Information Sciences 742 (2026) 123030

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Versatile and harmless deepfake proactive forensics via conditional watermarking

Xiaoshuai Wu[a], Xin Liao[a,*], Jie Zhang[b], Mingyue Chen[a], Yufeng Wu[a], Jinlin Guo[c]

[a] College of Cyber Science and Technology, Hunan University, Changsha, China
[b] Centre for Frontier AI Research, Agency for Science, Technology and Research, Singapore
[c] Faculty of System Engineering, National University of Defense Technology, Changsha, China

HIGHLIGHTS

- Motivation: Clarify the differences and limitations between watermarking real and deepfake images.
- Framework: Present a brand-new feature modulation-based conditional watermarking framework.
- Application: Accomplish versatile and harmless proactive forensics within a unified watermarking framework.

ARTICLE INFO

Keywords:
Data hiding
Image watermarking
Deepfake forensics

ABSTRACT

With the surge of deepfake-generated images, recent studies suggest robust watermarking to protect face images, allowing for provenance tracking through watermark extraction by the *authorized party*. However, the watermarked real image may be manipulated by a *malicious party* to convert it into a deepfake image, whereas the watermarked deepfake image may be misclassified as real by an *unauthorized third party*. To complement the forensics ecosystem, we present a conditional watermarking framework, which includes a simple yet effective condition network that accepts the target condition as input and generates the condition vectors to modulate the base watermarking network. By injecting conditions into the watermark decoder, the embedded watermark can be extracted flexibly at different levels of robustness. Moreover, by injecting conditions into the watermark encoder, we can responsibly control the detection results of the watermarked image by other passive detectors. Extensive experiments demonstrate the effectiveness of the proposed framework on typical forensic cases, including proactive detection for authenticating real images and detectability enhancement for regulating deepfake images. Quantitatively, the conditioned decoder achieves an average bit error rate below 1 % under common distortions and around 50 % under malicious distortions; meanwhile, the conditioned encoder helps improve the detection accuracy of the passive detector to above 99 %.

1. Introduction

Deepfake technology poses serious security risks by enabling the creation of authentic-seeming facial content [1]. Despite its positive potential in applications such as entertainment, education, and art, deepfake technology is also notorious for malicious uses like spreading misinformation, manipulating opinions, and forging evidence. Therefore, distinguishing between real and deepfake-generated images has become an important and urgent issue. Consequently, most studies in passive forensics focus on developing

* Corresponding author.
Email address: xinliao@hnu.edu.cn (X. Liao).

https://doi.org/10.1016/j.ins.2025.123030
Received 30 October 2025; Received in revised form 20 December 2025; Accepted 22 December 2025
Available online 26 December 2025

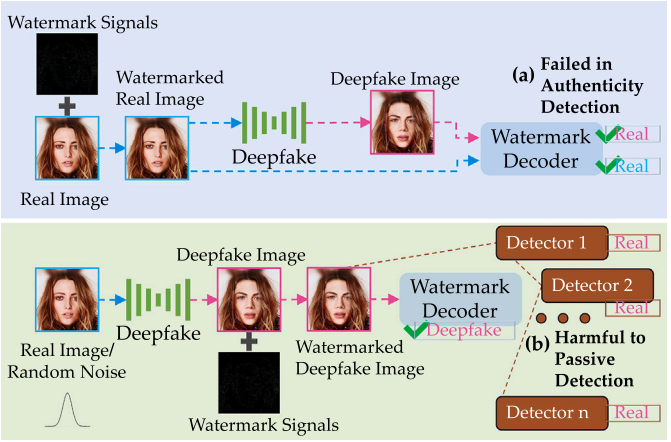


Fig. 1. Overall differences between watermarking real and deepfake images.

data-driven, image-level deepfake detectors to identify images that have already been created [2]. However, due to the increasing realism of deepfake images, fewer detectable forgery signals are left for passive detectors to learn. Hence, recent research is pivoting toward proactive methods to protect real images against foreseeable manipulations [3,4].

The research on proactive methods can be roughly divided into two main categories, namely proactive defense and proactive forensics. Both of them proactively insert invisible signals into the real images that need protection, but they differ in their underlying mechanisms and purposes. The first category is based on adversarial perturbation, which is the most direct way to completely disable the creation of deepfake images by attacking the deepfake models [5,6]. However, it is noteworthy that while the adversarial attacks against deepfake models prevent high-quality deepfake generation, the resulting abnormal outputs may attract unwanted attention and also disable the benign utilization of deepfakes. The second category builds upon encoder-decoder-based image watermarking, enabling the *authorized party* to embed and extract the watermark for distinct forensic purposes, such as provenance tracking and copyright protection via robust watermarking, or tamper detection via fragile watermarking [7–9]. Moreover, robust watermarking can be applied to deepfake images to regulate their use by embedding the provenance evidence that explicitly identifies them as deepfake-generated [10–12]. Nevertheless, it is proposed to recognize the nuanced differences between watermarking real and deepfake images, as illustrated in Fig. 1.

In the top panel of Fig. 1, when the watermarked real image is manipulated by the *malicious party*, the overly robust watermark can still be correctly extracted even after the manipulation, making it difficult to proactively detect the authenticity of the image in a blind setup where the original image is unavailable. While the semi-fragile watermarking, FaceSigns [7], allows the extracted watermark to change after deepfake manipulation, it cannot provide the ground-truth watermark for comparison in authentication, nor can it track provenance once the watermarked image is manipulated. An intuitive solution involves embedding robust and semi-fragile watermarks in sequence; however, the two signals may significantly affect each other, and embedding these independent watermarks into separate regions inevitably increases the complexity of implementation. Therefore, it is of great importance to propose a versatile proactive forensics solution for authenticating real images, which accomplishes both provenance tracking and proactive detection immediately using the extracted watermark.

In the bottom panel of Fig. 1, when applying watermarking to regulate the deepfake image, the watermark signals can interfere with the forgery signals used for passive detection, severely decreasing the detection accuracy of a wide range of passive detectors. Due to the coexistence of proactive watermark embedding and passive deepfake detection in real-life scenarios, the watermarked deepfake image could be processed by the *unauthorized third party* using downstream passive detectors if the watermark decoder is not publicly available. Unfortunately, although the watermark signals are invisible to the human eye, they can interfere with other passive detectors, which are more widely deployed than a single watermark decoder. This can be explained by the fact that the watermark signals are prone to overlap with the forgery signals used for passive detection. As a result, the watermarked deepfake image is more likely to be predicted as real by most detectors designed to identify fake patterns [13]. Therefore, it is of great significance to propose a harmless proactive forensics solution for regulating deepfake images, which achieves the purposes of provenance tracking and detectability enhancement simultaneously.

However, the conventional robust and fragile watermarking schemes are heuristically designed to withstand common digital distortions [14,15] or to detect local pixel-wise alterations [16,17]. As such, they are less relevant for countering the sophisticated semantic manipulations generated by modern deepfake models. Although some learning-based watermarking frameworks have been developed for proactive forensics against deepfakes, several problems remain unsolved. For example, SepMark [8] incorporates two separable decoders into the watermarking network, one for provenance tracking and another for deepfake detection, but this increases the model complexity and impacts the training of the single encoder. In addition, AdvMark [9], a harmless proactive forensics solution that relies on supervised fine-tuning on real and deepfake-generated images, cannot be easily generalized to unseen types of

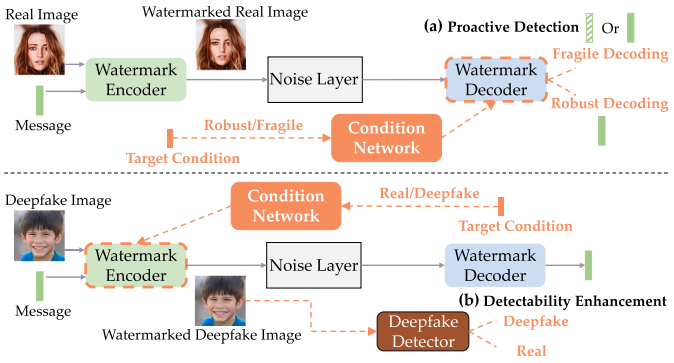


Fig. 2. Overview of our proposed ConMark. It consists of the base watermarking network and the condition network. The base watermarking network handles watermark encoding (embedding) and decoding (extraction), while the condition network adjusts the features of watermark encoding or decoding by changing the input condition.

deepfake images. More importantly, as far as we know, versatile and harmless forensics remain unaddressed in a unified watermarking framework, aiming to simultaneously achieve provenance tracking, proactive detection, and detectability enhancement.

To address the above non-trivial challenges, in this paper, we present a conditional watermarking framework to rule them all, which spans multiple functionalities, including provenance tracking, proactive detection, and detectability enhancement. The core of our proposed ConMark is a simple yet effective condition network that accepts the target condition as input and generates the condition vectors to modulate the base watermarking network. While the concept of conditioning is well-established in generative modeling, such as in conditional GANs and diffusion models, the conditionalization of the watermark decoder/encoder has not yet been explored in learning-based image watermarking, especially in the context of deepfake forensics. As depicted in Fig. 2, by changing the input condition, the features of watermark decoding and encoding can be effectively adjusted. To protect the authenticity of real images, the embedded watermark can be extracted flexibly at different levels of robustness via a single conditioned decoder, allowing proactive detection by matching the extracted watermark messages. For the regulation of deepfake images, the detection results of the watermarked image by deepfake detectors can be responsibly controlled through self-supervised training on only real images, enabling detectability enhancement by inputting the correct condition. Compared to prior art, we have the following new contributions:

1) We address versatile and harmless proactive forensics in a unified watermarking framework, including proactive detection for authenticating real images and detectability enhancement for regulating deepfake images, extending beyond the single-function of provenance tracking. To the best of our knowledge, this is the first work to recognize the fundamental distinction and inherent limitations between watermarking real and deepfake images, in the context of deepfake forensics.
2) We present a conditional watermarking framework, comprising the base watermarking network and the condition network, which brings a highly efficient, scalable, and elegant paradigm to the implementation of multipurpose watermarking through changing the target condition. Provenance tracking, together with proactive detection and detectability enhancement, is combined through the condition injection of the base watermark decoder and/or encoder via the condition network(s).
3) Extensive experiments conducted on real and deepfake images demonstrate the versatility of the proposed ConMark, including the ability to extract the embedded watermark with different levels of robustness using a single conditioned decoder, as well as the harmlessness of the watermarked image through a single conditioned encoder, which allows the image to be more easily distinguished by other passive detectors.

The remainder of this paper is organized as follows. Firstly, Section 2 briefly describes the foundational SepMark, which is used as the base watermarking network. This is followed by the proposed conditional watermarking framework in Section 3, including the condition injection of the base watermark decoder and/or encoder via the condition network(s). Next, the experimental results regarding visual quality, robustness, and harmlessness are analyzed in Section 4. Finally, Section 5 provides our conclusion.

2. Preliminaries

Before introducing the conditional watermarking framework, we briefly describe the watermark encoder, the random forward noise pool, and the watermark decoder, which serve as the main components of the base watermarking network shown in Fig. 4(a). For clarity, the main notations used throughout the paper are summarized in Table 1.

2.1. Watermark encoder

The encoder $\boldsymbol{E}$ receives the host image $I_{ho} \in \mathbb{R}^{B\times 3\times H\times W}$ and watermark message $M_{en} \in \{-\alpha, \alpha\}^{B\times L}$ (converted from $\{0,1\}^{B\times L}$) as inputs. The symbol B stands for the batch size, H and W represent the height and width of the image, while L and α represent the

Table 1
Notations.

I_{ho}	host image
M_{en}	encoded watermark message
I_{en}	watermarked image
I_{no}	noise distorted image
M_{rd}	robust decoded watermark message
M_{fd}	fragile decoded watermark message
c	target condition
$\boldsymbol{E}$	watermark encoder
$\boldsymbol{E}_c$	conditioned watermark encoder
$\boldsymbol{R}$	random forward noise pool
$\boldsymbol{D}_r$	robust watermark decoder
$\boldsymbol{D}_f$	fragile watermark decoder
$\boldsymbol{D}_c$	conditioned watermark decoder
$\boldsymbol{Ad}$	adversary discriminator
$\boldsymbol{\mathcal{D}}_{\{1,\ldots,n\}}$	deepfake detectors
$\boldsymbol{C}$	condition network

Fig. 3. A pictorial illustration of the threat model.

length and scaling factor of the message. In the main branch, I_{ho} is fed into a UNet-like structure. First, in the DOWN path, "conv-in-relu" with stride 2 halves the spatial dimensions, and "Double Conv (2× conv-in-relu)" doubles the number of feature channels. In the parallel branch, M_{en} is fed into several "Diffusion Block (FC layer-Conv)" to increase redundancy, expanding its length from L to $L \times L$. Next, in the UP path, nearest-neighbour interpolation doubles the spatial dimensions and "conv-in-relu" halves the number of feature channels, followed by the concatenation with both the corresponding feature maps from the DOWN path and the redundant message after the interpolation operation. Moreover, the "SE-Residual Block", inserted with Squeeze-and-Excitation [18], is applied to the concatenated features to enhance the interaction between different channels. Finally, the feature maps are further concatenated with the host image via a skip connection, and the output is passed through a 1×1 convolution to produce the watermarked image I_{en}.

2.2. Random forward noise pool

The random forward noise pool $\boldsymbol{R}$ randomly samples different distortions as pseudo-noise gap and adds it to the watermarked image I_{en}, resulting in the distorted image I_{no}. Since the simulated distortion struggles to replicate real complex distortion, it is proposed to directly obtain arbitrary real distortion $gap = I_{no} - I_{en}$ in a detached forward propagation. In this way, both standard forward and backward propagation can be realized through differentiable $I_{no} = I_{en} + gap$. It is also worth noting that the noise pool $\boldsymbol{R}$ randomly samples different distortions for each sample in the mini-batch rather than sampling only one distortion, which empirically benefits stable training.

2.3. Watermark decoder

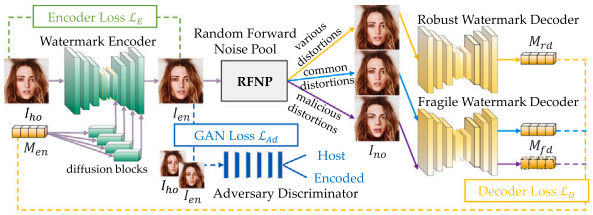
The separable decoders, $\boldsymbol{D}_r$ and $\boldsymbol{D}_f$, extract the messages M_{rd} and M_{fd} from the image I_{no} with different levels of robustness. The structural design of the watermark decoder simply borrows the UNet-like structure from the encoder. Note that after the last "SE-Residual Block" at the end of the UP path, the final output is fed into a 1×1 convolution to produce a single-channel residual image. After that, the residual image is downsampled to $L \times L$, and an "Inverse Diffusion Block (FC layer)" is used to extract the decoded watermark message of length L.

3. Proposed ConMark

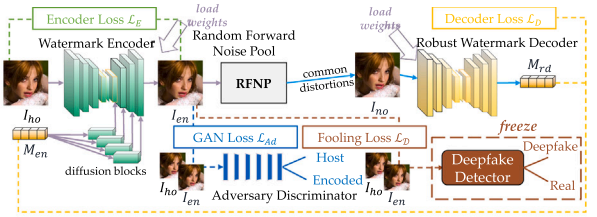
In this section, we elaborate on our proposed conditional watermarking and its applications to deepfake proactive forensics, involving proactive detection for authenticating real images and detectability enhancement for regulating deepfake images.

3.1. Threat model

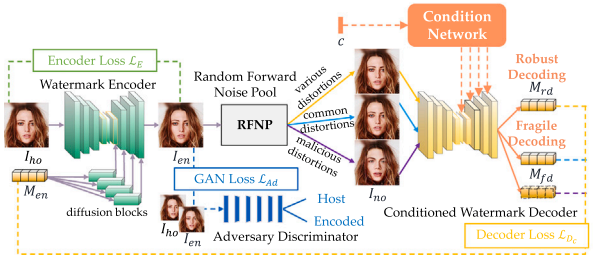
The threat model in Fig. 3 considers three parties: the authorized party, the malicious party, and the unauthorized third party. Let us detail the capabilities and goals of each.



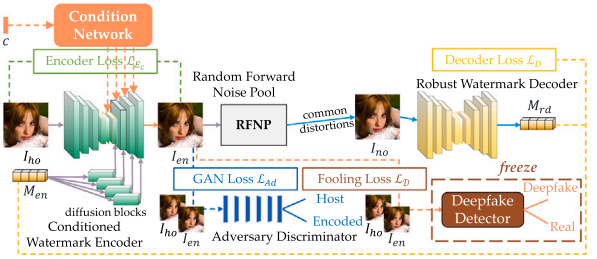
(a) Base Watermarking Network: SepMark



(b) Fine-tuned Watermarking Network: AdvMark



(c) Injecting Conditions into the Watermark Decoder: ConMark-D



(d) Injecting Conditions into the Watermark Encoder: ConMark-E

Fig. 4. Framework comparisons for a series of SepMark, AdvMark, and ConMark. (a) SepMark innovatively integrates a single encoder with two separable decoders, enabling the embedded watermark to be extracted separately at different levels of robustness. (b) AdvMark adversarially fine-tunes the watermark encoder and decoder in a way that the embedded watermark will be beneficial to downstream deepfake detection without requiring any tuning of in-the-wild detectors. (c) ConMark-D cleverly replaces the separable decoders with a single conditioned watermark decoder, where the proactive detection is achieved by flexibly modulating the features of watermark decoding. (d) ConMark-E interactively controls the detection results of the watermarked image by other passive detectors via a single conditioned watermark encoder, where the detectability enhancement is achieved by responsibly modulating the features of watermark encoding.

Authorized Party: The authorized party has access to the use of the watermark encoder and decoder. They are responsible for embedding the watermark (provenance evidence) into their published images, which may be either real or deepfake images, using the watermark encoder. Subsequently, any authorized party with access to the corresponding watermark decoder can extract the watermark for distinct purposes, such as provenance tracking, copyright protection, or tamper detection.

Malicious Party: The malicious party intends to manipulate the face images by altering identities, expressions, or attributes using well-trained deepfake models. Their goal is to transform real images into deepfake images, thereby undermining the authenticity of the image content and distorting the truth.

Unauthorized Third Party: The unauthorized party does not have access to the usage of the watermark encoder and decoder due to security issues. In such a case, the watermarked images are most likely processed by the unauthorized party using other downstream passive detectors, where the watermark encoder and decoder are either unknown or unavailable, or the decoder is mismatched with the encoder.

In the following, we will detail the forensic framework from the perspective of the authorized party.

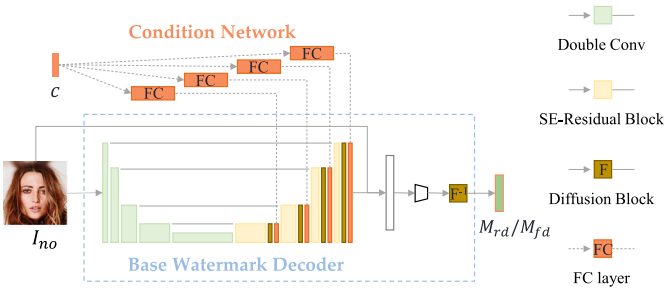
3.2. Overview and forensic cases

The conditional watermarking framework comprises the base watermarking network and the condition network. The base watermarking network is responsible for watermark encoding (embedding) and decoding (extraction), while the condition network modulates the features of watermark decoding and encoding. The foundational SepMark [8] is selected as the base watermarking network in this work. Compared to prior art, as depicted in Fig. 4, the main distinction lies in the additional condition network, which transforms the unconditioned decoder/encoder into a conditioned decoder/encoder.

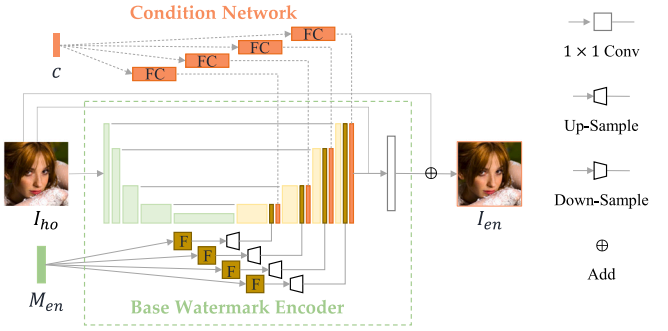
During training, the base watermarking network and the condition network are jointly optimized end-to-end. The condition network takes a target condition as input to generate the condition vectors, which are used to modulate the intermediate features of the base watermarking network. By injecting conditions into the watermark decoder, the embedded watermark can be extracted through either fragile or robust decoding, enabling proactive authentication of watermarked real images through the comparison of fragile and robust decoded messages. Similarly, by injecting conditions into the watermark encoder, the detection results of watermarked images by deepfake detectors can be controlled, thereby enhancing the forensic detectability of watermarked deepfake images by ensuring that the detectors yield accurate detection outcomes.

During inference, only the watermark encoder, the watermark decoder, and the condition network are adopted, and we can simply change the input condition to adjust the features of watermark decoding and encoding. Consequently, depending on the conditioned watermarking network, we can diversify the proactive forensics into the following cases:

1) If the watermark decoder is conditioned, the embedded watermark can be extracted flexibly at different levels of robustness. More precisely, the watermark message can be extracted correctly through either fragile or robust decoding under common distortions that have little effect on image semantics. In contrast, under malicious distortions such as deepfake, which is mainly



(a) Conditioned Watermark Decoder



(b) Conditioned Watermark Encoder

Fig. 5. Details on the conditioned watermark decoder and encoder. The condition network contains several independent fully-connected (FC) layers, placed at the end of each "SE-Residual Block" along the UP path. Each FC layer takes a target condition as input and generates the condition vector to modulate the base watermarking network.

addressed in our work, the watermark message can still be extracted correctly through robust decoding, while fragile decoding yields an incorrect message, implying that the authentic image has been altered since it was watermarked. Even without knowing the exact message of the original watermark, it is possible to compare the two decoded messages to determine whether the watermarked image has been manipulated after publication, and prevent the further spread of the suspicious image with mismatched watermarks.

2) If the watermark encoder is conditioned, the detection results of the watermarked image by other passive detectors can be responsibly controlled. In particular, when the ground-truth label of the host image is known, as is usually the case for the trustworthy publisher, the conditional watermark encoding can be leveraged to force downstream passive detectors to produce a correct prediction, even if the original prediction for the host image is incorrect. The ultimate goal here is to improve the forensic detectability of the image itself, where the obtained watermarked image is even easier to be detected by other unseen passive detectors. Even though the ground-truth label may be unknown, an alternative label can be directly derived from a surrogate detector, and non-interference is thus minimally achieved by keeping the detection result unchanged before and after watermarking.
3) If both the watermark encoder and decoder are conditioned, provenance tracking, proactive detection, and detectability enhancement can be realized simultaneously. This is universally applicable to both real and deepfake images, facilitating versatile and harmless proactive forensics in a unified watermarking framework.

3.3. Conditioned network

Fig. 5 illustrates the designs of our conditioned watermark decoder $\boldsymbol{D}_c$ and encoder $\boldsymbol{E}_c$, each incorporating a condition network to modulate the features of watermark decoding and encoding. Noticeably, the condition network $\boldsymbol{C}$ is extremely lightweight, consisting of only several independent fully-connected (FC) layers. Specifically, we perform the modulation operations at the end of each "SE-Residual Block" along the UP path. Each FC layer takes the condition $c \in \mathbb{R}^{B\times 2}$ as input and generates the condition vector $S_i \in \mathbb{R}^{B\times 2C_i}$, where C_i represents the channel number of the output feature in the i-th "SE-Residual Block". Since the target conditions in this work are discrete scalars, the input condition $c \in \mathbb{R}^{B\times 2}$ is represented using one-hot encoding. Concretely, for the input condition of the decoder, the one-hot vector $[1, 0]$ indicates the target for robust decoding, while $[0, 1]$ represents the target for fragile decoding. Similarly, for the input condition of the encoder, $[1, 0]$ denotes that the encoding should enable the downstream passive detectors to classify the watermarked image as real, whereas $[0, 1]$ implies that the encoding should make the detectors classify the watermarked image as deepfake-generated. The condition vector $S_i \in \mathbb{R}^{B\times 2C_i}$ is composed of two affine transformation parameters $\gamma_i, \beta_i \in \mathbb{R}^{B\times C_i}$. Accordingly, the base watermarking network is modulated through Feature-wise Linear Modulation (FiLM) [19] with γ_i and β_i:

$$\widetilde{F}_i = F_i \times \gamma_i + \beta_i, \tag{1}$$

where F_i is the output feature to be modulated in the base watermarking network, and $\widetilde{F}_i$ refers to the transformation result of the i-th modulation operation. Empirically, the base watermarking network and the condition network collaborate well, where the intermediate features of the base network can be effectively modulated by channel-wise scaling and shifting with only a few parameters. According to the forensic cases aforementioned in Section 3.2, three variants are implemented depending on the conditioned component, specifically the decoder, encoder, and both encoder-decoder.

3.4. Injecting conditions into the watermark decoder

As Fig. 5(a) illustrates, the conditioned decoder $\boldsymbol{D_c}$ receives both the distorted image I_{no} and the target condition c as inputs. To enable the extraction of the embedded watermark at different levels of robustness via a single conditioned decoder, as shown in Fig. 4(c), we design the following loss functions.

The robust decoded watermark message M_{rd}, extracted from the image I_{no} corrupted by either common or malicious distortions, should be identical to the original embedded message M_{en}:

$$\mathcal{L}_{D_{c1}} = L_2(M_{en}, M_{rd}) = L_2(M_{en}, \boldsymbol{D_c}(\phi; \boldsymbol{R}(\delta_c, \delta_m; I_{en}), [1, 0])), \tag{2}$$

where

$$I_{en} = \boldsymbol{E}(\theta; I_{ho}, M_{en}), \tag{3}$$

and θ, ϕ indicate the trainable parameters of the encoder and decoder, respectively, by default. δ_c, δ_m denote the common distortions and malicious distortions, respectively, which are possibly sampled by the random forward noise pool $\boldsymbol{R}$.

The fragile decoded watermark message M_{fd}, extracted from the commonly distorted image I_{no}, is expected to be consistent with the message M_{en}:

$$\mathcal{L}_{D_{c2}} = L_2(M_{en}, M_{fd}) = L_2(M_{en}, \boldsymbol{D_c}(\phi; \boldsymbol{R}(\delta_c; I_{en}), [0, 1])). \tag{4}$$

In contrast, the message M_{fd} should be inconsistent when extracted from the image I_{no} corrupted by malicious distortions:

$$\mathcal{L}_{D_{c3}} = L_2(0, M_{fd}) = L_2(0, \boldsymbol{D_c}(\phi; \boldsymbol{R}(\delta_m; I_{en}), [0, 1])), \tag{5}$$

where the L_2 distance between 0 and M_{fd} is optimized. It should be emphasized that the watermark message was converted from the binary representation $\{0, 1\}^{B\times L}$ into the zero-mean representation $\{-\alpha, \alpha\}^{B\times L}$. Therefore, by using 0 as the threshold, the extracted message can be converted back into the binary representation. Specifically, if a value in the zero-mean representation is greater than 0, the corresponding bit is recovered as 1; otherwise, it is recovered as 0. Suppressing the zero-mean watermark message to the decision boundary 0 makes the extracted M_{fd} close to random guesses. In total,

$$\mathcal{L}_{D_c} = \mathcal{L}_{D_{c1}} + \mathcal{L}_{D_{c2}} + \mathcal{L}_{D_{c3}}, \tag{6}$$

where three distinct input combinations are involved in optimizing the single conditioned decoder $\boldsymbol{D_c}$, namely: 1) the condition $[1, 0]$ and the noisy image $\boldsymbol{R}(\delta_c, \delta_m; I_{en})$; 2) the condition $[0, 1]$ and the image $\boldsymbol{R}(\delta_c; I_{en})$; 3) the condition $[0, 1]$ and the image $\boldsymbol{R}(\delta_m; I_{en})$.

Moreover, the watermarked image I_{en} should be visually similar to the original host image I_{ho}, and the watermark encoder $\boldsymbol{E}$ is therefore optimized by:

$$\mathcal{L}_E = L_2(I_{ho}, I_{en}) = L_2(I_{ho}, \boldsymbol{E}(\theta; I_{ho}, M_{en})). \tag{7}$$

Furthermore, to supervise the visual quality of watermarked image I_{en}, a discriminator $\boldsymbol{Ad}$ is trained alternately with the main encoder-decoder. PatchGAN [20] is used by default, where its parameters are updated by:

$$\mathcal{L}_{Adv} = L_2(\boldsymbol{Ad}(\psi; I_{ho}) - \overline{\boldsymbol{Ad}}(\psi; I_{en}), 1) + L_2(\boldsymbol{Ad}(\psi; I_{en}) - \overline{\boldsymbol{Ad}}(\psi; I_{ho}), -1), \tag{8}$$

where

$$\boldsymbol{Ad}(\psi; I_{en}) = \boldsymbol{Ad}(\psi; \boldsymbol{E}(I_{ho}, M_{en})), \tag{9}$$

and $\overline{\boldsymbol{Ad}}(I)$ denotes the average $\boldsymbol{Ad}(I)$ in the mini-batch, and ψ represents the trainable parameters of the discriminator. The relativistic GAN loss [21] here estimates the probability that the host image I_{ho} is more pristine (i.e., without watermark signals) than the watermarked image I_{en}, on average. Note that the discriminator also participates in updating the encoder $\boldsymbol{E}$:

$$\mathcal{L}_{Ad} = L_2(\boldsymbol{Ad}(I_{ho}) - \overline{\boldsymbol{Ad}}(I_{en}), -1) + L_2(\boldsymbol{Ad}(I_{en}) - \overline{\boldsymbol{Ad}}(I_{ho}), 1), \tag{10}$$

where

$$\boldsymbol{Ad}(I_{en}) = \boldsymbol{Ad}(\boldsymbol{E}(\theta; I_{ho}, M_{en})). \tag{11}$$

To summarize, the total loss for the unconditioned watermark encoder and conditioned decoder can be formulated as:

$$\mathcal{L}_{ConMark-D} = \lambda_1 \mathcal{L}_{Ad} + \lambda_2 \mathcal{L}_E + \lambda_3 \mathcal{L}_{D_c}, \tag{12}$$

where λ_1, λ_2, λ_3 are the weights for the respective loss terms, and they are set to $0.1, 1, 10$ by default.

3.5. Injecting conditions into the watermark encoder

As illustrated in Fig. 5(b), the conditioned encoder $\boldsymbol{E_c}$ receives the host image I_{ho}, watermark message M_{en}, and target condition c as inputs. We enforce the conditioned encoder $\boldsymbol{E_c}$ to first produce the watermark residuals and add them to the host image I_{ho}, finally producing the watermarked image I_{en}. Empirically, this change benefits the smooth production of the watermarked image across different conditions. Consequently, Eq. (7) is accordingly modified to:

$$\mathcal{L}_{E_c} = L_2(I_{ho}, I_{en}) = L_2(I_{ho}, I_{ho} + \boldsymbol{E_c}(\theta; I_{ho}, M_{en}, c)). \tag{13}$$

To responsibly control the detection results of the watermarked image by other passive detectors via a single conditioned watermark encoder, as can be seen in Fig. 4(d), we freeze a surrogate deepfake detector $\boldsymbol{D}_i$ and force the detector to predict the watermarked image I_{en} with the pseudo label determined by the target condition c, through updating the conditioned encoder $\boldsymbol{E_c}$:

$$\mathcal{L}_D = \mathcal{L}_{D_i} = \mathcal{F}(D_i(I_{en}), c) = \mathcal{F}(D_i(I_{ho} + \boldsymbol{E_c}(\theta; I_{ho}, M_{en}, c)), c), \tag{14}$$

where $\mathcal{F}$ denotes the loss function (e.g., binary cross-entropy loss) used to train the original detector $\boldsymbol{D}_i$, and the input condition c is randomly sampled from $[1, 0]$ and $[0, 1]$ for each sample in the mini-batch and then mapped to the pseudo labels 0 and 1, respectively. It is worth mentioning that during training in a self-supervised manner, only real images are required, and no deepfake-generated images are needed; this design helps to generalize to unseen types of deepfake images.

We also note that only the unconditioned decoder $\boldsymbol{D_r}$ is used here, as well as sampling from the common distortions; hence the decoder loss is expressed as:

$$\mathcal{L}_D = L_2(M_{en}, M_{rd}) = L_2(M_{en}, \boldsymbol{D_r}(\phi; \boldsymbol{R}(\delta_c; I_{en}))). \tag{15}$$

Furthermore, based on the observation of GAN loss during the training, we empirically discovered that under the input condition $[0, 1]$, the discrimination ability of the patch-based discriminator is relatively stronger than the generation ability of the conditioned watermark encoder. This leads to lower generation quality for watermarked images conditioned on $[0, 1]$ compared to those conditioned on $[1, 0]$, and increases the risk of training instability. Fortunately, inspired by R3GAN [22], we use $R_1 + R_2$ regularization to penalize the gradient norm of the discriminator on both non-watermarked and watermarked images. The combination of relativistic GAN loss [21] and zero-centered gradient penalties has been theoretically proven to ensure local convergence guarantees without necessitating other complex tricks. Moreover, the discriminator structure is modified to the global form by substituting the final convolutional layer of PatchGAN [20] with global average pooling and an FC layer. With these two configurations, we observed no training instability when injecting different conditions into the watermark encoder.

To sum up, the total loss for the conditioned watermark encoder and unconditioned decoder is equivalent to:

$$\mathcal{L}_{ConMark-E} = \lambda_1 \mathcal{L}_{Ad} + \lambda_2 \mathcal{L}_{E_c} + \lambda_3 \mathcal{L}_D + \lambda_4 \mathcal{L}_D, \tag{16}$$

where λ_4 is the weight for the fooling loss term, set to 0.1 by default. It is also worthy of note that we can simultaneously fool an ensemble of detectors, where Eq. (14) is changed to:

$$\mathcal{L}_D = \mathcal{L}_{D_{\{1,\dots,n\}}} = (\lambda_{D_1} \mathcal{L}_{D_1} + \cdots + \lambda_{D_n} \mathcal{L}_{D_n})/n, \tag{17}$$

in which $\{\lambda_{D_1}, \dots, \lambda_{D_n}\}$ is a set of weights for each detector.

3.6. Injecting conditions into both the encoder and decoder

In accordance with Fig. 6, building upon the constructions of ConMark-E, we transform the unconditioned watermark decoder $\boldsymbol{D_r}$ into a conditioned decoder $\boldsymbol{D_c}$ by incorporating an additional condition network. Without any other structural changes, the conditioned encoder $\boldsymbol{E_c}$ and the conditioned decoder $\boldsymbol{D_c}$ could be jointly trained from scratch, and the end-to-end training here aims to minimize:

$$\mathcal{L}_{ConMark-ED} = \lambda_1 \mathcal{L}_{Ad} + \lambda_2 \mathcal{L}_{E_c} + \lambda_3 \mathcal{L}_{D_c} + \lambda_4 \mathcal{L}_D. \tag{18}$$

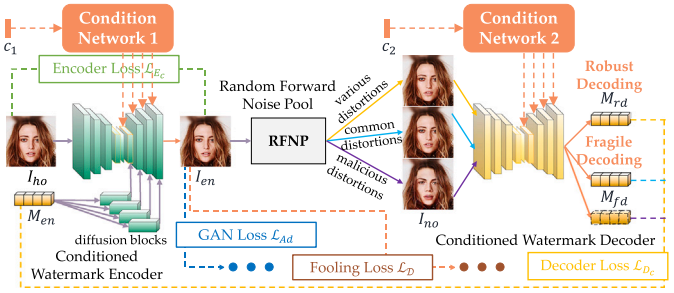


Fig. 6. Injecting conditions into both the encoder and decoder. ConMark-ED is universally applicable to both real and deepfake images, facilitating versatile and harmless proactive forensics in a unified watermarking framework.

Algorithm 1: Training procedure for ConMark-ED.

Input : Host image I_{ho}, watermark message M_{en}

Output: Optimized parameters θ for conditioned encoder $\boldsymbol{E_c}$, ϕ for conditioned decoder $\boldsymbol{D_c}$

Initialize the conditioned encoder $\boldsymbol{E_c}$, conditioned decoder $\boldsymbol{D_c}$, and adversary discriminator $\boldsymbol{Ad}$ with random parameters
Initialize the surrogate passive detector $\boldsymbol{\mathcal{D}}$ with pre-trained parameters
for each training iteration **do**
 # Train Conditioned Encoder-Decoder (End-to-End)
Prepare pseudo condition c, i.e., $[1,0]$ or $[0,1]$, for self-supervision
Generate the watermarked image: $I_{en} = I_{ho} + \boldsymbol{E_c}(\theta; I_{ho}, M_{en}, c)$
Generate three versions of distorted images: $I_{no_1} = \boldsymbol{R}(\delta_c, \delta_m; I_{en})$, $I_{no_2} = \boldsymbol{R}(\delta_c; I_{en})$, and $I_{no_3} = \boldsymbol{R}(\delta_m; I_{en})$
Extract the robust decoded message: $M_{rd} = \boldsymbol{D_c}(\phi; I_{no_1}, [1,0])$ and the fragile decoded message: $M_{fd} = \boldsymbol{D_c}(\phi; I_{no_2}, [0,1]) \vee \boldsymbol{D_c}(\phi; I_{no_3}, [0,1])$
Calculate the GAN loss $\mathcal{L}_{Ad}$ (Eq. 10), encoder loss $\mathcal{L}_{E_c}$ (Eq. 13), decoder loss $\mathcal{L}_{D_c}$ (Eq. 6), and fooling loss $\mathcal{L}_{\mathcal{D}}$ (Eq. 14)
Update θ and ϕ by minimizing:

$$\mathcal{L}_{ConMark-ED} = 0.1 \times \mathcal{L}_{Ad} + 1 \times \mathcal{L}_{E_c} + 10 \times \mathcal{L}_{D_c} + 0.1 \times \mathcal{L}_{\mathcal{D}}$$

 # Train Adversary Discriminator with Gradient Penalty ($R_1 + R_2$)
Compute the GAN loss for the discriminator: $\mathcal{L}_{Adv}$, i.e., Eq. (8)
Compute the gradient penalty terms: $\mathcal{L}_{GP1}$ and $\mathcal{L}_{GP2}$
Update ψ by minimizing:

$$\mathcal{L}_{Adv\_GP} = \mathcal{L}_{Adv} + 10 \times \mathcal{L}_{GP1} + 10 \times \mathcal{L}_{GP2}$$

end for

Alternatively, the well-trained parameters of ConMark-E can be used as the starting point for fine-tuning the unconditioned watermark decoder. More specifically, the parameters of the conditioned encoder $\boldsymbol{E_c}$ are fixed and kept unchanged, while only the unconditioned decoder $\boldsymbol{D_r}$ is fine-tuned using the condition network to transform it into the conditioned decoder $\boldsymbol{D_c}$. The objective of fine-tuning is thus to minimize:

$$\mathcal{L}_{ConMark-ED'} = \lambda_3 \mathcal{L}_{D_c}. \tag{19}$$

In this work, we choose to jointly train the conditioned encoder and decoder from scratch by default, as it provides a more qualified balance among visual quality, robustness, and harmlessness. In summary, the well-trained ConMark-ED is universally applicable to both real and deepfake images, facilitating versatile and harmless proactive forensics in a unified watermarking framework. Accordingly, the pseudocode for training ConMark-ED is provided in Algorithm 1.

3.7. An illustrative example

Taking ConMark-ED as an example, in which both the encoder and decoder are conditioned, we present a numerical description to illustrate its use cases. The inference procedure mainly consists of *Embedding Stage* and *Extraction Stage*, which can be briefly introduced as follows:

Embedding Stage:

1) For a given host image I_{ho}, the authorized party can first confirm the authenticity of the image before the watermark embedding (the watermark M_{en} can be arbitrary messages although we use *'Real'* or *'Deepfake'* for ease of illustration).
2) If I_{ho} is a real image, the watermark encoder $\boldsymbol{E_c}$ receives the image I_{ho}, the watermark *'Real'*, and the target condition $[1,0]$ as inputs and finally produces the watermarked image I_{en}.
3) In contrast, if I_{ho} is deepfake-generated, the encoder $\boldsymbol{E_c}$ receives the image I_{ho}, the watermark *'Deepfake'*, and the condition $[0,1]$ as inputs to produce the final watermarked image. This helps the downstream passive detectors more easily identify I_{en} as a deepfake-generated image. In particular, the detection accuracy of the watermarked images can be improved to above 99 %, compared to that of the original, non-watermarked images.

Extraction Stage:

1) Any authorized party with access to the watermark decoder $\boldsymbol{D_c}$ can perform the watermark extraction; in detail, the decoder first takes the image (i.e., I_{en} or its distorted version I_{no}) and the target condition $[1,0]$ as inputs to extract the robust decoded watermark M_{rd}.
2) If M_{rd} indicates *'Deepfake'*, the extracted watermark can confirm both provenance and that the image was deepfake-generated, and it is impractical to convert the deepfake image into a real image.
3) Otherwise, if M_{rd} is identical to *'Real'*, the extracted watermark can confirm provenance but cannot easily confirm that the image is authentic. In a fully blind setup, the fragile decoded watermark M_{fd} is further extracted (using the condition $[0,1]$)

Table 2
Experimental settings.

Image sources	real images from CelebA-HQ [23], deepfake-generated images from CelebA-Fake [9]
Data partition	24183/2993/(2824 × 2) for train/valuation/test
Image size	256 × 256
Message length	128
Watermark message	randomly generated binary sequence
Common distortions	Identity, JpegTest ($Q=50$), Resize ($p=50$ %), GaussianBlur ($k=3$, $\sigma=2$), MedianBlur ($k=3$), Brightness ($f=0.5$), Contrast ($f=0.5$), Saturation ($f=0.5$), Hue ($f=0.1$), Dropout ($p=50$ %), SaltPepper ($p=10$ %), GaussianNoise ($\sigma=0.1$)
Malicious distortions	SimSwap, GANimation, StarGAN ($Male$), StarGAN ($Young$), StarGAN ($BlackHair$), StarGAN ($BlondHair$), StarGAN ($BrownHair$)
Passive detectors	Xception [24], EfficientNet [25], CNND [26], FFD [27], PatchForensics [28], MultiAtt [29], RFM [30], RECCE [31], SBI [32], NPR [33]
Compared models	ConMark-D/-E/-ED (ours), MBRS [10], FIN [11], PIMoG [12], FaceSigns [7], SepMark [8], AdvMark [9]
Evaluation metrics	PSNR, SSIM, LPIPS [34], BER, ACC (detection)

Table 3
Comparison of visual quality for the watermarked image I_{en}. The best result is in bold and the second best is underlined.

Model	Real Images (CelebA-HQ)			Deepfake Images (CelebA-Fake)		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
MBRS [10]	40.2000	0.9584	0.0073	<u>40.7617</u>	<u>0.9580</u>	0.0089
FIN [11]	**43.1650**	<u>0.9668</u>	0.0154	**43.3532**	**0.9652**	0.0199
PIMoG [12]	37.7799	0.9312	0.0194	37.9980	0.9298	0.0219
FaceSigns [7]	32.3319	0.9211	0.0260	32.8455	0.9314	0.0248
SepMark [8]	38.5646	0.9328	0.0080	38.4787	0.9274	0.0111
AdvMark [9]	39.0292	0.9321	0.0115	36.9061	0.9163	0.0226
ConMark-D	39.7650	0.9454	<u>0.0059</u>	39.6694	0.9413	**0.0077**
ConMark-E	<u>42.5286</u>	**0.9673**	**0.0045**	39.6216	0.9482	<u>0.0086</u>
ConMark-ED	40.5075	0.9519	0.0064	38.5646	0.9334	0.0126

and matched with M_{rd}, to proactively detect the authenticity of the image. More specifically, the matching between M_{rd} and M_{fd} achieves an average bit error rate below 1 % under common distortions and around 50 % under malicious distortions.

4. Experiments

4.1. Implementation details

For clarity, Table 2 outlines the experimental settings. Since deepfake focuses on the human face, the experiments are mainly conducted on the real images from CelebA-HQ [23], and as a rule of thumb, 24183/2993/2824 face images are used for training, validation, and testing. Additionally, the testing set of the CelebA-Fake dataset [9], which includes 2824 manipulated face images, is employed to test the effectiveness on deepfake images across four categories, i.e., face swapping, expression reenactment, attribute editing, and entire synthesis. All the images are resized to 256×256, the length of the watermark message is set to 128, and the scaling factor α is fixed at 0.1. Following SepMark [8], the robustness is tested against both common and malicious distortions. SimSwap [35], GANimation [36], and StarGAN [37] represent typical face swapping, expression reenactment, and attribute editing, respectively. Similar to AdvMark [9], ten well-trained passive detectors are adopted to assess the harmlessness, which is often overlooked by prior works. Nine of the ten detectors align with the protocol of [9], and an additional detector, NPR [33], is included in the experiments. It is worth mentioning that the passive detectors used in the harmlessness evaluation all operate at the image level.

Our ConMark is implemented using PyTorch and executed on an NVIDIA RTX 4090. The entire training lasted for 100 epochs with a batch size of 16, and unless stated otherwise, the configurations and hyperparameters are the same as those in the base watermarking network [8].

4.1.1. Baselines

The most relevant models are FaceSigns [7], SepMark [8], and AdvMark [9], all designed for deepfake proactive forensics. We directly use their publicly available pre-trained models. Among the other models we compared, MBRS [10] and FIN [11] are common distortion-resistant models that withstand JPEG compression, and PIMoG [12] is a physical distortion-resistant model that resists screen shooting. To guarantee the same image size and message length, we had to train them on the CelebA-HQ dataset from scratch, based on their official codes. As far as we know, this is the first work to simultaneously accomplish provenance tracking, proactive detection, and detectability enhancement. Three variants are implemented depending on the conditioned component, which we termed ConMark-D, ConMark-E, and ConMark-ED, respectively. Unless otherwise stated, the surrogate deepfake detector used in AdvMark and ConMark-E/-ED is Xception [24] by default.

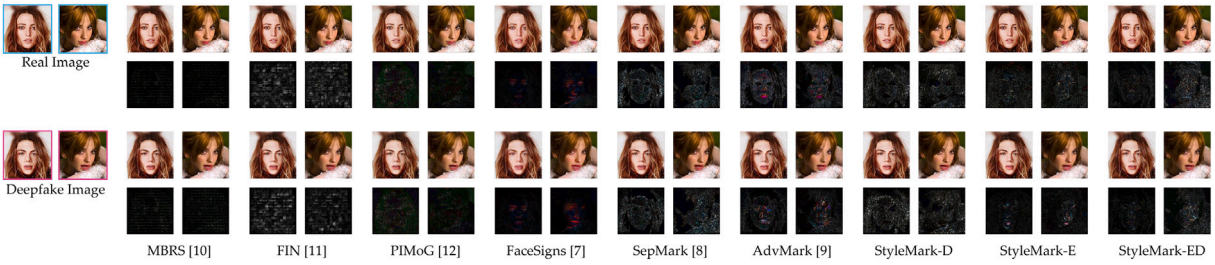


Fig. 7. Visualizations on watermarked images and normalized residuals. Best viewed in color, with zoom for details.

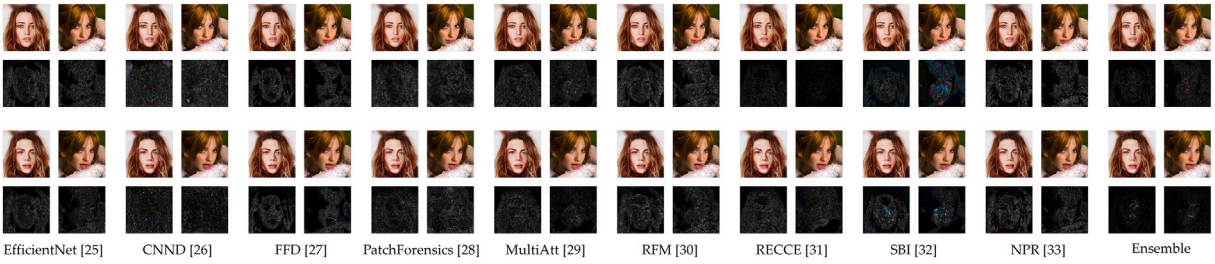


Fig. 8. Illustrations of distinct watermark patterns induced by different surrogate deepfake detectors. Best viewed in color, with zoom for details.

4.1.2. Evaluation metrics

For the evaluation of visual quality, we report the average PSNR, SSIM, and LPIPS [34] of the watermarked images throughout the testing set. For the robustness test, the average bit error rate (BER) is utilized as the default evaluation metric. Suppose that the embedded message $M_E \in \{0,1\}^{B\times L}$ and the extracted message $M_D \in \{0,1\}^{B\times L}$, formally,

$$BER(M_E, M_D) = \frac{1}{B} \times \frac{1}{L} \times \sum_{i=1}^{B} \sum_{j=1}^{L} |M_E^{i\times j} - M_D^{i\times j}| \times 100\ \%, \tag{20}$$

where $|\cdot|$ denotes the absolute value. The harmlessness is evaluated by the detection accuracy (ACC) of the detector D:

$$ACC(D) = \left(1 - \frac{1}{B} \sum_{i=1}^{B} |D(I^{i\times 3\times H\times W}) - y_{gt}^{i\times 1}|\right) \times 100\ \%, \tag{21}$$

where I indicates the input image to be detected, and y_{gt} means the ground-truth label.

4.2. Visual quality

We evaluate the visual quality of watermarked images from both quantitative and qualitative perspectives. As seen in Table 3, nearly all the variants of ConMark outperform their counterparts, SepMark and AdvMark, on both real and deepfake images. The numerical results also surpass those of PIMoG and FaceSigns, in terms of PSNR, SSIM, and LPIPS. It is not surprising that our ConMark-E/-ED performs comparably to MBRS and FIN on real images but not as well on deepfake images. This is mainly explained by our specially designed conditioned encoder, in which the watermark residuals are influenced by the input condition. Encouragingly, the LPIPS values are relatively low, and as visualized in Fig. 7, the images watermarked by ConMark-D/-E/-ED show no clear artifacts that can be observed by the naked eye. Moreover, the min-max normalized residuals amplify the pixel-wise differences between the watermarked and original images. By observation, FaceSigns exhibits localized patterns, AdvMark suffers from perceptible distortions in the watermarked deepfake image, and PIMoG shows color shifts, all of which underperform compared to ours. MBRS and FIN add the residuals primarily in the high-frequency regions of the image; in contrast, ours demonstrate structurally coherent residuals more concentrated in low-frequency components. By analysis, we speculate that the impressive performance mainly derives from our more advanced and stable GAN training, compared to other vanilla GAN-based and PSNR-oriented watermarking networks.

Additionally, the residual patterns of ConMark-E in Fig. 8 reveal that, even with the same embedded watermark messages, distinct patterns are observed across different conditions, i.e., between real and deepfake images, and across different detectors, specifically the one used to guide the self-supervised training. These findings suggest that the rationale behind the watermark residuals may be attributed to the real and fake patterns associated with the decision logic of the surrogate detector used.

Table 4
Robustness test (Bit Error Rate: %) for the commonly distorted image I_{no} on the real images (CelebA-HQ).

Distortion	MBRS [10]	FIN [11]	PIMoG [12]	FaceSigns [7]	SepMark [8]			ConMark-D			AdvMark [9]	ConMark-E	ConMark-ED		
					Robust	Fragile	Match	Robust	Fragile	Match			Robust	Fragile	Match
Identity	0.0000	0.0000	0.0047	0.0136	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0097	0.0097
JpegTest	0.0003	0.0000	10.1485	0.8258	0.0075	0.0069	0.0133	0.0307	0.0296	0.0044	0.0122	0.7063	0.1386	0.1441	0.0177
Resize	8.1727	1.7802	0.3704	1.0726	0.0000	0.0000	0.0000	0.0006	0.0006	0.0000	0.0000	0.0030	0.0000	0.0000	0.0000
GaussianBlur	2.1302	1.9354	1.8912	0.1671	0.0000	0.0274	0.0274	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0017	0.0017
MedianBlur	0.8366	0.7339	0.3071	0.0982	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0008	0.0000	0.0130	0.0130
Brightness	0.0725	0.2385	0.2902	0.0639	0.0017	0.0030	0.0047	0.0028	0.0033	0.0006	0.0025	0.0014	0.0000	0.0089	0.0089
Contrast	0.0982	0.1782	0.2971	0.0335	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0011	0.0011	0.0033	0.0022
Saturation	0.0011	0.0000	0.0066	0.7113	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0075	0.0075
Hue	0.0000	0.0000	0.0066	8.3780	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0041	0.0041
Dropout	0.0017	0.0122	0.3101	5.2621	0.0058	0.0000	0.0058	0.0000	0.0000	0.0000	0.0000	0.0014	0.0000	0.0011	0.0011
SaltPepper	45.8038	26.9216	44.2502	12.3238	0.0008	0.0003	0.0011	0.0011	0.0008	0.0003	0.0000	0.0003	0.0000	0.0000	0.0000
GaussianNoise	15.3785	0.1029	21.3662	0.8540	0.0578	0.0622	0.0930	0.2257	0.2318	0.0421	0.0575	0.9652	0.3118	0.3129	0.0409
Average Common	6.0413	2.6586	6.6041	2.4837	0.0061	0.0083	0.0121	0.0217	0.0222	**0.0040**	0.0060	0.1400	0.0376	0.0422	0.0089

Table 5
Robustness test (Bit Error Rate: %) for the commonly distorted image I_{no} on the deepfake images (CelebA-Fake).

Distortion	MBRS [10]	FIN [11]	PIMoG [12]	FaceSigns [7]	SepMark [8]			ConMark-D			AdvMark [9]	ConMark-E	ConMark-ED		
					Robust	Fragile	Match	Robust	Fragile	Match			Robust	Fragile	Match
Identity	0.0000	0.0000	0.0030	0.0277	0.0108	9.2500	9.2607	0.0196	16.0519	16.0494	0.0000	0.0000	0.0274	13.0635	13.0682
JpegTest	0.0008	0.0000	10.1751	0.9749	0.0097	0.0050	0.0119	0.0335	0.0382	0.0136	0.0562	1.2081	0.2457	0.2423	0.0299
Resize	3.8086	1.0147	0.1356	0.8341	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0050	0.0025	0.0180	0.0188
GaussianBlur	1.9885	1.6278	2.4627	0.1054	0.0003	0.2415	0.2418	0.0000	0.0999	0.0999	0.0000	0.0003	0.0019	0.3367	0.3386
MedianBlur	0.4498	0.4642	0.2722	0.0805	0.0003	0.0282	0.0285	0.0044	0.8161	0.8172	0.0000	0.0006	0.0061	0.3632	0.3677
Brightness	0.0813	0.1799	0.2999	0.0575	0.0227	9.5158	9.5352	0.0678	15.4488	15.4419	0.0000	0.0044	0.1242	11.7041	11.7254
Contrast	0.0794	0.1472	0.2769	0.0373	0.0050	7.0857	7.0901	0.0398	13.7424	13.7485	0.0000	0.0008	0.0647	10.5685	10.5884
Saturation	0.0000	0.0003	0.0277	0.6438	0.0100	8.9246	8.9346	0.0227	15.5954	15.5921	0.0000	0.0000	0.0335	12.3669	12.3733
Hue	0.0000	0.0000	0.0277	7.7688	0.0149	7.7646	7.7796	0.0249	15.7902	15.7896	0.0000	0.0000	0.0459	12.9847	12.9886
Dropout	0.0003	0.0066	0.2338	5.4823	0.2202	0.8167	1.0369	0.2553	16.3902	16.3623	0.0000	0.0008	0.2545	15.4181	15.4275
SaltPepper	46.2913	26.8757	44.6577	12.3501	0.0003	0.0000	0.0003	0.0011	0.0011	0.0011	0.0003	0.0000	0.1162	0.7433	0.8186
GaussianNoise	18.1029	0.1015	22.6142	1.1968	0.0813	0.0860	1.2145	0.2379	0.2374	0.0487	0.1610	1.7138	0.5162	0.5159	0.0639
Average Common	5.9002	2.5348	6.7655	2.4633	0.0313	3.6432	3.7612	0.0589	7.8510	7.8304	**0.0181**	0.2445	0.1199	6.5271	6.4841

4.3. Robustness test

Primarily, the robust decoding should have a low BER under both the common and malicious distortions, while the fragile decoding should have a low BER under the common distortions but a high BER under the malicious ones. Notably, we use the term "Match" in Tables 4–6 to denote the BER between the robust and fragile decoded messages.

4.3.1. Common distortions

Under the common distortions presented in Table 4, the robust decoding of SepMark, AdvMark, and our ConMark-D/-E/-ED achieves an average BER below 1 %, reaching stronger robustness than the other compared models. Moreover, the fragile decoding of SepMark and our ConMark-D/-ED also gains BER reduction compared to MBRS, FIN, PIMoG, and FaceSigns. Furthermore, although the BER of the single conditioned decoder in our ConMark-D/-ED is marginally higher than that of the two separable decoders in SepMark, the robust and fragile decoded messages are more closely matched when encountering common distortions due to the coupling between robust and fragile decoding. It is evident that the robust and fragile decoding extract nearly the same message under common distortions.

However, when directly applied to deepfake-generated images, as depicted in Table 5, the robust decoding shows minor fluctuations in the BER of each distortion compared to that on real images, while the fragile decoding of SepMark and ConMark-D/-ED deteriorates. Taking a closer look, these declines occur in SimSwap- and StarGAN-generated images and can be well mitigated by incorporating few-shot generated images during training. Despite this, the robust decoding alone is enough to achieve provenance tracking and proactive detection, based on the watermark message indicating it is a deepfake image.

4.3.2. Malicious distortions

Under the malicious distortions presented in Table 6, the robust decoding of SepMark and our ConMark-D/-ED has competitive robustness compared to MBRS, FIN, PIMoG, and AdvMark. As expected, the fragile decoding of FaceSigns, SepMark, and ConMark-D/-ED achieves an average BER around 50 %, close to random guesses. However, the others are overly robust against deepfake manipulations, particularly in facial expressions and attributes. Our ConMark-D/-ED successfully holds selective fragility against malicious distortions, and it can be concluded that robust and fragile decoding extract mismatched messages under malicious distortions.

Table 6
Robustness test (Bit Error Rate: %) for the maliciously distorted image I_{no} on the real images (CelebA-HQ).

Distortion	MBRS [10]	FIN [11]	PIMoG [12]	FaceSigns [7]	SepMark [8]			ConMark-D			AdvMark [9]	ConMark-E	ConMark-ED		
					Robust	Fragile	Match	Robust	Fragile	Match			Robust	Fragile	Match
SimSwap	48.9039	39.5632	38.5239	49.9463	7.9068	45.9117	45.6713	10.3214	50.2462	50.1840	24.9394	42.1092	12.9941	47.0803	46.4982
GANimation	1.6962	2.3834	3.3148	45.4172	0.0020	43.8524	43.8527	0.0003	48.7645	48.7642	0.0000	0.0003	0.0000	46.8252	46.8252
StarGAN ($Male$)	6.1446	3.6620	20.1487	50.2617	0.0033	45.4624	45.4641	0.0006	51.3439	51.3439	0.0075	0.1156	0.0039	43.8283	43.8266
StarGAN ($Young$)	6.6722	3.6465	20.0226	50.3649	0.0030	45.5319	45.5333	0.0008	51.4925	51.4922	0.0072	0.1220	0.0039	48.0328	48.0317
StarGAN ($BlackHair$)	9.2674	4.6034	24.8169	50.2576	0.0019	45.1299	45.1296	0.0008	51.6745	51.6748	0.0116	0.1632	0.0050	46.1593	46.1599
StarGAN ($BlondHair$)	3.8869	3.3784	14.8097	50.9829	0.0022	44.6953	44.6953	0.0014	50.1325	50.1311	0.0083	0.1201	0.0019	46.2940	46.2932
StarGAN ($BrownHair$)	5.6995	3.2636	19.8859	50.7884	0.0017	45.9950	45.9939	0.0003	50.7367	50.7370	0.0086	0.0932	0.0039	43.8283	43.8266
Average Malicious	11.7530	8.6429	20.2175	49.7170	**1.1316**	45.2255	45.1915	1.4751	50.6273	50.6182	3.5689	6.1034	1.8590	46.0069	45.9231

Table 7
Detection accuracy (Real/Fake ACC ↑) of well-trained deepfake detectors tested on the host images & watermarked images.

Detector	Host Images	JPEG	MBRS [10]	FIN [11]	PIMoG [12]	FaceSigns [7]	SepMark [8]	ConMark-D	AdvMark [9]	ConMark-E	ConMark-ED
Xception [24]	98.19/20.82	84.49/32.15	98.73/16.01	98.94/15.65	96.95/19.51	98.62/10.91	98.44/15.30	98.48/15.72	99.82/99.68	99.86/99.33	99.86/99.54
EfficientNet [25]	99.11/52.73	1.133/99.89	96.71/55.91	78.08/83.50	99.19/32.54	98.83/38.60	99.47/28.26	99.22/35.34	99.96/99.82	100.0/100.0	100.0/99.96
CNND [26]	99.96/27.87	99.89/18.06	99.93/27.83	99.93/25.81	99.86/29.64	99.82/23.09	99.96/24.11	99.96/22.66	99.89/97.84	100.0/99.93	100.0/99.96
FFD [27]	97.17/59.67	99.96/0.106	95.47/57.65	97.20/59.49	94.48/47.31	99.68/0.602	94.62/59.45	95.08/60.38	98.90/95.64	99.65/99.93	99.79/99.93
PatchForensics [28]	99.40/33.43	100.0/0.000	97.80/40.51	98.19/41.08	99.26/33.50	97.45/20.47	99.15/37.50	98.83/40.08	99.96/99.82	100.0/99.96	100.0/99.96
MultiAtt [29]	85.23/20.36	91.25/13.39	88.17/15.55	88.46/14.73	91.11/12.82	75.71/28.90	87.75/18.02	85.62/19.65	99.86/99.93	99.96/99.89	99.93/99.93
RFM [30]	39.41/82.33	98.87/0.212	40.97/82.72	37.96/83.89	23.30/81.76	1.948/99.40	36.30/81.06	37.25/83.50	99.15/100.0	100.0/100.0	100.0/100.0
RECEE [31]	54.36/60.38	92.92/15.16	73.37/36.19	65.44/45.75	75.60/46.46	55.17/62.78	59.81/51.70	55.81/56.06	99.96/99.96	100.0/100.0	99.93/100.0
SBI [32]	87.57/30.06	90.93/17.17	92.78/18.73	95.40/16.08	93.84/16.75	89.59/23.37	89.13/22.45	88.74/24.54	99.58/98.83	99.82/99.96	99.86/99.96
NPR [33]	98.73/99.86	100.0/0.000	16.01/99.54	99.54/99.86	3.364/99.43	0.000/100.0	18.02/99.65	13.14/99.75	27.02/99.75	76.20/100.0	100.0/100.0
Average ACC	85.91/48.75	85.94/19.61	79.99/45.06	85.91/48.58	77.70/41.97	71.68/40.81	78.27/43.75	77.35/45.77	92.41/99.13	97.55/99.90	**99.94/99.92**

Table 8
Detection accuracy (Real/Fake ACC ↑) tested on ConMark-E watermarked images under the cross-detector settings.

Detector	Xception	EfficientNet	CNND	FFD	PatchForensics	MultiAtt	RFM	RECEE	SBI	NPR	Ensemble
Xception [24]	99.86/99.33	99.33/20.61	99.36/22.77	98.83/15.69	98.55/18.45	99.43/31.41	98.26/19.55	99.65/23.09	99.68/25.85	98.76/17.67	99.68/97.73
EfficientNet [25]	99.68/64.24	100.0/100.0	99.50/49.26	99.72/38.53	99.47/41.15	99.50/45.43	98.65/35.62	98.65/28.43	99.43/56.41	99.19/41.47	99.65/72.34
CNND [26]	99.96/31.59	99.96/27.37	100.0/99.93	99.96/25.35	99.93/28.54	99.93/26.38	99.96/28.26	99.93/31.66	99.93/29.82	99.96/26.88	99.96/30.28
FFD [27]	95.08/64.41	96.00/63.74	94.90/59.24	99.65/99.93	99.08/86.72	96.28/62.75	95.75/65.65	94.05/64.91	94.79/64.34	95.11/62.61	94.79/70.75
PatchForensics [28]	98.62/36.30	99.58/36.15	99.15/55.67	99.58/56.73	100.0/99.96	99.01/37.18	99.72/39.91	99.29/34.92	99.33/35.69	99.11/41.18	98.55/48.83
MultiAtt [29]	89.45/35.45	87.50/26.38	86.86/19.90	87.22/17.74	84.77/20.29	99.96/99.89	85.02/22.27	96.39/57.58	90.37/41.40	85.87/19.65	98.94/96.49
RFM [30]	36.76/82.47	33.53/87.46	35.38/85.69	38.31/85.66	41.18/88.39	38.46/86.15	100.0/100.0	35.87/88.21	37.61/85.06	37.22/81.23	99.54/99.89
RECEE [31]	63.03/76.24	65.40/65.55	61.30/55.81	65.26/52.80	57.05/58.29	79.64/90.65	53.65/59.67	100.0/100.0	72.27/87.11	58.43/56.37	77.02/95.18
SBI [32]	96.60/60.48	94.72/36.37	93.06/33.46	91.64/20.22	87.36/28.36	96.78/52.73	83.18/34.49	98.65/53.15	99.82/99.96	89.87/26.06	96.39/73.62
NPR [33]	12.46/99.33	22.20/99.54	16.36/99.33	51.06/99.65	22.70/99.68	21.60/99.61	25.00/99.65	13.81/99.50	34.67/99.15	76.20/100.0	44.05/96.71
Average ACC	79.15/64.98	79.82/56.32	78.59/58.11	83.12/51.23	79.01/56.98	83.06/63.22	83.92/50.51	83.63/58.15	82.79/62.48	83.97/47.31	**90.86/78.18**

4.4. Harmlessness test

For a comprehensive evaluation, we report the Real/Fake ACC on the separate real and deepfake image subsets. The higher the value, the better the detection performance.

4.4.1. Intra-detector evaluation

When the watermark network and deepfake detector are in the same camp, i.e., the detector is already known by the watermarking developer, the surrogate detector used can be consistent with the downstream detector. By observing Table 7, we note that the detector, without any tuning, achieves significant performance gains in detecting the images watermarked by AdvMark and our ConMark-E/-ED. Other compared models act similarly to image processing operations that are harmful to the detector, as they distort the host image in different ways. By contrast, AdvMark and our ConMark-E/-ED turn the watermark from harmful signals into beneficial ones, ensuring above 99 % accuracy for the detector in the intra-detector setting. This result also holds when multiple detectors are included during training (see the "Ensemble" column in Table 8). One special case is the state-of-the-art detector NPR, where we observe that although it performs excellently on unprocessed host images, once the images are distorted by watermarking, they tend to be predicted as fake; in this respect, the self-supervised training of ConMark-E surpasses the supervised fine-tuning of AdvMark.

4.4.2. Cross-detector evaluation

Table 8 shows that the images watermarked by our ConMark-E can be more easily distinguished by other unseen detectors. For example, using Xception as the surrogate detector, the detection performance is higher than that without watermarking, on average (except for NPR). We notice that ConMark-E may ruin the detection on a few unseen detectors, which is likely due to the different decision logic shared by the detectors (see Fig. 8). E.g., most detectors mainly identify distinct fake patterns in the images, while

Table 9
Ablation experiments. The locations of condition injection.

DOWN Path	UP Path	Modulated Number	PSNR	BER (Average Common/Malicious)		
				Robust	Fragile	Match
×	✓	{1, 2, 3, 4}	39.7650	0.0217/1.4751	0.0222/50.6273	0.0040/50.6182
✓	×	{1, 2, 3, 4}	38.9064	0.0080/1.3254	0.0191/54.6658	0.0143/54.6262
✓	✓	{1, 2, 3, 4}	39.4980	0.0107/1.2932	0.0106/47.3716	0.0029/47.3378
×	✓	{ 2, 3, 4}	39.7878	**0.0097**/**1.2027**	**0.0096**/50.1035	**0.0020**/50.0687
×	✓	{ 2, 3 }	39.2579	0.0203/1.5249	0.0215/47.1342	0.0067/47.0368
×	✓	{1 }	**40.4213**	0.0285/1.6691	0.0288/47.1844	0.0044/47.1098

Table 10
Ablation experiments. The approaches of condition injection.

Modulation	Real Images (CelebA-HQ)			Deepfake Images (CelebA-Fake)		
	PSNR	BER	ACC (Xception)	PSNR	BER	ACC (Xception)
FiLM [19]	42.5286	**0.1400**	99.86	39.6216	**0.2445**	**99.33**
AdaIN [38]	42.5650	0.4323	**99.93**	**39.7159**	0.7219	99.19
ModConv [39]	**44.0267**	1.8899	99.68	38.5646	1.7779	98.90

the exception RECCE seeks to discriminate what is real. Encouragingly, we can include an ensemble of detectors, such as Xception, MultiAtt, and RFM; in the cross-detector setting, we obtain an overall improved detection performance. Note that ConMark-ED is omitted since it yields similar results.

4.5. Ablation study

4.5.1. Locations of condition injection

Without loss of generality, we conduct the ablation study on the CelebA-HQ dataset using ConMark-D. We mainly study the locations for injecting conditions into the base watermark decoder. The results are presented in Table 9. Specifically, the UNet-like decoder has an UP path and a DOWN path, where we modulate the output features of "SE-Residual Block" and "Double Conv" blocks, respectively. We also numbered all the blocks (both modulated and non-modulated) from left to right for each path. The results indicate that it is relevant to inject conditions into either the UP path or the DOWN path, with the former yielding a closer match between the robust and fragile decoded messages. Injecting conditions into both paths leads to slightly better performance, but it also doubles the complexity. In addition, we found that the condition injection remains dependable despite variations in both the position and the number of feature modulations. We opt for the UP path for simplicity and generality, although the best results may not be obtained through a full path.

4.5.2. Approaches of condition injection

To demonstrate the adaptability of the condition injection, we integrated other feature modulation approaches, such as AdaIN [38] and Modulated Convolution (ModConv) [39], which were originally designed for image synthesis, into our conditional watermarking framework. Notably, AdaIN and ModConv replace the instance normalization and convolutional block within each "SE-Residual Block" of the conditioned watermark encoder in ConMark-E, respectively. The results in Table 10 show that our framework benefits from all the feature modulation approaches. In particular, all of them successfully adjust the features of watermark encoding according to the input condition, as the watermarked real and deepfake images are correctly classified by the passive detector. It is unsurprising that AdaIN and ModConv present higher visual metrics, since in principle they focus on high-fidelity image synthesis. However, in the task of watermark extraction, FiLM [19] outperforms AdaIN and ModConv by a clear margin in terms of average BER (under common distortions). Considering the remarkable visual quality, robustness, and harmlessness, we hence opt for FiLM as the approach for condition injection.

4.6. Complexity analysis

The condition network we use is extremely lightweight, consisting of only a few FC layers. The additional parameters come from the condition network, and the total introduced parameter count is calculated as $2 \times 2 \times (128 + 64 + 32 + 16) = 960$. As a result, the additional computational overhead on the entire model is nearly negligible. Compared to the two separable decoders in SepMark, the single conditioned decoder in our ConMark-D is more efficient in terms of parameters and memory usage. Moreover, AdvMark requires both the real and deepfake images for supervised fine-tuning, while our ConMark-E is trained in a self-supervised manner, requiring only the real images and no deepfake images. Through condition injection, our conditional watermarking framework provides a fundamentally more efficient, scalable, and elegant solution than the common approach of training multiple separate networks.

Table 11 presents the detailed efficiency evaluations, including the number of parameters, floating-point operations (FLOPs), running time, and memory usage. Note that the watermark encoding and decoding statistics are counted separately and calculated once for comparison. The results confirm that the condition network introduces almost no additional computational cost. Except

Table 11
Efficiency evaluations based on the number of parameters (M), FLOPs (G), running time (ms), and memory usage (MB).

Model	Watermark encoder				Watermark decoder			
	Parameter	FLOPs	Time	Memory	Parameter	FLOPs	Time	Memory
MBRS [10]	0.68	33.49	5.43	2040.37	20.32	27.09	5.91	980.74
FIN [11]	0.81	**7.12**	8.22	544.49	**0.81**	7.12	8.19	544.49
PIMoG [12]	**0.59**	7.45	**3.70**	329.80	0.92	30.10	**5.32**	917.80
FaceSigns [7]	10.98	N/A	N/A	N/A	7.89	N/A	N/A	N/A
SepMark [8]	10.96	7.71	7.88	**298.11**	$4.42_{\times 2}$	**4.17**	6.44	$181.27_{\times 2}$
AdvMark [9]	10.96	7.71	7.92	**298.11**	4.42	**4.17**	6.51	**181.26**
ConMark-D	10.96	7.71	7.88	**298.11**	4.42	**4.17**	6.65	181.27
ConMark-E	10.96	7.71	8.04	298.12	4.42	**4.17**	6.45	**181.26**
ConMark-ED	10.96	7.71	8.04	298.12	4.42	**4.17**	6.69	181.27

Table 12
Additional experiments: Expanding to multi-class classification.

Condition	PSNR	BER	ACC (DNA)				
			Real	ProGAN	MMDGAN	SNGAN	InfoMaxGAN
[1, 0, 0, 0, 0]	39.8093	0.0073	**98.58**	0.000	1.381	0.000	0.035
[0, 1, 0, 0, 0]	39.7669	**0.0038**	0.000	**99.08**	0.921	0.000	0.000
[0, 0, 1, 0, 0]	**40.2766**	0.0066	0.000	0.035	**99.96**	0.000	0.000
[0, 0, 0, 1, 0]	40.1224	0.0109	0.000	0.106	24.47	**75.42**	0.000
[0, 0, 0, 0, 1]	40.1456	0.0123	0.000	0.000	7.330	0.000	**92.67**

for SepMark's two decoders, which double the parameters and memory, ConMark-D/-E/-ED perform comparably to SepMark and AdvMark across all four metrics due to their similar watermarking network structure. Despite having a larger number of trainable parameters, the watermark encoder results in fewer FLOPs than MBRS but slightly more than FIN and PIMoG. For the watermark decoder, it has fewer trainable parameters than MBRS and the fewest FLOPs among the models, demonstrating the high efficacy. Moreover, both the watermark encoder and decoder excel in memory usage. Regarding the running time, the watermark encoding and decoding take about 8 ms and 6.5 ms per image, respectively, making them suitable for practical applications.

4.7. Extending to other applications

To validate the scalability of the framework, we expand the conditioned watermark encoder's target condition from binary classification (Real/Fake) to multi-class classification. The deepfake attribution model DNA [40] is used in the experiments. Specifically, the dimension of the input condition is increased to five, corresponding to different attribution results, namely Real, ProGAN [23], MMDGAN [41], SNGAN [42], and InfoMaxGAN [43]. It should be noted that the watermarking network was trained on the real images (CelebA-HQ) while tested exclusively on the deepfake images (CelebA-Fake). As Table 12 demonstrates, the fine-grained attribution results of the watermarked images can be responsibly controlled by changing the input condition, which is helpful for regulating the use of deepfake images via harmless watermarking. Inspired by the success of the conditioned encoder, it is possible to implement tamper localization by increasing the target condition dimension of the conditioned watermark decoder. If the input condition is targeted for tamper localization, the conditioned decoder would output a fakeness prediction map to reflect the location and strength of the tampered region. However, it can be expected that, compared to the image template used for localization [44], the bit sequence would require the tamper mask to be involved in the training to achieve locality and fragility.

5. Conclusion and future work

In this paper, we propose a conditional watermarking framework, dubbed ConMark, which brings a highly efficient, scalable, and elegant paradigm to the implementation of multipurpose watermarking. At its core is a simple yet effective condition network that generates the condition vectors to modulate the intermediate features of the base watermarking network. This additional condition network transforms the unconditioned watermark decoder/encoder into a conditioned decoder/encoder, enabling the features of watermark decoding and encoding to be efficiently adjusted by changing the input condition. Depending on the conditioned component, three variants are implemented for versatile and harmless deepfake proactive forensics, extending beyond the single function of provenance tracking. Regarding the authenticity of real images, the proactive detection is achieved by extracting the embedded watermark at different levels of robustness via a single conditioned decoder. Regarding the regulation of deepfake images, the detectability enhancement is achieved by controlling the detection results of the watermarked image by other passive detectors via a single conditioned encoder. Moreover, to the best of our knowledge, this is the first work to simultaneously accomplish provenance tracking, proactive detection, and detectability enhancement, where both the encoder and decoder are conditioned.

Albeit conceptually novel, several future directions are still worth studying. First, the selective fragility is more challenging than the undoubted robustness. This means the watermark can still be reliably extracted by the robust decoding, whereas the fragile decoding may fail to distinguish deepfake manipulations that were never seen during network training. Second, the harmlessness

performance in the cross-detector setting needs to be explored further. Although it achieves nearly perfect performance with specific passive detectors, it is not yet publicly detectable or verifiable by arbitrary detectors. Lastly, the conditioned network is currently based on image watermarking, with the need to expand to video and multi-modal watermarking. Future work could explore incorporating visual-audio cues to help create a more trustworthy information ecology.

CRediT authorship contribution statement

Xiaoshuai Wu: Writing -– review & editing, Writing -– original draft, Software, Methodology, Conceptualization. **Xin Liao:** Writing – review & editing, Supervision, Investigation. **Jie Zhang:** Writing – review & editing, Supervision, Investigation. **Mingyue Chen:** Writing – review & editing, Investigation. **Yufeng Wu:** Writing -– review & editing, Investigation. **Jinlin Guo:** Writing – review & editing, Investigation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work is supported by National Key R&D Program of China (Grant Nos. 2024YFF0618800, 2022YFB3103500), National Natural Science Foundation of China (Grant No. U22A2030), Hunan Provincial Funds for Distinguished Young Scholars (Grant No. 2024JJ2025), and the Hunan Provincial Key Research and Development Program (Grant Nos. 2024AQ2027, 2025AQ2022).

Data availability

Data will be made available on request.

References

[1] F. Juefei-Xu, R. Wang, Y. Huang, Q. Guo, L. Ma, Y. Liu, Countering malicious deepfakes: survey, battleground, and horizon, Int. J. Comput. Vis. 130 (7) (2022) 1678–1734.
[2] Z. Xia, T. Qiao, M. Xu, N. Zheng, S. Xie, Towards Deepfake video forensics based on facial textural disparities in multi-color channels, Inf. Sci. 607 (2022) 654–669.
[3] J. Zhu, R. Kaplan, J. Johnson, L. Fei-Fei, Hidden: hiding data with deep networks, in: Proceedings of the European Conference on Computer Vision, 2018, pp. 657–672.
[4] R. Wang, F. Juefei-Xu, M. Luo, Y. Liu, L. Wang, Faketagger: robust safeguards against deepfake dissemination via provenance tracking, in: Proceedings of the ACM International Conference on Multimedia, 2021, pp. 3546–3555.
[5] Z. He, W. Wang, W. Guan, J. Dong, T. Tan, Defeating Deepfakes via adversarial visual reconstruction, in: Proceedings of the ACM International Conference on Multimedia, 2022, pp. 2464–2472.
[6] Y. Dai, J. Fei, F. Huang, Idguard: robust, general, identity-centric POI proactive defense against face editing abuse, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 11934–11943.
[7] P. Neekhara, S. Hussain, X. Zhang, K. Huang, J. McAuley, F. Koushanfar, Facesigns: semi-fragile watermarks for media authentication, ACM Trans. Multimed. Comput. Commun. Appl. 20 (11) (2024) 1–21.
[8] X. Wu, X. Liao, B. Ou, Sepmark: deep separable watermarking for unified source tracing and deepfake detection, in: Proceedings of the ACM International Conference on Multimedia, 2023, pp. 1190–1201.
[9] X. Wu, X. Liao, B. Ou, Y. Liu, Z. Qin, Are watermarks bugs for deepfake detectors? Rethinking proactive forensics, in: Proceedings of the International Joint Conference on Artificial Intelligence, 2024, pp. 6089–6097.
[10] Z. Jia, H. Fang, W. Zhang, Mbrs: enhancing robustness of dnn-based watermarking by mini-batch of real and simulated JPEG compression, in: Proceedings of the ACM International Conference on Multimedia, 2021, pp. 41–49.
[11] H. Fang, Y. Qiu, K. Chen, J. Zhang, W. Zhang, E.-C. Chang, Flow-based robust watermarking with invertible noise layer for black-box distortions, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, 2023, pp. 5054–5061.
[12] H. Fang, Z. Jia, Z. Ma, E.-C. Chang, W. Zhang, Pimog: an effective screen-shooting noise-layer simulation for deep-learning-based watermarking network, in: Proceedings of the ACM International Conference on Multimedia, 2022, pp. 2267–2275.
[13] U. Ojha, Y. Li, Y.J. Lee, Towards universal fake image detectors that generalize across generative models, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 24480–24489.
[14] R. Ridwan, M.N. Kabir, Robust color image watermarking using dual embedding via Schur decomposition, Int. J. Adv. Comput. Inform. 1 (1) (2025) 39–47.
[15] H. Wang, X. Tian, Y. Xia, A robust dual color image blind watermarking scheme in the frequency domain, Multimed. Tools Appl. 83 (13) (2024) 38711–38735.
[16] N.C. Onn, S.N. Avivah, M.M. Alam, A. Ahmad, Secure and effective image integrity and copyright protection using two-layer authentication with integer wavelet transform, Int. J. Adv. Comput. Inform. 1 (1) (2025) 13–27.
[17] R. Bouarroudj, F.Z. Bellala, F. Souami, High capacity and reversible fragile watermarking method for medical image authentication and patient data hiding, J. Med. Syst. 48 (1) (2024) 98.
[18] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7132–7141.
[19] E. Perez, F. Strub, H. De Vries, V. Dumoulin, A. Courville, Film: visual reasoning with a general conditioning layer, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, 2018, pp. 3942–3951.
[20] P. Isola, J.-Y. Zhu, T. Zhou, A.A. Efros, Image-to-image translation with conditional adversarial networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1125–1134.
[21] A. Jolicoeur-Martineau, The relativistic discriminator: a key element missing from standard GAN, in: Proceedings of the International Conference on Learning Representations, 2019, pp. 1–10.
[22] N. Huang, A. Gokaslan, V. Kuleshov, J. Tompkin, The GAN is dead; long live the GAN! a modern GAN baseline, in: Proceedings of the Advances in Neural Information Processing Systems, vol. 37, 2024, pp. 44177–44215.
[23] T. Karras, T. Aila, S. Laine, J. Lehtinen, Progressive growing of gans for improved quality, stability, and variation, in: Proceedings of the International Conference on Learning Representations, 2018, pp. 1–12.

[24] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, M. Nießner, Faceforensics++: learning to detect manipulated facial images, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 1–11.
[25] D. Li, W. Wang, H. Fan, J. Dong, Exploring adversarial fake images on face manifold, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 5789–5798.
[26] S.-Y. Wang, O. Wang, R. Zhang, A. Owens, A.A. Efros, Cnn-generated images are surprisingly easy to spot…for now, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 8695–8704.
[27] H. Dang, F. Liu, J. Stehouwer, X. Liu, A.K. Jain, On the detection of digital face manipulation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 5781–5790.
[28] L. Chai, D. Bau, S.-N. Lim, P. Isola, What makes fake images detectable? Understanding properties that generalize, in: Proceedings of the European Conference on Computer Vision, 2020, pp. 103–120.
[29] H. Zhao, W. Zhou, D. Chen, T. Wei, W. Zhang, N. Yu, Multi-attentional deepfake detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 2185–2194.
[30] C. Wang, W. Deng, Representative forgery mining for fake face detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 14923–14932.
[31] J. Cao, C. Ma, T. Yao, S. Chen, S. Ding, X. Yang, End-to-end reconstruction-classification learning for face forgery detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 4113–4122.
[32] K. Shiohara, T. Yamasaki, Detecting deepfakes with self-blended images, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 18720–18729.
[33] C. Tan, Y. Zhao, S. Wei, G. Gu, P. Liu, Y. Wei, Rethinking the up-sampling operations in Cnn-based generative network for generalizable deepfake detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 28130–28139.
[34] R. Zhang, P. Isola, A.A. Efros, E. Shechtman, O. Wang, The unreasonable effectiveness of deep features as a perceptual metric, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 586–595.
[35] R. Chen, X. Chen, B. Ni, Y. Ge, Simswap: an efficient framework for high fidelity face swapping, in: Proceedings of the ACM International Conference on Multimedia, 2020, pp. 2003–2011.
[36] A. Pumarola, A. Agudo, A.M. Martinez, A. Sanfeliu, F. Moreno-Noguer, Ganimation: anatomically-aware facial animation from a single image, in: Proceedings of the European Conference on Computer Vision, 2018, pp. 818–833.
[37] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, J. Choo, Stargan: unified generative adversarial networks for multi-domain image-to-image translation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 8789–8797.
[38] X. Huang, S. Belongie, Arbitrary style transfer in real-time with adaptive instance normalization, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1501–1510.
[39] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, T. Aila, Analyzing and improving the image quality of Stylegan, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 8110–8119.
[40] T. Yang, Z. Huang, J. Cao, L. Li, X. Li, Deepfake network architecture attribution, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, 2022, pp. 4662–4670.
[41] M. Bińkowski, D.J. Sutherland, M. Arbel, A. Gretton, Demystifying Mmd Gans, in: Proceedings of the International Conference on Learning Representations, 2018, pp. 1–15.
[42] T. Miyato, T. Kataoka, M. Koyama, Y. Yoshida, Spectral normalization for generative adversarial networks, in: Proceedings of the International Conference on Learning Representations, 2018, pp. 1–12.
[43] K.S. Lee, N.-T. Tran, N.-M. Cheung, Infomax-Gan: improved adversarial image generation via information maximization and contrastive learning, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2021, pp. 3942–3952.
[44] X. Zhang, R. Li, J. Yu, Y. Xu, W. Li, J. Zhang, Editguard: versatile image watermarking for tamper localization and copyright protection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 11964–11974.