the weights parameters of DNN, \( w \mid \delta_t \) use different coding orders according to the underlying GOP structure. Each frame in the coding order is mapped to one time step in the input video based on the type of GOP structure \( k \in \{ \) non-hierarchical, hierarchical-P, hierarchical-B \}. We note that this time step, denoted by \( t' \), may not be equivalent to the display order \( t \), i.e., the order of frames in which the video is captured. For a given \( k \), the \( n \)-th coding order in the \( g \)-th GOP is mapped to time step \( t' \) using a deterministic function \( m_k(g,n,t') \). Here, \( 1 \leq n \leq G \) and \( 0 \leq g \leq \lfloor \frac{T}{T_0} \rfloor \) where \( G \) denotes the number of frames in the GOP.

Accordingly, we formulate the video encoder as a function \( E_k(\bar{x}_{t'},P_{t'},\lambda) = b_t \) that accepts an input \( \bar{x}_{t'} \) and produces an output bitstream \( b_t \). The conversion is performed using several reference frames stored in the decoded frames’ buffer:

\[ P_{t'} = \{ y_{m_k(g,1)}, \ldots, y_{m_k(g,n-1)} \} \]

where \( P_{t'} = \emptyset \) when \( n = 1 \). Here, \( \lambda \) controls the trade-off between the bit-rate (\( R(\cdot) \)) and the distortion (\( D(\cdot) \)). The video decoder is formulated as a function \( D_k(b_t,P_{t'},\lambda) = y_{t'} \) that accepts the encoded bitstream \( b_t \) and reconstructs the decoded frame \( y_{t'} \).

The end-to-end video compression model \( [26, 44, 74] \) is trained to solve the following optimization problem over the \( t' \)-th frame \( x_{t'} \):

\[ \min_{b_{t'}} R(b_{t'}) + \lambda \cdot D(x_{t'},y_{t'}), \]

where \( w \) represents the weights parameters of DNN, \( R(\cdot) \) indicates the number of bits allocated for the \( t' \)-th frame, and \( D(\cdot) \) measures the distortion between the input frame \( x_{t'} \) and its reconstruction \( y_{t'} \). The distortion \( D(\cdot) \) is often measured using the mean squared error (MSE) \( [26, 44, 74] \).

Adversarial Attacks on Video Compression. Let \( \Delta = \{ \delta_1, \ldots, \delta_T \} \in \mathbb{R}^{T \times W \times H \times C} \) denote a perturbation set for a given video \( X \). We denote the resulting adversarial video by \( X = \{ \bar{x}_1, \ldots, \bar{x}_T \} \in \mathbb{R}^{T \times W \times H \times C} \), which consists of adversarial frames \( \bar{x}_t = x_t + \delta_t \).

Upon receiving the adversarial input \( \bar{x}_{t'} \), the encoder outputs an adversarial bitstream \( b_{t'} \) using perturbed reference frames stored in the adversarial decoded frames’ buffer \( P_{t'} = \{ y_{m_k(g,1)}, \ldots, y_{m_k(g,n-1)} \} \). Finally, the decoder reconstructs the adversarial decoded frame \( \bar{y}_{t'} \) from the perturbed bitstream \( b_{t'} \).

Figure 5 demonstrates RoVISQ attacks to the video compression modules, along with the corresponding inputs, outputs, and the adversarial perturbation.

Recall the coding of each frame \( \bar{x}_{t'} \) is conditioned upon the coding of its preceding frame stored in the (adversarial) decoded frames’ buffer \( P_{t'} \). This temporal coding constructs a chain of dependency between all adjacent frames in each GOP. To account for the inter-frame dependency (IFD) within a video sequence, our proposed adversary focuses on the coding performance of one GOP unit. Specifically, we construct our attacks based on the following adversarial R-D model considering the IFD in for the \( g \)-th GOP:

\[ \max_{\Delta_g} \frac{1}{G} \sum_{t' = m_k(g,1)}^{m_k(g,G)} (R(\bar{y}_{t'}) + \lambda \cdot D(x_{t'},\bar{y}_{t'})) \]

where \( \Delta_g \in \mathbb{R}^{G \times W \times H \times C} \) is the perturbation for the \( g \)-th GOP.

We quantify the video compression performance based on two important QoE factors for back-end subscribers: \( \{ Q_0, Q_1 \} \). Here, \( Q_0 \) denotes the end-users’ expectations for network bandwidth and \( Q_1 \) represents the expectation for video distortion. Using the perturbation in Equation 2, we estimate the QoE factors at intervals of GOP size \( G \) as follows:

\[ Q_0(\bar{B}_g) = \frac{1}{G} \sum_{b_t \in \bar{B}_g} R(\bar{b}_t) \]

\[ Q_1(X_g,\bar{Y}_g) = \frac{1}{G} \sum_{y_t \in \bar{Y}_g} D(x_{t'},\bar{y}_{t'}), \]

where \( X_g, \bar{B}_g, \bar{Y}_g \) denote the set of input frames, adversarial bitstreams, and (adversarial) reconstructed frames over the time steps in the \( g \)-th GOP, i.e., \( t' \in [m_k(g,1), m_k(g,G)] \). We formulate RoVISQ attacks to manipulate the above QoE factors. Note that our definition of \( Q_0, Q_1 \) captures the IFD through an implicit conditioning on the GOP temporal coding function \( m_k(\cdot) \). As such, our perturbations can adapt to any given GOP structure without need for reformulating the attack. Below we explain how each of our attacks are constructed.

1- Video Quality Attack. The adversarial perturbation \( \Delta_g \) is determined by solving the following optimization problem:

\[ \max_{\Delta_g} Q_1(X_g,\bar{Y}_g) \quad \text{s.t.} \quad E_0 < \epsilon_0. \]

Here, \( E_0 = \| Q_0(\bar{B}_g) - Q_0(B_g) \|_2 \) is the difference between the expected bit-rate of the clean and perturbed videos.

2- Bandwidth Attack. The objective of the bandwidth attack is to find \( \Delta_g \) that maximizes the bit-rate while keeping the video distortion unaltered as follows:

\[ \max_{\Delta_g} Q_0(\bar{B}_g) \quad \text{s.t.} \quad E_1 < \epsilon_1, \]

where \( E_1 = \| Q_1(X_g,\bar{Y}_g) - Q_1(X_g,Y_g) \|_2 \) is the difference between the expected distortion of clean and perturbed videos. \( Y_g \) is the set of clean reconstructed frames in the \( g \)-th GOP.

3- RD Attack. \( \Delta_g \) is found by solving the following optimization problem that affects both QoE factors at the same time:

\[ \max_{\Delta_g} Q_0(\bar{B}_g) + \lambda \cdot Q_1(X_g,\bar{Y}_g). \]

**Compression-Robust Classifier Attacks.** Figure 5 illustrates a video classification module appended to the video compression...
pipeline. Video classification relies on a discriminant function \( F(Y) \) that takes as input a perturbed decoded video clip \( \tilde{Y} = \{y_1, \ldots, y_T\} \in \mathbb{R}^{T \times W \times H \times C} \) and outputs a probability distribution over a set \( K \) of class labels. \( F_c(\tilde{Y}) \) indicates the probability of the input video belonging to a specific class \( c \in K \). The video classifier \( C \) maps an input \( \tilde{Y} \) to the class with the maximum probability: \( C(\tilde{Y}) = \arg\max_{c \in K} F_c(\tilde{Y}) \). The untargeted attack against \( C \) generates perturbations such that \( C(\tilde{Y}) \neq C(Y) \), whereas the targeted attack aims at \( C(\tilde{Y}) = c^* (\neq C(Y)) \) for a predetermined incorrect class \( c^* \in K \), where \( Y \) is the clean decoded video clip. The adversarial loss \( L_{adv} \) for the untargeted and targeted attacks can be written as:

\[
L_{adv} = \begin{cases} 
F_c(Y) - \max_{c \neq C(Y)} F_c(\tilde{Y}) & \text{(Untargeted)} \\
\max_{c \neq c^*} (F_c(\tilde{Y}) - F_{c^*}(\tilde{Y})) & \text{(Targeted)}
\end{cases}
\]  

(8)

where \( \tilde{Y} \) is the perturbed decoded video. Both attacks gain success if and only if \( L_{adv} < 0 \). To create versatile perturbations that are robust to compression, we formulate RoVISQ attacks against video classification with consideration for both video compression and classification. Specifically, the adversarial perturbation \( \Delta_g \) is computed using the same formula in Equations 5\cite{5}, 6\cite{6} with a new constraint added for \( L_{adv} < 0 \).

### B. Offline Attack Methodology

In the offline attack scenario, injecting the adversarial perturbations is not latency bound. The attacker can therefore use the entire captured video, before creating and adding the adversarial perturbations. To generate the perturbations, the adversary leverages the proposed approximations of QoE to maximize the following adversarial loss function:

\[
\max_{\Delta_g} L_{comp}(g) \text{ s.t. } \|\Delta_g\|_{\infty} \leq \epsilon_c
\]

\[
L_{comp}(g) = \begin{cases} 
E_0 + \lambda \cdot Q_1(X_g, \tilde{Y}_g) & \text{if } \xi = 0 \\
Q_0(B_g) + \lambda \cdot E_1 & \text{if } \xi = 1 \\
Q_0(B_g) + \lambda \cdot Q_1(X_g, \tilde{Y}_g) & \text{if } \xi = 2,
\end{cases}
\]  

(9)

where \( \xi \) determines the attack type: \( \xi = 0 \) presents the video quality attack, \( \xi = 1 \) is the bandwidth attack, and \( \xi = 2 \) is the RD attack. To ensure the injected noise is imperceptible to humans, the norm of the perturbation is upper bounded by a pre-defined small value \( \epsilon_c \) for all attacks. We use the iterative FGSM (I-FGSM)\cite{35} method to solve Equation 9 as follows:

\[
\hat{X}_{n+1} = \text{clip}_{\rho}(\hat{X}_n + \frac{\epsilon_c}{\rho} \text{sign}(\nabla L_{comp}(g)))
\]

(10)

where \( \rho \) is the total number of iterations. Here, \( \hat{X}_0 \) is equal to the original (benign) video \( X_g \) and the final perturbed video \( \hat{X}_n \) will be equal to \( \hat{X}_\rho \) from the last iteration of Equation 10.

Equation 7 delivers the perturbations for downgrading the robustness (QoE) of the video compression system itself. The adversary can optionally choose to repurpose the RoVISQ attacks to also affect a downstream video recognition system. In such scenarios, we integrate \( L_{adv} \), to simultaneously derive perturbations on video compression and classification. Therefore, we convert Equation 7 to the following objective function:

\[
\max_{\Delta} L_{total} = \frac{1}{T/G} \sum_{g=0}^{T/G} L_{comp}(g) - \beta \cdot L_{adv},
\]  

(11)

where \( \beta \) adjusts the scale of the two loss functions. We use a similar method as Equation 10 to solve the above maximization problem. Specifically, we replace the loss function with \( L_{total} \) to generate perturbations for both compression and classification models.

### C. Online Attack Methodology

Online adversarial attack is particularly challenging for several reasons. Firstly, the perturbed decoded frames from video encoder and decoder are conditioned on \( \lambda \) which controls the compression rate. Since \( \lambda \) is determined by the end-user’s network traffic, it is difficult for the adversary to infer the value of \( \lambda \) in real time. Secondly, the display and coding orders vary according to the GOP structure set for the victim video compression. Even if the adversary knows the GOP structure used by the system, they can not predict the order of the frames in the live video stream. As such, computing the QoE factors in Equations 3\cite{3}, 4\cite{4} are not possible in the online attack. Finally, the adversary may not be able to align the image perturbations with the video sequence, which reduces the attack performance due to a boundary effect\cite{37}, \cite{71}. If not addressed properly, the said challenges hinder that attacker’s success and degrade the functionality of the adversarial perturbation.

We thus propose a novel attack that is universally applicable to live video streams using pre-trained perturbations that are generated offline. Our universal perturbations are agnostic to compression ratio, GOP structure, and input, which is suitable for online attack. Algorithm 1 outlines the training phase of our online attack. In crafting the universal perturbation, we adapt the perturbation to all variations of compression ratio and GOP structures (\( G = \{\text{non-hierarchical}, \text{hierarchical-P}, \text{hierarchical-B}\} \)). These variations are captured in a universal loss \( L_{univ} \), which is used to derive the final perturbation. Additionally, we address the boundary effect using a temporal transformation function \( \Gamma(\cdot) \) for the 3D perturbation with shifting variable \( \tau \)\cite{37}, \cite{71}.

Let us denote the universal adversarial perturbation for a GOP by \( \Phi = \{\phi_1, \phi_2, \ldots, \phi_G\} \in \mathbb{R}^{G \times W \times H \times C} \). We define a permutation function \( \Gamma(\Phi, \tau) \) which produces a cyclic temporal shift of the original perturbation \( \Phi \) by an offset \( \tau \in [0,G-1] \):

\[
\Gamma(\Phi, \tau) = \{\phi_{1, \tau} \mod G, \phi_{2, \tau} \mod G, \ldots, \phi_{G, \tau} \mod G\}
\]  

\[
= \{\phi_{(1+\tau) \mod G}, \phi_{(2+\tau) \mod G}, \ldots, \phi_{(G-1+\tau) \mod G}, \phi_{\tau}\}.
\]

Once the temporal shift is applied to the perturbation, the adversarial video of the \( g \)-th GOP can be computed as \( \hat{X}_g = \{x_1 + \phi_{1, \tau}, \ldots, x_G + \phi_{G, \tau}\} \). Note that \( \hat{X}_g \) receives all \( G \) possible temporal shift transformations during training with a high probability, therefore ensuring the generalizability of \( \Phi \). Instead of applying the adversarial perturbation to one video clip, we obtain the universal perturbation by averaging the value across all training videos available to the attacker. When computing the adversarial loss, the adversary can set \( \beta \) to zero in Equation 11 to limit the scope of the attack to the compression model only, or a non-zero value to attack a downstream video classification system. In the latter scenario, we consider the case where \( L_{adv} \) is untargeted. We iteratively update \( \Phi \) with the I-FGSM method as shown in Equation 10.

We note that prior works on video recognition attacks\cite{37}, \cite{71} do not consider video compression in their threat model for the universal attack. Therefore, they have the disadvantage compared to our formulation that the adversarial perturbation for video recognition is not robust to video compression. Their attack performance against video classifiers drops dramatically as the compression rate increases,
as we show in our evaluations (see Figure 9). To break down video compression, our perturbations are learned in consideration of the inter-frame dependencies and error propagation.

V. ATTACK EVALUATION

We first evaluate RoVISQ attacks on video compression in Section V-B. We further provide a user study in Section V-C to investigate the effect of RoVISQ attacks on user QoE. Finally, in Section V-D, we evaluate our compression-robust attack against a downstream video classifier by considering the video compression and classification modules in one framework.

A. Experimental Setup

RoVISQ Attacks on Video Compression. We use the Vimeo-90K dataset [72] to train the video compression models. This dataset contains 89,800 video clips with 7 frames each, with a resolution of 448 × 256. We crop the video sequences into a resolution of 256 × 256 before training. As our victim video compression module, we consider three state-of-the-art models with different structures, namely, DVC [44], HLVC [74], and FVC [26]. We implement each model through an open source framework [25] for video compression. We train the benign models with different λ values (λ = 256, 512, 1024, 2048) using the loss function in Equation 11.

To generate the adversarial perturbations for the offline and online scenarios, we use the loss function in Equation 11 and the pseudo code in Algorithm 1 with β = 0. We set I-FGSM iterations to 50 and εc to 102, following previous studies [37], [53], [69], [71]. For the offline attack, we assume the adversary adds perturbations to the raw video before compression is performed (offline attack scenario 1 in Section III-A). For the online attack, we use the Vimeo-90K dataset to train our universal perturbations. To evaluate the attacks, the video sequences in the HEVC datasets [60], i.e., classes B, C, and D are used. HEVC class B is a high resolution (1920 × 1080) dataset while classes C and D have lower resolutions of 834 × 480 and 416 × 240, respectively. Following prior work [20], [43], [74], we test the HEVC datasets on the first 100 frames and set G to 10 for non-hierarchical and hierarchical-P GOPs. For hierarchical-B GOP, we set G to 11, and the last frame of each GOP is reused as the first frame of the next GOP. Video quality is measured as peak signal-to-noise ratio (PSNR) and the bit-rate is calculated by bits per pixel (Bpp).

Baselines on Video Compression. We compare RoVISQ attacks on video compression models with adaptive Gaussian noise: ϵg ∼ N(0, σ2). Specifically, since the I frame strongly affects P and B frames in the GOP [28], we evaluate two scenarios, where the I frame either receives higher noise or equal, compared to the P and B frames. In summary, we use two types of Gaussian noises: (1) Case I (σI = σP = σB = εc), (2) Case II (σI = 2 · εc, σP = σB = εc). We also include the traditional video coding methods, H.264 [70], H.265 [69], as our baseline and show that RoVISQ attacks also degrade the performance of legacy methods. We follow the command line in [44] and use FFmpeg with default mode to implement H.264 and H.265.

RoVISQ Attacks on Video Classification. We benchmark three state-of-the-art video classification models including 13D [13], SlowFast [21], and TPN [75]. We leverage the DVC framework as the backbone for video compression. We assume the attacker decodes a video that has already been compressed with DVC, then injects perturbation and re-encodes it (offline attack scenario 2 in Section III-A). Our attacks are evaluated on the human action recognition dataset UCF-101 [59] and the hand gesture recognition dataset 20BN-JESTER (Jester) [47]. UCF-101 includes 13320 videos from 101 human action categories (e.g., diving, biking, blow-drying hair, yo yo). Jester includes 27 gestures recorded by crowd-sourced workers (e.g., sliding hand left, sliding two fingers left, zooming in with full hand, zooming out with full hand).

To generate the offline and online adversarial perturbations, we set β = 0.1 and εc = 102 [37], [53], [69], [71] in Equation 11 and Algorithm 1. We set MaxIter to 50 steps for I-FGSM. Following the evaluation setup in prior work [56], we randomly pick the target class for our targeted attacks. For assessing both targeted and untargeted attacks, we randomly chose one and four videos from each human action category for UCF-101 and Jester datasets, respectively.

Baselines on Video Classification. We compare the performance of RoVISQ attacks with the following state-of-the-art adversarial attacks on video recognition: SAP [69], GEO-TRAP [36], C-DUP [37], and U3D [71]. Similar to the threat model for our attacks, we insert the perturbations created by prior work to the inputs of the video compression and classification. The perturbed videos are then compressed and reconstructed by video compression. Finally, they are fed to the victim classifiers. We define the attack success rate as the portion of misclassified test samples for the untargeted attack. For the targeted scenario, the attack success rate is the portion of samples mapped to the adversary’s desired class.

B. RoVISQ Attacks on Video Compression

In this Section, we analyze the effect of our proposed attacks on the performance of video compression. Specifically, we analyze two attack scenarios, namely white-box and black-box settings. In the white-box attack scenario, the victim video compression model is open-source and the attacker has complete knowledge of the underlying DNNs used. Figure 10 demonstrates the PSNR-based R-D performance evaluated on the HEVC standard test sequences (class B, class C). The RoVISQ attacks enclosed in this Figure follow the white-box threat model explained above. Each plot corresponds...
to a video compression model trained with various values of $\lambda \in \{256, 512, 1024, 2048\}$. As $\lambda$ decreases, the bit-rate of the video compression model improves but the video quality deteriorates. We report the results for our video quality, bandwidth, and RD attacks in both offline and online settings. Intuitively, the online attacks are slightly less powerful than the image-specific offline attacks. It is worth noting that even though the training dataset used for generating the universal perturbation is entirely different from the dataset on which it is tested on, the online attack remains highly effective.

We further study the black-box attack scenario, i.e., when the attacker does not know the architecture or weights of the DNN used for video compression. In this setting, we show that our universal perturbations trained on a surrogate open-source video compression model are transferable to unseen systems. Table I encloses the $R$-$D$ performance of the video compression systems before and after applying the black-box online attack. Below we analyze the effects of our various attacks in white-box and black-box scenarios in detail.

**Analysis of the Video Quality Attack.** As shown in Figure 6-left, RoVISQ attacks can break the $R$-$D$ model of pre-trained video encoders by a large margin. Specifically, our attacks lower the video quality by up to 5.43dB when $\lambda = 2048$ while maintaining the Bpp level. Our attack performance improves when $\lambda$ is bigger. This is because the bit-rate increases with $\lambda$ and therefore it is less likely for the perturbations to be removed during compression. Figure 6 also shows that video compression is vulnerable to our universal online attack. When applying the online perturbation in the white-box scenario, the PSNR drops by 2.30-3.24dB. When applying the universal perturbation to unseen victim models, i.e., in the black-box setting, the average PSNR drop is 2.41dB which is only 0.41dB lower than the average PSNR drop of the white-box attack. This shows that RoVISQ online perturbations can be used to effectively attack any arbitrary video compression model.

| Analyze of the Bandwidth Attack.** Figure 6-middle demonstrates the performance of our proposed bandwidth attack in the white-box setting. As seen, our adversarial perturbations successfully reduce the compression rate while maintaining the PSNR level. Table I: PSNR (dB) and Bpp of victim video compression models on the HEVC class C dataset after applying RoVISQ online perturbations in the black-box attack setting. Reported values are averaged across different encoding parameters $\lambda \in \{256, 512, 1024, 2048\}$. Architectural details of the evaluated models are enclosed in the Appendix Table X. |
TABLE II: PSNR and Bpp of conventional compression methods after applying RoVISQ universal perturbations, evaluated on the HEVC class C dataset.

<table>
<thead>
<tr>
<th>Video Quality Attack</th>
<th>Bandwidth Attack</th>
<th>RD Attack</th>
<th>Gaussian Noise</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PSNR (dB)</strong></td>
<td><strong>Bpp</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H.265</td>
<td>-3.47</td>
<td>-1.55</td>
<td>-3.62</td>
</tr>
<tr>
<td>H.264</td>
<td>-3.19</td>
<td>-1.03</td>
<td>-3.48</td>
</tr>
<tr>
<td><strong>Bpp</strong></td>
<td>+54.5%</td>
<td>+78.4%</td>
<td>+73.8%</td>
</tr>
<tr>
<td>H.265</td>
<td>+34.7%</td>
<td>+65.2%</td>
<td>+61.8%</td>
</tr>
</tbody>
</table>

models to hide the attack. Gaussian noise deviates a lot from the R-D curve of the original models and also increases the Bpp level much less than our bandwidth attack. In order to maintain D, the perturbed video should not be much different from before. However, Gaussian noise differs by up to 2.39dB from the original video quality, which is considerable. When compared with the benign compression models, our proposed offline bandwidth attack dramatically increases the Bpp level by up to 2.12 ×, 2.03 ×, and 2.37 × on the HEVC class B, C, and D while reducing the PSNR by only 0.01dB. Similarly, our white-box online attack increases the Bpp level by up to 1.61 × while maintaining the PSNR difference within 0.55dB. Our online attack increases the Bpp of unseen models by up to 1.52 ×, which is on average only 5.6% lower than the Bpp of the white-box scenario.

Analysis of the RD Attack. Figure 6 right shows the performance of our white-box RD attack. We confirm that RD attack can achieve all the objectives of the video quality and bandwidth attacks. When compared with the video quality attack, RD attack lowers the PSNR by 0.44dB more, on average. When compared to the bandwidth attack, the increase in the Bpp is on average 6.3% lower. This is due to the inherent trade-off between the bit-rate and distortion in the perturbation loss (Equation 1), which is controlled by λ; a higher value of λ shifts the optimization towards lower PSNR, thereby resulting in a higher Bpp compared to the bandwidth attack. When applying our online RD perturbations to unseen models (black-box attack), the average PSNR drop and Bpp increase are -2.48dB and 30.2%, compared to the -3.41dB and 37.5% achieved by the white-box online attack.

Comparison with Baselines. As shown in Figure 6, our proposed perturbations outperform the baselines by a large margin, decreasing the PSNR by up to 4.52dB, or increasing the BPP by 2.12 ×. Without considering the QoE factors in the optimization problem, the evaluated baselines cannot critically hurt video compression. Especially, while Gaussian Noise Case II outperforms case I, it contains twice the noise level compared to our perturbation in the I frames. Nevertheless, our perturbations achieve a much higher attack performance by leveraging the QoE factors of video compression.

Attack on Conventional Compression Methods. We were curious to know whether our trained universal perturbations are also effective against conventional, non-DNN-based, compression methods, i.e., H.264 [70] and H.265 [60]. Therefore, in addition to DNN-based video compression models, we show that our online perturbations can also be used for black-box attack on traditional video compression methods, i.e., H.264 [70] and H.265 [60].

As shown in Table II, the attack performance decreases for conventional video-compression methods compared to DNN-based systems evaluated in Figure 6. However, the attack performance still outperforms the noise baseline by a large margin. It can be concluded that RoVISQ perturbations automatically learn to target low-level and intrinsic properties of video compression which are common between DNN-based and conventional methods. Our perturbations are thus transferable and can downgrade video compression performance even when it is not performed by a DNN.

C. Effect of RoVISQ Attacks on User Experience

Attack Visualization. Figure 7 shows an example of RoVISQ attacks against DVC video compression. As seen, the performance of the video coder is significantly degraded in terms of either PSNR or Bpp, after compressing the perturbed video. Specifically, the benign compressed video and the adversarial video perturbed with the bandwidth attack show similar image quality after compression, although the compression ratio differs by 1.8 ×. We can also see that the video quality attack keeps the Bpp constant but adds several noise-like artifacts to the compressed video that degrade the visual quality. RD attack generates more distortion in the decoded frame compared to the video quality attack while requiring a higher Bpp (although lower than the bit-rate achieved by the bandwidth attack). We note that the disturbance observed in the video is not uniform for all frames: the disturbance is higher for frames which have a low inter-frame correlation, such as those visualized in Figure 7(b,d). It is inherently difficult to perform spatiotemporal predictions on dynamic video sequences containing fast moving objects or sudden scene changes [76]. Due to inaccurate motion compensation, the magnitude of the generated residual is large and the contextual information of the residual frame is largely lost after video compression. When our perturbation is added to such dynamic videos, the quality of the decoded frame is further deteriorated due to error propagation and accumulation of the decoded error.

User Study. We conduct a public survey to study the effect of our attacks on users’ QoE. We recruited 55 volunteers aged between
You Tube, on a daily basis. Each user is shown video sequences in we can see a clear shift in the distribution of scores to higher values As seen, participants assign low noise scores to benign videos, with way. To avoid any bias in the gathered answers, the participants ((https://sites.google.com/view/demo-of-rovisq/home)).

The portion of scores equal to (more visible noise and lower video quality). For the video quality ((https://sites.google.com/view/demo-of-rovisq/home)).

in the majority of answers validates that our proposed attacks success-

itive since RD attack drops the PSNR by a larger margin compared to

fully degrade users’ QoE, leading to denial or interruption of service.

20-40 from different backgrounds, i.e., college students, graduate students, postdoctoral researchers, and industry professionals. We chose the background and age range such that the target demographic has relatively high exposure to various video streaming services, e.g., YouTube, on a daily basis. Each user is shown video sequences in HEVC class B and class C. Each set contains three video clips: 1) the original benign video, 2) the video with RoVISQ’s video quality attack, and 3) the video attacked using RoVISQ’s RD perturbations.

The participants were asked to rate the noise level for each video clip based on a Likert scale from 1 to 5 where a score of 1 corresponds to no perceptible noise and a score of 5 shows highly perceptible noise. As clarified in the survey, a score of 5 would mean the user is likely to abandon the streaming service due to high levels of noise disrupting the video quality. The category of each video clip, i.e., benign versus video quality or RD attacks, is not revealed to the users and the videos are not annotated in any way. To avoid any bias in the gathered answers, the participants are not aware of RoVISQ methodology and the overarching goal of the paper. Specifically, the users do not know which videos are generated by RoVISQ, whether RoVISQ is a defense or an attack, and what the nominal score for each video clip is.

Figure 8 shows the distribution of scores selected by the participants for the original video clips as well as the attacked versions. As seen, participants assign low noise scores to benign videos, with a score of 1 appearing in 82.2% of the responses. This result establishes the baseline for video quality: the users unanimously agree that the quality of the benign videos included in the survey are adequately high. Once RoVISQ attacks are applied to the same videos, however, we can see a clear shift in the distribution of scores to higher values (more visible noise and lower video quality). For the video quality attack, the distribution of the 4 and 5 scores is nearly the same, constituting 37.4% and 41.9% of the participant answers. For the RD attack, the portion of scores equal to 5 increases to 61.1%. This is intuitive since RD attack drops the PSNR by a larger margin compared to the video quality attack (see Figure 9). Observing a noise score of 5 in the majority of answers validates that our proposed attacks successfully degrade users’ QoE, leading to denial or interruption of service.

2The video clips used in the user survey are visible on the project website (https://sites.google.com/view/demo-of-rovisq/home).

Results for video quality and RD attacks are provided in Appendix B.

D. RoVISQ Attacks on Video Classification

We evaluate the success rate of our bandwidth attack when directed towards a downstream video classifier and provide comparisons with state-of-the-art attacks on video classification. We define the attack success rate as the portion of misclassified test samples for the untargeted attack. For the targeted attack, success rate is the portion of samples mapped to the adversary’s desired class. The convergence curves for generating targeted and untargeted perturbations using our losses are shown in Appendix C. The attacks in this section not only affect the downstream video classifier but also degrade the performance of the video compression system. We provide an analysis of this effect in Appendix B.

We first evaluate our attacks in the white-box scenario, i.e., when the attacker has complete knowledge of the underlying video classification model. Figure 9 demonstrates the attack success rate for white-box RoVISQ attacks, along with two baseline white-box attacks on video classification, namely, SAP [69] and C-DUP [37]. As seen, our attack consistently achieves the highest success rate. In particular, we obtain over 90% success rate on the UCF-101 and Jester datasets. This is in contrast to prior works, which largely lose their performance once the video is compressed, especially in that targeted scenario. We additionally consider an online attack scenario, where video data is continuously generated and a video clip is classified in real time from the generated video sequence. Even in the complex real-time scenario, our universal perturbations can be injected into both UCF-101 and Jester to achieve a high success rate up to 85.4% and 87.5%, respectively.

To explain why RoVISQ outperforms other baselines, we visualize some example video snapshots before and after compression in Figure 10. As observed in Figure 10(b), the compressed video clips lose some texture information and high-frequency components when compared with the original perturbed video in Figure 10(a). This effect hinders the attack success rate of prior works, causing the perturbations to be removed after compression.

We further analyze the performance of RoVISQ attacks in the black-box scenario, i.e., when the attacker does not have access to the video classifier. In this setting, we train our universal perturbations on a surrogate video classifier, and use the trained perturbations to attack an unseen victim model. The attack success rate obtained by RoVISQ and two state-of-the-art black-box attacks on video...
with prior work on adversarial video classification. Attacks are conducted in the black-box untargeted scenario on the Jester dataset. The names inside parentheses in the attack column are the surrogate video classifiers used to train the RoVISQ universal perturbations.

**TABLE III: Performance of RoVISQ bandwidth attack compared to previous attacks.** As shown, the proposed adversarial perturbations are transferable to unseen video classification models, outperforming previous attacks, namely, GeoTrap [36] and U3D [71], by 27.8-66.8% and 22.2-65.8%, respectively.

VI. RESILIENCY TO DEFENSE SCHEMES

In this section, we comprehensively evaluate different defense mechanisms against RoVISQ attacks. There are very few defenses available for adversarial video classification. We evaluate the recent defense proposed in [43] which relies on adversarial training (AT). We make slight modifications to [43] to make it applicable to our attack scenario. In addition, we implement new defense mechanisms that rely on signal transformations to remove adversarial perturbations from the input video and audio domains [27], [29], [32], [40]. Signal transformation aims at reconstructing the original signal while removing the adversarial perturbations.

**TABLE IV: PSNR and Bpp of adversarially trained (AT) [46] DVC on the HEVC class C Dataset.**

![Image](image_url)

**Fig. 11: Overview of the evaluated defenses.** The signal transformation aims at reconstructing the original signal while removing the adversarial perturbations.

obtain a clean signal that is similar to the original benign input. We evaluate two signal transformations, namely, JPEG compression [6], [29], [42], [54] and the state-of-the-art video denoising method [16]. We use DVC [44] compression and the HEVC (class C) and UCF-101 datasets for the defense evaluations.

A. Adversarial Training

The performance of AT depends on the adversary’s knowledge regarding the attack algorithms. We assume that the defender has complete white-box knowledge about the perturbations crafted by our attacks. AT expands the training dataset by including all of the adversarial examples generated by the adversary and trains the video compression model on the augmented training dataset. Specifically, the defender adds perturbed videos from every \( \lambda \in \{256, 512, 1024, 2048\} \) for each RoVISQ attack (bandwidth, video quality, and RD) to the clean training dataset. Due to space limitations, we present the results for selected benchmarks in what follows.

**Defense for Video Compression.** As shown in Table IV, the incorporation of adversarial images inside the training comes at a cost of 18.5% higher Bpp and 1.64dB lower PSNR even if the underlying victim model is not attacked. Moreover, the adversarial training cannot protect the video compression model against RoVISQ attacks. As an example, for the offline attack in Table IV, the video quality attack to the benign model would drop the PSNR by 3.5dB. This drop is reduced to 2.5dB with the adversarial training. However, due to the original 1.64dB drop of PSNR after adversarial training, the total effect on PSNR is even worse than the no defense scenario. For the bandwidth attack, the Bpp is increased by 99.5% when no defense is present and 88.5% with adversarial training. However, the original 18.5% increase in Bpp due to adversarial training defeats the purpose of the defense. Similar trends can be observed for RD attack. We include additional variations of the datasets evaluated against AT in Appendix Table XII.

**TABLE V: Benchmark w vs. w/o Defense PSNR (dB) Bpp PSNR (dB) Bpp**

**Defense for Video Classification.** We now present the defense results against RoVISQ attacks for the video compression and classification system. Here, we benchmark our bandwidth attack. Table V shows the classification accuracy of the original
We assume that the defender executes JPEG compression as a preprocessing before performing video compression. To compress a perturbed image, the high (low) frequency DCT coefficients are usually scaled more (less). The coefficients are then rounded to the nearest integers by performing a quantization controlled by compression factor (CF). CF directly controls the trade-off between image quality and compression rate, as such, we perform our evaluations with two different values of CF, i.e., 20 and 40.

### B. JPEG Compression

JPEG is a popular lossy compression that applies discrete cosine transform (DCT) on images with different frequencies. We assume that the defender executes JPEG compression as a preprocessing before performing video compression. JPEG compression cannot remove the effect of our perturbations. When JPEG is used as preprocessing before video compression, we observe that our proposed video quality attack is applied in the offline and online settings, the PSNR is dropped by \( \sim 2 \text{dB} \) even in the absence of the attack. This is because the texture of the input video which contains high-frequency information is also removed after JPEG coding. As shown, the effect of JPEG compression on attack effectiveness is very negligible, i.e., our attacks can still degrade the QoE factors as intended. This trend can be observed for both our online and offline attacks.

#### C. Video Denoising

Denoising is a fundamental video processing algorithm that removes sensor imperfections. We assume that the defender has a noise removal algorithm in the front-end sources, e.g., cameras, before applying the video compression. We use a state-of-the-art DNN-based blind denoiser (ViDeNN) that does not assume any prior environmental conditions, e.g., color and light. As such, ViDeNN can adapt to the external changes where there is a different distribution for each frame.

### Defense for Video Compression

To evaluate RoVISQ attacks against JPEG compression, we apply our offline and online attacks on UCF-101 datasets. As shown in Table VII, applying JPEG compression reduces the accuracy of the downstream video classifier by 2.5-7.4\% even in the absence of any adversarial attack. However, JPEG compression cannot remove the effect of our perturbations. Table VII summarizes the attack success rate on video classification when JPEG is used as preprocessing before video compression. We see that JPEG fails to defend against our targeted attacks in most cases. In addition, Table V shows that compared to the AT defense against untargeted attacks, the performance of JPEG compression is relatively worse. The main reason is that JPEG compression removes adversarial instances on a frame-by-frame basis but it cannot suppress error propagation using a time-series analysis.

### TABLE V: Effect of various defense methods on the accuracy (ACC) of video classifiers on clean data along with the targeted attack success rate (ASR). Results are gathered on UCF-101 dataset.

<table>
<thead>
<tr>
<th>Video Classifier</th>
<th>Defense</th>
<th>w/o Defense</th>
<th>w Defense</th>
<th>Drop (%)</th>
<th>ASR (%)</th>
<th>w/o Defense</th>
<th>w Defense</th>
</tr>
</thead>
<tbody>
<tr>
<td>SlowFast</td>
<td>AT</td>
<td>65.8</td>
<td>8.0</td>
<td>74.3</td>
<td>82.7</td>
<td>81.8</td>
<td>91.0</td>
</tr>
<tr>
<td></td>
<td>JPEG</td>
<td>67.7</td>
<td>6.5</td>
<td>73.3</td>
<td>69.8</td>
<td>68.5</td>
<td>82.7</td>
</tr>
<tr>
<td></td>
<td>Denoising</td>
<td>65.8</td>
<td>7.2</td>
<td>72.3</td>
<td>69.8</td>
<td>68.5</td>
<td>82.7</td>
</tr>
<tr>
<td>TPN</td>
<td>AT</td>
<td>69.3</td>
<td>5.5</td>
<td>76.3</td>
<td>73.3</td>
<td>68.5</td>
<td>82.7</td>
</tr>
<tr>
<td></td>
<td>JPEG</td>
<td>75.3</td>
<td>8.5</td>
<td>78.3</td>
<td>76.3</td>
<td>68.5</td>
<td>82.7</td>
</tr>
<tr>
<td></td>
<td>Denoising</td>
<td>69.3</td>
<td>5.5</td>
<td>76.3</td>
<td>73.3</td>
<td>68.5</td>
<td>82.7</td>
</tr>
<tr>
<td>3D</td>
<td>AT</td>
<td>64.2</td>
<td>6.2</td>
<td>70.2</td>
<td>66.2</td>
<td>68.5</td>
<td>82.7</td>
</tr>
<tr>
<td></td>
<td>JPEG</td>
<td>70.2</td>
<td>6.2</td>
<td>73.2</td>
<td>69.2</td>
<td>68.5</td>
<td>82.7</td>
</tr>
<tr>
<td></td>
<td>Denoising</td>
<td>64.2</td>
<td>6.2</td>
<td>70.2</td>
<td>66.2</td>
<td>68.5</td>
<td>82.7</td>
</tr>
</tbody>
</table>

### TABLE VI: Untargeted Attack success rate (ASR) on video classification in the presence of various defense techniques. Results are gathered on the UCF-101 dataset.

<table>
<thead>
<tr>
<th>Video Classifier</th>
<th>Defense</th>
<th>ASR (%)</th>
<th>ASR (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SlowFast</td>
<td>AT</td>
<td>67.1</td>
<td>53.2</td>
</tr>
<tr>
<td></td>
<td>JPEG</td>
<td>72.3</td>
<td>64.6</td>
</tr>
<tr>
<td></td>
<td>Denoising</td>
<td>73.3</td>
<td>64.1</td>
</tr>
<tr>
<td>TPN</td>
<td>AT</td>
<td>64.2</td>
<td>58.2</td>
</tr>
<tr>
<td></td>
<td>JPEG</td>
<td>70.9</td>
<td>61.2</td>
</tr>
<tr>
<td></td>
<td>Denoising</td>
<td>71.8</td>
<td>63.8</td>
</tr>
<tr>
<td>3D</td>
<td>AT</td>
<td>75.8</td>
<td>65.3</td>
</tr>
<tr>
<td></td>
<td>JPEG</td>
<td>80.8</td>
<td>72.2</td>
</tr>
<tr>
<td></td>
<td>Denoising</td>
<td>82.7</td>
<td>68.5</td>
</tr>
</tbody>
</table>

### TABLE VII: PSNR and Bpp of DVC on videos preprocessed by JPEG compression on the HEVC class D Dataset, where CF means compression factor.

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>CF</th>
<th>PSNR (dB)</th>
<th>Bpp</th>
<th>PSNR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVC</td>
<td>20</td>
<td>31.14</td>
<td>0.28</td>
<td>31.24</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>29.26</td>
<td>0.21</td>
<td>29.26</td>
</tr>
<tr>
<td>Video Quality (Offline)</td>
<td>20</td>
<td>-3.35</td>
<td>+0.7%</td>
<td>-3.42</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>-3.14</td>
<td>+0.6%</td>
<td>-3.24</td>
</tr>
<tr>
<td>Video Quality (Online)</td>
<td>20</td>
<td>-2.95</td>
<td>+19.1%</td>
<td>-3.05</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>-2.76</td>
<td>+18.4%</td>
<td>-2.86</td>
</tr>
<tr>
<td>Bandwidth (Offline)</td>
<td>20</td>
<td>-0.45</td>
<td>+86.7%</td>
<td>-0.51</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>-1.45</td>
<td>+34.7%</td>
<td>-1.51</td>
</tr>
<tr>
<td>Bandwidth (Online)</td>
<td>20</td>
<td>-0.45</td>
<td>+86.7%</td>
<td>-0.51</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>-1.45</td>
<td>+34.7%</td>
<td>-1.51</td>
</tr>
<tr>
<td>RD (Offline)</td>
<td>20</td>
<td>-2.95</td>
<td>+31.8%</td>
<td>-3.00</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>-2.79</td>
<td>+28.6%</td>
<td>-2.85</td>
</tr>
<tr>
<td>RD (Online)</td>
<td>20</td>
<td>-2.95</td>
<td>+31.8%</td>
<td>-3.00</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>-2.79</td>
<td>+28.6%</td>
<td>-2.85</td>
</tr>
</tbody>
</table>

scheme, even when the compression factor CF is set to a high value of 40. Note that when CF=40 in Table VII, we can observe that the compression performance of the clean video is reduced by \( \sim 2 \text{dB} \) even in the absence of the attack. This is because the texture of the input video which contains high-frequency information is also removed after JPEG coding. As shown, the effect of JPEG compression on attack effectiveness is very negligible, i.e., our attacks can still degrade the QoE factors as intended. This trend can be observed for both our online and offline attacks.
observed that the video quality attack is largely resilient against denoising. Additionally, our bandwidth and RD attacks, particularly in the offline setting, still significantly affect the video compression after video denoising. It is worth noting that the high computational cost of ViDeNN makes it impractical for real-time defense on video content. We conclude that transforming the input signal through video denoising cannot prevent RoVISQ attacks. As shown in Figure 7, even if the defender denoises the perturbed input, the error is still propagated from the perturbed input to the compressed video.

**Defense for Video Classification.** The denoising operation, by nature, is a lossy transformation that reduces the accuracy of the downstream classifier. As shown in Table V, denoising the benign streaming video can lead to 4-7.5% lower classification accuracy even when the input is not adversarial. Tables V and VI report the attack success rate for targeted and untargeted scenarios when the perturbed video is passed through the denoiser. As can be observed, the targeted attack still maintains a high success rate of 75.3-81.8% which is 10.3-16.7% lower than when no defense is present. On the untargeted attack scenarios, the denoising reduces the attack success rate by 12.2-24% but it remains well above 60%, thus still posing a critical problem for the classification service. Our results show that denoising the video cannot successfully remove the effect of injected adversarial perturbations.

### VII. Related Work

**DNN-based Video Compression.** In the past decades, plenty of handcrafted image and video compression standards were proposed, such as JPEG [67], JPEG 2000 [58], H.264 [70], and H.265 [60]. Most of these methods follow handcrafted algorithms to remove redundancies in both spatial and temporal domains. Recently, DNN-based video compression frameworks have attracted a lot of attention [26], [44], [74]. Especially, the video compression framework in [44] achieves impressive results by replacing all the components in the standard H.264/H.265 video compression codecs with DNNs. Deep learning-based video compression techniques rely on convolutional neural networks (CNNs) for their three main design components: (1) motion estimation network for estimating the temporal motion, (2) motion compensation network to generate the predicted frame, and (3) auto-encoder style network for compressing the motion and residual data. MPEG experts have already started discussions about a new video codec that takes advantage of DNN technology [1]. It has also been adopted in industry [2]. An important motivation is DNN’s superior performance compared to legacy methods. Another motivation seen in industry is using DNN accelerators on mobile devices, e.g., Qualcomm Snapdragon 888 processor [2]. Compared to handcrafted video compression standards [60], [70], CNN-based end-to-end optimized video compression significantly reduces the redundancies in video motion [44]. Furthermore, the R-D optimization adopted in DNN-based video compression enables higher compression efficiency by directly using the number of bits in the optimization procedure. To estimate the bit-rates, context models [7], [8], [41], [48], [65] are learned for the adaptive arithmetic coding method which compresses discrete-valued data to bit-rates closely approaching the entropy of the representation. More recently, Yang et al. [74] shows that hierarchical GOP structure can make use of the advantageous information from high-quality frames by improving the temporal predictive coding. Furthermore, Hu et al. [26] propose to perform all operations (motion estimation, motion compensation, motion compression, and residual compression) in the feature space, which demonstrates better video compression performance.

**DNN-based Video Classification.** DNNs are widely adopted in live video analysis services such as video activity recognition for health care [24], video surveillance [62], and self-driving cars [31]. One of the widely used approaches for video activity recognition is an inflated three-dimensional (3D) network [13]. This method builds upon a pre-trained image classification model by inflating the convolutional and pooling kernels with an additional temporal dimension. Doing so enables the proposed spatiotemporal convolutions to treat spatial structures and temporal events separately, creating a two-stream approach. Slowfast [21] also adopts a two-stream approach and improves the accuracy for action detection by digesting the input video at different temporal resolutions. More recently, Yang et al. [73] propose a generic feature-level Temporal Pyramid Network (TPN) to model speed variations amid different actions.

**Adversarial Attack and Defense on Image Domain.** The adversarial attack was first proposed by [64] aiming to fool a victim model with small perturbations. Goodfellow et al. [23] developed the fast gradient sign method (FGSM) to calculate the perturbation by following the direction of gradients. Kurakin et al. [35] extended FGSM to an iterative approach, called iterative FGSM (I-FGSM), and showed higher attack success rates. Recently, adversarial attacks using image compression [19] have been proposed that drop critical information from images. This method optimizes over a trainable quantization table by minimizing the adversarial loss.

In response to adversarial attacks, many prior works suggest adversarial training to improve the robustness of the victim DNN [20], [49], [53], [56], [66]. Nevertheless, adversarial training is known to negatively affect the accuracy on clean samples, thus leading to the much-debated trade-off between accuracy and robustness. Recently, compression-based defense algorithms [9], [29], [42], [54] have been proposed which use lossy compression techniques such as JPEG coding to remove the adversarial perturbations. In the video domain, research on defenses against adversarial attacks using video compression is not explored in prior methods.

**Adversarial Attack and Defense on Video Domain.** There are only a few studies that propose adversarial attack on video action recognition models. Wei et al. [69] propose an optimization-based method to generate adversarial perturbation for the CNN+RNN video classifier [18]. Pony et al. [35] present a flickering attack that changes the color of each frame to obtain an adversarial effect and fool the video recognition model. Li et al. [36] employ standard geometric transformations for query-efficient black-box attacks. The attacks proposed in [9], [36], [53], [55] are all performed offline. In this context, the video is decomposed into frames which then undergo adversarial attacks, similar to the image domain.

### TABLE VIII: PSNR and Bpp of DVC on videos preprocessed by DNN-based denoiser on the HEVC class C Dataset.

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>w Defense PSNR (dB)</th>
<th>Bpp</th>
<th>w/o Defense PSNR (dB)</th>
<th>Bpp</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVC [44]</td>
<td>29.74</td>
<td>0.28</td>
<td>31.24</td>
<td>0.27</td>
</tr>
<tr>
<td>Video Quality (Offline)</td>
<td>-3.23</td>
<td>+0.5%</td>
<td>-3.52</td>
<td>+0.8%</td>
</tr>
<tr>
<td>Video Quality (Online)</td>
<td>-2.76</td>
<td>+14.3%</td>
<td>-3.05</td>
<td>+19.9%</td>
</tr>
<tr>
<td>Bandwidth (Offline)</td>
<td>-0.12</td>
<td>+64.8%</td>
<td>-0.01</td>
<td>+99.5%</td>
</tr>
<tr>
<td>Bandwidth (Online)</td>
<td>-0.43</td>
<td>+21.8%</td>
<td>-0.39</td>
<td>+35.7%</td>
</tr>
<tr>
<td>RD (Offline)</td>
<td>-3.81</td>
<td>+56.8%</td>
<td>-4.21</td>
<td>+85.3%</td>
</tr>
<tr>
<td>RD (Online)</td>
<td>-2.63</td>
<td>+18.4%</td>
<td>-3.10</td>
<td>+33.5%</td>
</tr>
</tbody>
</table>
Li et al. [37] train a universal perturbation generator offline and use it to attack real-time video classification systems. The most recent study [71] generates universal 3-dimensional perturbations to subvert real-time video classification systems using a surrogate DNN model. The proposed perturbations in the above studies [36], [37], [53], [69], [71] lose effect once the video is passed through the compression pipeline. As such, they cannot be applied in real-world live video streaming and classification scenarios where video compression is a crucial step, as shown in Figure 1. The proposed attacks on video classification in this work, however, can withstand several lossy signal transformations, e.g. compression and denoising.

There are very few defenses that protect against adversarial attacks on videos [43]. Lo et al. [43] perform adversarial training with multiple independent batch normalization layers to learn different perturbation types. Denoising [16] is also a fundamental video processing method to remove noise from a camera sensor, such as mobile phones and surveillance cameras. In this paper, we show that RoVISQ attacks are robust to the proposed adversarial training and DNN-based denoiser.

VIII. CONCLUSION, LIMITATIONS, AND FUTURE WORK

This paper presents the first systematic study on adversarial attacks to deep learning-based video compression systems. Our attacks, dubbed RoVISQ, are the first to manipulate the Rate-Distortion (R-D) relationship of the video compression model for influencing the user quality of experience (QoE). RoVISQ attacks are formalized as well-defined optimization problems, thus resulting in a dramatic degradation of video quality and compression ratio. We also propose novel targeted and untargeted attacks against a downstream deep learning-based video classification system. While the video compression framework inherently invalidates most adversarial examples, our attack is robust to compression by considering the temporal coding structure of the GOP and encoding parameters. Our comprehensive experiments show that our attacks outperform noise baselines and previously proposed attacks in both offline and online settings. Furthermore, our attacks still maintain high success rate in the presence of various defenses, such as adversarial training, video denoising, and JPEG coding.

While RoVISQ shows great success in attacking various video compression systems, the perturbations designed in this work are purely digital. An interesting future work is investigating physical sensor-based attacks where the video is perturbed during generation, e.g., using LED light manipulations at the camera. Another area of future studies is expanding factors that affect the user QoE in video compression and classification systems, potentially by performing large-scale user studies. Establishing effective defense strategies against our attacks is also a crucial and promising direction.

ACKNOWLEDGEMENT

We would like to thank the anonymous NDSS reviewers for their valuable feedback. We also thank Mohammad Samrakh for fruitful discussions. This work was supported by the U.S. Army/Department of Defense award number W911NF2020267.

REFERENCES


When targeting both BPP and PSNR, our RD attack still outperforms Tables IXa, IXb, IXc. The provided numbers here correspond to the reasons: (1) HLVC and FVC use post-processing techniques based

(2) HLVC and FVC employ hierarchical temporal coding structures as mentioned in Section [11]. By exploiting temporal information form multiple perturbed reference frames, the hierarchical structure becomes more vulnerable to error than non-hierarchical structure.

Unseen Victim Models for Black-box Attack. Table X encloses the architectural details of victim video compression models used in our black-box attack evaluations (see Section V-B, Table I). New models are constructed by altering the architecture of a state-of-the-art video compression model, e.g., DVC [44], denoted by “Template Model” in Table X. Each model uses the GOP structure of the corresponding template model. The new models are then trained from scratch following the training setup of their template model. We train four variations for each new model using different encoding parameters $\lambda \in \{256, 512, 1024, 2048\}$.

APPENDIX B

RoVISQ Attacks on Video Classification

Below we provide the success rate of our video quality and RD attacks when applied to a downstream video classifier. Similar to the analysis performed in Section V-D, we evaluate our attacks in the white-box and black-box settings in Figure [13] and Table XI respectively. All attacks in this section are conducted on the Jester activity recognition dataset. For each attack scenario, we compare our attack success rate with the prior state-of-the-art adversarial attacks for video classification. Overall, similar trends can be observed as those in Section V-D. RoVISQ perturbations maintain a high attack success rate in both white-box and black-box scenarios while the effect of other adversarial attacks is removed after video compression. Notably, RD attack obtains the best attack success rate. This is intuitive since our formulation of the bandwidth and video quality attacks enforce constraints on the perturbations to ensure only one QoE factor is changed (see Equations 5-6). Our formulation for RD attack, however, does not have such constraints (see Equation 7) and therefore has more flexibility in finding a more effective perturbations.

Effect on QoE. Our proposed attacks on downstream video classifiers also affect the $R$-$D$ curve of the video compression system. In Figure [13] we show the effect of our attacks for video compression and classification on the QoE of the DVC video coder. We observe that our adversarial attacks on video compression and classification can selectively deteriorate the benign compressor’s $R$ and $D$ as explained in Section V-B. In the example of the Jester dataset, our bandwidth attack increases the Bpp by up to $2.4 \times$ while changing the video quality by only $0.1 \text{dB}$. Our video quality attack lowers the PSNR by $4.84 \text{dB}$ while changing the Bpp by only $1.47\%$. Our RD attack reduces the PSNR by $4.96 \text{dB}$ and increases Bpp by $2.2 \times$. Although the online universal attack has lower performance than our offline attack, it lowers PSNR by up to $3.5 \text{dB}$ and increases Bpp by up to $101.42\%$ on Jester dataset.

APPENDIX C

Convergence Analysis of the Proposed Attacks

The convergence curves of perturbation generation using our losses for targeted and untargeted attacks are demonstrated in Figure [15]. As seen, both losses converge rapidly, and achieve $\sim 100\%$ attack success rate with 20 FGSM iterations. There is a slight difference in the convergence speed among the bandwidth, video quality, and RD attacks. Since the video quality and RD attacks inject perturbations that downgrade image quality, their overall convergence speed is a bit faster compared to the bandwidth attack.
TABLE IX: PSNR and Bpp changes after applying RoVISQ attacks on the (a) DVC [44], (b) HLVC [74], and (c) FVC [26] compression models. Reported values are averaged over different encoding parameters $\lambda \in \{256, 512, 1024, 2048\}$.

![Graph](image1)

(a) Untargeted Attack
(b) Targeted Attack

Fig. 13: Attack success rate of our compression-robust classifier perturbations compared to state-of-the-art adversarial attacks on video classification, i.e., SAP [69] and C-DUP [37]. Attacks are conducted in the white-box scenario on three video classification models, i.e., I3D [13], SlowFast [21], and TPN [73] using the Jester dataset.

![Graph](image2)

Fig. 14: Experimental results of RoVISQ attacks on video compression and classification systems when tested on the UCF-101 and Jester test sets. Each graph contains the results of video compression for the case where $\lambda \in \{256, 512, 1024, 2048\}$.
TABLE X: Architecture details of victim video compression models used in the black-box evaluations of Table I. We use the $n_1 \Rightarrow n_2$ notation where $n_1$ is the number of layers/kernels for the corresponding module in the template model and $n_2$ is the altered number of layers/kernels in the new victim model.

<table>
<thead>
<tr>
<th>Victim Model</th>
<th>Attack</th>
<th>Attack Success Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GeoTrap [36]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TPN [73]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SlowFast [21]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3D [13]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

TABLE XI: RoVISQ video quality and RD attacks compared with prior work on adversarial video classification. Attacks are conducted in the black-box untargeted scenario on the Jester dataset. The names inside parentheses in the attack column are the surrogate video classifiers used to train the RoVISQ universal perturbations.

<table>
<thead>
<tr>
<th>Victim Model</th>
<th>Attack</th>
<th>Attack Success Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GeoTrap [36]</td>
<td>Video Quality (I3D)</td>
<td>74.1, 77.5, 82.4, 87.0</td>
</tr>
<tr>
<td>TPN [73]</td>
<td>Video Quality (TPN)</td>
<td>75.9, 78.7, 83.3, 86.1</td>
</tr>
<tr>
<td>SlowFast [21]</td>
<td>Video Quality (TPN)</td>
<td>75.9, 79.6, 83.3, 86.1</td>
</tr>
<tr>
<td>3D [13]</td>
<td>Video Quality (TPN)</td>
<td>75.9, 79.6, 83.3, 86.1</td>
</tr>
</tbody>
</table>

TABLE XII: PSNR and Bpp of adversarially trained (AT) [46] DVC on the HEVC class B and C Datasets.

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Dataset</th>
<th>PSNR (dB)</th>
<th>Bpp</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVC [44]</td>
<td>Class B</td>
<td>31.14</td>
<td>0.20</td>
</tr>
<tr>
<td></td>
<td>Class D</td>
<td>29.28</td>
<td>0.28</td>
</tr>
<tr>
<td>Video Quality (Offline)</td>
<td>Class B</td>
<td>-2.61</td>
<td>+0.8%</td>
</tr>
<tr>
<td>Video Quality (Online)</td>
<td>Class B</td>
<td>-1.83</td>
<td>+14.8%</td>
</tr>
<tr>
<td>Bandwidth (Offline)</td>
<td>Class B</td>
<td>-0.13</td>
<td>+75.9%</td>
</tr>
<tr>
<td>Bandwidth (Online)</td>
<td>Class B</td>
<td>-0.58</td>
<td>+28.3%</td>
</tr>
<tr>
<td>RD (Offline)</td>
<td>Class B</td>
<td>-3.13</td>
<td>-72.4%</td>
</tr>
<tr>
<td>RD (Online)</td>
<td>Class B</td>
<td>-1.81</td>
<td>+26.2%</td>
</tr>
</tbody>
</table>

TABLE XIII: Accuracy (ACC) of video classification on clean videos, along with the attack success rate (ASR) of compression-robust classifier perturbations. The defense is adversarial training on the Jester dataset. Here, “T” and “U” denote targeted and untargeted attacks, respectively.

<table>
<thead>
<tr>
<th>Video Classifier</th>
<th>Type</th>
<th>Attack</th>
<th>ASR (%)</th>
<th>ACC (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SlowFast</td>
<td>T Offline</td>
<td>71.1</td>
<td>94.2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>U Offline</td>
<td>72.5</td>
<td>96.8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>U Online</td>
<td>64.3</td>
<td>83.6</td>
<td></td>
</tr>
<tr>
<td>TPN [73]</td>
<td>T Offline</td>
<td>70.7</td>
<td>93.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>U Offline</td>
<td>75.5</td>
<td>97.3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>U Online</td>
<td>68.3</td>
<td>82.6</td>
<td></td>
</tr>
<tr>
<td>3D [13]</td>
<td>T Offline</td>
<td>65.3</td>
<td>92.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>U Offline</td>
<td>66.7</td>
<td>97.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>U Online</td>
<td>56.8</td>
<td>86.5</td>
<td></td>
</tr>
</tbody>
</table>