On the Robustness of Neural-Enhanced Video Streaming against Adversarial Attacks

Qihua Zhou[1], Jingcai Guo[1,2*], Song Guo[3*], Ruibin Li[1], Jie Zhang[1], Bingjie Wang[1], Zhenda Xu[1]
[1]The Hong Kong Polytechnic University, Hong Kong
[2]The Hong Kong Polytechnic University Shenzhen Research Institute, Shenzhen, China
[3]The Hong Kong University of Science and Technology, Hong Kong
{qi-hua.zhou, ruibin.li, bingjie-daniel.wang, jackal.xu}@connect.polyu.hk
{jc-jingcai.guo, jie-comp.zhang}@polyu.edu.hk, songguo@cse.ust.hk

Abstract

The explosive growth of video traffic on today's Internet promotes the rise of *Neural-enhanced Video Streaming* (NeVS), which effectively improves the rate-distortion trade-off by employing a cheap neural super-resolution model for quality enhancement on the receiver side. Missing by existing work, we reveal that the NeVS pipeline may suffer from a practical threat, where the crucial codec component (*i.e.,* encoder for compression and decoder for restoration) can trigger adversarial attacks in a man-in-the-middle manner to significantly destroy video recovery performance and finally incurs the malfunction of downstream video perception tasks. In this paper, we are the first attempt to inspect the vulnerability of NeVS and discover a novel adversarial attack, called *codec hijacking*, where the injected invisible perturbation conspires with the malicious encoding matrix by reorganizing the spatial-temporal bit allocation within the bitstream size budget. Such a zero-day vulnerability makes our attack hard to defend because there is no visual distortion on the recovered videos until the attack happens. More seriously, this attack can be extended to diverse enhancement models, thus exposing a wide range of video perception tasks under threat. Evaluation based on state-of-the-art video codec benchmark illustrates that our attack significantly degrades the recovery performance of NeVS over previous attack methods. The damaged video quality finally leads to obvious malfunction of downstream tasks with over 75% success rate. We hope to arouse public attention on codec hijacking and its defence.

Introduction

Driven by the remarkable surge of today's Internet video traffic, the *Neural-enhanced Video Streaming* (NeVS) (Yeo et al. 2022; Liu et al. 2021; Dasari et al. 2022) has emerged as a fundamental infrastructure to accommodate modern video-centric services across the network, including Zoom meeting (Zoom 2023), tiktok short-form videos (TikTok 2023) and YouTube live (YouTube 2023). As shown in Figure 1, the NeVS pipeline involves the collaboration between two sides. On the content delivery side (*i.e.,* the client), raw data are downscaled from original high-resolution (HR) frames into low-resolution (LR) ones and encoded into a compressed video for network transmission in streaming. On the content receiver side (*i.e.,* the server), the compressed video is fed into a cheap neural super-resolution (SR) model for quality enhancement (Yeo et al. 2022; Zhang et al. 2022; Nguyen et al. 2022; Wang et al. 2022). The final restored HR video holds adequate visual quality as the original version, and thus can be applied for video analytics in different downstream tasks. Such a pipeline greatly improves the *rate-distortion* trade-off, *i.e.,* reducing streaming traffic while not incurring a quality drop of the restored video, which is the core objective of NeVS systems.

Recently, optimizing the pipeline of NeVS has become a hot topic, such as improving the encoder-decoder (*i.e.,* codec) efficiency (Du et al. 2022; Dasari et al. 2022), reducing steaming latency (Yeo et al. 2018) and elaborating SR enhancement (Zhang et al. 2021; Nguyen et al. 2022). However, its security vulnerability has not been well explored. Missing by existing work, we reveal that NeVS easily suffers from a practical threat, where the crucial video codec component can trigger the adversarial attack to greatly destroy the restored video quality and finally cause the malfunction of downstream perception tasks.

Nevertheless, launching a successful adversarial attack on NeVS is not easy. Most existing works either rely on completely image-level perturbation (Wei et al. 2022; Yue et al. 2021) or deteriorate the video model accuracy based on specific vision tasks (Chen et al. 2022; Jia et al. 2021; Hwang et al. 2021). They fail to NeVS due to the unawareness of video codec. Essentially, the codec handles the video frames by allocating most bits to low-frequency information while compressing the high-frequency ones, so as to obtain the best balance between video size and restoration quality. However, previous codec-unaware adversarial attacks usually add high-frequency perturbation into frames (Yue et al. 2021; Choi et al. 2022). In this case, the encoding procedure serves as a noise filter to remove the injected perturbation, making conventional attacks fail to NeVS. This insight is also verified by our preliminary experiments (in Figure 3) and inspires us to build a *codec-aware* adversarial attack.

This paper is the first work to holistically inspect the vulnerability of NeVS and we discover a novel adversarial attack, called *codec hijacking*. Codec hijacking can be launched by controlling two factors: (1) searching and in-

*Corresponding author.

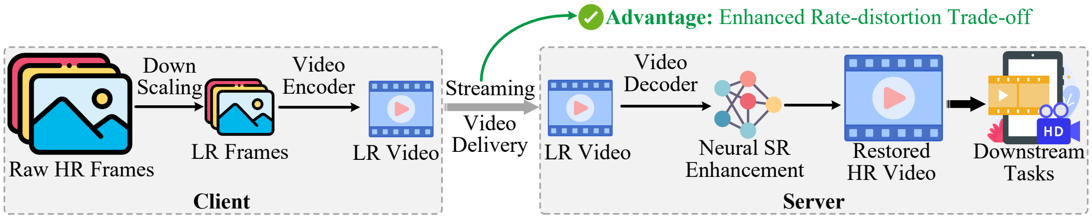


Figure 1: The overview of NeVS pipeline, including downscaling, video codec (encoder+decoder) and neural SR enhancement.

jecting invisible perturbation on LR frames before video encoding, (2) controlling the macroblock-level (*i.e.,* patch) *Quantization Parameter* (QP) (Schwarz, Marpe, and Wiegand 2007) matrix to generate malicious encoding, which conceals the injected perturbation in normal circumstances but can trigger the attack to significantly damage the quality of restored HR videos. We reveal that *only jointly optimizing these two factors can successfully attack the NeVS pipeline*. Codec hijacking camouflages the injected perturbation as common noise and triggers the attack only when invoking the malicious encoding. By invoking codec hijacking, the adversary can expose the entire NeVS system to a catastrophic zero-day vulnerability, which is hard to defend because there is no visual distortion on the restored video until the attack happens. Extensive experiments based on the typical UDM10 (Yang et al. 2019) benchmark demonstrate that codec hijacking significantly deteriorates the restoration performance of NeVS using different SR models, and finally results in downstream task malfunction with over 75% success rate, including multiple object tracking and human pose estimation. Overall, our key contributions are as follows.

- **Novel and critical attack on NeVS pipeline.** To the best of our knowledge, we are the first to study the vulnerability in NeVS. We discover a novel attack paradigm, codec hijacking, to effectively break the rate-distortion trade-off between quality preservation and traffic saving.
- **Covert and effective trigger inside codec.** We reveal that the codec component conceals the vulnerability and can trigger adversarial attacks to significantly degrade the recovered video quality without visual distortion on the intermediate frames. This attack paradigm captures the essentials of video streaming, *i.e.,* spatial-temporal encoding for frame bitrate controlling, to successfully fool the entire streaming pipeline, which cannot be achieved by transferring existing image-based attacks to videos.
- **Ubiquitous threat to diverse video perception tasks.** The proposed attack paradigm is hard to defend due to its zero-day vulnerability property. Extensive experiments show that diverse SR enhancement models and video perception tasks are vulnerable to such attacks, thus requiring public attention on its defence.

Preliminary

We first introduce the NeVS pipeline. Then, we briefly review conventional codec-unaware adversarial attacks and discuss why they fail to NeVS via preliminary experiments.

Neural-enhanced Video Streaming

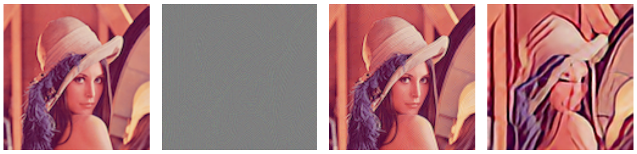
With the unprecedented boom of low-end devices (*e.g.,* mobile phones and IoT sensors), data are continuously generated on the client side and transmitted from the client to server (Huang et al. 2020). Video streaming enables the server to utilize videos and conduct analytics in real-time without having to completely receive all the frames. The contents of video streaming include live broadcasts, virtual conferences, YouTube user-generated content, surveillance and manufacturing in industry (Zheng, Zuo, and Zhang 2020). However, the video perception tasks require massive computations that client hardware cannot afford. Therefore, the raw frames are encoded as videos and then sent to the remote server for subsequent downstream tasks. To fully reduce streaming traffic while preserving the video quality, it comes to the rise of *Neural-enhanced Video Streaming* (NeVS), which significantly improves rate-distortion trade-off by employing a cheap super-resolution model for quality enhancement on the server side. As shown in Figure 1, the core objective of NeVS is to achieve an improved *rate-distortion* trade-off, *i.e.,* reducing streaming traffic while maximizing the restoration quality.

Conventional Adversarial Attacks

Existing adversarial attack works most focus on image-level perception tasks, *i.e.,* image classification (Wei et al. 2022), object detection (Tu et al. 2020), semantic segmentation (He et al. 2020) and super-resolution restoration (Yin et al. 2018; Choi et al. 2019; Yue et al. 2021), while the threat in videos has been less explored. When adversarial attack extends to videos, it is also first explored on video classifiers. Jiang *et al.* (Jiang et al. 2019) proposed the first black-box attack utilizing tentative perturbations and *Natural Evolution Strategies* (NES) to calibrate gradient. Recently, Jia *et al.* (Jia et al. 2021) explored the black-box attack by utilizing IoU scores in both current and historical frames. Consequently, most adversarial attack methods either rely on complete image-level perturbation or deteriorate the video model accuracy based on specific visual tasks. As a result, they are *unaware of the impact of video codec* to encode the frame sequences and fail to the scenarios of NeVS.

Adversarial Attacks on NeVS Pipeline

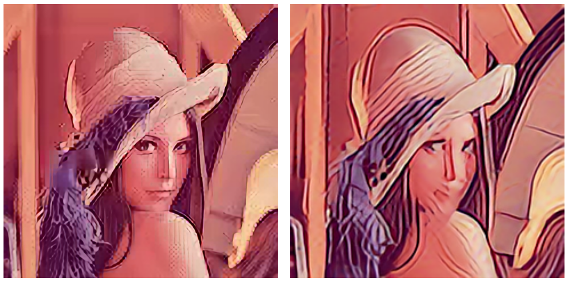
Attack visualization. Attacking the NeVS requires: (1) injecting perturbation into the original LR frame, (2) conducting malicious encoding to conceal the perturbation and damage the final restored HR frame. We use Figure 2 to visual-



(a) Original. (b) Perturbation. (c) Attacked. (d) Restored.

Figure 2: Visualization of adversarial attacks via (a) original LR frames, (b) image-level perturbation, (c) attacked LR frames, and (d) restored HR frames, where the rate-distortion trade-off of SR enhancement is explicitly broken.

ize the perturbation, attacked LR frame and final restored HR frame for a better understanding of the attack rationale. Given an original LR frame (Figure 2(a)), we can observe that the injected perturbation is visually imperceptible (Figure 2(b)) and it is hard to detect the perturbation by looking at the attacked LR frame (Figure 2(c)). However, after the neural SR enhancement, the final restored HR frame is significantly damaged, with unexpected moire textures and blurs (Figure 2(d)). This kind of distortion *not only deteriorates the perception experience of human, but also leads to catastrophic malfunction of downstream tasks.*



(a) Codec-unaware attack. (b) Our codec hijacking.

Figure 3: Comparison of the adversarial attack effectiveness based on the widely-used H.264 and H.265 standard.

Why not existing attack methods? On top of injecting perturbation, conducting malicious encoding is the key to successfully launching the attack on NeVS, *i.e., video codec is the essential trigger*. We conduct preliminary experiments in Figure 3 to verify an important point: existing codec-unaware attack methods fail to NeVS scenarios. Here is a control group with two frames. The frame in Figure 3(a) follows the codec-unaware attack scheme by solely injecting image-level perturbation. Meanwhile, the frame in Figure 3(b) injects the same perturbation while launching the proposed codec hijacking to conduct malicious encoding. We can observe that Figure 3(a) still holds good visual quality while Figure 3(b) is significantly damaged. The results motivate us to capture the impact of video codec and *establish a codec-aware adversarial attack to truly deteriorate the NeVS pipeline.*

Methodology: Codec Hijacking

Observing the vulnerability of the NeVS pipeline, we discover a novel adversarial attack by exploiting the video encoder-decoder (*i.e.,* codec) procedure, which is named *codec hijacking*. Codec hijacking is launched by two steps: (1) searching and injecting invisible perturbation into LR frames before video encoding, (2) controlling the macroblock-level (*i.e.,* patch) QP matrix to generate malicious encoding, which conceals the injected perturbation in normal circumstances but can trigger the attack to significantly damage the quality of restored HR videos. It is worth noting that solely injecting perturbation without the coordination of malicious encoding cannot bring distortion to the final restored video – *video codec is the trigger*. This indicates that the adversary can successfully camouflage the injected perturbation as common noise by using normal encoding instead of the malicious version, thus deceiving the NeVS system that there is no risk of adversarial attacks.

Formulation of Attack Pipeline

Given the entire video frame set $\mathbf{X}$, we use X_i to denote an original raw HR frame with index i, where $X_i \in \mathbf{X}$ and i identifies the sequence order for video encoding. For yielding less streaming traffic, a downscaling module is adopted to shrink the spatial size of X_i and transfer it as an LR frame x_i, *e.g.,* from 1920×1080 pixels to 480×270 pixels with a $4\times$ scaling ratio. We describe the downscaling procedure as $x_i = \texttt{Downscale}(X_i)$. Then, the proposed codec hijacking calculates the most effective perturbation δ_i and injects it into x_i to generate the *semi-attacked* LR frame $\hat{x}_i$. Note that the injected perturbation should be visually imperceptible, which means the similarity gap between x_i and $\hat{x}_i$ is bounded by the L_∞-norm constraint, *i.e.,* $\|x_i - \hat{x}_i\|_\infty \leq \alpha$, where α is the distortion upper bound following the *Iterative Fast Gradient Sign Method* (I-FGSM) (Kurakin, Goodfellow, and Bengio 2017; Choi et al. 2019) update rule.

By conducting malicious video encoding on $\hat{x}_i$, codec hijacking generates the *fully-attacked* LR video and transmits its bitstream through the network. The server receives the compressed LR video streaming, decodes and feeds it into the SR model for quality enhancement. As codec hijacking has triggered the malicious encoding based on perturbation injection, the restored HR video generated by the SR model will be significantly destroyed, usually with moire textures and blurs. This catastrophic deterioration of restoration quality breaks the *rate-distortion* advantages of NeVS systems.

From each downscaled LR frame x to the final restored HR video, the attack procedure on NeVS can be formulated as: $f(x;\delta,\mathcal{Q}) = \texttt{SR}(\texttt{Decode}(\texttt{Encode}(x;\delta,\mathcal{Q})))$, where δ and $\mathcal{Q}$ represent the injected perturbation and malicious encoding, respectively. Only simultaneously optimizing the perturbation δ and encoding $\mathcal{Q}$ can successfully launch the adversarial attacks of codec hijacking. Given a frame index i, we can use $f(x_i)$ and $f(x_i;\delta_i,\mathcal{Q}_i)$ to denote the frames of the final restored HR video, in normal and attacked NeVS, respectively. Thus, the objective of codec hijacking is to maximize distortion (*i.e.,* minimize similarity) of the frames inside the restored HR videos, which is defined as

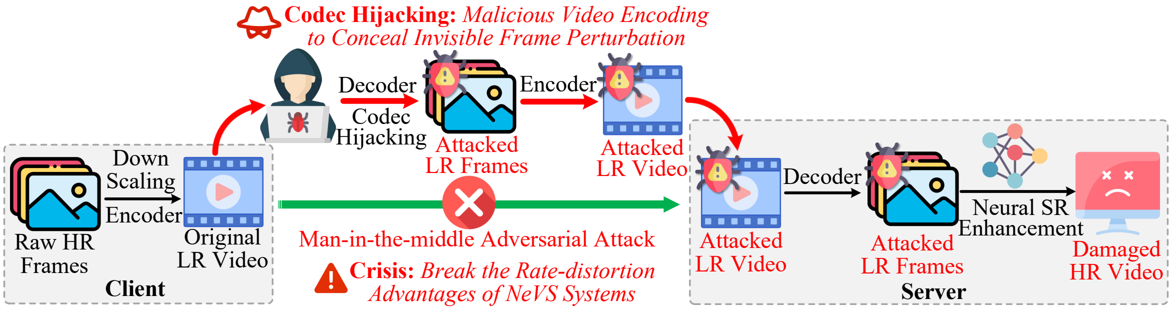


Figure 4: The overview of how codec hijacking launches a man-in-the-middle adversarial attack on the NeVS pipeline, which is established upon modern codec protocols, including H.264/AVC, H.265/HEVC and H.266/VVC. Thus, the codec information can be accessed by probing original LR videos, such as using “ffprobe” (ffprobe 2023), and the attacker can easily control QP by using the standard built-in APIs provided by the codec protocols.

$\min : \texttt{Sim}(f(x), f(x; \delta, \mathcal{Q}))$, where $\texttt{Sim}$ is the similarity function between normal and damaged frames. Optimizing the above target requires (1) searching perturbation δ and (2) conducting malicious encoding $\mathcal{Q}$.

Iterative Search of Segment Perturbation

Indeed, injecting independent perturbation for each frame can provide best the attack effectiveness. This frame-wise calculation will yield a huge computation overhead that may break the real-time streaming requirements. We need to obtain the perturbation in a relatively coarse-grained manner. In modern video codecs (*e.g.,* H.264/AVC (H.264 2023) and H.265/HEVC (H.265 2023)), a video is usually encoded as a series of segments, each of which is a group of frames with three types: I-, P- and B-frames (Ding et al. 2021). Encoding videos into segments can help to refresh video quality and recover from bitstream errors. Therefore, we treat a segment as the basic unit for searching perturbation. All the LR frames belonging to the same segment share the perturbation. This helps codec hijacking achieve a good trade-off between perturbation effectiveness and computation overhead.

Given a segment S_j with index j, we assume that there are N frames inside this segment, where the n-th frame is defined as x_n. With the segment perturbation δ_j, we can obtain the attacked LR frame $\hat{x}_n$ as: $\hat{x}_n = \texttt{Clip}_{0,1}(x_n + \delta_j)$, where the clipping function $\texttt{Clip}$ is defined as: $\texttt{Clip}_{a,b}(X) = \min(\max(X, a), b)$. By analyzing the visual characteristics of the entire segment, we measure the average frame distortion caused by segment perturbation δ_j as:

$$\mathcal{L}(\delta_j) = \frac{1}{N} \sum_{n=1}^{N} \texttt{Sim}(f(x_n), f(x_n; \delta_j, \mathcal{Q}_n)). \quad (1)$$

Following the principle of I-FGSM (Kurakin, Goodfellow, and Bengio 2017; Choi et al. 2019) mentioned above, we can initialize δ_j^0 by a Gaussian distribution noise and iteratively optimize the global perturbation as:

$$\delta_j^{t+1} = \texttt{Clip}_{-\alpha,\alpha}\left(\delta_j^t + \frac{\alpha}{T}\text{sign}\nabla\mathcal{L}(\delta_j^t)\right), \quad (2)$$

where the superscript t, T and $\text{sign}\nabla\mathcal{L}(.)$ represent the iteration index, the maximum number of iterations and the sign of the gradient, respectively. We stop the optimization procedure at iteration index T and generate the final segment perturbation δ_j^T. Note that the distortion upper bound α restricts the maximum degree of segment perturbation and avoids noticeable changes on the attacked LR frames.

Malicious Encoding via QP Matrix Controlling

For video size compression, the encoder adopts the intra- and inter-frame prediction to remove both spatial and temporal redundancy of the frames, which follows the classical predictive coding paradigm (Li, Li, and Lu 2021; Jiang et al. 2022) on frame residuals. To estimate the motion vector across frames, each frame is resolved into a series of macroblocks, *i.e.,* usually a patch with 16×16 pixels (some fine-grained estimation may use 8×8 or 4×4 patches). These macroblocks serve as the basic units for encoding and can be organized as a 2D matrix to represent the entire frame, where each macroblock separately corresponds to a spatial region.

Based on the residual prediction and motion estimation, the encoding procedure can transfer the frame from spatial space to frequency space, where the *Discrete Cosine Transform* (DCT) (Ding et al. 2021; Liu et al. 2020) and its variants are widely adopted. With such a transformation, the encoding procedure tends to allocate most bits to the macroblocks with low-frequency coefficients and compress the high-frequency information. As previous codec-unaware adversarial attacks usually add high-frequency perturbation to the input frames (Yue et al. 2021; Choi et al. 2022), the encoding procedure serves as a noise filter and smooths the image-level perturbation injected in frames, making conventional attack methods fail to NeVS scenarios. Recall the preliminary experiments in Figure 3, simply adopting vanilla encoding (*e.g.,* H.264/AVC (H.264 2023) and H.265/HEVC (H.265 2023)) cannot successfully launch the adversarial attacks on NeVS, where the final restored HR frames only suffer from a tiny distortion on the visual quality. This phenomenon requires us to hijack the encoding procedure and conceal the injected perturbation.

Actually, the macroblock-wise bit allocation is a kind of data quantization and is controlled by the *Quantization Parameter* (QP), which is the index used to derive a scaling matrix (Schwarz, Marpe, and Wiegand 2007). In the widely-

Algorithm 1: Distortion-oriented QP Matrix Controlling

```
Input: original HR frame $X$, bitstream size budget $\mathcal{B}$.
Output: adversarial pair of $< \mathcal{Q}, \delta >$.
 1: $epochs \leftarrow E$;                          ▷ Set the maximum epochs.
 2: $\mathcal{S} \leftarrow 32$;                   ▷ Set the initial QP controlling stride.
 3: $\mathcal{Q} \leftarrow 0$;                    ▷ Uniformly initialize QP matrix.
 4: $x \leftarrow \texttt{Downscale}(X)$;          ▷ Interpolation, e.g., bicubic.
 5: $\mathcal{L}^*_{codec} \leftarrow \texttt{Sim}(f(x), X)$;  ▷ Vanilla codec's distortion.
 6: $\mathcal{L}^*_{atk} \leftarrow 0$;            ▷ Initial distortion by attack.
 7: while $epochs$ do
 8:     $flag \leftarrow \texttt{TRUE}$;           ▷ A flag for reducing search stride.
 9:     for $m_k \in x$ using zigzag scanning do
10:         if $q_k + \mathcal{S} \leq 51$ then    ▷ Current stride is feasible.
11:             $q_k \leftarrow q_k + \mathcal{S}$;  ▷ Increase the QP value.
12:             Get bitstream size $v$ with current QP matrix;
13:             if $v \geq \mathcal{B}$ then        ▷ Bitstream size control.
14:                 continue;                    ▷ The bitstream is oversize.
15:             $\mathcal{L}_{codec} \leftarrow \texttt{Sim}(f(x), X)$;
16:             Search perturbation $\delta$ by using Eq. (2);
17:             $\mathcal{L}_{atk} \leftarrow \texttt{Sim}(f(x), f(x; \delta, \mathcal{Q})$;
18:             if $\mathcal{L}_{codec} \leq \mathcal{L}^*_{codec}$ and $\mathcal{L}_{atk} \geq \mathcal{L}^*_{atk}$ then
19:                 $\mathcal{L}^*_{atk} \leftarrow \mathcal{L}_{atk}$;  ▷ Accept QP adjustment.
20:                 $flag \leftarrow \texttt{FALSE}$;  ▷ Retain stride.
21:             else
22:                 $q_k \leftarrow q_k - \mathcal{S}$;  ▷ Reject and reset QP.
23:         else                                 ▷ No adjustment space under current stride.
24:             continue;                        ▷ Adjust the next macroblock.
25:     if $flag$ is TRUE then
26:         $\mathcal{S} \leftarrow \mathcal{S}/2$;   ▷ Reduce the search stride.
27:     if $\mathcal{S} < 1$ then
28:         break;                               ▷ Stop when stride is smaller than 1.
29:     $epochs \leftarrow epochs - 1$;            ▷ Remaining epochs.
30: return $< \mathcal{Q}, \delta >$;
```

used H.264 and H.265 encoding, QP ranges from 0 to 51. Given a macroblock, the lower the QP is, the more bits will be allocated to it, thus with better visual quality. However, a lower QP value will also lead to a larger bitstream size of the macroblock. Therefore, the essential to make a good rate-distortion trade-off inside NeVS is to determine a proper bit allocation strategy on each macroblock. This property inspires us to generate malicious encoding by controlling the macroblock-wise QP matrix, which directly impacts how much perturbation can remain after encoding. We called this procedure the *distortion-oriented QP matrix controlling*. Given a clean frame x inside a segment and its bitstream size budget $\mathcal{B}$, our objective is to find the most effective adversarial pair $< \mathcal{Q}, \delta >$, *i.e.,* the malicious encoding matrix and the corresponding perturbation, to successfully launch the adversarial attack. We use Algorithm 1 to describe the rationale of QP matrix controlling based on the widely-used H.264 and H.265 standards.

Assuming a downscaled LR frame x is divided as K macroblocks, we denote a macroblock and its QP value as m_k and q_k, respectively, with the macroblock index k. The entire QP matrix corresponding to x is denoted as $\mathcal{Q}$. Each LR frame holds an individual QP matrix, which is iteratively optimized to maximize the visual distortion to the final restored HR frame. Here, the algorithm ensures two requirements. The first is to figure out all potential QP matrices that the final restored HR frame holds adequate visual quality as the origin HR one (the left condition in line 18). This guarantees that solely adopting malicious encoding of QP matrix without perturbation will *not* bring perceptible differences. The second is simply injecting perturbation without malicious encoding still cannot successfully bring visual deterioration, consistent as the observation in preliminary experiments. This property makes the perturbation hidden in the frame, thus is hard to defend. However, once we conduct malicious encoding and perturbation injection simultaneously, the final restored HR frames will be significantly damaged, compared with the restoration from clean one (the right condition in line 18). Since the bitstream size budget is maintained (in line 13), the video delivery system cannot detect this attack by checking the streaming bitrate, which effectively improves the attack success rate. By scanning all the frames, we can conduct malicious encoding on the entire video and make codec hijacking as the trigger for adversarial attacks. Note that we use binary search to restrict the computational complexity of Algorithm 1 as $O(K \times \log_2 \mathcal{S})$, where K and $\mathcal{S}$ represent the macroblock number inside the frame and the initial QP controlling stride, respectively. Consequently, the QP matrix controlling algorithm makes codec hijacking fast enough to attack the NeVS pipeline.

Experiments

Experimental Setups

NeVS benchmark and enhancement models. We employ the typical UDM10 (Yang et al. 2019) benchmark, covering the downstream tasks of object tracking and human pose estimation. The video codecs are based on the widely-used H.264/AVC (H.264 2023) and H.265/HEVC (H.265 2023) standards. Thus, the bitstream size budget (upper bound) is the size achieved by vanilla codecs on clean data, with constant QP 23, medium preset, and yuv420p pixel format. As to the video enhancement module, we consider 11 typical super-resolution models with diverse architectures and parameter sizes, including EDSR (Lim et al. 2017), EUSR (Choi et al. 2020), DBPN (Haris, Shakhnarovich, and Ukita 2018), RCAN (Zhang et al. 2018), MSRN (Li et al. 2018), 4PP-EUSR (Choi et al. 2020), ESRGAN (Wang et al. 2018), RRDB (Wang et al. 2018), CARN (Ahn, Kang, and Sohn 2018), FRSR (Soh et al. 2019) and NATSR (Soh et al. 2019). These models covers both image-level and video-level SR enhancement. For example, the methodologies of ESRGAN and EDSR can be extended into the design of video-wise SR models, including BasicVSR (Chan et al. 2021) and BasicVSR++ (Chan et al. 2022). Therefore, the selection of SR models matches the realistic deployment of NeVS systems.

Attack baselines and performance measurement. We inspect the robustness of the NeVS against our codec hijacking by measuring the frame similarity, using metrics of *Peak Signal-to-noise Ratio* (PSNR) and *Structural Similarity Index Method* (SSIM). To check the invisible perturbation, we measure the similarity between the clean and attacked

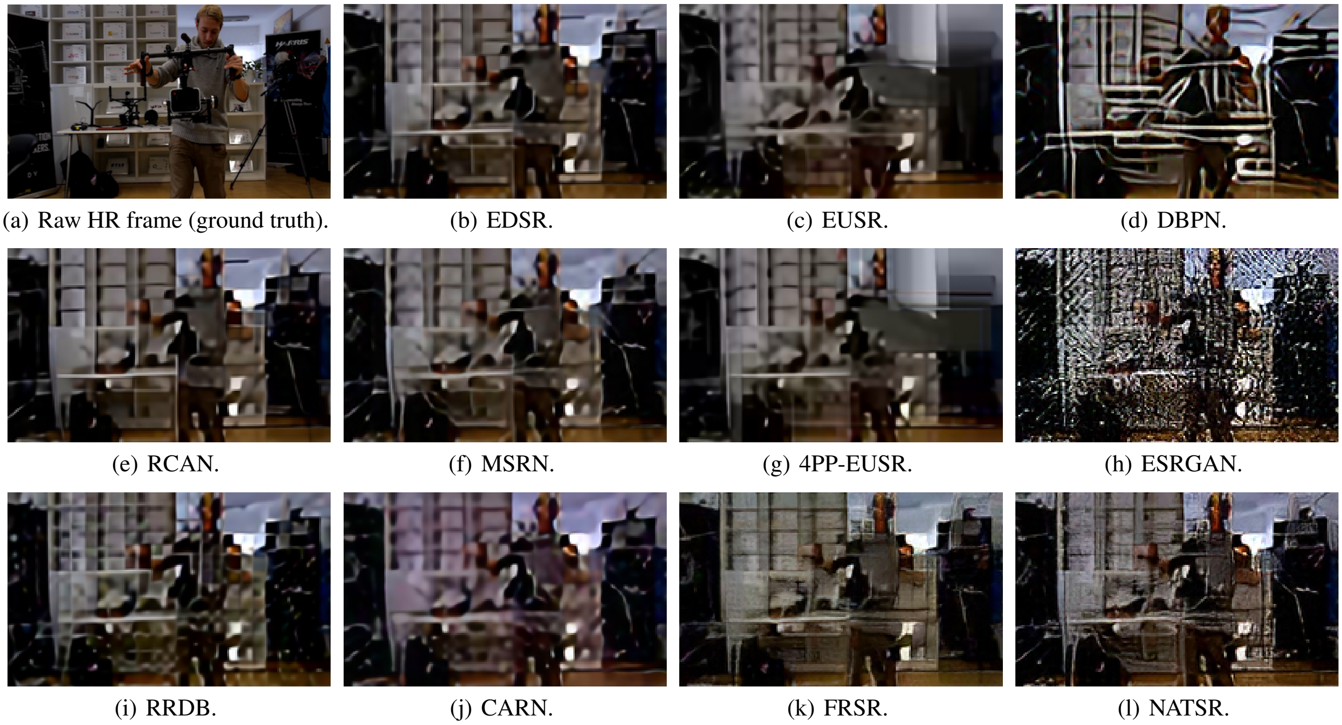


Figure 5: Visual comparison of restored HR frames when launching codec hijacking on different SR enhancement models. The raw HR frame describes that a man is adjusting the photography equipment in a room, with shelves in the background.

LR frames. To check the restoration quality, we measure the similarity between the HR frames restored from clean and attacked LR frames. As to the comparison of attack effectiveness between codec hijacking and previous methods, we take three typical baselines: *Iterative Fast Gradient Sign Method* (I-FGSM) (Choi et al. 2019) *Projected Gradient Descent* (PGD) (Madry et al. 2018) and *Feature Disruptive Attack* (FDA) (Ganeshan, S., and Radhakrishnan 2019). We also inspect how codec hijacking impacts the average prediction accuracy of downstream tasks. The attack success rate (%) measures the probability when the accuracy with attacks is at least half lower than that without attacks.

	Attacked Frames		Restored Frames	
Model	PSNR	SSIM	PSNR	SSIM
EDSR	43.06	0.9979	20.97	0.6043
EUSR	35.87	0.9935	20.05	0.5145
DBPN	39.44	0.9949	20.67	0.5667
RCAN	45.32	0.9987	21.01	0.6127
MSRN	41.41	0.9971	19.89	0.5239
ESRGAN	33.90	0.9829	14.13	0.2276
RRDB	32.52	0.9773	20.22	0.5303
CARN	33.99	0.9822	20.96	0.5679
FRSR	35.84	0.9934	19.13	0.4749
NATSR	35.81	0.9933	19.41	0.4848

Table 1: The attacks are hard to detect since there are no perceptible changes on the attacked LR videos.

Attack Impacts on Video Restoration Quality

We inspect how codec hijacking impacts the NeVS performance under different enhancement models. As shown in Table 1, we compare the frame similarity before and after codec hijacking, by checking the attacked LR frames and restored HR frames, respectively. The similarity measurement covers the L_2-based PSNR and human-perception-oriented SSIM. After perturbation injection, the attacked LR frames hold high similarity to the original clean ones, indicating that perturbation injection brings imperceptible changes to the frame content and cannot be detected visually. However, the final restored frames are significantly damaged with huge similarity deterioration, especially to the GAN-based models (*e.g.,* ESRGAN and NATSR). This is because these methods often generate high-frequency information to enhance visual content, where the injected perturbation is amplified by codec hijacking. This phenomenon is best to understand by checking the visual comparison in Figure 5. Compared with the original clean frame, the restored frames in 11 enhancement models consistently suffer from fatal quality distortion, with unexpected moire textures and blurs. This type of distortion not only compromises the visual experience for humans but also eventually results in catastrophic malfunctions of downstream tasks.

Attack Impacts on Downstream Tasks

Apart from the visualization of codec hijacking's attack effectiveness, we further demonstrate how the damaged HR videos mislead downstream perception tasks, *i.e.,* multiple object tracking and human pose estimation. We take ESRGAN as the SR example for illustration purposes.

Setting	Pedestrian	Station	Market	Street
Original	82.9	83.4	79.5	80.3
Attacked	41.2	37.8	31.7	33.2
Success Rate	79.3	81.9	75.1	76.7

Table 2: Malfunction of object tracking on MOTA (%).

Multiple object tracking. Table 2 shows the performance comparison based on the challenging Multiple Object Tracking dataset (MOTChallenge 2023), where the key metric is *Multiple Object Tracking Accuracy* (MOTA) (Bernardin and Stiefelhagen 2008). Overall, codec hijacking breaks the model accuracy with over 75% attack success rate.

Setting	Head	Elbow	Wrist	Knee	Ankle
Original	97.8	90.2	87.1	96.5	91.9
Attacked	38.2	34.8	32.3	36.4	33.7
Success Rate	84.1	85.6	86.7	82.9	83.3

Table 3: Malfunction of human pose estimation on PCK (%).

Human pose estimation. We deploy human pose estimation on the Human3.6M dataset, with 3.6 million video frames. The key metric is *Percentage of Correct Keypoints* (PCK) (Yang and Ramanan 2013). As shown in Table 3, codec hijacking significantly deteriorates the estimation accuracy of different skeleton joints, with over 82% attack success rate.

Ablation Studies

Comparison of attack effectiveness. We compare the attack effectiveness between our codec hijacking and the baselines. As shown in Figure 6, we can observe that all the baselines fail to damage the restored video frames, where slight moire textures and blurs exist. This is because these baselines cannot adapt to the inherent codec of NeVS, which serves as a high-frequency noise filter to remove the injected perturbation. In contrast, our codec hijacking (Figure 6(e)) can retain the perturbation by controlling the QP matrix, thus finally degrading the restoration quality. Note that codec hijacking will also be almost deactivated if we remove the malicious encoding procedure (Figure 6(f)). This comparison verifies our insight that being aware of video codec is the key to successfully launching adversarial attacks on NeVS.

Computational efficiency and time cost. In practice, the adversary can launch codec hijacking to tamper with the LR videos before streaming to the server. Although this attack occurs in the data preparation stage and is *not* time-sensitive, we still restrict the computational overhead of the attack algorithm in two aspects. First, instead of searching for independent perturbation for each frame, we capture the inter-frame similarity and make frames belonging to the same segment share the perturbation. As the segment number is much smaller than the frame number (often in a ratio of $1/100$), searching segment-level perturbation is fast. Second, we use binary search to restrict the computational complexity of QP controlling in each segment as $O(K \times \log_2 \mathcal{S})$, where K and $\mathcal{S}$ represent the macroblock number (*e.g.,* in hundred scales for 1080p) and the initial QP controlling stride (*e.g.,* often a constant of 32), respectively. Thus, the entire computational overhead of the attack algorithm is light, *e.g.,* less than 300ms for a 10-second standard 1080p video. This time cost is feasible for most video-centric applications, especially when compared with the inherent video encoding process, which is $30-40\times$ slower than the attack algorithm.

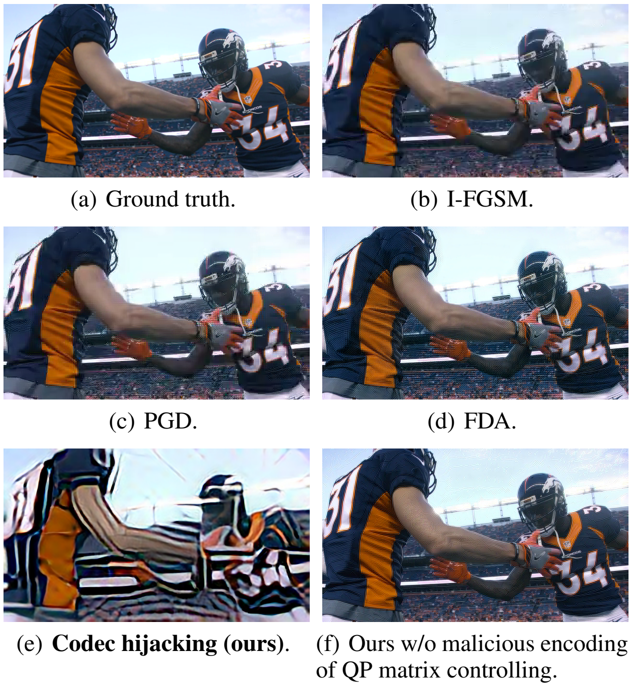

(a) Ground truth.
(b) I-FGSM.
(c) PGD.
(d) FDA.
(e) **Codec hijacking (ours)**.
(f) Ours w/o malicious encoding of QP matrix controlling.

Figure 6: The vulnerability comparison between previous attack methods and codec hijacking. Zoom in for best view.

Conclusion

NeVS has become a fundamental infrastructure to handle video perception applications across the network, where its robustness has not been well explored by previous work. This paper is the first attempt to inspect the vulnerability of NeVS. It reveals that the inherent codec can be the essential trigger to launch covert adversarial attacks, which significantly break the rate-distortion advantages of NeVS. We discover a novel and codec-aware adversarial attack, called codec hijacking, which jointly optimizes the perturbation injection and malicious encoding to launch a successful attack, by exploiting the spatial-temporal prediction and macroblock bit-rate controlling inside the codec component. Codec hijacking exposes the streaming pipeline to a catastrophic zero-day vulnerability, which is hard to defend because there is no visual distortion on the restored video until the attack happens. Evaluations show that codec hijacking explicitly deteriorates the video restoration quality and leads to the malfunction of diverse downstream perception tasks.

Acknowledgements

This research was supported by fundings from the Hong Kong RGC General Research Fund (No. 152211/23E, No. 152203/20E, 152244/21E, 152169/22E, 152228/23E), the National Natural Science Foundation of China (No. 62102327), the PolyU Internal Fund (No. P0043932), the Key-Area Research and Development Program of Guangdong Province (No. 2021B0101400003), Hong Kong RGC Research Impact Fund (No. R5060-19, No. R5034-18), Areas of Excellence Scheme (AoE/E-601/22-R), Shenzhen Science and Technology Innovation Commission (JCYJ20200109142008673).

References

Ahn, N.; Kang, B.; and Sohn, K. 2018. Fast, Accurate, and Lightweight Super-Resolution with Cascading Residual Network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 11214 of *Lecture Notes in Computer Science*, 256–272. Springer.

Bernardin, K.; and Stiefelhagen, R. 2008. Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics. *EURASIP J. Image Video Process.*, 2008.

Chan, K. C. K.; Wang, X.; Yu, K.; Dong, C.; and Loy, C. C. 2021. BasicVSR: The Search for Essential Components in Video Super-Resolution and Beyond. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4947–4956. IEEE.

Chan, K. C. K.; Zhou, S.; Xu, X.; and Loy, C. C. 2022. BasicVSR++: Improving Video Super-Resolution with Enhanced Propagation and Alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 5962–5971. IEEE.

Chen, K.; Wei, Z.; Chen, J.; Wu, Z.; and Jiang, Y. 2022. Attacking Video Recognition Models with Bullet-Screen Comments. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 312–320. AAAI Press.

Choi, J.; Kim, J.; Cheon, M.; and Lee, J. 2020. Deep learning-based image super-resolution considering quantitative and perceptual quality. *Neurocomputing*, 398: 347–359.

Choi, J.; Zhang, H.; Kim, J.; Hsieh, C.; and Lee, J. 2019. Evaluating Robustness of Deep Image Super-Resolution Against Adversarial Attacks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 303–311. IEEE.

Choi, J.; Zhang, H.; Kim, J.; Hsieh, C.; and Lee, J. 2022. Deep Image Destruction: Vulnerability of Deep Image-to-Image Models against Adversarial Attacks. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, 1287–1293. IEEE.

Dasari, M.; Kahatapitiya, K.; Das, S. R.; Balasubramanian, A.; and Samaras, D. 2022. Swift: Adaptive Video Streaming with Layered Neural Codecs. In *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 103–118. USENIX Association.

Ding, D.; Ma, Z.; Chen, D.; Chen, Q.; Liu, Z.; and Zhu, F. 2021. Advances in Video Compression System Using Deep Neural Network: A Review and Case Studies. *Proceedings of the IEEE (Proc. IEEE)*, 109(9): 1494–1520.

Du, K.; Zhang, Q.; Arapin, A.; Wang, H.; Xia, Z.; and Jiang, J. 2022. AccMPEG: Optimizing Video Encoding for Accurate Video Analytics. In *Proceedings of the Machine Learning and Systems (MLSys)*. mlsys.org.

ffprobe. 2023. ffprobe Documentation. https://ffmpeg.org/ffprobe.html. Accessed: 2023-12-22.

Ganeshan, A.; S., V. B.; and Radhakrishnan, V. B. 2019. FDA: Feature Disruptive Attack. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 8068–8078. IEEE.

H.264. 2023. H.264 Official Website. https://www.itu.int/rec/T-REC-H.264. Accessed: 2023-12-22.

H.265. 2023. H.265 Official Website. https://www.itu.int/rec/T-REC-H.265. Accessed: 2023-12-22.

Haris, M.; Shakhnarovich, G.; and Ukita, N. 2018. Deep Back-Projection Networks for Super-Resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1664–1673. Computer Vision Foundation / IEEE Computer Society.

He, Y.; Rahimian, S.; Schiele, B.; and Fritz, M. 2020. Segmentations-Leak: Membership Inference Attacks and Defenses in Semantic Image Segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 12368 of *Lecture Notes in Computer Science*, 519–535. Springer.

Huang, J.; Samplawski, C.; Ganesan, D.; Marlin, B. M.; and Kwon, H. 2020. CLIO: enabling automatic compilation of deep learning pipelines across IoT and cloud. In *Proceedings of the Annual International Conference on Mobile Computing and Networking (MobiCom)*, 58:1–58:12. ACM.

Hwang, J.; Kim, J.; Choi, J.; and Lee, J. 2021. Just One Moment: Structural Vulnerability of Deep Action Recognition against One Frame Attack. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 7648–7656. IEEE.

Jia, S.; Song, Y.; Ma, C.; and Yang, X. 2021. IoU Attack: Towards Temporally Coherent Black-Box Adversarial Attack for Visual Object Tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6709–6718. IEEE.

Jiang, L.; Ma, X.; Chen, S.; Bailey, J.; and Jiang, Y. 2019. Black-box Adversarial Attacks on Video Recognition Models. In *Proceedings of the ACM International Conference on Multimedia (MM)*, 864–872. ACM.

Jiang, X.; Peng, X.; Zheng, C.; Xue, H.; Zhang, Y.; and Lu, Y. 2022. End-to-End Neural Speech Coding for Real-Time Communications. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 866–870. IEEE.

Kurakin, A.; Goodfellow, I. J.; and Bengio, S. 2017. Adversarial Machine Learning at Scale. In *Proceedings of the International Conference on Learning Representations (ICLR)*. OpenReview.net.

Li, J.; Fang, F.; Mei, K.; and Zhang, G. 2018. Multi-scale Residual Network for Image Super-Resolution. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 11212 of *Lecture Notes in Computer Science*, 527–542. Springer.

Li, J.; Li, B.; and Lu, Y. 2021. Deep Contextual Video Compression. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, 18114–18125.

Lim, B.; Son, S.; Kim, H.; Nah, S.; and Lee, K. M. 2017. Enhanced Deep Residual Networks for Single Image Super-Resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1132–1140.

Liu, D.; Li, Y.; Lin, J.; Li, H.; and Wu, F. 2020. Deep Learning-Based Video Coding: A Review and a Case Study. *ACM Comput. Surv.*, 53(1): 11:1–11:35.

Liu, J.; Lu, M.; Chen, K.; Li, X.; Wang, S.; Wang, Z.; Wu, E.; Chen, Y.; Zhang, C.; and Wu, M. 2021. Overfitting the Data: Compact Neural Video Delivery via Content-aware Feature Modulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 4611–4620. IEEE.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *Proceedings of the International Conference on Learning Representations (ICLR)*. OpenReview.net.

MOTChallenge. 2023. Multiple Object Tracking Benchmark. https://motchallenge.net/. Accessed: 2023-12-22.

Nguyen, M.; Çetinkaya, E.; Hellwagner, H.; and Timmerer, C. 2022. Super-resolution based bitrate adaptation for HTTP adaptive streaming for mobile devices. In *Proceedings of the Mile-High Video Conference (MHV)*, 70–76. ACM.

Schwarz, H.; Marpe, D.; and Wiegand, T. 2007. Overview of the Scalable Video Coding Extension of the H.264/AVC Standard. *IEEE Trans. Circuits Syst. Video Technol.*, 17(9): 1103–1120.

Soh, J. W.; Park, G. Y.; Jo, J.; and Cho, N. I. 2019. Natural and Realistic Single Image Super-Resolution With Explicit Natural Manifold Discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 8122–8131. IEEE.

TikTok. 2023. TikTok Official Website. https://www.tiktok.com/about. Accessed: 2023-12-22.

Tu, J.; Ren, M.; Manivasagam, S.; Liang, M.; Yang, B.; Du, R.; Cheng, F.; and Urtasun, R. 2020. Physically Realizable Adversarial Examples for LiDAR Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 13713–13722. IEEE.

Wang, X.; Yu, K.; Wu, S.; Gu, J.; Liu, Y.; Dong, C.; Qiao, Y.; and Loy, C. C. 2018. ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 11133, 63–79. Springer.

Wang, Z.; Luo, Z.; Hu, M.; Wu, D.; Cao, Y.; and Qin, Y. 2022. Revisiting super-resolution for internet video streaming. In *Proceedings of the 32nd Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*.

Wei, Z.; Chen, J.; Goldblum, M.; Wu, Z.; Goldstein, T.; and Jiang, Y. 2022. Towards Transferable Adversarial Attacks on Vision Transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2668–2676.

Yang, W.; Xia, S.; Liu, J.; and Guo, Z. 2019. Reference-Guided Deep Super-Resolution via Manifold Localized External Compensation. *IEEE Trans. Circuits Syst. Video Technol.*, 29(5): 1270–1283.

Yang, Y.; and Ramanan, D. 2013. Articulated Human Detection with Flexible Mixtures of Parts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(12): 2878–2890.

Yeo, H.; Jung, Y.; Kim, J.; Shin, J.; and Han, D. 2018. Neural Adaptive Content-aware Internet Video Delivery. In *Proceedings of the USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 645–661. USENIX Association.

Yeo, H.; Lim, H.; Kim, J.; Jung, Y.; Ye, J.; and Han, D. 2022. NeuroScaler: neural video enhancement at scale. In *Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication (SIGCOMM)*, 795–811. ACM.

Yin, M.; Zhang, Y.; Li, X.; and Wang, S. 2018. When Deep Fool Meets Deep Prior: Adversarial Attack on Super-Resolution Network. In *Proceedings of the ACM Multimedia Conference on Multimedia Conference (MM)*, 1930–1938.

YouTube. 2023. YouTube Live Streaming Official Website. https://www.youtube.com/live. Accessed: 2023-12-22.

Yue, J.; Li, H.; Wei, P.; Li, G.; and Lin, L. 2021. Robust Real-World Image Super-Resolution against Adversarial Attacks. In *Proceedings of the ACM Multimedia Conference (MM)*, 5148–5157. ACM.

Zhang, A.; Wang, C.; Han, B.; and Qian, F. 2021. Efficient Volumetric Video Streaming Through Super Resolution. In *Proceedings of the International Workshop on Mobile Computing Systems and Applications (HotMobile)*, 106–111. ACM.

Zhang, A.; Wang, C.; Han, B.; and Qian, F. 2022. YuZu: Neural-Enhanced Volumetric Video Streaming. In *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 137–154. USENIX Association.

Zhang, Y.; Li, K.; Li, K.; Wang, L.; Zhong, B.; and Fu, Y. 2018. Image Super-Resolution Using Very Deep Residual Channel Attention Networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 11211 of *Lecture Notes in Computer Science*, 294–310. Springer.

Zheng, H.; Zuo, W.; and Zhang, L. 2020. BS-MCVR: Binary-sensing based Mobile-cloud Visual Recognition. In *Proceedings of the ACM International Conference on Multimedia (MM)*, 1339–1347. ACM.

Zoom. 2023. Zoom Meeting Official Website. https://zoom.us/. Accessed: 2023-12-22.