Data Mining and Knowledge Discovery (2025) 39:62
https://doi.org/10.1007/s10618-025-01129-z

Spatiotemporal attention-based real-time video watermarking

Quan Yan[1] · Yuanjing Luo[2] · Zhangdong Wang[1] · Junhua Xi[1] · Geming Xia[1] · Zhiping Cai[1]

Received: 6 February 2025 / Accepted: 2 July 2025 / Published online: 1 August 2025

Abstract

As streaming media becomes prevalent, the demand for real-time video copyright protection has increased. Digital watermarking, a common copyright protection technique, has been widely used in copyright validation in various media. However, most of the existing video watermarking schemes follow the paradigm of image watermarking, focusing mainly on the impact of watermark embedding on visual perception and its robustness in channel transmission while neglecting the importance of efficiency. To efficiently protect the digital rights of streaming media, this article proposes an Efficient deep video Watermarking model based on Spatiotemporal Attention mechanism and patch sampling (EWSA). A spatiotemporal attention mechanism is employed to enhance watermark imperceptibility by embedding the watermark into texture and insensitive regions. Additionally, embedding efficiency is improved by sampling patches of video frames rather than embedding watermarking in entire frames. The performance of our model on three datasets through goal-oriented, three-stage training validates the effectiveness of the proposed EWSA, which achieves embedding speed approximately $2 \sim 3$ times faster than other deep watermarking methods.

Keywords Blind video watermarking · Deep learning · Patch sampling · Spatiotemporal attention

1 Introduction

With the rapid development of streaming media, live video streaming via Tiktok, YouTube, and other distribution platforms is becoming increasingly popular and widespread. While this popularity brings convenience to users, it also makes these

Extended author information available on the last page of the article videos easy targets for theft and misuse, such as misleading advertising[1] and plagiarism of works, posing new challenges to copyright protection for online platform and video publishers. To address this challenge, invisible watermarking technology has emerged, which not only protects copyrights without compromising the user experience (Patel et al. 2021; Savakar and Ghuli 2019; Joshi et al. 2018), but also shows great potential for application in content tracking, authenticity verification, and ownership identification with the rise of Artificial Intelligence Generated Content (AIGC) and Non-fungible Tokens (NFTs) (Zhang et al. 2023; Luo et al. 2023a, c; Yoo et al. 2022).

In recent years, various watermarks such as Fang et al. (2022), Lu et al. (2022), Wu et al. (2021), Fang et al. (2020) have been proposed. Video watermarking follows the paradigm of image watermarking that has achieved remarkable success and can be delineated into traditional watermarking and deep watermarking. Traditional watermarking methods are usually based on the principles of signal processing, information theory, and cryptography, embedding and extracting watermark messages in the original spatial or transform domain (Asikuzzaman et al. 2016; Dey et al. 2012). But these methods rely heavily on hand-designed algorithms or rules, leading to a lack of robustness against various types of distortions simultaneously (Asikuzzaman and Pickering 2017). To resolve this predicament, extensive research has been conducted in recent years, leading to significant advancements in the field of robust blind watermarking. In recent years, deep learning-based watermarking methods have been increasingly developed (Luo et al. 2023b; Ye et al. 2023; Zhang et al. 2023), which employ neural networks to process watermarking information and learn an adaptive and wiser way to embed watermarking information. This enables them to overcome the constraints of traditional watermarking methods in the face of complex attacks and improve the robustness of watermarking.

Despite advances in imperceptibility and robustness, deep watermarking techniques still fall short in real-time applications. In particular, scenarios like live video demand extremely high data processing capabilities and rapid response times (Sharma and Mir 2022). Consequently, achieving real-time performance while maintaining watermark embedding efficiency, imperceptibility, and robustness has become an urgent challenge.

To cope with this problem, we propose an Efficient deep video Watermarking model based on Spatiotemporal Attention mechanism and patch sampling (EWSA). First, we introduce a spatiotemporal attention mechanism to embed the watermark into texture and insensitive regions, enhancing the imperceptibility and robustness of the watermark. The spatiotemporal attention mechanism better captures the spatial and temporal features of the video, thereby improving the watermark embedding effectiveness. Second, we employ a patch sampling method to embed the watermark only in a small part of the frame, rather than the entire frame, thus improving the watermark embedding speed. This design not only enhances watermark embedding efficiency simply and effectively but also alters fewer video frames compared to the full-frame watermark embedding scheme, resulting in better visual quality of the

[1] https://www.asiaiplaw.com/section/in-depth/how-live-streaming-can-expose-you-to-charges-of-copyright-infringement.

watermarked video. Finally, we design a watermark localization module to locate the watermarked patches for videos with embedded watermark messages.

Figure 1 illustrates our proposed method compared to other watermarking methods in terms of peak signal-to-noise ratio (PSNR), frames per second (FPS), and bit accuracy ratio (BAR) metrics. In summary, the main contributions are listed as follows:

- We present an efficient deep video watermarking model (EWSA) that addresses the limitations of current schemes in meeting the requirements of real-time video watermarking scenarios.
- For the first time, a spatiotemporal attention mechanism is employed to embed the watermark into the texture and insensitive regions of the video. The patch sampling method is utilized to enhance embedding efficiency, simultaneously the watermark localization module is designed to ensure robustness.
- Experimental results on different datasets demonstrate the effectiveness of EWSA. We optimize the embedding efficiency while maintaining the imperceptibility and robustness of the watermarking. The results demonstrate EWSA achieves embedding speed approximately $2 \sim 3$ times faster than other deep watermarking methods.

The rest of this paper is organized as follows. In Sect. 2, we review the related work on video watermarking, patch sampling, and attention mechanism. In Sect. 3, we introduce the detailed network architecture. Section 4 presents training strategy. The experimental settings and results are presented in Sect. 5. Finally, Sect. 7 concludes this paper.

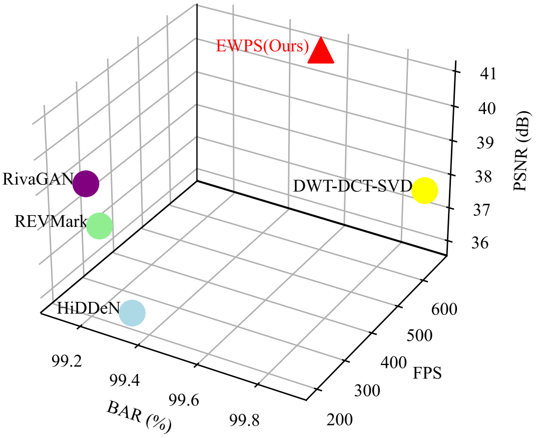


Fig. 1 PSNR-FPS-BAR (imperceptibility-efficiency-robustness) comparison with different watermarking methods, including the deep learning-based watermarking method HiDDeN (Zhu et al. 2018), RivaGAN (Zhang et al. 2019) and REVMark (Zhang et al. 2023), the traditional video watermarking method DWT-DCT-SVD (Dey et al. 2012) and our EWSA

2 Related work

2.1 Traditional and deep learning-based video watermarking

Video watermarking technology can generally be categorized into traditional and deep learning-based approaches (Kumar et al. 2020). Table 1 provides a comparative summary of recent works in the field, highlighting key differences between traditional methods, deep learning-based methods, and our proposed EWSA approach.

Traditional watermarking methods are usually based on the principles of signal processing, information theory, and cryptography. These methods are typically classified into three main categories: spatial domain, transform domain, and compressed domain (Asikuzzaman and Pickering 2017; Chen et al. 2023; Chang et al. 2022; Huan et al. 2022). *Traditional techniques usually use artificial heuristics and manually designed strategies for watermark embedding, which are subject to limitations such as lack of flexibility and adaptability.*

Recently, deep learning-based approaches have garnered increasing attention from researchers in the field of watermarking. HiDDeN (Zhu et al. 2018) pioneered deep learning for end-to-end image watermarking, featuring an encoder, decoder, and distortion network for differentiable simulation.

Compared to image watermarking, deep video watermarking faces greater challenges due to the complexity of video data and susceptibility to various attacks. DVMark (Luo et al. 2023b) introduces a multiscale design for robustness against video distortions, integrating a watermark detector and adapting to compression for reliability and practicality. RIVIE (Jia et al. 2022) leverages differentiable 3D rendering for video information hiding, mimicking camera capture and combining motion

Table 1 Comparative analysis of video watermarking methodologies

Aspect	Traditional methods (e.g., Chen et al. 2023; Huan et al. 2022; Chang et al. 2022; Liu et al. 2023)	Deep learning-based methods (e.g., Luo et al. 2023b; Jia et al. 2022; Zhang et al. 2023, 2024)	Our EWSA
Architecture	Signal processing, transform domain, cryptography-based	Encoder-decoder networks, attention mechanisms, multi-scale designs	Spatiotemporal attention, patch sampling, lightweight localization network
Attack types	Specific attacks (e.g., compression, cropping)	Various attacks (e.g., compression, temporal distortions, spatial distortions)	Comprehensive: compression, temporal (frame drop/swap), spatial (crop, resize)
Computational cost	Low: No parameters, fast processing	High: 0.21M–2.73M params, 46.52G–107.67G FLOPs	Moderate: 0.40M params, 26.92G FLOPs
Datasets	Custom or small-scale datasets	Kinetics-600, custom video datasets	Kinetics-600, Hollywood2, MGTV_WM
Pros	Less computational power, faster processing time	Adapts to various attacks with strong generalizability	Faster processing speed and more robust to different attacks
Cons	Targets specific attacks, limited generalizability	High computational complexity and slower processing speed	–

synthesis with temporal correlation to maintain visual quality over time. REVMark (Zhang et al. 2023) aims to improve robustness, particularly against H.264/AVC compression, with a focus on temporal feature extraction for efficient feature analysis while preserving visual quality.

Despite the advancements made by these methods, few deep learning-based watermarking approaches address embedding efficiency, and most are unable to simultaneously achieve high performance in terms of visual quality, robustness, and embedding efficiency.

2.2 Patch sampling related watermarking

Several image watermarking methods embed information into sub-image rather than whole image for different destination. DIPW (Luo et al. 2023a) introduces a patch-based deep watermarking framework designed to combat artwork plagiarism by embedding watermarks in non-overlapping image patches. Jia et al. (2022) propose a novel approach to invisible information hiding through sub-image encoding, which is robust in offline-to-online photography scenarios. DWSF (Guo et al. 2023) is a robust deep dispersed watermarking framework that employs a block-based embedding strategy, synchronization module, and message fusion to enhance watermark resilience against various image manipulations.

However, these works focus on image watermarking, and to date, there has been limited research on patch-based video watermarking. Inspired by recent literature on video quality evaluation using patch sampling (Wu et al. 2022), our approach adopts a patch sampling strategy to enhance the efficiency of video watermark embedding, thereby improving both robustness and processing speed in dynamic video environments.

2.3 Attention mechanism

The attention mechanism, which imitates the human eyes to receive features in conspicuous areas, is widely used in computer vision fields such as classification, detection, and segmentation (Liu et al. 2022; Zamir et al. 2022). D-LKA (Azad et al. 2024) introduces an attention mechanism that leverages large convolution kernels and deformable convolutions to enhance medical image segmentation, providing improved performance and computational efficiency in processing volumetric context in both 2D and 3D images. Similarly, CVANet (Cvanet 2024) utilizes pixel attention mechanisms to enhance single image super-resolution, simulating human visual perception to focus on detail reconstruction and achieve superior performance in image quality and resolution.

While most attention-based techniques in computer vision have focused on spatial features of images, our approach combines both spatial and temporal features in video. By designing a spatiotemporal attention mechanism, we embed watermarks in a way that takes into account both the spatial content of individual frames and the temporal consistency across video frames, offering improved robustness and imperceptibility in dynamic video sequences.

3 The proposed method

To meet the demands of real-time video watermark embedding and enhance watermark embedding efficiency without compromising imperceptibility and robustness, we propose an efficient deep video watermarking framework with patch sampling (EWSA), as illustrated in Fig. 2. Similar to previous works (Luo et al. 2023b; Zhang et al. 2023; Jia et al. 2022), EWSA consists primarily of the encoder EN_θ, the decoder DE_φ, and the distortion network *DN*. However, unlike these previous works, the carrier of the watermark messages in EWSA is not the entire video frames but rather the frame patches. Additionally, EWSA includes a novel module: the watermark location network (LO_γ). In this context, θ, φ, and γ represent the trained parameters of the encoder, decoder, and watermark location network, respectively. These parameters are iteratively updated during training to achieve optimal model performance. We explain these modules in more detail below, sequentially elaborating on their designs.

3.1 Patch sampling

The primary objective of this study is to design a model that meets the requirements of real-time video watermark embedding. Reducing the input data size for the deep model is an effective approach to minimizing inference time. Therefore we design patch sampling embedding instead of whole frame embedding. Specifically, we randomly select $H \times W$-sized patches from each frame of the video V_{co} forming cover patches P_{co} with size $L \times H \times W$ (the default input size for Encoder).

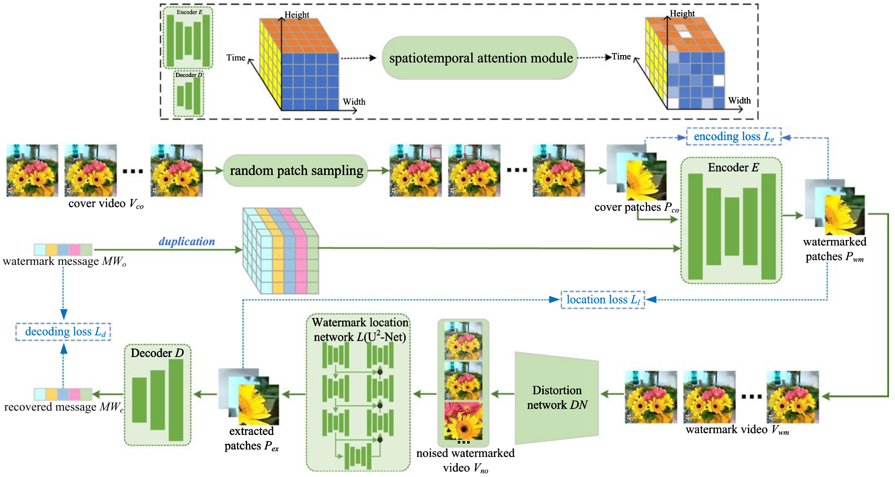


Fig. 2 The framework of our proposed EWSA. The black dashed box is a schematic representation of the spatiotemporal attention mechanism. The blue dashed line indicates the loss. The green solid line indicates the direction of data flow in the overall frame

3.2 Spatiotemporal attention

To achieve invisible video watermarking, we introduce a spatiotemporal attention mechanism that handles both spatial and temporal dimensions. Unlike traditional attention mechanisms that operate solely on spatial dimensions, our approach integrates temporal dynamics, enhancing the invisibility of video watermarking.

We employ 3D convolutions to capture spatio-temporal features. The spatiotemporal attention module consists of two 3D convolutional layers that process the video frames collectively, taking into account the temporal correlations between frames. The output of the 3D convolutional layers is an attention mask of size (*L*, *H*, *W*, *X*), where *L* represents the temporal dimension (number of frames), *W* and *H* are the spatial dimensions (width and height), and *X* is the data dimension corresponding to the watermark bits. For each pixel in the video frames, the spatiotemporal attention module generates a probability distribution over the data dimensions. The spatiotemporal attention mechanism is shown schematically in the black dashed box diagram in Fig. 2, the color shades of the output indicate varying probabilities. This distribution is used to determine the significance of each bit of the watermark at different locations within the video frames. The spatiotemporal attention mechanism can be formally expressed as:

$$F_a = \sigma(Conv3D(Conv3D(P_{co}))), \tag{1}$$

where σ indicates the softmax function, *Conv*3*D* represents the convolutional operation. The spatiotemporal attention weight map F_a will be used to fuse watermarking messages in the encoder.

The spatiotemporal attention biases the model towards embedding different bits of the watermark in various textures across the video frames.

3.3 Encoder

The architecture of encoder network EN_θ is shown in Fig. 3. Encoder EN_θ takes cover patches P_{co} and watermark messages WM_o as input and produces the watermarked patches P_{wm} as output:

$$P_{wm} = EN(P_{co}, WM_o), \tag{2}$$

where $EN(\cdot)$ represents the encoding process. The dimension of P_{co} and P_{wm} are $L \times H \times W \times C$, where L is the length of video (measured in frames/patches), H denotes height, *W* denotes width, and *C* represents channel number.

The watermark messages with the length of *X* take the form of a string of binary bits $WM\{0,1\}^m(m = X)$. Before merging the watermark messages with the video features, we begin by repeating the watermark messages across the spatial-temporal dimensions, the repeated watermark block has shape $L \times H \times W \times X$. Then, we combine it with the spatiotemporal attention weight map F_a along the channel dimension to get the fusion feature F_f.

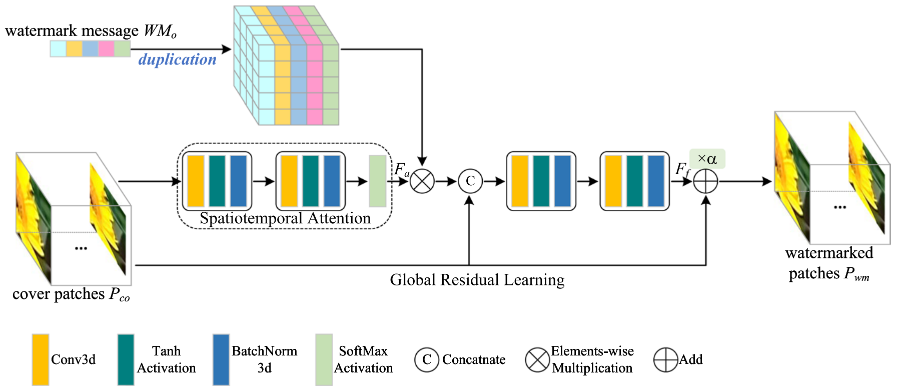


Fig. 3 Structure of encoder

E_θ uses the global residual skip connection to produce P_{wm} according to the equation:

$$P_{wm} = P_{co} + \alpha F_f, \tag{3}$$

where α is a modifiable embedding strength factor.

Note that random spatial crop is applied at the same coordinates in all L consecutive frames, so the encoder still receives a temporally ordered tube and the watermark can be embedded through the spatiotemporal attention module.

3.4 Distortion network

The distortion network is used to train the watermark to withstand diverse distortions, ensuring that the watermark messages can be recovered with precision even when exposed to a range of distortions. To prevent embedding watermarking messages in sensitive areas, we learn the distribution of differentiable distortions, as these distortions are more conducive to backpropagation. In DE_φ, the distorted patches are generated by using different types of distortion:

$$P_{no} = DN(P_{wm}, D_t). \tag{4}$$

During training, each distortion D_t is chosen at random with an equal chance at each step of training. The encoder EN_θ and decoder DE_φ update their parameters and both learn to be robust to a variety of different distortions simultaneously in response to randomly injected distortions. Figure 4 demonstrates samples of the eight distortions discussed in this paper.

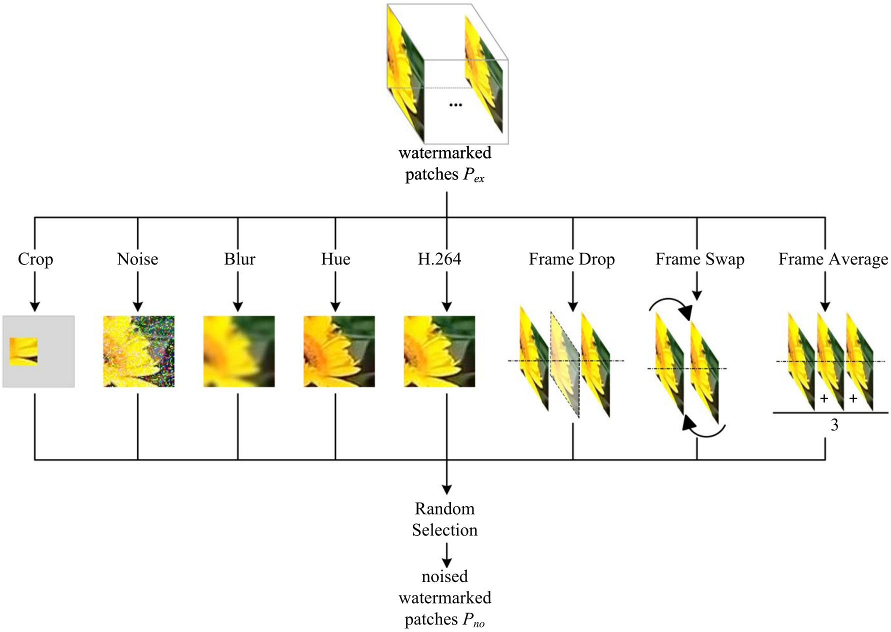


Fig. 4 Structure of distortion network

3.5 Watermark location network

In order to identify the encoded patch within the full distorted frame, the framework incorporates a localization network LO_{γ} positioned between the distortion network *DN* and the decoder DE_{φ}.

The input to the watermark localization network LO_{γ} is the frames of distorted watermarked video V_{no}. And the watermarked video is obtained by splicing the watermarked patches generated by the encoder back into the original cover video. The localization network LO_{γ}, a lightweight variant of salient object detection model U^2-Net† (Qin et al. 2020), is designed to segment encoded patches by learning spatial features, including texture and edge patterns and color and luminance variations, induced by watermark embedding.

The distorted encoded patches are extracted from frames of distorted video by location network:

$$P_{ex} = LO(V_{no}), \tag{5}$$

where $LO(\cdot)$ represents the process of watermark location.

3.6 Decoder

Given the watermarked patches extracted from the watermarked videos by the watermark location network P_{ex}, the decoder DE_{φ} extracts the embedded watermark messages:

$$WM_e = DE(P_{ex}), \tag{6}$$

where $DE(\cdot)$ represents the decoding process. As shown in Fig. 5, the decoder utilizes the same saptiotemporal attention module to extract the watermark from the watermarked video. It aggregates the spatial and temporal information, ensuring accurate recovery of the watermark. we use convolution, BatchNorm, and Tanh activation to extract video features. To ensure the output WM_e is the same length as the original watermark messages WM_o, we apply an elements-wise multiplication.

4 Training strategy

4.1 Loss function

During the training process, each module has a different optimization objective. The encoder is dedicated to make the watermarked video achieve good visual quality, the watermark localization module is to find the watermarked patches from a watermarked video, and the decoder aims to recover the watermark messages.

Visual loss Encoder training aims to generate invisible watermarked video patches P_{wm}, and the encoding loss function L_e minimises the distance between P_{co} and P_{wm} by updating the parameter θ. We regulate the pixel-wise modification between P_{co} and P_{wm} by the Mean Squared Error (MSE) loss:

$$L_e(P_{co}, P_{wm}) = \frac{1}{L \cdot H \cdot W} \sum_{i=0}^{L-1} \sum_{j=0}^{H-1} \sum_{k=0}^{W-1} [P_{co_i}(j,k) - P_{wm_i}(j,k)]^2, \tag{7}$$

where $P_{co_i(j,k)}$ and $P_{wm_i}(j,k)$ denote the pixel value at position (*j*, *k*) of the i- th patch.

Location loss The purpose of localization network $L\gamma$ is to accurately determine the exact position of the watermarked block.

We use Binary Cross-Entropy (BCE) loss and Intersection over Union (IoU) loss to minimize the distance between the predicted patch position and the true patch position. Thus, the overall loss of the segmentation model is formulated as:

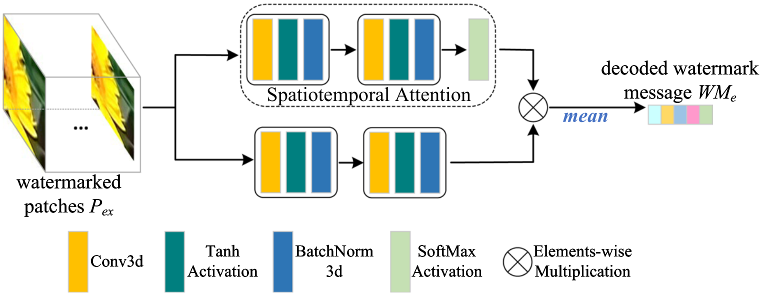

Spatiotemporal Attention
decoded watermark message WM_e
mean
...
watermarked patches P_{ex}
Conv3d
Tanh Activation
BatchNorm 3d
SoftMax Activation
Elements-wise Multiplication

Fig. 5 Structure of decoder

$$L_l(P_p, P_g) = L_{BCE}(P_p, P_g) + \omega L_{IoU}(P_p, P_g), \tag{8}$$

where $P_p \in [0,1]^{H\times W}$ is the predicted mask for the watermarked frame, $P_g \in \{0,1\}^{H\times W}$ is the ground truth watermarked patch of embedding mask, and ω is a weight to balance BCE loss and IoU loss.

$$L_{BCE}(P_p, P_g) = -\sum_{(j,k)}[P_g(j,k)logP_p(j,k) + (1-P_g(j,k))log(1-P_p(j,k))]. \tag{9}$$

$$\begin{aligned} L_{IoU}(P_p, P_g) =& 1 - \frac{P_g \cap P_p}{P_g \cup P_p} \\ =& 1 - \frac{\sum_{j=0}^{H-1}\sum_{k=0}^{W-1} P_g(j,k)P_p(j,k)}{\sum_{j=0}^{H-1}\sum_{k=0}^{W-1}[P_g(j,k) + P_p(j,k) - P_g(j,k)P_p(j,k)]}, \end{aligned} \tag{10}$$

where $P_g(j,k) \in \{0,1\}$ is the ground truth label of the pixel at position (*j*, *k*), and $P_p(j,k) \in [0,1]$ denotes the predicted probability that position (*j*, *k*) is watermarked.

Decoding loss Decoder D_φ aims to minimize the difference between the extracted watermark WM_e and the original watermark WM_o. We utilize BCE loss as decoding loss which is reduced by updating the parameter φ:

$$L_d(WM_o, WM_e) = -\frac{1}{X}[WM_o logWM_e + (1-WM_e)log(1-WM_o)]. \tag{11}$$

Algorithm 1 Three-stage training

```
Input: Training set of cover videos $V_{co}$ and watermark messages $WM_o$.
Output: Trained encoder $EN_\theta$, decoder $DE_\varphi$, and watermark location network $LO\gamma$.
Initialize encoder $EN_\theta$ and decoder $DE_\varphi$ with random value.
Initialize watermark location network $LO\gamma$.
if Stage-1 training then
    while $Step < max_{steps}$ do
        -Embed watermark messages: $P_{wm} = EN(P_{co}, WM_o)$.
        -Add distortion: $P_{no} = DN(P_{wm}, D_t)$.
        -Extract watermark messages: $WM_e = DE(P_{no})$.
        -Update the parameters of encoder and decoder: minimize $L_{ed}$.
    end while
end if
if Stage-2 training then
    while $Step < max_{steps}$ do
        -Extract watermarked patch: $P_{ex} = LO(V_{no})$.
        -Update the parameters of watermark location network: minimize $L_l$.
    end while
end if
if Stage-3 training then
    while $Step < max_{steps}$ do
        -Extract patch: $P_{ex} = LO(V_{no})$.
        -Extract watermark messages: $WM_e = DE(P_{ex})$.
        -Update the parameters of decoder: minimize $L_d(WM_o, WM_e)$.
    end while
end if
```

4.2 Three-stage training

Simultaneously training all losses results in challenges in achieving convergence for the watermark locating and decoding losses. Designing a training strategy that strikes a compromise between robustness, visual quality, and embedding efficiency is necessary.

We take advantage of stage training, which was inspired by Liu et al. (2019), Jia et al. (2022), Luo et al. (2023a), Fang et al. (2023). The training strategy is summarized in Algorithm 1.

Stage-1 In the first stage, the encoder and decoder are initially trained with distortion network. To sum up, the optimization of this stage is to minimize

$$L_{ed} = \lambda_1 L_e(P_{co}, P_{wm}) + \lambda_2 L_d(WM_o, WM_e), \tag{12}$$

where λ_1 and λ_2 are the weights to balance $L_e(P_{co}, P_{wm})$ and $L_d(WM_o, WM_e)$. As a result, a pretrained encoder and an decoder are provided, with the former being in charge of embedding watermarks. After encoding the watermark messages into the patches, splice the watermarked patches P_{wm} back into the original video to get the watermarked video V_{wm}.

Stage-2 In the second stage, to supervise the regression of the location of the watermarked patches P_{wm} in watermarked video V_{wm}, we use P_{wm} and V_{wm} to generate the training dataset for the location network. The ground truth is binary masks. The value of 0 corresponds to background pixels, indicating the unwatermarked area, while the value of 1 represents the foreground watermarked pixels. The optimization of this stage is to minimize location loss $L_l(P_p, P_g)$.

Stage-3 In the third stage, we put the watermarked patches output from the watermark localization network as distorted patches into the distortion network to optimize the decoding accuracy of the decoder. This is because the accuracy of the watermark localization network has an impact on the performance of the decoder. At this stage, the optimization objective is to minimize $L_d(WM_o, WM_e)$.

5 Experiments

5.1 Basic setup

5.1.1 Datasets

The experiments are performed on three distinct video datasets: Kinetics-600, Hollywood2, and MGTV_WM. Below, we provide a concise introduction to them.

- *Kinetics-600* (Carreira et al. 2018; Carreira and Zisserman 2017). Kinetics-600 contains over 500K video clips, each lasting 10 s. The dataset encompasses a total of 600 categories, with each category containing at least 600 films or more. The video resolution is variable but roughly around 570×320.
- *Hollywood2* (Marszalek et al. 2009). Hollywood2 is a human behavioral action

video dataset containing 3669 video clips from 69 movies with a total video length of about 20.1 h. The video resolution is variable but roughly around 570× 320.

- *MGTV_WM* (Chen et al. 2023). MGTV_WM is the most recent video watermarking evaluation dataset with a resolution of 720×1280 or 480×848. It contains 1000 video clips, and each clip lasts for about 30 s.

It is worth noting that Kinetics-600 is employed for both training and testing in our experiments, whereas Hollywood2 and MGTV_WM are exclusively utilized for testing. In particular, the watermarking model (comprising the encoder and decoder) and baselines (Zhang et al. 2019; Zhu et al. 2018; Dey et al. 2012) are trained on 1000 randomly cropped video clips with dimension 128×128×8×3 in the Kinetics-600 training set, and the models are evaluated on the Kinetics-600 validation set, Hollywood2 and MGTV_WM. For each input video clip, there is a corresponding secret watermark that is randomly sampled from the binary distribution $WM\{0,1\}^m (m = 96)$. To build the training set for the watermark location network LO_γ, we first resize every watermarked video frame to 256×256 to ensure uniform spatial resolution. In each resized frame we select one random patch and embed the watermark in that region, recording its coordinates. Using these coordinates we create a 256×256 binary mask whose pixels are set to 1 inside the watermarked patch and to 0 elsewhere. Each (frame, mask) pair is then used as one training sample for LO_γ.

5.1.2 Implementation

Our method is implemented with PyTorch 1.13.1, Intel(R) Core(TM) i9-9900K @ 3.60GHz, 64.00 GB RAM, and an NVIDIA GeForce GTX 2080 Ti GPU (with 11GB memory) is used for both training and testing. We set batch size $= 8$, video size 128×128×8×3, and ADAM optimizer with an initial learning rate of 1e-4 during training. In three-stage training, we set 400, 100, and 300 epochs for the three stages, respectively. And for other parameters, we have the following settings: the weight parameters of Eq. (12), $\lambda_1 = 1$, $\lambda_2 = 1$; the watermark strength $\alpha = 0.05$.

5.1.3 Metrics

We evaluate deep video watermarking based on efficiency, imperceptibility, and robustness.

- *Efficiency* is measured by the embedding speed, quantified in frames per second (FPS) (Chen et al. 2023), representing the number of frames processed per second.
- *Imperceptibility* assesses the visual quality of the watermarked video compared to the original. We use Peak Signal-to-Noise Ratio (PSNR) (Huynh-Thu and Ghanbari 2008) and Learned Perceptual Image Patch Similarity (LPIPS) (Zhang et al. 2018) as metrics.
- *Robustness* is evaluated by extraction accuracy, measured using the Bit Accuracy

Ratio (BAR) (Chen et al. 2023), which quantifies the accuracy of recovered watermark messages.

Higher FPS, PSNR, and BAR indicate better performance, while lower LPIPS is preferable. These metrics provide insight into the method's applicability across different scenarios.

5.1.4 Baselines

We evaluate our model by comparing it to the deep learning-based watermarking method HiDDeN (Zhu et al. 2018), RivaGAN (Zhang et al. 2019), REVMark (Zhang et al. 2023) and traditional video watermarking method DWT-DCT-SVD (Dey et al. 2012).

Performance is assessed using the identical video size (128×128×8×3) and payload (96 bits) to ensure a fair comparison.

5.2 Efficiency

Efficiency is assessed based on the frames per second (FPS), which measures the number of frames that are embedded every second. Model parameters and floating point operations are used to assist in evaluating the computational cost of different watermarking methods. The number of parameters indicates the model size, the number of floating point operations per second (FLOPs) provides the time complexity, and the number of frames processed per second provides the throughput.

Table 2 demonstrates the comparison of our EWSA with baseline methods under the condition that the video resolution is 128×128 and EWSA takes the patch size of 64×64. The results in the table show that our approach is the most efficient at embedding among deep watermarking methods and is close to the traditional method DWT-DCT-SVD.

To assess the effectiveness and practicality of the model, we conduct a test by randomly selecting patches of varying sizes from the three datasets, i.e., Kinetics-600, Hollywood2, MGTV_WM. The results are presented in Fig. 6. The videos in the MGTV_WM are separated into two distinct categories based on their resolutions: 720×1280 and 480×848. While the videos in Hollywood2 and Kinetics-600 have varying resolutions, both are around 570×320, smaller than the resolution of videos in MGTV_WM. The results show that the watermark embedding speed peaks around the patch size of 32×32, which is 1047.62, 1080.38, and 207.99 FPS on Hollywood2, Kinetics-600, and MGTV_WM respectively. This occurs due to the insufficient efficiency increase gained from using patch size smaller than 32×32, which does not

Table 2 Embedding efficiency comparison

Method	Param.(M) ↓	FLOPs(G) ↓	FPS ↑
DWT-DCT-SVD	N/A	N/A	**663.93**
HiDDeN	**0.21**	54.27	211.21
RivaGAN	0.41	107.67	196.47
REVMark	2.73	46.52	305.31
EWSA (Ours)	0.40	**26.92**	661.33

"N/A": not applicable

The best result is in bold

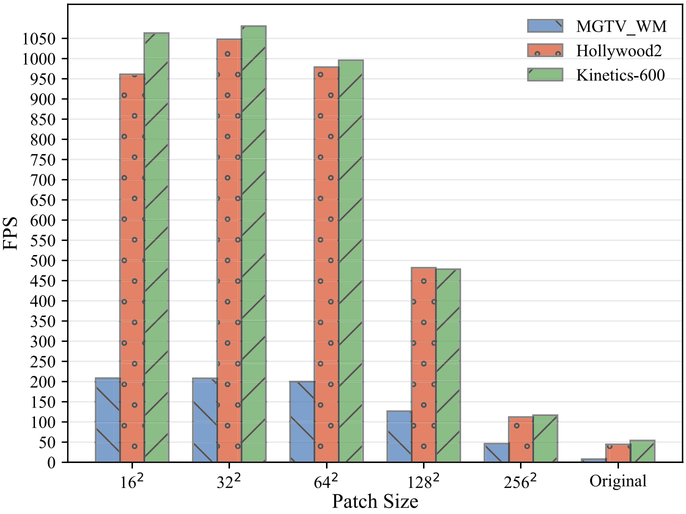


Fig. 6 Watermark embedding efficiency of EWSA for different sized patches on different datasets. The X-axis tick "Original" indicates the original video size

compensate for the time lost in patch sampling and reinserting watermarked patches into the original video frame. As the patch size increases beyond 32×32, the speed at which the watermark is embedded slows down.

We also observe that the efficiency of using the MGTV_WM dataset is much lower overall than that of the other two datasets in Fig. 6. The lower FPS on MGTV_WM than Hollywood2 and Kinetics-600 is primarily due to its higher resolutions. The EWSA embedding pipeline, which includes patch sampling, encoding, and patch integration, incurs greater overhead on larger frames. Sampling and integration involve accessing and modifying the full frame, leading to increased memory access times, data transfer costs, and potential cache misses, particularly under the hardware constraints of an NVIDIA GTX 2080 Ti with 11GB memory. Although patch sampling reduces computational load by 10-fold (e.g., 146.146 FPS vs. 10.970 FPS for MGTV_WM at 128×128, Table 5), the proportional benefit diminishes for MGTV_WM, as a 32×32 patch (1,024 pixels) represents only 0.11% of a 720×1280 frame compared to 0.56% of a 570×320 frame, amplifying the dominance of full-frame processing overhead.

We discuss the effectiveness of EWSA in a real-time scenario. Taking 25 FPS as the standard for real-time (Li et al. 2022), the method of patch sampling in sizes that are 256×256 or less can meet the requirement for real-time embedding speed on all test datasets. With a patch size of 256×256, the results of EWSA on the Hollywood2, Kinetics-600, MGTV_WM datasets, achieve embedding rates of 112.167, 116.643, 44.306, and 46.101 FPS, respectively.

5.3 Imperceptibility

The imperceptibility comparison of EWSA and baselines is presented in Table 3. EWSA exhibits superior performance in both PSNR and LPIPS metrics. In terms of

Table 3 Imperceptibility comparison

Dataset	Method	PSNR(dB) ↑	LPIPS × 100 ↓
Hollywood2	HiDDeN	35.56	8.92
	RivaGAN	**40.56**	6.89
	REVMark	36.22	8.05
	DWT-DCT-SVD	36.28	12.85
	EWSA (Ours)	39.33	**4.61**
Kinetics-600	HiDDeN	35.90	9.20
	RivaGAN	39.43	6.12
	REVMark	37.21	7.83
	DWT-DCT-SVD	37.62	13.94
	EWSA (Ours)	**40.92**	**5.45**
MGTV_WM	HiDDeN	35.22	8.54
	RivaGAN	38.52	4.79
	REVMark	37.18	8,79
	DWT-DCT-SVD	36.16	14.35
	EWSA (Ours)	**40.05**	**4.84**

The best result is in bold

PSNR, EWSA outperforms the traditional and deep learning-based methods in most datasets. For instance, on the MGTV_WM dataset, EWSA achieves a PSNR of 40.05 dB, which is 3.89 dB higher than DWT-DCT-SVD, 4.83 dB higher than HiDDeN, 1.53 dB higher than RivaGAN, and 2.87 dB higher than REVMark. This demonstrates EWSA's superior imperceptibility.

The LPIPS metric, which evaluates perceptual similarity, also shows that EWSA excels in imperceptibility. EWSA achieves the lowest LPIPS score across all datasets, indicating that the watermark is perceptually less noticeable. For example, EWSA on the Hollywood2 dataset achieves an LPIPS score of 4.61, which is significantly lower than the second-best RivaGAN (6.89), further confirming its superior perceptual quality.

The invisibility of EWSA is visualized in Fig. 7, which contains the original cover video frame patch, visualization of the attention mask, and the encoded video frame patch. As shown in Fig. 7b and d, the encoded frame patch is almost identical to the original video frame patch, indicating the imperceptibility of the watermark. The attention masks for different frame patches are shown in Fig. 7c, with varying colors indicating different levels of attention. Areas with a reddish hue are designated for high-strength watermarking, which corresponds to regions with intricate textures and low visibility, while other areas are allocated for low-strength watermarking. In addition, Fig. 7 shows the video frame difference before and after watermark embedding for the EWSA and baselines. From the difference figure Fig. 7e, g, i, k, m, it is clear that EWSA exhibits the best watermark imperceptibility, while DWT-DCT-SVD shows the largest differences and the poorest imperceptibility.

An additional experiment is carried out to examine the effect of various patch sizes on imperceptibility. The findings are displayed in Fig. 8. The X-axis ticks patch size "Original" indicates the original video size, i.e., the watermark is embedded in the whole frame, not in a patch. The figure demonstrates that increasing the patch size negatively impacts the imperceptibility. This is because larger patch sizes result in a higher ratio between the embedded watermark area and the entire video frame.

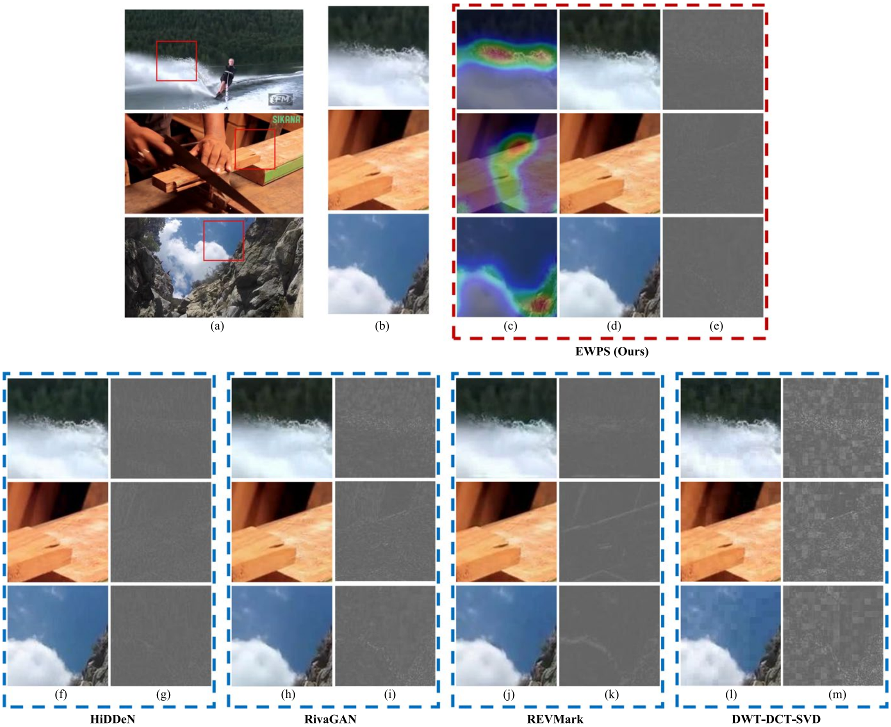
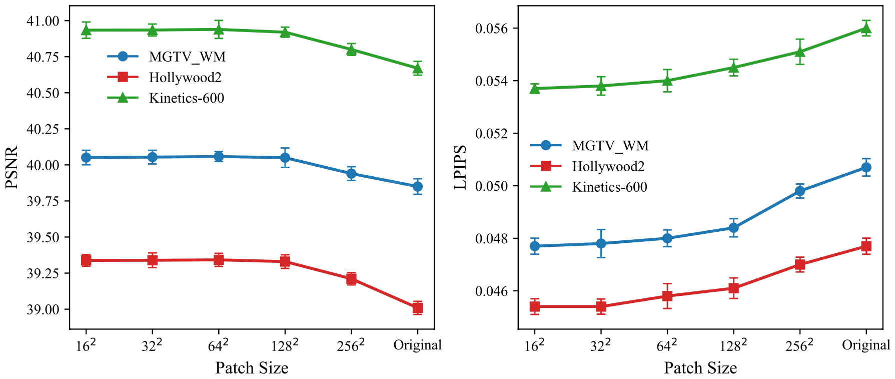


Fig. 7 Samples of watermarked videos generated by our method and the comparison methods HiDDeN, RivaGAN, REVMark, and DWT-DCT-SVD. **a** Patched cover video frame; **b** Cover patch; **c** Attention mask; **d** Watermarked patch of EWSA; **e** Difference between before and after watermarking for EWSA; **f** Watermarked patch of HiDDeN; **g** Difference between before and after watermarking for HiDDeN; **h**Watermarked patch of RivaGAN; **i** Difference between before and after watermarking for RivaGAN; (j) Watermarked patch of REVMark; **k**Difference between before and after watermarking for REVMark; **l** Watermarked patch of DWT-DCT-SVD; **m** Difference between before and after watermarking for DWT-DCT-SVD

5.4 Robustness

In order to achieve strong robustness, it is necessary for the extracted watermark messages to closely resemble the original watermark messages even in the presence of distortion. This implies that the watermark distortion rate should be low and the bit accuracy ratio (BAR) should be high. As explained fully in the Sect. 3.4, we assess the robustness of EWSA and baselines against eight distortions and Identity. Where Identity denotes that there is no distortion to the watermarked videos.

As shown in Table 4, our method outperforms both the HiDDeN, RivaGAN and DWT-DCT-SVD in most of the tested distortions. We also note that while the DWT-DCT-SVD method is fairly robust to some distortions such as frame drop and frame swap, it is not robust to other types of distortions such as frame average or random crop. Temporal distortions such as frame drop and frame swap, which do not modify the pixel values of the frames, have little effect on the accuracy of watermark extraction for HiDDeN and DWT-DCT-SVD. This is because these two methods are



(a) PSNR of EWPS

(b) LPIPS of EWPS

Fig. 8 The imperceptibility of watermarked videos with the different patch sizes. The X-axis tick “Original" indicates the original video size

two-dimensional operations on videos, meaning that each video frame is processed individually like an image.

5.5 Comparative analysis of trade-offs

Our systematic evaluation reveals fundamental trade-offs among three critical dimensions of video watermarking: computational efficiency, perceptual imperceptibility, and operational robustness, as visualized in Fig. 1 and described in Table 1. The key findings are summarized as follows:

- *Traditional vs. Deep Learning Methods* The traditional method DWT-DCT-SVD achieves superior frame rates (663.93 FPS, Table 2) due to non-parametric operations, yet demonstrates significant vulnerability to spatial distortions—notably achieving only 50.61% bit accuracy under random crop attacks (Table 4). In contrast, EWSA balances these aspects via lightweight architecture and patch-based processing. It sacrifices merely 0.4% in FPS while achieving a 13.1% improvement in average bit accuracy ratio (BAR).
- *Robustness Specialisation* REVMark demonstrates superior compression resistance (94.24% BAR under H.264, Table 4) through optical flow estimation network (Zhang et al. 2023). However, this specialization comes at the cost of model complexity (2.73M parameters vs. 0.40M for EWSA, Table 2) and perceptual quality degradation (36.22 dB PSNR vs. 39.33 dB for EWSA, Table 3). Our spatiotemporal attention mechanism achieves comprehensive robustness (97.04% average BAR, Table 4) while maintaining computational efficiency and superior imperceptibility.
- *Adversarial Training Trade-offs* While RivaGAN’s adversarial training framework enhances pixel fidelity (40.56 dB PSNR on Hollywood2, Table 3), this approach demonstrates compromised robustness against common distortions (95.63% average BAR vs. 97.04% for EWSA, Table 4). EWSA circumvents this

Table 4 Comparison of bit accuracy ratio i.e. BAR (%)↑

Dataset	Method	Attacks									AVG.
		Identity	H.264 (CRF=22)	Frame Average (N=3)	Frame Drop (*p*=0.5)	Frame Swap (*p*=0.5)	Gaussian Blur ($\sigma = 2.0$)	Gaussian Noise ($\sigma = 0.04$)	Random Crop (*p*=0.4)	Random Hue (d=1.0)	
Hollywood2	HiDDeN	98.97	75.61	95.02	98.42	98.69	75.91	90.13	91.71	97.18	91.29
	RivaGAN	99.71	87.40	93.96	99.67	97.35	96.63	95.83	93.52	96.61	95.63
	REVMark	99.43	**94.24**	97.85	99.73	**99.89**	**97.64**	**99.71**	81.16	78.89	94.28
	DWT-DCT-SVD	**99.89**	78.73	60.78	**99.85**	99.88	76.18	95.40	50.61	94.35	83.96
	EWSA (Ours)	99.70	90.08	**98.58**	98.09	98.94	97.28	97.64	**95.29**	**97.74**	**97.04**
Kinetics-600	HiDDeN	99.33	78.67	93.37	99.31	98.43	75.89	93.96	91.51	**97.83**	92.03
	RivaGAN	99.20	85.26	92.97	98.48	97.08	95.45	95.66	93.75	96.01	94.87
	REVMark	99.12	**95.06**	98.72	**99.80**	**99.83**	**97.96**	**99.30**	80.02	80.77	94.51
	DWT-DCT-SVD	**99.90**	84.69	59.93	99.71	99.79	76.68	98.00	49.66	93.07	84.60
	EWSA (Ours)	99.54	89.67	**99.12**	97.34	98.25	97.09	95.43	**96.67**	97.10	**96.69**
MGTV_WM	HiDDeN	99.41	80.89	95.39	99.29	98.93	72.09	91.51	91.25	97.34	91.79
	RivaGAN	99.73	87.19	93.52	98.83	98.27	93.75	95.45	94.76	95.83	95.26
	REVMark	99.03	**93.67**	97.79	99.64	99.78	**98.61**	**99.58**	82.47	84.93	95.06
	DWT-DCT-SVD	**99.98**	77.20	59.28	**99.93**	**99.81**	79.16	94.24	51.83	93.72	83.91
	EWSA (Ours)	99.87	89.34	**98.82**	99.21	98.38	96.40	97.33	**95.95**	**98.58**	**97.10**

The best result is in bold

limitation through perceptually-aware embedding in low-saliency regions (Fig. 7c), eliminating the stability issues inherent in adversarial optimization paradigms.

Practical recommendations

1. *Real-time streaming* EWSA's patch-based design ensures real-time watermark embedding while maintaining robustness against common distortions.
2. *Compression-heavy environments* REVMark is preferred when H.264 resistance is critical, despite higher computational costs (46.52G FLOPs vs. 26.92G for EWSA).
3. *Legacy systems* DWT-DCT-SVD is suitable for resource-constrained environments where only basic robustness, such as tolerance to frame drops, is required.

These trade-offs highlight EWSA's value as a balanced solution for dynamic video environments.

5.6 Ablation study

We carried out ablation experiments to assess the effectiveness of the EWSA components we designed, such as patch sampling, the spatiotemporal attention module, and the watermark location network.

5.6.1 Effect of patch sampling

As mentioned in Sect. 3.1, we embed the watermark messages after sampling random patches of video frames rather than embedding them into the whole video frame. To evaluate its effectiveness, we compare it to the full-size frame embed method. Since our network can receive videos of different sizes, we test the watermark embedding speed of full-size video frames by feeding them directly into the already-trained encoder. And the model with patch sampling for comparison testing uses the patch size of 128×128. The results are shown in Table 5. It is evident that the embedding efficiency of the model with patch sampling is considerably superior to that without patch sampling, with an improvement of 9.87, 7.86, and 12.32 times on Hollywood2, Kinetics-600, and MGTV_WM, respectively. on every dataset, with a nearly tenfold difference in embedding efficiency. This demonstrates that patch sampling contributes to the efficiency of watermark embedding.

5.6.2 Effect of spatiotemporal attention module

The encoder specifically develops the attention module as an essential component to guide the watermark embedding process. To assess the effect of the attention module, two watermarking models are devised for comparison. The initial model incorporates

Table 5 The effect of patch sampling

Dataset	Patch sampling	FPS ↑
Hollywood2	✗	44.339
	✓	**481.913**
Kinetics-600	✗	53.987
	✓	**478.228**
MGTV_WM	✗	10.970
	✓	**146.146**

The best result is in bold

Table 6 The effect of attention module

Dataset	Attention module	PSNR(dB) ↑	LPIPS×100 ↓
Hollywood2	✗	38.52	9.94
	✓	**39.33**	**4.61**
Kinetics-600	✗	38.06	10.84
	✓	**40.92**	**5.45**
MGTV_WM	✗	38.74	9.10
	✓	**40.05**	**4.84**

The best result is in bold

an attention module, while the encoder of the other model combines a sequence of Conv3d, ReLU, and BatchNorm3d operations.

Table 6 shows the comparison between the two models. We can see that compared to module without attention module, module with attention module can significantly enhance the visual quality of EWSA. This demonstrates that attention module contributes to imperceptibility.

5.6.3 Effect of watermark location network

This section follows Jia et al. (2022) and compares the results of using a watermark localization network with those of manually locating (without the location network). When using the watermark localization network, the network detects watermarked patches from videos subjected to random attacks and then recovers the watermark messages through a decoder. Without location network, we rely on manual identification of watermarked patches and employ a decoder to directly retrieve the watermark messages.

The comparison results are presented in Table 7. Upon initial examination, it is evident that the model with location network consistently outperforms that without location network. For example, when watermarked videos are distorted by Random Crop (p=0.4), the model with location network performs 8.62% better than the model without location network. Manual locating is less exact than the watermark localization network due to the high imperceptibility of watermarks, resulting in lower decoding accuracy. The results indicate that the localization network is highly efficient in acquiring more precise watermark patches for watermarked videos.

Table 7 The effect of watermark location network. Comparison of bit accuracy ratio i.e. BAR (%) ↑

Dataset	Watermark location network	Attacks									AVG.
		Identity	H.264 (CRF=22)	Frame Average (N=3)	Frame Drop (p=0.5)	Frame Swap (p=0.5)	Gaussian Blur ($\sigma = 2.0$)	Gaussian Noise ($\sigma = 0.04$)	Random Crop (p=0.4)	Random Hue (d=1.0)	
Hollywood2	✗	94.04	87.67	96.12	94.34	92.25	90.09	95.43	86.67	93.10	92.19
	✓	**99.70**	**90.08**	**98.58**	**98.09**	**98.94**	**97.28**	**97.64**	**95.29**	**97.74**	**97.04**
Kinetics-600	✗	97.53	86.33	90.40	90.93	89.91	95.85	93.75	92.65	92.10	92.16
	✓	**99.54**	**89.67**	**99.12**	**97.34**	**98.25**	**97.09**	**95.43**	**96.67**	**97.10**	**96.69**
MGTV_WM	✗	97.31	88.62	88.04	95.92	96.39	94.36	90.49	85.68	88.68	91.72
	✓	**99.87**	**89.34**	**98.82**	**99.21**	**98.38**	**96.40**	**97.33**	**95.95**	**98.58**	**97.10**

The best result is in bold

6 Discussion

6.1 Application

The proposed EWPS offers significant advantages in real-time video applications. Unlike traditional and existing deep learning-based watermarking methods, EWPS achieves faster embedding speed and better imperceptibility while maintaining robustness. Figures 6 and 8 show that our model, trained on a 128×128 resolution dataset, performs consistently across various resolutions, eliminating the need for retraining on different datasets. This adaptability is crucial for modern streaming platforms with variable video resolutions. Additionally, by adjusting patch sizes, EWPS meets real-time requirements, ensuring optimal embedding speed and video quality. These features make EWPS an invaluable tool for video copyright protection in dynamic streaming environments.

6.2 Limitation

While the EWPS model significantly improves embedding efficiency, it does present certain limitations. First, the robustness needs to be improved because the performance of robustness is average. Second, although embedding efficiency is enhanced, the need to locate watermark-embedded regions introduces a time cost during extracting watermarking. This localization step can impact the overall performance, potentially offsetting the benefits gained during the embedding phase. Addressing these limitations is crucial for further optimizing the model for practical, real-time applications.

6.3 Future work

There are several promising directions for future research. First, integrating the watermark-embedding process with video-compression standards such as H.264/AVC and H.265/HEVC would allow embedding to run in parallel with compression, greatly improving system efficiency. Second, it is important to enhance the robustness of the watermark against practical attack methods such as camera re-recording and screen capturing, so that protection remains effective in real-world scenarios. Third, an open challenge is to increase resilience to copy-paste (region-duplication) attacks, in which an adversary duplicates or relocates a watermarked region within the same video. Developing localization and extraction mechanisms that remain reliable under such tampering is a key direction for future work. These advancements will further solidify the utility and effectiveness of the proposed EWPS model.

7 Conclusion

This paper proposes a blind video watermarking framework called EWSA, which aims to improve the embedding speed by using patch sampling. Furthermore, the framework uses the spatiotemporal attention mechanism to guide the watermark embedding in inconspicuous texture regions, ensuring high visual quality. The framework also incorporates a watermark localization network and uses stage training to gradually enhance the robustness of the watermark. Experiments conducted on three datasets demonstrate that EWSA achieves superior performance in balancing efficiency, imperceptibility, and robustness. The speed of embedding can be controlled by adjusting the patch size, allowing for the generalizability of EWSA in videos with arbitrary resolutions. Therefore, EWSA has potential applications across various fields, including online videos, live streaming, and medical image system.

Author contributions Quan Yan wrote the main manuscript text. Yuanjing Luo, Zhangdong Wang, Junhua Xi, Zhiping Cai, and Geming Xia prepared figures and guiding the revision of manuscript.

Funding This work is supported by the National Natural Science Foundation of China(62172155, 2472434, 62402171), and the Science and Technology Innovation Program ofHunan Province (2022RC3061).

Data availability No datasets were generated or analysed during the current study.

Code availability The code for experiments is available at the following url: https://github.com/QuestaYan/EWSA

Declarations

Conflict of interest The authors declare no Conflict of interest.

References

Asikuzzaman M, Pickering MR (2017) An overview of digital video watermarking. IEEE Trans Circuits Syst Video Technol 28(9):2131–2153

Asikuzzaman M, Alam MJ, Lambert AJ, Pickering MR (2016) Robust dt cwt-based dibr 3d video watermarking using chrominance embedding. IEEE Trans Multimedia 18(9):1733–1748

Azad R, Niggemeier L, Hüttemann M, Kazerouni A, Aghdam EK, Velichko Y, Bagci U, Merhof D (2024) Beyond self-attention: Deformable large kernel attention for medical image segmentation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pp. 1287–1297

Carreira J, Noland E, Banki-Horvath A, Hillier C, Zisserman A (2018) A short note about kinetics-600. arXiv preprint arXiv:1808.01340

Carreira J, Zisserman A (2017) Quo vadis, action recognition? a new model and the kinetics dataset. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6299–6308

Chang Q, Huang L, Liu S, Liu H, Yang T, Wang Y (2022) Blind robust video watermarking based on adaptive region selection and channel reference. In: Proc. of the ACM International Conference on Multimedia, pp. 2344–2350

Chen S, Malik A, Zhang X, Feng G, Wu H (2023) A fast method for robust video watermarking based on zernike moments. IEEE Trans Circuits Syst Video Technol 33(12):7342–7353

Chen J, Yu Y, Song S, Wang X, Yang J, Xue Y, Lao Y (2023) Acm multimedia 2023 grand challenge report: Invisible video watermark. In: Proc. of the 31st ACM International Conference on Multimedia, pp. 9630–9634

Cvanet: Cascaded visual attention network for single image super-resolution. Neural Networks **170**, 622–634 (2024)

Dey N, Das P, Roy AB, Das A, Chaudhuri SS (2012) Dwt-dct-svd based intravascular ultrasound video watermarking. In: Proc. of the World Congress on Information and Communication Technologies, pp. 224–229

Fang H, Zhang W, Ma Z, Zhou H, Sun S, Cui H, Yu N (2020) A camera shooting resilient watermarking scheme for underpainting documents. IEEE Trans Circuits Syst Video Technol 30(11):4075–4089

Fang H, Chen K, Qiu Y, Liu J, Xu K, Fang C, Zhang W, Chang E-C (2023) Denol: A few-shot-sample-based decoupling noise layer for cross-channel watermarking robustness. In: Proc. of the ACM International Conference on Multimedia, pp. 7345–7353

Fang H, Jia Z, Ma Z, Chang E-C, Zhang W (2022) Pimog: An effective screen-shooting noise-layer simulation for deep-learning-based watermarking network. In: Proc. of the ACM International Conference on Multimedia, pp. 2267–2275

Guo H, Zhang Q, Luo J, Guo F, Zhang W, Su X, Li M (2023) Practical deep dispersed watermarking with synchronization and fusion. In: Proc. of the ACM International Conference on Multimedia, pp. 7922–7932

Huan W, Li S, Qian Z, Zhang X (2022) Exploring stable coefficients on joint sub-bands for robust video watermarking in dt cwt domain. IEEE Trans Circuits Syst Video Technol 32(4):1955–1965

Huynh-Thu Q, Ghanbari M (2008) Scope of validity of psnr in image/video quality assessment. Electron Lett 44(13):800–801

Jia J, Gao Z, Zhu D, Min X, Hu M, Zhai G (2022) Rivie: Robust inherent video information embedding. IEEE Trans Multimedia 25:7364–7377

Jia J, Gao Z, Zhu D, Min X, Zhai G, Yang X (2022) Learning invisible markers for hidden codes in offline-to-online photography. In: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2273–2282

Joshi S, Pande J, Singh B (2018) Watermarking of audio signals using iris data for protecting intellectual property rights of multiple owners. Int J Inf Technol 10:559–566

Kumar S, Singh BK, Yadav M (2020) A recent survey on multimedia and database watermarking. Multimedia Tools and Applications 79:20149–20197

Liu J, Fan X, Jiang J, Liu R, Luo Z (2022) Learning a deep multi-scale feature ensemble and an edge-attention guidance for image fusion. IEEE Trans Circuits Syst Video Technol 32(1):105–119

Liu Q, Yang S, Liu J, Zhao L, Xiong P, Shen J (2023) An efficient video watermark method using blockchain. Knowl-Based Syst 259:110066

Liu Y, Guo M, Zhang J, Zhu Y, Xie X (2019) A novel two-stage separable deep learning framework for practical blind watermarking. In: Proc. of the ACM International Conference on Multimedia, pp. 1509–1517

Li J, Wang H, Yang L, Liu D (2022) Dst-based video watermarking robust to lossy channel compression. In: Proc. of the IEEE International Workshop on Multimedia Signal Processing, pp. 1–6

Lu J, Ni J, Su W, Xie H (2022) Wavelet-based cnn for robust and high-capacity image watermarking. In: Proc. of the IEEE International Conference on Multimedia and Expo, pp. 1–6

Luo Y, Zhou T, Cui S, Ye Y, Liu F, Cai Z (2023) Fixing the double agent vulnerability of deep watermarking: A patch-level solution against artwork plagiarism. IEEE Trans Circuits Syst Video Technol 34(3):1670–1683

Luo X, Li Y, Chang H, Liu C, Milanfar P, Yang F (2023) Dvmark: a deep multiscale framework for video watermarking. IEEE Transactions on Image Processing

Luo Y, Zhou T, Liu F, Cai Z (2023) Irwart: Levering watermarking performance for protecting high-quality artwork images. In: Proc. of the ACM Web Conference, pp. 2340–2348

Marszalek M, Laptev I, Schmid C (2009) Actions in context. In: Proc. of IEEE Conference on Computer Vision and Pattern Recognition, pp. 2929–2936. IEEE

Patel R, Lad K, Patel M, Desai M (2021) An efficient dct-sbpm based video steganography in compressed domain. Int J Inf Technol 13:1073–1078

Qin X, Zhang Z, Huang C, Dehghan M, Zaiane OR, Jagersand M (2020) U2-net: Going deeper with nested u-structure for salient object detection. Pattern Recogn 106:107404

Savakar DG, Ghuli A (2019) Robust invisible digital image watermarking using hybrid scheme. Arab J Sci Eng 44(4):3995–4008

Sharma VK, Mir RN (2022) An enhanced time efficient technique for image watermarking using ant colony optimization and light gradient boosting algorithm. Journal of King Saud University-Computer and Information Sciences 34(3):615–626

Wu H, Liu G, Yao Y, Zhang X (2021) Watermarking neural networks with watermarked images. IEEE Trans Circuits Syst Video Technol 31(7):2591–2601

Wu H, Chen C, Hou J, Liao L, Wang A, Sun W, Yan Q, Lin W (2022) Fast-vqa: Efficient end-to-end video quality assessment with fragment sampling. In: Proc. of the European Conference on Computer Vision, vol. 13666, pp. 538–554

Ye G, Gao J, Wang Y, Song L, Wei X (2023) Itov: Efficiently adapting deep learning-based image watermarking to video watermarking. In: Proc. of the International Conference on Culture-Oriented Science and Technology, pp. 192–197

Yoo I, Chang H, Luo X, Stava O, Liu C, Milanfar P, Yang F (2022) Deep 3d-to-2d watermarking: Embedding messages in 3d meshes and extracting them from 2d renderings. In: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10031–10040

Zamir SW, Arora A, Khan S, Hayat M, Khan FS, Yang M-H (2022) Restormer: Efficient transformer for high-resolution image restoration. In: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5728–5739

Zhang KA, Xu L, Cuesta-Infante A, Veeramachaneni K (2019) Robust invisible video watermarking with attention. arXiv preprint arXiv:1909.01285

Zhang Z, Wang H, Wang G, Wu X (2024) Hide and track: Towards blind video watermarking network in frequency domain. Neurocomputing 579:127435

Zhang R, Isola P, Efros AA, Shechtman E, Wang O (2018) The unreasonable effectiveness of deep features as a perceptual metric. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 586–595

Zhang Y, Ni J, Su W, Liao X (2023) A novel deep video watermarking framework with enhanced robustness to h. 264/avc compression. In: Proc. of the ACM International Conference on Multimedia, pp. 8095–8104

Zhu J, Kaplan R, Johnson J, Fei-Fei L (2018) Hidden: Hiding data with deep networks. In: Proc. of the European Conference on Computer Vision, pp. 657–672

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Quan Yan[1] · Yuanjing Luo[2] · Zhangdong Wang[1] · Junhua Xi[1] · Geming Xia[1] · Zhiping Cai[1]

✉ Junhua Xi
hjh17@nudt.edu.cn

✉ Zhiping Cai
zpcai@nudt.edu.cn

Quan Yan
yanquan21@nudt.edu.cn

Yuanjing Luo
luoyuanjing@csuft.edu.cn

Zhangdong Wang
wangzd@nudt.edu.cn

Geming Xia
xiageming@163.com

[1] College of Computer Science and Technology, National University of Defense Technology, Changsha 410073, Hunan, China

[2] College of Computer and Mathematics, Central South University of Forestry and Technology, Changsha 410004, Hunan, China