A Novel Deep Video Watermarking Framework with Enhanced Robustness to H.264/AVC Compression

Yulin Zhang
Sun Yat-sen University
zhangylin5@mail2.sysu.edu.cn

Jiangqun Ni*
Sun Yat-sen University
Peng Cheng Laboratory
issjqni@mail.sysu.edu.cn

Wenkang Su
Sun Yat-sen University
Guangzhou University
suwk3@mail.sysu.edu.cn

Xin Liao
Hunan University
xinliao@hnu.edu.cn

ABSTRACT

The recent success of deep image watermarking has demonstrated the potential of deep learning for watermarking, which has drawn increasing attention to deep video watermarking with the objective to improve its robustness and perceptual quality. Compared to images, video watermarking is much more challenging due to the rich structures of video data and the diversity of attacks in video transmission pipeline. The existing deep video watermarking schemes are far from satisfactory in dealing with temporal attacks, e.g., frame averaging, frame dropping and transcoding. To this end, a novel deep framework for **R**obustness **E**nhanced **V**ideo water**mark**ing (REVMark) is proposed in this paper, aiming at improving the overall robustness, especially in dealing with H.264/AVC compression, while maintaining good visual quality. REVMark has an encoder/decoder structure with a pre-processing block (TAsBlock) to effectively extract the temporal-associated features on aligned frames. To ensure the end-to-end robust training, a distortion layer is integrated into the REVMark to resemble various attacks in real-world scenarios, among which, a new differentiable simulator of video compression, namely DiffH264, is developed to approximately simulate the process of H.264/AVC compression. In addition, the mask loss is incorporated to guide the encoder to embed the watermark in the human-imperceptible regions, thus improving the perceptual quality of the watermarked video. Experimental results demonstrate that the proposed scheme can outperform other SOTA methods while achieving 10× faster inference.

CCS CONCEPTS

• **Security and privacy** → **Security services**; **Social aspects of security and privacy**.

KEYWORDS

video watermarking, deep learning, temporal features, video compression simulator

ACM Reference Format:
Yulin Zhang, Jiangqun Ni, Wenkang Su, and Xin Liao. 2023. A Novel Deep Video Watermarking Framework with Enhanced Robustness to H.264/AVC Compression. In *Proceedings of the 31st ACM International Conference on Multimedia (MM '23), October 29–November 3, 2023, Ottawa, ON, Canada.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3581783.3612270

1 INTRODUCTION

Digital video technology has been widely applied in various fields, such as instant messaging, digital surveillance, digital entertainment, and digital advertising. In recent years, the popularity of short videos on social networks has sparked a growth in the development of digital video technology. As a critical role in digital content security, digital watermarking [9, 12, 17, 24, 39] provides a variety of protections for digital media, including but not limited to owner identification, copyright protection, authenticity verification, and tracking. In watermarking pipeline, the watermark message is robustly embedded into digital media to survive multiple attacks such as noise, filtering, compression, statistical averaging attacks, etc. In addition to the aforementioned attacks, digital video watermarking [2, 3, 7, 8, 27] also needs to handle various temporal attacks, including frame dropping, video compression, etc. To ensure the overall robustness, conventional approaches typically resort to transforming techniques, such as Discrete Cosine Transform (DCT) [26, 31, 44], Discrete Fourier Transform (DFT) [10, 28, 35], Discrete Wavelet Transform (DWT) [5, 6, 22] and hybrid transform [15, 18, 34, 40, 43], to obtain the transform domain coefficients of the digital signal and then apply watermarking according to well-designed rules. However, these methods are often complex and limited to certain specific attacks.

Thanks to advances in deep learning, end-to-end deep watermarking frameworks [13, 19, 23, 36, 41] have been developed to ensure the overall robustness while maintaining quality. These frameworks typically consist of an encoder, a decoder, and a differentiable distortion layer. With such effective and practical architecture, deep learning-based image watermarking methods such as HiDDeN [45], ReDMark [1], and StegaStamp [33] have proven success in improving the robustness against various image attacks. Due to the complexity of the video data and the diversity of attacks

[*] Corresponding author.

MM '23, October 29–November 3, 2023, Ottawa, ON, Canada.

ACM ISBN 979-8-4007-0108-5/23/10…$15.00
https://doi.org/10.1145/3581783.3612270

in digital video transmission, deep video watermarking research remains a challenging task and has attracted increasing attention from researchers. DVMark [25] proposed a multi-scale end-to-end framework wherein the distortion layer contained a series of spatial and temporal attacks for video, namely frame averaging, frame dropping, frame swapping, random crop, color dithering, Gaussian noise, Gaussian blur, and simulated video compression. As the real video compression is non-differentiable, DVMark trained a network as differentiable proxy to simulate the compressed video for a given input. Based on these designs, DVMark reported encouraging experimental performance. RIVIE [20] proposed a spatio-temporal generator to hide information into the video sequence through a frame-by-frame manipulation. The involved distortion layer adopted differentiable 3D rendering to simulate the camera imaging process in which a screen image was captured by a camera. This approach is capable of facilitating the communication from screens to cameras without visual quality degradation.

In general, existing deep video watermarking approaches are not adequately robust against temporal attacks, especially video compression. This can be attributed to the insufficient exploitation of video temporal features and the lack of dedicated video compression simulators for end-to-end robust training. To this end, we propose a novel deep framework for **R**obustness **E**nhanced **V**ideo water**mark**ing (REVMark), and the main contributions of our work are summarized as follows:

- The proposed REVMark exhibits enhanced robustness to various attacks, especially H.264/AVC compression, along with promising visual quality.
- The proposed temporal-associated feature extraction block (TAsBlock) enables feature extraction on aligned frames, which can deliver well-founded features to REVMark for robust message embedding and extraction.
- We construct a differentiable video compression simulator named DiffH264, which can be directly integrated into the distortion layer to enable the robust training of REVMark against H.264/AVC compression.
- The mask loss for deep video watermarking is formulated by jointly exploiting the spatial and temporal masks, which encourages the encoder to embed message in the human-imperceptible regions, thus improving the perceptual quality of watermarked videos.

2 PRELIMINARIES

2.1 H.264/AVC Coding

H.264/AVC [37] is currently the popular video compression standard. The H.264/AVC encoder uses both intra-frame and inter-frame compression techniques to improve compression efficiency while maintaining high video quality. The technical details are depicted in Figure 1.

Intra-frame compression exploits the spatial redundancy within a video frame to compress the first frame in the Group Of Pictures (GOP). Specifically, based on variable block-size partition, the pixel values of the current coding block are predicted by the neighboring reconstructed blocks. Subsequently, the prediction residual is obtained by subtracting the prediction block from the current block, and then transformed, quantized, and entropy coded to achieve compression. To obtain the reconstructed block for further prediction, the prediction residual is reconstructed by dequantization and inverse transformation on quantization coefficients.

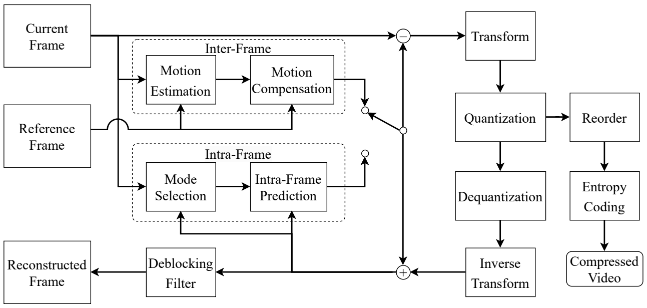

Figure 1: H.264/AVC encoder block.

Inter-frame compression exploits the temporal redundancy between adjacent video frames to compress the frames other than the first one in the GOP. First, a frame (or two frames for bi-prediction) is selected as the reference frame in the GOP. Based on the variable block-size partition, the motion vectors (MVs) between the current coding frame and the reference frame are searched in motion estimation process, and then the prediction of the current frame is obtained by applying motion compensation to the reference frame. With motion estimation and motion compensation techniques, the H.264/AVC encoder only needs to compress the MVs and the residual between the current frame and its prediction, significantly reducing the bit rate. Finally the reconstructed frame is obtained following a similar approach to the reconstruction process in intra-frame compression.

To enhance the robustness of a deep video watermarking framework against H.264/AVC compression, the compression distortion should be introduced into the distortion layer to facilitate the joint learning of the encoder and decoder. Considering that real H.264/AVC compression is not differentiable, it is crucial to integrate a differentiable H.264/AVC compression simulator into the deep video watermarking framework for robust training.

2.2 Differentiable JPEG Approximation

Introducing differentiable JPEG compression into the training process of deep watermarking frameworks has been acknowledged as a promising technique to enhance the robustness against the anticipated distortion caused by JPEG attack. With the aim of achieving this, Zhu *et al.* [45] suggested employing JPEG-Mask and JPEG-Drop, to roughly approximate the JPEG compression in a differentiable approach. However, there is a significant disparity between this approximation and real JPEG compression, leading to suboptimal results. To faithfully simulate the JPEG compression process, Shin *et al.* [32] proposed a differentiable approximation in accordance with the main procedures in real JPEG compression. Given that the rounding operation in the quantization process of JPEG compression is non-differentiable, in [32], it was approximated by a piecewise function, i.e.,

$$round(x) = \begin{cases} x^3, & |x| < 0.5 \\ x, & |x| \geq 0.5 \end{cases}. \tag{1}$$

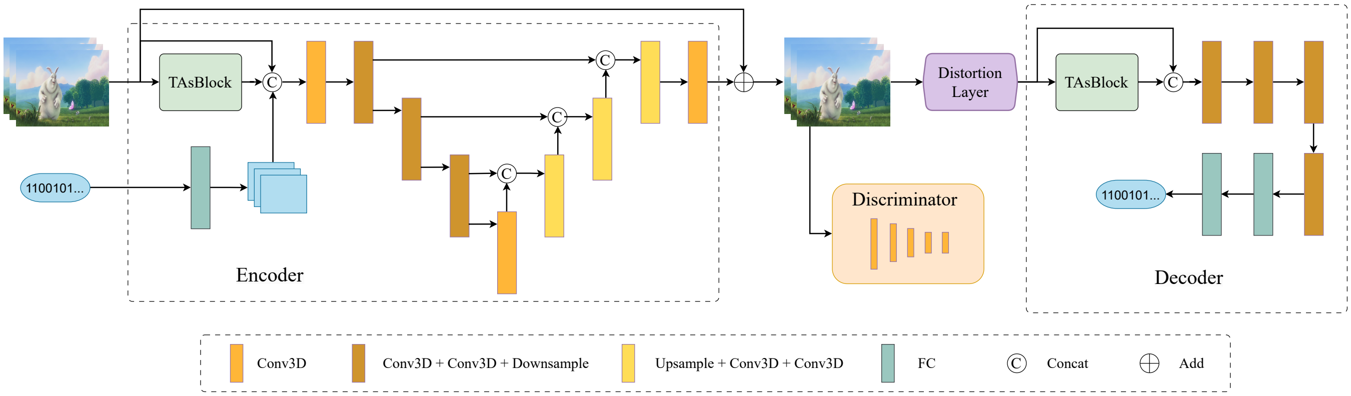

Figure 2: The overall framework of REVMark. Firstly, for an input video sequence and watermark message, the encoder generates the scaled watermark residual and adds it to the original input. Then the watermarked video undergoes various attacks in the differentiable distortion layer. Finally, the decoder recovers the message from the distorted watermarked video.

This differentiable scheme achieved a faithful and modularized simulation of JPEG compression, thus enabling the watermarking network in end-to-end robust training.

2.3 Optical Flow Estimation

As an important task in computer vision, optical flow estimation methods [4, 14, 16] are commonly utilized to compute the motion between two frames in a video, which has been widely applied in various fields, e.g., motion estimation, video compression, video stabilization, and target tracking. Optical flow is a two-dimensional vector field that consists of displacement vectors representing the motion of pixel points from the first frame to the second frame. The advent of deep learning has brought about enhanced accuracy and robustness in optical flow estimation [11, 29, 38]. Ranjan *et al.* [29] combined spatial pyramids with deep learning to propose SPyNet, which is capable of estimating optical flow more accurately through a coarse-to-fine architecture. As a lightweight model, SPyNet exhibits flexibility and efficiency, which allows it to be embedded into a variety of applications. As such, a pre-trained SPyNet is leveraged in our framework for optical flow estimation.

3 PROPOSED FRAMEWORK

The proposed REVMark consists of an encoder, a differentiable distortion layer and a decoder, and its overall robustness is improved by end-to-end training, as the architecture illustrated in Figure 2. To improve the robustness of REVMark in temporal domain, we propose a **T**emporal-**As**sociated feature extraction **Block** (TAsBlock) to serve as a component of the encoder and decoder. To enhance the robustness of REVMark against video compression attack, the differentiable video compression simulator DiffH264 is designed. In addition, to further boost the visual quality of the watermarked video, the mask loss for video watermarking is formulated. The specific design will be expounded sequentially below.

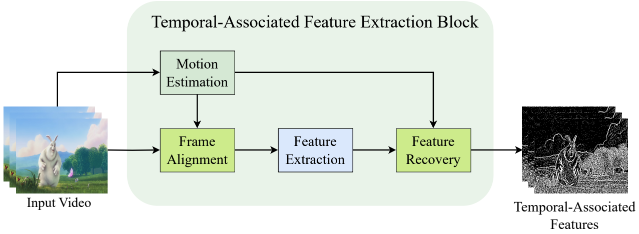

Figure 3: The structure of the proposed TAsBlock. In motion estimation, the optical flow is obtained and then employed for frame alignment and feature recovery.

3.1 Temporal-Associated Feature Extraction Block

Due to the pixel shift between adjacent frames caused by object motion in video, the correlation of temporally adjacent pixels is weakened, which poses an obstacle for the convolutional network to extract effective temporal features in regions with motion. Therefore, it is necessary to move the shifted pixels back to the original positions (noted as frame alignment) before feature extraction. To this end, we develop a temporal-associated feature extraction block (TAsBlock), as shown in Figure 3. Specifically, first, a frame in the input video sequence is selected as the target frame, and the optical flow estimation network SPyNet[29] is introduced to estimate the optical flow between the target frame and each of the other non-target frames. Next, the non-target frames are aligned towards the target frame through warping operation. After that, two successive convolution operations are performed on the frame-aligned video sequence to extract the temporal-associated features of the same size as the input. Finally, an inverse operation of frame alignment (noted as feature recovery) is performed on the temporal-associated features to spatially match the input video for fusion in subsequent procedures. It should be noted that the feature recovery operation only adjusts the position of feature points and does not change their values.

The introduction of TAsBlock in the video watermarking task is dedicated to providing well-founded temporal features to the encoder and decoder. The frame alignment can be regarded as an operation to transform video data into the deformation space, enabling a well-constructed representation of the original video. The succeeding feature extraction on the frame-aligned video is designed to characterize the correlations and prior knowledge of the video sequence. Specifically, there are temporal differences between aligned frames, which can be viewed as the inter-frame prediction residual in video compression. Note that lossy compression on the prediction residual results in temporal distortion of the compressed video. To sum up, the temporal distortion can be predicted by the temporal differences between aligned frames. Therefore, with the features captured on the frame-aligned video sequence, the encoder can estimate the distortion that would result from video compression and thus predictably "select" the embedding domain. In addition, the frame-aligned video sequence can be considered as a series of aligned images, which allows the decoder to perform multi-image noise estimation and denoising, aiming to boost the decoding accuracy of the watermark message.

3.2 Encoder

The encoder takes the original video sequence and watermark message as input and outputs content-adaptive watermark residual. As structured in Figure 2, the encoder employs U-net[30] as the backbone with 3D convolutional layers and LeakyReLU as the basic building blocks. Integrating the proposed TAsBlock, the encoder can robustly embed the watermark message to video through a multi-scale generation approach.

The watermark message takes the form of a string of binary bits. In message pre-processing, the message goes through a fully connected layer to obtain a message vector. Then it is reshaped and upsampled in spatial-temporal dimensions to fit the video shape $C \times T \times H \times W$. In watermark embedding, the enlarged message is concatenated with the input video sequence and the extracted temporal-associated features to be fed into the multi-scale encoder. Then the generated watermark residual on each frame is scaled to unit length to equally distribute the embedding modifications among all frames. Note that the PSNR of the watermarked video is fixed through the scaling operation. Finally, the scaled watermark residual is multiplied by a strength factor α and added to the original video to obtain the watermarked video, i.e.,

$$F_t^w = F_t + \widetilde{R}_t = F_t + \alpha \times \frac{clip_\sigma(R_t)}{\|clip_\sigma(R_t)\|_2}, \tag{2}$$

where F_t denotes the t^{th} frame of the video, F_t^w denotes the corresponding watermarked frame and R_t denotes the corresponding watermark residual generated by encoder, $clip_\sigma$ is the sigma-clipping method to remove outliers from R_t for stabilized training.

3.3 Distortion Layer

To improve the robustness of REVMark in spatial and temporal domains, the distortion layer is designed to comprise a range of differentiable spatial and temporal attacks, namely frame averaging, frame dropping, frame swapping, random crop, Gaussian noise, 3D Gaussian blur, and simulated video compression. In the training phase, the distortion layer carries out one of these attacks each time according to their respective selection probability. In order to make REVMark more attentive to the distortion caused by video compression, the selection probability of video compression simulator is set to 0.86, and the selection probabilities of the other attacks are equally distributed. Note that the distortion layer is disabled at the beginning of the training phase to exclusively induce high accuracy of message extraction.

We will elaborate a detailed explanation for each attack method in the distortion layer. For frame averaging, the temporal moving window size N is set to 3. For frame dropping, a frame is dropped with the probability $p = 0.5$. In frame swapping, a frame is swapped with its immediate neighbor with the probability $p = 0.5$. In random crop, the frames are cropped in width and height with the proportion $p = 0.4$. The standard deviation σ of 3D Gaussian blur and Gaussian noise is set to 2.0 and 0.04 respectively, and the kernel size k of 3D Gaussian blur is set to $3 \times 3 \times 3$.

Differentiable H.264/AVC Compression. The proposed differentiable H.264/AVC compression simulator (DiffH264) is composed of two parts, i.e., the simulated intra-frame and inter-frame compression. Each frame in the video sequence will go through the simulated intra-frame or inter-frame compression to acquire the reconstructed frame, which is the simulation of the distorted frame under real H.264/AVC compression. In pre-processing of the simulator, the video sequence is converted to YUV color space. Once the frames have been reconstructed, they will be converted back to RGB color space.

The simulated intra-frame compression is applied to the reference frame of the input video sequence. It is randomly selected to address the situation that different GOP may be adopted in real H.264/AVC compression. Then the reference frame is split to non-overlapping 8×8 blocks. In H.264/AVC, intra-frame compression is based on variable block-size partition and is sequentially applied to each partition block, that is, the current coding block can only be predicted after the previous coding block has been reconstructed. This mechanism improves the compression efficiency but is quite time-consuming when implemented to deep learning pipelines. As a compromise, we choose to simultaneously reconstruct all of the 8×8 blocks in reference frame. In our design, firstly, we implement discrete cosine transformation on the reference frame F_{ref}. For a block $B_{i,j}$ at the position (i, j) in F_{ref}, its DCT coefficients are obtained by

$$C_{i,j} = dct(B_{i,j}). \tag{3}$$

Afterwards, the differentiable quantization is performed on the DCT coefficients $C_{i,j}$ to simulate the lossy compression step

$$Q_{i,j} = round(\frac{C_{i,j}}{T_{quant}}), \tag{4}$$

where T_{quant} represents the quantization table and $round$ is the differentiable function in Equation 1. We employ the function $round$ to approximate the non-differentiable rounding operation in the quantization process to achieve the differentiable quantization. Finally the reconstructed block can be acquired through dequantization and inverse discrete cosine transformation

$$\widehat{B}_{i,j} = idct(Q_{i,j} \odot T_{quant}), \tag{5}$$

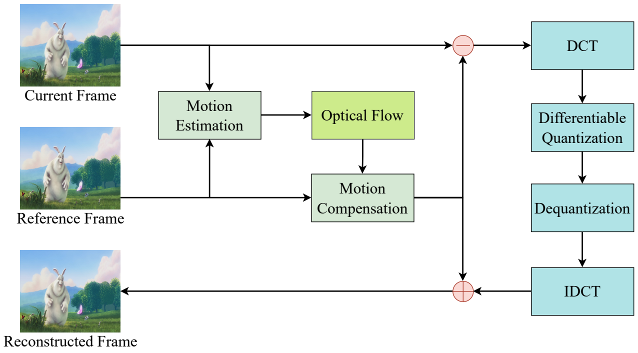

Figure 4: The simulated inter-frame compression in DiffH264. The optical flow is obtained by an optical flow estimation network and is employed for motion compensation.

where ⊙ represents the Hadamard product (element-wise product). In summary, at the frame level, the simulated intra-frame compression can be represented as

$$\widehat{F}_{ref} = C_{intra}(F_{ref}). \tag{6}$$

The simulated inter-frame compression is applied to the non-reference frames in video sequence, as the procedures illustrated in Figure 4. In specific, each non-reference frame is fed into the optical flow network $\mathcal{N}$ coupled with the reference frame to estimate the optical flow between the two frames:

$$MV_{cur} = \mathcal{N}(F_{ref}, F_{cur}), \tag{7}$$

where F_{cur} denotes the current non-reference frame to be compressed. Then a warping operation is performed on the reference frame to acquire the prediction of the current frame

$$F_{pred} = warp(F_{ref}, MV_{cur}), \tag{8}$$

which simulates the motion compensation process in video compression through a differentiable manner. Subsequently the prediction residual between F_{cur} and F_{pred} is reconstructed by the aforementioned simulated intra-frame compression

$$\widehat{F}_{res} = C_{intra}(F_{cur} - F_{pred}). \tag{9}$$

Finally, the reconstruction of the current frame is obtained,

$$\widehat{F}_{cur} = C_{inter}(F_{ref}, F_{cur}) = F_{pred} + \widehat{F}_{res}, \tag{10}$$

which reflects the influence of temporal distortion on the current frame.

This design conforms the basic concept of H.264/AVC, which can act as a dedicated and interpretable approximation of real video compression.

3.4 Decoder

As shown in Figure 2, the decoder consists of a TAsBlock, 3D convolutional layers and fully connected layers, which takes the distorted watermarked video as input to recover the watermark message robustly. During the decoding process, the temporal-associated features of the input video are extracted by TAsBlock and concatenated with the input video for subsequent fusion. Then the convolutional layers and fully connected layers are sequentially applied to extract the message bits. By jointly training the decoder and encoder, the watermark message can be accurately extracted.

3.5 Discriminator

To prompt the generation quality of the encoder, a 5-layer 3D CNN is constructed as the video discriminator to distinguish the original video from the generated watermarked video. By the adversarial training, the encoder and discriminator are alternately optimized and the quality of the watermarked video is consequently improved.

3.6 Loss Function

The proposed loss function comprises three parts: the message loss, the adversarial loss, and the mask loss. The message loss is adopted to ensure the decoding accuracy of the watermark message and is calculated by the cross-entropy between the input message m and the decoded message m',

$$\mathcal{L}_{msg} = CrossEntropy(m, m'). \tag{11}$$

The adversarial loss is leveraged to enhance the visual quality of the watermarked video. Through adversarial training, the generator loss $\mathcal{L}_{adv}$ and the discriminator loss $\mathcal{L}_D$ are alternately optimized, which are defined as

$$\mathcal{L}_{adv} = \mathcal{D}(V^w), \tag{12}$$

$$\mathcal{L}_D = \mathcal{D}(V) - \mathcal{D}(V^w), \tag{13}$$

where V and V^w refer to the input and watermarked video respectively, and $\mathcal{D}$ refers to the discriminator.

The mask loss is developed to improve the perceptual quality of the watermarked video via spatial and temporal masks to guide the encoder to embed message in regions where human eye is insensitive. Considering that human eye is less sensitive to the regions with complex textures, we first construct a spatial mask based on the texture information, i.e.,

$$\mathcal{M}_S(F_t) = |\mathcal{S}^h * F_t| + |\mathcal{S}^v * F_t|, \tag{14}$$

$$\widetilde{\mathcal{M}_S}(F_t) = \frac{clip_\sigma(\mathcal{M}_S(F_t))}{\|clip_\sigma(\mathcal{M}_S(F_t))\|_2}, \tag{15}$$

where F_t is the t^{th} frame of the video, $\mathcal{S}^h$ and $\mathcal{S}^v$ denotes the horizontal 3 × 3 Sobel operator and vertical 3 × 3 Sobel operator, respectively. The corresponding spatial mask loss is formulated as

$$\mathcal{L}_{M_S}(\widetilde{R}_t) = \|\max(0, |\widetilde{R}_t| - \widetilde{\mathcal{M}_S}(F_t))\|_2, \tag{16}$$

where $\widetilde{R}_t$ denotes the final watermark residual of the t^{th} frame. Supervised by $\mathcal{L}_{M_S}$, the encoder is encouraged to embed message in regions with rich textures as a result of the increased modification loss in smooth regions. On the other hand, human eye is also sensitive to temporal flickers in the video, thus a temporal mask is constructed as the difference between two adjacent video frames:

$$\mathcal{M}_T(F_t, F_{t+1}) = |F_{t+1} - F_t|, \tag{17}$$

by which, the corresponding temporal mask loss is formulated as

$$\mathcal{L}_{M_T}(F^w_t, F^w_{t+1}) = \\ \|\max(0, \mathcal{M}_T(F^w_t, F^w_{t+1}) - \mathcal{M}_T(F_t, F_{t+1}))\|_2. \tag{18}$$

According to the temporal mask loss, when the temporal difference between two adjacent frames of the original video is insignificant, the temporal difference of the watermarked video in these regions

Table 1: Comparison of decoding accuracy (%) between REVMark and DVMark under different attacks.

Framework	Frame Average ($N = 3$)	Frame Drop ($p = 0.5$)	Frame Swap ($p = 0.5$)	Random Crop ($p = 0.4$)	Gaussian Noise ($\sigma = 0.04$)	Gaussian Blur ($\sigma = 2.0$)	H.264/AVC ($CRF = 22$)
DVMark	98.10	98.99	99.35	**97.06**	98.56	98.09	92.94
REVMark	**99.98**	**99.81**	**99.98**	83.31	**99.97**	**99.82**	**95.84**

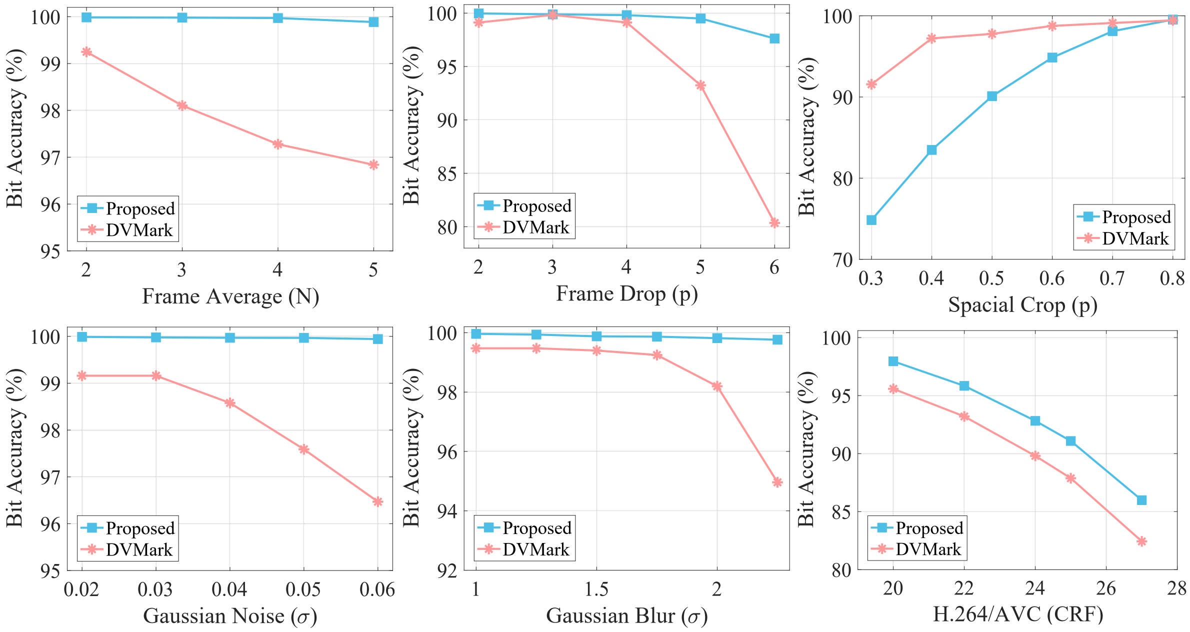

Figure 5: Decoding accuracy of REVMark and DVMark under different attacks and attack intensities.

should also be insignificant, without producing temporal flickers. Combining the spatial mask loss $\mathcal{L}_{M_S}$ and the temporal mask loss $\mathcal{L}_{M_T}$, the full mask loss $\mathcal{L}_{mask}$ is defined as:

$$\mathcal{L}_{mask} = \frac{\lambda_{M_S}}{NT}\sum_{t=0}^{T}\mathcal{L}_{M_S}{}^2(\widetilde{R}_t) + \frac{\lambda_{M_T}}{N(T-1)}\sum_{t=0}^{T-1}\mathcal{L}_{M_T}{}^2(F_t^w, F_{t+1}^w). \tag{19}$$

where N denotes the number of pixels in a frame.

Finally, the joint encoder/decoder loss function is summarized as follows:

$$\mathcal{L}_{total} = \lambda_1\mathcal{L}_{msg} + \lambda_2\mathcal{L}_{adv} + \lambda_3\mathcal{L}_{mask}. \tag{20}$$

4 EXPERIMENTS

4.1 Experimental Settings

In this section, the dataset, training details and evaluation metrics in our experiments are introduced, with reference to which the performance of REVMark and its variants are evaluated.

Dataset. We train and evaluate the proposed REVMark on the Kinetics-400 dataset [21]. All the videos are cropped to the size of 128×128, and each input video sequence is composed of 8 cropped frames, with a payload of 96 bits for message embedding.

Training Details. The whole watermarking framework is implemented by PyTorch and trained on one NVIDIA GTX TITAN X GPU for a total of 60K steps. During encoder/decoder training, the batch size is set to 16 and the learning rate is set to $1e^{-4}$. We choose Adam as the optimizer and apply an exponential decay with decay rate 0.5 every 30K steps. For adversarial training, the discriminator is trained for one step after every encoder/decoder step, with the initial learning rate of $1e^{-5}$ and the same weight decay strategy to encoder/decoder.

In the training phase, λ_{M_S} and λ_{M_T} in mask loss is set to 200 and 5 respectively, and the message loss weight λ_1 is fixed to 1. To ensure that the message is extracted accurately, λ_2 and λ_3 are initialized as 0 and the watermark strength factor α is initialized as 20 in the beginning of the training phase. Then λ_2 and λ_3 are gradually increased to $5e^{-4}$ and 10 while α is gradually decreased to 6.2 to promote the quality of the watermarked video.

Evaluation Metrics. As comparative experiments, the performance of REVMark and the state-of-the-art (SOTA) approach, i.e., DVMark [25], is evaluated under the same video size and payload. We measure the bit accuracy of the decoded message, the PSNR and LPIPS loss [42] of the watermarked video for quantitative evaluation.

4.2 Robustness Evaluation

Robustness is evaluated by the decoding accuracy under various attacks, including frame averaging, frame dropping, frame swapping,

Table 2: Quality comparison of the watermarked videos generated by REVMark and DVMark.

Framework	PSNR ↑	LPIPS [42]×100 ↓
DVMark	37.0	5.70
REVMark	**37.5**	**2.96**

random crop, Gaussian noise, 3D Gaussian blur, and H.264/AVC compression. Table 1 collects the corresponding performance comparison of the proposed REVMark and DVMark under specific attack intensity. It is observed that benefiting from the proposed TAsBlock and DiffH264, REVMark shows superior performance to DVMark under most attacks, especially video compression.

For random crop attack, REVMark demonstrates inferior performance, which is due to the considerable difference between REVMark and DVMark in terms of message pre-processing. In particular, DVMark expands the message by duplicating every bit along the spatio-temporal dimensions before embedding. This mechanism enables every bit of the message to spread over the entire video sequence. REVMark maps and up-samples the message to the shape of the video sequence. As a result, every bit of the message is limited to local areas after pre-processing, which weakens the robustness of REVMark against random spatial crop attack. However, considering that the message pre-processing in DVMark will results in higher computational demand and greater difficulty for encoder/decoder training, we still choose to map and up-sample the message to accelerate the convergence and facilitate network training.

To further evaluate the overall robustness of REVMark, we report the decoding accuracy under various attacks at different intensities, as shown in Figure 5. It can be seen that REVMark is consistently robust to a wide range of attack intensities. It becomes the most pronounced for temporal attacks and Gaussian noise, when compared to DVMark. Although the decoding accuracy is susceptible to the Constant Rate Factor (CRF) in video compression, it can be reliably maintained at modest CRFs. More importantly, the decoding accuracy can be competitive with DVMark at modest random crop intensity, e.g., $p \geq 0.7$.

4.3 Video Quality Evaluation

The video quality comparison of REVMark and DVMark with respect to the above robustness performance is presented in Table 2. It shows that REVMark exhibits superior performance in both PSNR and LPIPS loss of generated watermarked video, especially in LPIPS loss. The watermarked video samples in Figure 6 indicate that the encoder of REVMark indeed generates content-adaptive watermark residual, which contributes to boosting the perceived quality of the watermarked video.

4.4 Robustness-Quality-Payload Trade-off

In this section, we investigate the trade-off among robustness, video quality and payload in watermarking applications. In order to evaluate the robustness of our watermarking framework under various watermarked video qualities, we adjust the watermark strength α without re-training the framework and evaluate the decoding accuracy under H.264/AVC compression. The experimental results are summarized in Table 3, providing insights into selecting appropriate watermarking strength for specific applications.

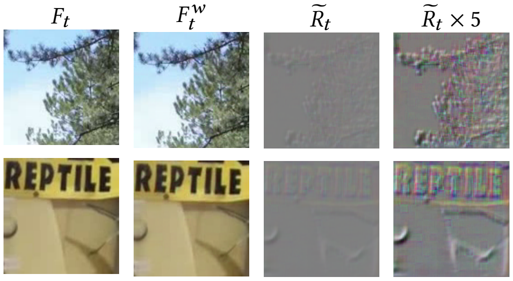

Figure 6: Samples of the watermarked videos generated by the proposed REVMark.

Table 3: PSNR versus decoding accuracy (%) of REVMark under H.264/AVC Compression.

PSNR	38.62	38.08	37.53	36.98	36.47
Acc.	93.97	95.07	95.84	96.74	97.37
LPIPS	2.30	2.67	2.96	3.29	3.81

Table 4: Payload (bits) versus decoding accuracy (%) of REVMark under H.264/AVC Compression.

Payload	64	80	96	112	128
Acc.	96.67	96.55	95.84	95.55	94.94
PSNR	37.57	37.57	37.53	37.59	37.53
LPIPS	2.84	3.19	2.96	4.03	3.82

To explore the trade-off between payload and robustness, we train our framework for each separate payload and evaluate the decoding accuracy under H.264/AVC compression. From the experimental results in Table 4, it can be observed that the proposed framework maintains strong robustness even at high levels of payload.

4.5 Ablation Study

The ablation experiments are carried out to investigate the effectiveness of the proposed components in REVMark, including TAsBlock, DiffH264 and the mask loss. In the ablation study of each component, the variants of REVMark are constructed and trained under the same experimental settings. In the evaluation of robustness, the quality of the watermarked video is fixed by finetuning the variants and adjusting the watermark strength α. Correspondingly, when the quality of the watermarked video is evaluated, the decoding accuracy is fixed by adjusting α.

Effect of TAsBlock. We design two variants including the complete REVMark and REVMark with TAsBlock removed, and then report the decoding accuracy of each trained variant under four different attack methods, i.e., frame dropping, Gaussian noise, 3D Gaussian blur and H.264/AVC compression. As presented in Table 5, armed with TAsBlock, the decoding accuracy of REVMark

Table 5: Decoding accuracy (%) of REVMark with (w/) and without (w/o) TAsBlock under four different attack methods.

Variant	Frame Drop ($p = 0.5$)	Gaussian Noise ($\sigma = 0.10$)	Gaussian Blur ($\sigma = 2.0$)	H.264/AVC ($CRF = 22$)
w/o TAsBlock	99.62	98.91	**99.87**	94.31
w/ TAsBlock	**99.81**	**99.73**	99.82	**95.84**

Table 6: Decoding accuracy (%) of REVMark trained with different video compression simulators under four different attack methods.

Simulator	Frame Drop ($p = 0.5$)	Gaussian Noise ($\sigma = 0.04$)	Gaussian Blur ($\sigma = 2.0$)	H.264/AVC ($CRF = 22$)
CompNet [25]	99.65	99.89	99.80	79.12
DiffJPEG [32]	99.79	99.31	74.31	87.44
DiffH264	**99.81**	**99.97**	**99.82**	**95.84**

Table 7: The video quality of REVMark with (w/) and without (w/o) the proposed mask loss $\mathcal{L}_{mask}$.

Variant	PSNR ↑	LPIPS [42] ×100 ↓
w/o mask loss	37.4	4.70
w/ mask loss	**37.5**	**2.96**

against H.264/AVC compression is visibly elevated. Consistent with our expectations, the proposed TAsBlock can extract well-founded temporal features on the frame-aligned video sequence, which allows the encoder and decoder to take advantage of this effective information and facilitates enhanced robustness of REVMark.

Effect of DiffH264. To verify the effectiveness of the proposed DiffH264, we construct the differentiable proxy network in DVMark (denoted as CompNet) and differentiable JPEG [32] (denoted as DiffJPEG) for comparison. They are employed to act as the video compression simulator in the distortion layer to train the proposed REVMark. As shown in Table 6, compared to CompNet and DiffJPEG, DiffH264 can significantly enhance the robustness of REVMark against H.264/AVC compression.

We conduct an analysis for the performance. The CompNet attempts to learn the transformation from the original video to the compressed video, however, it is difficult to directly model the transformation mechanism of H.264/AVC compression, such as the quantization distortion of DCT coefficients and the temporal distortion from inter-frame compression. The DiffJPEG is also incapable of handling temporal distortion. The proposed DiffH264 complies with the main principles of H.264/AVC to systematically simulate the intra-frame and inter-frame compression with a differentiable approximation. This enables it to accomplish a faithful video compression simulator and promote the robustness of REVMark against H.264/AVC compression.

Effect of the Mask Loss. To explore the contribution of the proposed mask loss, we construct two variants of REVMark based on whether applying the mask loss or not. The quality evaluation of the generated watermarked video is presented in Table 7. As can be seen, the video quality, particularly in terms of the perceptual quality, demonstrates promising augmentation with assistance of the mask loss. This can be attributed to the careful consideration of human visual perception that is undertaken in the proposed mask loss.

Table 8: Comparison between REVMark and DVMark in terms of model parameters, floating point operations (FLOPs), and training steps.

Framework	Param. (M)	FLOPs (G)	Training steps (K)
DVMark	22.45	2589	3000
REVMark	8.38	242	60

4.6 Comparison in Computational Cost

We conduct extension experiments to evaluate the computational cost, and report the comparison results between REVMark and DVMark in terms of model parameters, floating point operations (FLOPs), and training steps. The evaluations are conducted on the encoder, the distortion layer and the decoder of both frameworks. As compared in Table 8, our proposed REVMark decreases the model parameters and achieves significant computation reduction over the SOTA approach, and still maintains better performance.

5 CONCLUSION

In this paper, a novel end-to-end video watermarking framework named REVMark is proposed, which consists of an encoder, a differentiable distortion layer and a decoder. Specifically, the multi-scale encoder and decoder are constructed by 3D convolutional layers and cooperated with the proposed TAsBlock for effective temporal feature extraction. To enhance the robustness of REVMark against video compression, the DiffH264 is developed and implemented in the distortion layer to simulate the intra-frame and inter-frame compression of H.264/AVC on a differentiable basis. Moreover, the mask loss is formulated to boost the perceptual quality of the watermarked video, in which the spatial and temporal masks are provided to guide the encoder to embed watermark message with imperceptible modifications. The experimental results indicate significant improvements in robustness of REVMark compared to previous deep video watermarking approach, i.e., DVMark, in the meantime satisfactory video quality is maintained. Further experiments illustrate the high efficiency of REVMark. This framework has potential applications in various fields such as copyright protection, tracking, active forensics, as well as proactive defense.

ACKNOWLEDGMENTS

This work is supported in part by National Natural Science Foundation of China under Grants U22A2030, U1936212 and 62202507, in part by the Key-Area Research and Development Program of Guangdong Province under Grant 2020B0101360001, in part by Natural Science Foundation of Guangdong Province under Grant 2022A1515011209, and in part by the Peng Cheng Laboratory Project under Grant PCL2021A02.

REFERENCES

[1] Mahdi Ahmadi, Alireza Norouzi, Nader Karimi, Shadrokh Samavi, and Ali Emami. 2020. ReDMark: Framework for residual diffusion watermarking based on deep networks. *Expert Systems with Applications* 146 (2020), 113157.

[2] Md Asikuzzaman, Md Jahangir Alam, Andrew J Lambert, and Mark Richard Pickering. 2014. Imperceptible and robust blind video watermarking using chrominance embedding: A set of approaches in the DT CWT domain. *IEEE Transactions on Information Forensics and Security* 9, 9 (2014), 1502–1517.

[3] Md Asikuzzaman, Md Jahangir Alam, Andrew J Lambert, and Mark R Pickering. 2016. Robust DT CWT-based DIBR 3D video watermarking using chrominance embedding. *IEEE Transactions on Multimedia* 18, 9 (2016), 1733–1748.

[4] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. 2004. High accuracy optical flow estimation based on a theory for warping. In *Proceedings of European Conference on Computer Vision (ECCV)*. Springer, 25–36.

[5] Patrizio Campisi and Alessandro Neri. 2005. Video watermarking in the 3D-DWT domain using perceptual masking. In *Proceedings of IEEE International Conference on Image Processing (ICIP)*, Vol. 1. IEEE, I–997.

[6] Pik-Wah Chan and Michael R Lyu. 2003. A DWT-based digital video watermarking scheme with error correcting code. In *Proceedings of International Conference on Information and Communications Security (ICICS)*. Springer, 202–213.

[7] Qinwei Chang, Leichao Huang, Shaoteng Liu, Hualuo Liu, Tianshu Yang, and Yexin Wang. 2022. Blind robust video watermarking based on adaptive region selection and channel reference. In *Proceedings of the 30th ACM International Conference on Multimedia (ACM MM)*. 2344–2350.

[8] Lino E Coria, Mark R Pickering, Panos Nasiopoulos, and Rabab Kreidieh Ward. 2008. A video watermarking scheme based on the dual-tree complex wavelet transform. *IEEE Transactions on Information Forensics and Security* 3, 3 (2008), 466–474.

[9] Ingemar Cox, Matthew Miller, Jeffrey Bloom, and Chris Honsinger. 2002. Digital watermarking. *Journal of Electronic Imaging* 11, 3 (2002), 414–414.

[10] Frederic Deguillaume, Gabriela Csurka, Joseph JK O'Ruanaidh, and Thierry Pun. 1999. Robust 3D DFT video watermarking. In *Proceedings of Security and Watermarking of Multimedia Contents*, Vol. 3657. SPIE, 113–124.

[11] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. 2015. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 2758–2766.

[12] Oleg Evsutin and Kristina Dzhanashia. 2022. Watermarking schemes for digital images: Robustness overview. *Signal Processing: Image Communication* 100 (2022), 116523.

[13] Han Fang, Zhaoyang Jia, Zehua Ma, Ee-Chien Chang, and Weiming Zhang. 2022. PIMoG: An effective screen-shooting noise-layer simulation for deep-learning-based watermarking network. In *Proceedings of the 30th ACM International Conference on Multimedia (ACM MM)*. 2267–2275.

[14] David Fleet and Yair Weiss. 2006. Optical flow estimation. *Handbook of Mathematical Models in Computer Vision* (2006), 237–257.

[15] Palak Garg, Lakshita Dodeja, and Mayank Dave. 2019. Hybrid color image watermarking algorithm based on DSWT-DCT-SVD and Arnold transform. In *Proceedings of International Conference on Signal Processing and Communication (ICSC)*. Springer, 327–336.

[16] Frank C Glazer. 1987. *Hierarchical motion detection.* University of Massachusetts.

[17] Mohamed Hamidi, Mohamed El Haziti, Hocine Cherifi, and Driss Aboutajdine. 2015. A blind robust image watermarking approach exploiting the DFT magnitude. In *Proceedings of 2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA)*. IEEE, 1–6.

[18] Hwai-Tsu Hu and Ling-Yuan Hsu. 2017. Collective blind image watermarking in DWT-DCT domain with adaptive embedding strength governed by quality metrics. *Multimedia Tools and Applications* 76, 5 (2017), 6575–6594.

[19] Jun Jia, Zhongpai Gao, Kang Chen, Menghan Hu, Xiongkuo Min, Guangtao Zhai, and Xiaokang Yang. 2020. RIHOOP: Robust invisible hyperlinks in offline and online photographs. *IEEE Transactions on Cybernetics* 52, 7 (2020), 7094–7106.

[20] Jun Jia, Zhongpai Gao, Dandan Zhu, Xiongkuo Min, Menghan Hu, and Guangtao Zhai. 2022. RIVIE: Robust inherent video information embedding. *IEEE Transactions on Multimedia* (2022).

[21] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. 2017. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950* (2017).

[22] Hongmei Liu, Nuo Chen, Jiwu Huang, Xialing Huang, and Yun Q Shi. 2002. A robust DWT-based video watermarking algorithm. In *Proceedings of 2002 IEEE International Symposium on Circuits and Systems (ISCAS)*, Vol. 3. IEEE, III–III.

[23] Junxiong Lu, Jiangqun Ni, Wenkang Su, and Hao Xie. 2022. Wavelet-based CNN for robust and high-capacity image watermarking. In *Proceedings of 2022 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 1–6.

[24] Zhe-Ming Lu, Dian-Guo Xu, and Sheng-He Sun. 2005. Multipurpose image watermarking algorithm based on multistage vector quantization. *IEEE Transactions on Image Processing* 14, 6 (2005), 822–831.

[25] Xiyang Luo, Yinxiao Li, Huiwen Chang, Ce Liu, Peyman Milanfar, and Feng Yang. 2023. DVMark: a deep multiscale framework for video watermarking. *IEEE Transactions on Image Processing* (2023).

[26] S Maheswari, K Rameshwaran, and KM Malarselvi. 2015. DCT-PCA based watermarking on E-governance documents. *Research Journal of Applied Sciences, Engineering and Technology* 9, 7 (2015), 507–511.

[27] Hannes Mareen, Johan De Praeter, Glenn Van Wallendael, and Peter Lambert. 2018. A scalable architecture for uncompressed-domain watermarked videos. *IEEE Transactions on Information Forensics and Security* 14, 6 (2018), 1432–1444.

[28] Chi-Man Pun. 2006. A novel DFT-based digital watermarking system for images. In *Proceedings of IEEE International Conference on Signal Processing (ICSP)*, Vol. 2. IEEE.

[29] Anurag Ranjan and Michael J Black. 2017. Optical flow estimation using a spatial pyramid network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 4161–4170.

[30] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Proceedings of International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*. Springer, 234–241.

[31] Soumitra Roy and Arup Kumar Pal. 2017. A blind DCT based color watermarking algorithm for embedding multiple watermarks. *AEU-International Journal of Electronics and Communications* 72 (2017), 149–161.

[32] Richard Shin and Dawn Song. 2017. Jpeg-resistant adversarial images. In *Proceedings of NIPS 2017 Workshop on Machine Learning and Computer Security*, Vol. 1. 8.

[33] Matthew Tancik, Ben Mildenhall, and Ren Ng. 2020. Stegastamp: Invisible hyperlinks in physical photographs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2117–2126.

[34] Rohit Thanki, Ashish Kothari, and Deven Trivedi. 2019. Hybrid and blind watermarking scheme in DCuT–RDWT domain. *Journal of Information Security and Applications* 46 (2019), 231–249.

[35] Matthieu Urvoy, Dalila Goudia, and Florent Autrusseau. 2014. Perceptual DFT watermarking with improved detection and robustness to geometrical distortions. *IEEE Transactions on Information Forensics and Security* 9, 7 (2014), 1108–1119.

[36] Eric Wengrowski and Kristin Dana. 2019. Light field messaging with deep photographic steganography. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 1515—-1524.

[37] Thomas Wiegand, Gary J Sullivan, Gisle Bjontegaard, and Ajay Luthra. 2003. Overview of the H.264/AVC video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology* 13, 7 (2003), 560–576.

[38] Jonas Wulff and Michael J Black. 2015. Efficient sparse-to-dense optical flow estimation using a learned basis and layers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 120–130.

[39] Innfarn Yoo, Huiwen Chang, Xiyang Luo, Ondrej Stava, Ce Liu, Peyman Milanfar, and Feng Yang. 2022. Deep 3d-to-2d watermarking: Embedding messages in 3d meshes and extracting them from 2d renderings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 10031–10040.

[40] Aditi Zear, Amit Kumar Singh, and Pardeep Kumar. 2018. A proposed secure multiple watermarking technique based on DWT, DCT and SVD for application in medicine. *Multimedia Tools and Applications* 77 (2018), 4863–4882.

[41] Chaoning Zhang, Adil Karjauv, Philipp Benz, and In So Kweon. 2021. Towards robust deep hiding under non-differentiable distortions for practical blind watermarking. In *Proceedings of the 29th ACM International Conference on Multimedia (ACM MM)*. 5158–5166.

[42] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 586–595.

[43] Yifeng Zhang, Yingying Li, and Yibo Sun. 2019. Digital watermarking based on joint DWT–DCT and OMP reconstruction. *Circuits, Systems, and Signal Processing* 38 (2019), 5135–5148.

[44] Zhen Zhou, Shuyu Chen, and Guiping Wang. 2017. A robust digital image watermarking algorithm based on dct domain for copyright protection. In *Proceedings of International Symposium on Smart Graphics (SG)*. Springer, 132–142.

[45] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. 2018. Hidden: Hiding data with deep networks. In *Proceedings of European Conference on Computer Vision (ECCV)*. 657–672.

A ADDITIONAL EXPERIMENTAL RESULTS

Considering the practicality in real-world scenarios, we will evaluate the effectiveness of the proposed REVMark on videos of varying resolutions and lengths. In our experiments, instead of re-training the framework, we directly apply it to pre-processed videos.

Table 9: Decoding accuracy (%) of REVMark under Gaussian noise and Gaussian blur in terms of various video resolutions.

Resolution	Gaussian Noise ($\sigma = 0.04$)	Gaussian Blur ($\sigma = 2.0$)
128×128	99.97	99.82
256×256	99.89	99.82
512×512	99.90	99.84
1280×720	99.95	99.95

Table 10: Detector accuracy (%) of watermarked frames generated by REVMark under four different attack methods.

Frame Drop ($p = 0.5$)	Gaussian Noise ($\sigma = 0.04$)	Gaussian Blur ($\sigma = 2.0$)	H.264/AVC ($CRF = 22$)
99.9	99.5	99.7	97.2

A.1 Apply to Larger Videos

In order to apply REVMark to videos of various resolutions, the videos are resized in pre-processing to fit the input of REVMark. In watermark embedding, the generated watermark residuals are resized to the shape of the original videos and the quality of the watermarked video is fixed by adjusting the watermark strength α. Experimental results conducted on larger videos are presented in Table 9, which validates the effectiveness of REVMark across different video resolutions.

A.2 Apply to Longer Videos

For longer videos, we adopt a segmentation operation to divide them into sequences and subsequently applying the framework to each sequence. Considering that certain attacks may induce temporal asynchrony within videos, we develop a watermark detector as an auxiliary module of the decoder, which is specifically designed to identify the watermarked frame and consequently locate the entire watermarked video segment. The detector is a 2D CNN constructed by 8 convolutional layers, a global average pooling layer and a fully connected layer. In the training process of the detector, the parameters of the pre-trained encoder are fixed to generate watermarked videos as training data, and the cross-entropy loss function is adopted. Since it is trained independently, the real-world attacks can be applied in training without considering differentiability to boost the robustness.

To demonstrate that REVMark is applicable for long videos with the assistance of the detector, we conduct experiments to assess the detection accuracy of the detector across different attacks. Experimental results in Table 10 indicate that the trained detector achieves high levels of accuracy in identifying watermarked frames, thereby allowing us to locate the watermarked video segment. By incorporating the detector as an auxiliary module, the proposed REVMark can effectively be extended to process long videos.