Encoded Feature Enhancement in Watermarking Network for Distortion in Real Scenes

Han Fang, Zhaoyang Jia, Hang Zhou, Zehua Ma and Weiming Zhang

***Abstract*—Deep-learning based watermarking framework has been extensively studied recently. The main structure of such framework is an encoder, a noise layer and a decoder. By training with different distortion sets in the noise layer, the whole network can realize different robustness. However, such framework has a huge drawback that the noise layer must be differentiable, otherwise it cannot be trained end-to-end. But for practical use, much distortions are non-differentiable, so such framework cannot be applied. To address such limitations, this paper propose a triple-phase watermarking framework for practical distortions. The proposed framework consists of three phases including a noise-free initial phase, a mask-guided frequency enhancement phase and an adversarial-training phase. Phase 1 aims to initialize an encoder to embed watermark with high visual quality and a decoder to extract the watermark. In order to generate high quality watermarked image, we design the just noticeable difference (JND)-mask image loss in phase 1 to guide the encoder. At phase 2, based on the investigation of the encoded features and distortions, we propose a mask-guided frequency enhancement algorithm to enhance the encoded feature which ensures the survival of such features after distortion, so that there will be enough features to be learned in phase 3. And phase 3 aims to train a stronger decoder to extract the watermark from the image after practical distortions. The combination of these 3 phases can well handle the non-differentiable problems and make the whole network trainable. Various experiments indicate the superior performance of the proposed scheme in the view of traditional differentiable image processing distortion robustness and practical non-differentiable distortion robustness.**

***Index terms*—**Deep-learning Watermarking, practical distortions triple-phase, mask-guided frequency enhancement.

I. Introduction

As an important branch of data hiding technology [1]–[4], digital watermarking [5]–[9] has been widely studied. For robust watermarking scheme, the most important property is robustness, which refers to the extraction accuracy of the watermark against different distortions. To acquire the strong robustness, traditional watermarking schemes often embed the watermark into robust coefficients in spatial domain [10], [11] and frequency domain [1], [12].

In the recent years, inspired by the success of deep learning in many tasks, a few deep neural network (DNN) based end-to-end watermarking architectures [13]–[16] were proposed. The DNN based architecture consists of three main parts: encoder, noise layer and decoder. The encoder tries to embed the watermark into the host image, the noise layer aims to add the distortion to the watermarked image, the decoder extracts the watermark from the watermarked image and the distorted image. Since the whole architecture is trained in an end-to-end way, the key to be trainable is that the noise layer must be differentiable. Only in this way the gradient can be propagated back in the whole network. However, when facing the non-differentiable distortions, such architectures cannot be applied. And nowadays, more and more practical distortions are presented in a non-differentiable way, such as style transferring, screen-shooting and so on.

A few schemes designed for non-differetiable distortions had been proposed recently. As for JPEG compression distortion, Zhu *et.al.* [13] designed a noise layer to approximate the JPEG compression. Tancik *et.al.* [14] proposed to use several differentiable operations to simulated the print-shooting distortion and add them to the noise layer. Meanwhile, to resist the screen-shooting distortions, Wengrowski *et.al.* [16] added a camera-display transfer function (CDTF) network in the noise layer. However, these three solutions have the same drawbacks that the noise simulation can only ensure the similarity of the forward process, but the gradient propagated back by simulated process is not necessarily the same as the actual distortions. So when applying such network into practice, the performance will be worse than the simulated results.

Instead of simulating the noise layer, Liu *et.al.* [15] proposed a two-stage separable watermarking architecture. In stage II, the decoder is separably fine-tuned by distorted data in order to obtained the target robustness. However, only enhancing the decoder is not enough when facing serious distortions, because once the watermark signal is seriously damaged by the distortion, the decoder cannot obtain enough watermark features for decoding even with adversarial training.

Hence, in order to realize the robustness against non-differentiable distortions, we propose a novel triple-phase watermarking framework. The whole architecture consist of three main phases. At phase-1, a noise-free end-to-end encoder and decoder is trained, which aims to generate a cooperative encoder and decoder to embed and extract the watermark. And for better visual quality, we proposed a just noticeable difference (JND)-mask-guided image loss to cope with traditional mse-loss, which effectively guides the encoder training. But the encoded features is not strong enough against various distortions. So we investigate the feature's changing before and after distortions and propose a mask-guided frequency enhancement algorithm at phase-2 to produce more robust

Han Fang is with School of Computing, National University of Singapore, Singapore. Zhaoyang Jia, Hang Zhou, Zehua Ma and Weiming Zhang are all with CAS Key Laboratory of Electromagnetic Space Information, University of Science and Technology of China, Hefei, 230026, China. (e-mail: fanghan@nus.edu.sg, zhangwm@ustc.edu.cn)

This work was supported in part by the Natural Science Foundation of China under Grant 62072421, 62002334, 62121002 and U20B2047, Anhui Science Foundation of China under Grant 2008085QF296, and by Exploration Fund Project of University of Science and Technology of China under Grant YD3480002001.

features based on phase-1, so that the watermark signal can be preserved after distortions. At phase-3, a set of images are embedded and enhanced by phase-1 and phase-2 and further attacked by the target non-differentiable distortions to generate the adversarial training dataset. Based on that, the decoder is further fine-tuned to extract the watermark from the distorted images.

In summary, the contributions of the proposed network are as follows:

1). We investigate the biggest drawbacks of the existing deep-learning watermarking architecture and propose a traditional-deep-learning combination-based triple-phase watermarking framework, with which, the adaptation ability of neural networks and the feature enhancement ability of traditional frequency enhancement can be effectively combined. Therefore, the robustness against practical distortion which is usually non-differentiable can be well guaranteed.

2). Based on the analysis of human visual system, we propose a just noticeable difference (JND)-mask-guided image loss. With the constrain of such loss, the algorithm can generate higher quality watermarked images in the same epoch iteration compared with the traditional MSE-Loss.

3). According to the investigation of the encoded features as well as the distortions, we designed a mask-guided frequency enhancement algorithm to enhance the encoder, based on which, the encoded feature that carries watermark signal can be better preserved after non-differentiable distortions. So that the decoder will get enough feature to be trained in phase-3.

4). Various experimental results indicate the outstanding performance against not only the traditional image processing distortions but also various black-box non-differentiable distortions compared with the state-of-the-art algorithms.

The remaining of this paper are organized as follows. In Section II, we mainly discuss the related work of the proposed scheme. Section III introduces the architecture of the proposed watermarking scheme. The corresponding experimental results are indicated in Section IV and Section V. Section VII concludes the paper.

II. Related Work

A. Traditional watermarking scheme

Traditional watermarking schemes are extensively studied since 1994, Schyndel *et.al.* [17] first defined the word "watermark", which marked the birth of digital watermarking technology. Then, many spatial domain based watermarking and frequency domain based watermarking schemes are proposed in the last few years. Spatial domain based watermarking schemes mainly modify the pixel value or the pixel distribution to embed the watermark. Among them, the histogram-based embedding [18], [19] and the template based embedding [10], [11], [20] are the most common algorithms.

For the frequency domain based schemes, the most commonly used domains are DCT domain [8], DFT domain [1], [12] and DWT domain [21], [22]. Since the modification of the frequency coefficients can better balance the visual quality and the robustness, the frequency domain based schemes are much more widely used than spatial domain based schemes. However, traditional watermarking scheme only use the handcrafted features for embedding and extracting, though such features are robust to certain distortions, they do not make full use of the characteristics of the host image.

B. Deep learning based watermarking scheme

Recently, many deep learning based watermarking algorithms [13]–[16] have been proposed. Specifically, Zhu *et.al.* [13] proposed an auto-encoder like data hiding network. By jointly training the encoder, decoder as well as the differentiable noise layer, the resilience against image processing distortions can be achieved. Ding *et. al.* [23] proposed an up-sampler and down-sampler based architecture to separately convert the image and watermark to hidden layer then further embed the watermark. Chen *et. al.* [24] simulated the JPEG compression distortion with a DCT transformation layer and a 3D noise-mask quantization operation, with which the JPEG robustness can be improved. Mellimi *et. al.* [25] proposed a DNN-based extraction network combined with a traditional DWT-based embedding scheme. Since the embedding feature is handcrafted designed, it does not make full use of the strength of DNN in embedding part. Ahmadi *et. al.* [26] proposed a block-based end-to-end watermark framework to embed the watermark. But such framework can only adapt to differentiable distortions.

Tancik *et.al.* [14] simulated the distortions of print-shooting process with several differentiable operations such as color reconstruction and Gaussian noising then further added it into the noise layer. Recently, Wengrowski *et.al.* [16] produce an image dataset of screen-to-camera image pair, then propose a camera-display transfer function (CDTF) network to simulate the camera shooting process. By replacing the noise layer with the CDTF network, the proposed network can realize the screen-to-camera resilience.

The aforementioned algorithm are based on one-stage end-to-end architecture, which aims to use the differentiable operation to replace the non-differentiable distortions. However, the performance of the network trained by simulated noise often degrades when facing real noise because of the imperfect simulation. Therefore, Liu *et.al.* [15] proposed a two-stage separable watermarking architecture. By adversarial training the decoder in stage II, the robustness with some distortions is greatly improved. However, since they do not enhance the encoder but only fine-tune the decoder, the encoded feature may be erased after several strong distortions, so even after fine-tuning, the decoder cannot effectively extracted the watermark.

III. Proposed Framework

A. Motivations

The biggest drawback of DNN-based watermarking framework is that the noise layer must be differentiable, otherwise the network cannot be trained end-to-end.

The reason is that the loss needs to be propagated back to the encoder through the noise layer. However, we found that encoder and decoder can be trained separately. After initializing the encoder by a noise-free training process, we

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3149641, IEEE Transactions on Multimedia

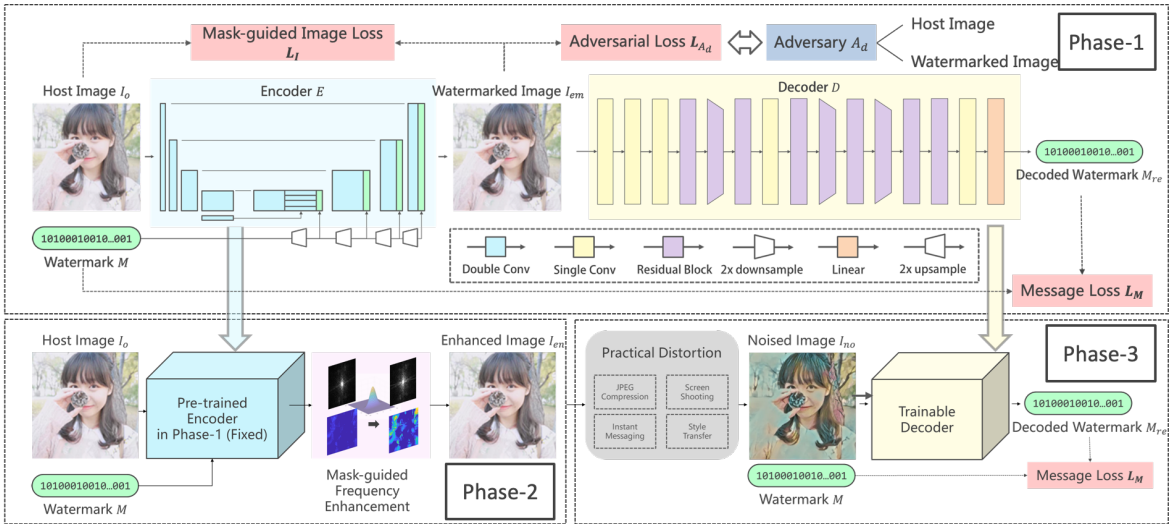

Fig. 1: The framework of the whole system. It consist of three main phases. In Phase-1, the encoder and the decoder are trained end-to-end without noise layer, the encoder tries to encoded the message through a u-net like architecture into the host image, where the decoder aims to recognize the embedded feature and extracted the message. In Phase-2, the mask-guided frequency enhancement algorithm is used for enhancing the encoded feature that is generated by the pre-trained encoder in Phase-1. At Phase-3, the practical distortions are applied to a series of enhanced images to generate the training dataset for decoder training, with which, the decoder is trained and the loss propagates back only through the decoder, aiming to extract the feature from the distorted image.

can embed the watermark with the encoded features. And for decoder, as long as the encoded feature can be preserved after the distortion, the decoder can learn the corresponding features to realize the decoding process.

To achieve this goal, we design a triple-phase watermarking framework. In phase-1, we first initialize one encoder and decoder. Then in phase-2, we should enhance the encoded features for strong robustness via the proposed mask-guided frequency enhancement algorithm. After that, in phase-3, we further train the decoder based on the real distorted watermarked image dataset, which is generated with phase-1 and phase-2 and the corresponding practical distortions. In this way, even the distortion is non-differentiable, the decoder can still extract the watermark successfully.

B. Framework

The framework of the proposed scheme is shown in Fig. 1, which consists of three phases and six main parts: (1) the message of length L, which will be reshaped to the same size of the hidden layer and further concatenated to the hidden layer of the encoder; (2) the encoder E with parameters θ_E, which will be fed with the host image $I_o \in \mathbb{R}^{C\times H\times W}$ and the reshaped message $M \in \{0,1\}$ to generate the embedded image $I_{em} \in \mathbb{R}^{C\times H\times W}$; (3) the decoder D with parameters θ_D, which receives I_{em}, and recovers the encoded message $M_{re} \in \{0,1\}$. (4) the adversary A_d with parameters θ_{A_d}, which tries to judge whether the I_{em} is an embedded image or not; (5) the mask-guided frequency enhancement algorithm, which is applied to enhance the embedded feature to generate the enhanced image I_{en}; (6) the practical distortion part, which tries to add the distortions on a series of I_{en}s to produce noised images I_{no}s for further adversarial training.

In practical use, the host image is fed into the pre-trained encoder and then enhanced by mask-guided frequency enhancement operation to generate the watermarked image. And the decoding procedure is carried out by the decoder after phase-3.

1) Watermark Reshaping: The message of length L is first filled with '0' (if necessary), then it is reshaped and up-sampled to the size of the encoder hidden layer respectively. As can be seen in Fig. 1, in the proposed framework, the message should be up-sampled 4 times.

2) Encoder: The encoder we adopt in the proposed scheme is U-Net [27] like architecture. Specifically, three "double-conv" (2∗conv-bn-relu-maxpool) blocks first progressively downsample I_o to $H/8 \times W/8$ feature maps, then a global $H/32\times W/32$ feature block is obtained by using an extra convolutional layer. Then the global feature block as well as the reshaped watermark layer is concatenated to the $H/8 \times W/8$ feature maps. Finally, several "up-double-conv" (2∗up-conv-bn-relu-maxpool) blocks upsample the $H/8 \times W/8$ feature maps back to the original size to get the encoded image I_{em} where the watermark layer with size $H/4\times W/4$, $H/2\times W/2$ and $H \times W$ are concatenated to the upsampled hidden layer respectively.

To better constrain the image quality of the encoded image, we propose an JND-mask-based image loss to give different weights to different pixels. Because the eye's sensitivity to different texture and color are varies. The weight mask we utilize is JND [28] which is proved to successfully represent the characteristics of the human visual system.

The corresponding equations are shown as follows:

$$\mathrm{JND}(x,y) = \lambda_1 \times f_1(\mathrm{bg}(x,y), mg(x,y)) + \lambda_2 \times f_2(bg(x,y)) \tag{1}$$

where

$$f_1(x,y) = 0.001xy + 0.115y - 0.1x + \lambda \tag{2}$$

$$f_2(x) = \begin{cases} T_0 \times \left(1 - \sqrt{\frac{x}{127}}\right) + 3 & \text{if } x \le 127 \\ \gamma \times (x - 127) + 3 & \text{otherwise} \end{cases} \tag{3}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3149641, IEEE Transactions on Multimedia

where f_1 is the spatial masking component, f_2 determines the visibility threshold according to the background luminance. $bg(x, y)$ and $mg(x, y)$ are average background luminance and maximum weighted average of luminance differences around the pixel at (x, y), respectively. And the bg and mg is determined by:

$$bg = I \otimes B \tag{4}$$

and

$$\begin{aligned} mg &= \max_{i=1,2,3,4} |g_k| \\ g_k &= I \otimes G_k \end{aligned} \tag{5}$$

where $\otimes$ indicates the convolution operation, and

$$G_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 3 & 8 & 3 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & -3 & -8 & -3 & -1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} G_2 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 8 & 3 & 8 & 0 \\ 1 & 3 & 0 & -3 & -1 \\ 0 & 0 & -3 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix} \tag{6}$$

$$B = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 2 & 2 & 1 \\ 1 & 2 & 0 & 2 & 1 \\ 1 & 2 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{7}$$

And $G_3 = G_2^T, G_4 = G_1^T$. In this paper, T_0, γ, λ, λ_1 and λ_2 is set as $17, 3/128, 1/2, 2$ and 3 respectively, which is same as the original settings in [28]. For more specific information about the equations, please refer to [28]. The JND image of the host color image is calculated channel by channel and further scaled between 0 and 1.

Besides, in human visual system (HVS), the sensitivity of human eyes to blue components is much lower than that of green and red components [29], we should constrain the modification to concentrate more on the blue components. So in loss function, we give different channel with different weights. In this paper, we use a 3-channel mask M_a with the same size of I_o to realize such constrain,

$$M_a[r, g, b] = [\delta_r \times JND_r, \delta_g \times JND_g, \delta_b \times JND_b] \tag{8}$$

where $\delta_r, \delta_g, \delta_b$ indicate the weight of each components and JND_r, JND_g, JND_b represent the 3 channel of the JND image. In this paper, $\delta_r, \delta_g, \delta_b$ is set as $5, 10, 1$. Then the encoder network is trained in a fully supervised way to make I_o and I_{em} more similar, the object of the encoder is to minimize the mask-guided Mean Squared Error (MSE) distance between I_o and I_{em} by updating θ_E:

$$\mathcal{L}_E = M_a * MSE(I_o, I_{em}) = M_a * MSE(I_o, E(\theta_E, I_o, M)) \tag{9}$$

3) Decoder: The structure of the proposed decoder is shown as Fig. 1. The decoder D aims to recover the encoded message from I_{em}. We apply Res-Net [30] like network which is proved to be useful in classification tasks for the decoding process. Specifically, five "single-conv" (conv-bn-relu-maxpool) blocks, five "residual" blocks and one "linear" block is applied to compose the decoder, where the down-sample operation is carried out in the "residual" blocks.

The objective of D is to minimize the difference between M_{re} and the original watermark M by updating θ_D:

$$L_D = MSE(M, M_{re}) = MSE(M, D(\theta_D, I_{no})) \tag{10}$$

4) Adversary: For better image quality of the encoded image, we utilize the adversarial network to judge whether the encoded image is similar enough to the host image. The encoded network is trying to generate the high quality I_{em} to mistake the judgement of the adversarial network. So $\mathcal{L}_{A_d}$ loss is used to improve the image quality of I_{em} by updating θ_{A_d}:

$$\mathcal{L}_{A_d} = log(1 - A_d(\theta_{A_d}, I_{em})) = log(1 - A_d(\theta_{A_d}, E(\theta_E, I_o, M))) \tag{11}$$

Besides, θ_{A_d} should also give a correct binary classification results between I_{em} and I_o. So such goal is realized by updating θ_{A_d} with the following loss function:

$$\mathcal{L}_{A_d} = log(1 - A_d(\theta_{A_d}, I_o)) + log(A_d(\theta_{A_d}, E(I_o, M))) \tag{12}$$

In this paper, we use the PatchGAN [31] as A_d by default. And after end-to-end training with E, D and A_d, the initialized encoder and decoder is obtained.

5) Mask-guided Frequency Enhancement: After initializing the encoder and the decoder, the watermark can be embedded into the host image. We believe the residual image (RI) carries the encoded feature and represents the watermark signal. RI is defined by

$$RI = I_{em} - I_o \tag{13}$$

In order to successfully extract watermark signal, the encoded feature should be well preserved after distortion. But the initialized encoder may not able to create strong encoded features for distortion. So in phase-2, we have to enhance the encoded feature to make it more robust to various distortions.

Besides, since the encoder in phase-1 is trained with the visual mask loss, the encoded feature is trained to be adaptive to host image. So in order to realize better visual quality, the enhancement cannot greatly change the characteristics of the encoded features.

To achieve this goal, we proposed a mask-guided frequency enhancement algorithm. Specifically, after obtaining the watermarked image in phase-1, we first get the RI of it, and centralize the RI, as shown in Eq. (14).

$$RI_c = \frac{RI - \mu_{RI}}{\sigma_{RI}} \tag{14}$$

where μ_{RI}, σ_{RI} indicates the mean and standard deviation of RI respectively. After that, we apply 2-D DFT (Discrete Fourier transform) on blue channel of RI_c, as shown in Eq. (15).

$$F_{RI}^{blue}(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} RI_c^{blue}(x, y) e^{-j2\pi\left(\frac{ux}{M} + \frac{vy}{N}\right)} \tag{15}$$

where (x, y) and (u, v) indicates the coordinates of pixel and Fourier coefficient respectively. M, N indicate the width and height of RI_c. F_{RI} represent the Fourier coefficients matrix of

RI_c^{blue}. After 2-D DFT, we perform a weighted enhancement process on F_{RI}^{blue}, as shown in Eq. (16).

$$Fw_{RI}^{blue} = F_{RI}^{blue} \times W_F \tag{16}$$

where W_F indicates the weight matrix, which can be formulated by Eq. (17).

$$\begin{aligned} g(x,y) &= \frac{1}{2\pi\sigma^2} e^{-\frac{\left(x^2+y^2\right)}{2\sigma^2}} \\ W_F &= \frac{g - \min(g)}{\max(g) - \min(g)} \times \beta \end{aligned} \tag{17}$$

β is the enhance factor that adjust the visual quality and robustness performance. In this paper, σ is set as 20, and the size of W_F is same as RI. Then we applied the 2-D inverse DFT on Fw_{RI}^{blue} to get the enhance blue channel of RI, noted as RI_e^{blue}, as shown in Eq. (18).

$$RI_e^{blue}(x,y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} Fw_{RI}^{blue}(u,v) e^{j2\pi\left(\frac{ux}{M} + \frac{vy}{N}\right)} \tag{18}$$

Note that the enhancement process only applied to the blue channel of RI. After Fourier coefficients enhancement, the generated RI_e may differs from RI, so in order to keep the visual quality, we should also make a visual constrain which is same as the constrain in phase-1 on RI_e, as Eq. (19) illustrated.

$$RI_{final} = RI_e \times (1 - JND) \tag{19}$$

So the final enhanced-watermarked image can be obtained by

$$I_{en} = I_o + RI_{final} \tag{20}$$

The importance of phase-2 and the analysis for frequency enhancement will be explained in detail at Section III-C.

C. The Analysis of Phase-2

The main procedure to improve the robustness in the proposed framework is the enhancement process of phase-2. In this section, we will first explain how the idea of phase-2 comes and why phase-2 is important in detail. Then we will show and discuss the most important features in the enhancement process in phase-2.

1) Analysis of two-stage training: In [15], Liu *et. al.* have proposed a two-stage training strategy that is directly applying extra training (the training process at phase-3 in the proposed scheme) after phase-1 which can relieve the limitation of differentiable noise layer. But there are still some distortions such as JPEG compression that cannot be well handled. The reason is that after the distortion, the encoded features in phase-1 are not guaranteed to be preserved. Once the distortion is too strong to eliminate the encoded features, the decoder in phase-3 cannot obtain enough features to learn, so the performance will be bad. Therefore, the key to guarantee the robustness is ensuring the survival of the encoded features in phase-1 after distortion. To achieve that, we have to add a phase between phase-1 and phase-3 to enhance the encoded features and make it is robust enough for distortions. Now the question is how the enhancement process can be achieved. We divide the analysis of enhancement process into two steps: 1). What features are more likely to be survived from the distortion? 2). How to enhance such features?

2) Analysis of encoded features: The first thing we need to determine is what features are conducive to survival in non-differentiable distortion. In the first phase, the encoded features generated by the encoder changes greatly with the iterative training process. But we find not all the encoded features trained with different epoch can survive from the distortion. Such conclusion is illustrated by the following experiments. We perform the two-stage training process (phase-3 training after the phase-1 initialization) with different epoch of pre-trained encoder and distortion of JPEG compression(QF=50).

Before phase-3 training, we should align visual quality of watermarked images to compare the performance in a more fair way. Specifically, we normalized the encoded features to 0 mean and 15 variance with all epochs, and then added it to the original image to conduct the encoding process. In this way, the visual quality of the watermarked images (measured by PSNR) is set at the same level. After that, we conduct phase-3 training on these images with JPEG compression(QF=50), and the results are shown in Table I.

TABLE I: The extraction accuracy against JPEG compression with different pre-trained encoder after normalization.

Epoch	30	40	60	80	100
PSNR	28.95403	29.03016	28.94537	29.0864	29.15976
Acc	95.46%	94.48%	88.67%	83.61%	78.48%

We can see that after nomarlization, the PSNR values are set to the same level of 29.1 ± 0.2dB. And from the accuracy result we can see that even when the PSNR is in the same level, the extraction accuracy trained with different encoders still varies. The encoded features with smaller epoch are more conducive to survive from distortion, which resulted to the larger extraction accuracy. So next, we will explore the differences of encoded features with different epochs, and analyze why encoded features with small epochs are more likely to survive from distortion.

Since we design the loss to encourage the network to modify more on blue channel, which means blue channel will trained to carry more information. So our subsequent feature analysis will be carried out in B-channel image of RI. Specifically, we take the B-channel images of RI under different epochs and performed Fourier transform on them. The results are shown in the Fig. 2.

As we can see in Fig. 2, with the training process going on, the RI changes significantly. In spatial domain, it changes to be adaptive to the image which will result to better visual quality. But in Fourier domian, the distribution of coefficients with large absolute value changes from the low-middle coefficient concentrated form to low-to-high coefficients uniformity form.

According to [1], the middle and low frequency coefficients are more robust than high frequency coefficients. That's the reason why training with pre-trained encoder of smaller epoch can obtain better robustness. So in order to better enhance the robustness, we should enhance the middle and low frequency coefficients of the image frequency spectrum.

3) Proposed Enhancement: There are two main constraints of encoded feature enhancement process in phase-2: 1). The

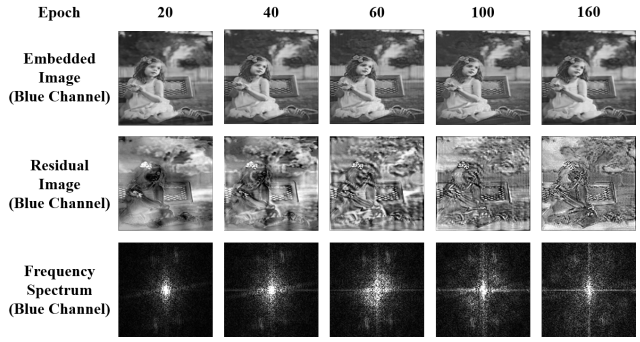

Fig. 2: The blue channel of residual image as well as the corresponding frequency spectrum generated by pre-encoder with different training epochs.

enhancement process should maintain the RI's texture to ensure the visual quality of the encoded image. 2). The enhancement process should improve the robustness under the premise of visual quality.

For 1), since we have implemented mask-guided image loss in phase-1, so in phase-2, the enhancement should still be mask-guided to keep the visual quality.

For 2), we need to enhance the middle and low frequency coefficients of RI and weaken the high frequency coefficients of RI, so that the robustness of RI can be better improved.

To meet the two above-mentioned demands, we propose the operations as shown in Eq. (13)-Eq. (20) to realize the enhancement procedure in phase-2. In this way, the mask-guided frequency enhancement can effectively achieve the requirements of visual quality and robustness.

After we get the watermarked image in phase-1, we first calculate the RI of such image. Then we centralize the RI, as shown in Eq. (14). There are two main reasons for centralization: 1) It will not change the texture of RI, so that the visual quality of watermarked image will not be significantly degraded. 2) It enables the frequency coefficients enhancement to be performed on a standardized distribution, so that we can utilize an enhancement factor to adjust the enhancement effect (robustness and visual quality). It's worth noted that we use Gaussian distribution matrix as the weight matrix. The reason for using Gaussian distribution is that it can achieve the effect of middle and low frequency coefficients enhancement and high frequency coefficients attenuation. And the enhance factor β controls the performance of enhancement. The specific experiment will be shown in Section V-B.

After that, we transform the enhanced Fourier coefficient matrix into spatial domain. Then we multiply the enhanced RI by the JND-mask in phase-1, as Eq. (19) shown.

D. Triple Phase Training

1) Phase-1: End-to-end noise free training: At phase-1, the end-to-end training without noise layer is adopted to generate a collaborative encoder and decoder. The training objective is to minimize:

$$\mathcal{L}_1 = \lambda_E \mathcal{L}_E + \lambda_D \mathcal{L}_D + \lambda_{A_d} \mathcal{L}_{A_d} \tag{21}$$

where λ_E, λ_D and λ_{A_d} are weights factors, and in this paper, we set $\lambda_E = 1$, $\lambda_D = 3$ and $\lambda_{A_d} = 0.001$ by default.

The primary target of phase-1 is to initialize an encoder which will be fixed at phase-2.

2) Phase-2: Mask-guided frequency enhancement for encoder: In phase 2, we first apply the encoder that is pre-trained in phase-1 to embed the watermark into host image. Then, the mask-guided frequency enhancement is adopted to enhance the encoded features. After that, the enhanced RI is further added into the host images to get the ultimate watermarked image. So the enhanced encoder is combined with the pre-trained encoder and the mask-guided frequency enhancement operation.

3) Phase-3: Adversarial training-based enhancement for decoder: In order to fine-tune the decoder to extract watermark from the distorted images, we should generated the distorted image dataset to train the decoder. Specifically, we embed a set of images with the enhanced encoder obtained in phase-2 and apply the practical distortion on them to generate the training dataset. Based on the dataset of distorted images, the decoder is trained to be adaptive to target distortions. In phase-3, only θ_D is updated by minimizing $\mathcal{L}_D$.

IV. Experimental Results

In this section, we will first briefly introduce the implementation details and the parameter selection. Then extensive experiments will be conducted to justify the performance of our method. Though the proposed architecture is designed for practical distortions, it can be used for not only the differentiable distortions but also the non-differentiable distortions. So we conduct the experiments on both differentiable and non-differentiable distortions to show effectiveness of the proposed scheme. Finally, more analysis and will be provided to justify our design.

A. Implementation Details

To train the network in phase-1, we randomly choose 10000 images from the COCO dataset [32] as our training dataset. The whole framework is implemented by PyTorch [33] and executed on NVIDIA RTX 2080ti. All images are reshaped to size of $128 \times 128 \times 3$. For gradient descent, Adam [34] is applied with default hyperparameters as the optimization method. Each model is trained for 200 epochs with a batch size of 32. In phase-3, we randomly choose 1000 images from COCO dataset and conduct the non-differentiable distortion on each image. So we use 10000 images in the COCO dataset [32] to train the nosie-free encoder and decoder in Phase-1. Then after getting the pre-trained encoder and decoder, we embed the watermark to 1000 different images from COCO dataset [32] with the operation in Phase-2 and obtain the 1000 watermarked images. After that, we perform the distortion on the 1000 watermarked images to generate the distorted datasets. Finally, we use the distorted datasets as the training datasets in Phase-3. All the test experiments are performed with the classical USC-SIPI image dataset [35].

To measure the visual quality of the watermarked image, we utilize PSNR as the default evaluation metrics. And for extracting accuracy and robustness evaluation, we directly use the extraction bit accuracy as the metric. We compare the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3149641, IEEE Transactions on Multimedia

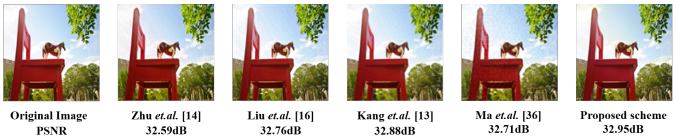

Fig. 3: The visual quality and PSNR with different watermarking schemes.

performance of the proposed framework with state-of-the-art deep-learning based algorithms [13], [15] and the traditional watermarking method [12], [36] which claims to be robust for most of the distortions.

For fair comparison, the length L of the random message M is set as 64 without error correction codes, and the enhancement factor β is set as 18. The PSNR of the each method is set in the same level of 32.5 ± 0.5 dB. The pre-trained encoder used for phase-2 is with the epoch of 30^{th}. The example of embedded images of different schemes are shown in Fig. 3.

B. Robustness Test

To test the robustness of the proposed framework, we conduct the experiments on not only the differentiable distortions but also the non-differentiable distortions. The specific distortions include 7 types of differentiable distortions: "Cropout", "Dropout", "Gaussian Noise", "Salt&Pepper", "Gaussian Blur", "Medium Blur" and "Resize" (as shown in Fig. 4) and 4 types of non-differentiable distortions: "JPEG Compression", "Style Transfer", "Screen-shooting", and "Instant Message Transmission" (as shown in Fig. 5). For differentiable distortion and "JPEG Compression" tests (examples are shown in supplementary materials), we compared the performance of our scheme with Zhu *et. al.* [13], Liu *et. al.* [15], Kang *et. al.* [12] and Ma *et. al.* [36], which are announced to be robust against such distortions. But for the rest of non-differentiable distortions, we only compare our scheme with Liu *et. al.* [15], Kang *et. al.* [12] and Ma *et. al.* [36] since Zhu *et. al.* [13] can only be adaptive to differentiable distortions.

1) Robustness against differentiable distortions:

a) Cropout Distortion: Cropout refers to the operation that crop a certain ratio of the image out and replace the cropped region with black image block. When fine-tuning the decoder in phase-3, we generate the distorted dataset with the cropout ratio uniformly selected from 25% to 35%. In testing frame, we changes the cropped ratio from 10% to 40% and conduct the robustness test. The experimental results are shown in Table II.

TABLE II: The extraction accuracy with different cropout ratios.

Ratio	10%	20%	30%	40%
Zhu *et. al.* [13]	85.6%	85.4%	85.2%	85.2%
Liu *et. al.* [15]	89.2%	89.3%	89.2%	88.6%
Kang *et. al.* [12]	83.4%	82.1%	78.9%	75.5%
Ma *et. al.* [36]	92.6%	91.4%	92.0%	89.6%
Proposed	**98.3%**	**97.9%**	**97.1%**	**95.3%**

As can be seen in Table II, the proposed scheme maintains the highest extraction accuracy in all the crop ratios compared with the other four schemes. The extraction accuracy of the proposed scheme are all higher than 95%, which indicates the great robustness against cropout distortions.

b) Dropout Distortion: Dropout indicates the operation of dropping and zeroing a certain ratio of image pixels. For phase-3 training, we randomly select the ratio from 15% to 25% to generate the training dataset. And for testing, we change the ratio of dropout from 5% to 25% to show the robustness against dropout distortion. The results are shown in Table III.

TABLE III: The extraction accuracy with different dropout ratios.

Ratio	5%	10%	15%	20%	25%
Zhu *et. al.* [13]	71.9%	63.5%	59.4%	60.6%	59.3%
Liu *et. al.* [15]	86.5%	87.6%	88.0%	87.9%	87.6%
Kang *et. al.* [12]	75.8%	70.3%	65.2%	64.6%	61.7%
Ma *et. al.* [36]	90.6%	88.4%	86.3%	83.9%	80.8%
Proposed	**97.1%**	**97.1%**	**97.9%**	**97.4%**	**96.9%**

We can see from Table III that the proposed scheme is robust to dropout distortions since the extraction accuracy are higher than 96%. Besides, compared with other four schemes, the performance of the proposed scheme against dropout attack is much better.

c) Gaussian Noise Distortion: For Gaussian noise distortion, the adversarial training data for phase-3 is generated with the variance of 0.01. And the testing variance of the noise ranges from 0.001 to 0.01. The results are indicated in Table IV.

TABLE IV: The extraction accuracy with Gaussian noise.

σ	0.001	0.002	0.005	0.01
Zhu *et. al.* [13]	79.8%	75.2%	73.9%	68.7%
Liu *et. al.* [15]	89.6%	90.0%	89.2%	86.5%
Kang *et. al.* [12]	83.2%	83.0%	81.2%	76.6%
Ma *et. al.* [36]	92.7%	91.9%	**92.4%**	**92.8%**
Proposed	**92.8%**	**93.3%**	91.8%	90.5%

As seen in Table IV, the robustness against Gaussian noise distortion is at least 2% higher than [12], [13], [15] in all variance. But for [36], the performance against the Gaussian noise with $\sigma = 0.005$ and $\sigma = 0.01$ is better than the proposed methods. It's mainly because [36] is embedding according to the statistical features, so Gaussian noise may not heavily influence such feature. Besides, since the network is trained with the Gaussian noise of 0.01 variance, for the noise with variance of 0.001, 0.002 and 0.005, the proposed scheme shows the robustness too. This indicates that training with strong noise distortion will make the network to be adaptive to the weak noise distortion.

d) Salt & Pepper Noise Distortion: Similar as the Gaussian noise, Salt & Pepper noise is commonly used too in watermarking attack. To resist such distortion, we add the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3149641, IEEE Transactions on Multimedia

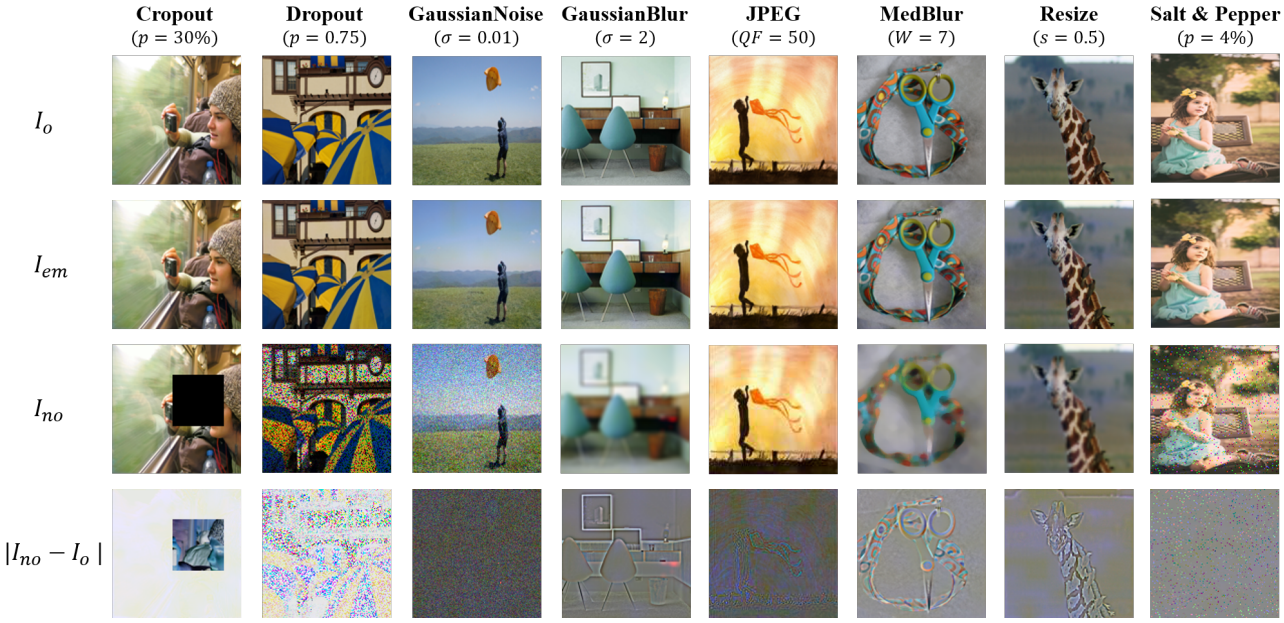

Fig. 4: The robustness tests on traditional image processing distortions. We show the visual quality of original image I_o, the enhanced image I_{en} and the distorted image I_{no} attacked by eight different types of traditional distortions: Cropout, Dropout, Gaussian Noise, Gaussian Blur, JPEG Compression, Medium Blur, Resize and SaltPepper.

noise with density of 0.05 for adversarial training. And we test the extraction performance with the density of 0.01, 0.02, 0.03, 0.04 and 0.05. The extraction results are shown in Table V.

TABLE V: The extraction accuracy with salt & pepper distortion.

Density	0.01	0.02	0.03	0.04	0.05
Zhu *et. al.* [13]	85.0%	85.2%	81.2%	76.9%	71.0%
Liu *et. al.* [15]	88.8%	88.7%	89.1%	88.6%	88.3%
Kang *et. al.* [12]	81.4%	79.7%	77.0%	76.6%	74.1%
Ma *et. al.* [36]	92.7%	92.9%	91.8%	91.4%	89.5%
Proposed	**97.7%**	**97.3%**	**97.7%**	**97.1%**	**97.0%**

It is easy to see from Table V that the proposed scheme get better performance than the compared schemes in salt & pepper noise distortion. With different density of noise, the proposed scheme all can maintain more than 97% accuracy, which indicates the great salt & pepper noise resilience of the proposed scheme.

e) Gaussian Blur Distortion: For Gaussian blur distortion, we generate the training dataset with the variance 2. And for testing stage, we conduct the Gaussian blurring operation with variance from 0 to 2 to show the robustness. The accuracy are shown in Table VI.

As seen in Table VI, the robustness against Gaussian blur distortion is higher than other four methods in all variance except for the variance of 2. With the variance of 2, the accuracy of the proposed schemes is almost same as Liu *et. al.* [15]. But for other variance, the proposed scheme performs much better.

f) Medium Blur Distortion: We utilize the medium blur operation with window 7×7 to generate the dataset for phase-3. And in testing frame, we test the performance with window

TABLE VI: The extraction accuracy with Gaussian blur distortion.

σ	0	0.5	1	2
Zhu *et. al.* [13]	79.2%	84.8%	81.9%	72.1%
Liu *et. al.* [15]	89.8%	68.8%	84.8%	**92.1%**
Kang *et. al.* [12]	84.4%	84.3%	82.8%	82.3%
Ma *et. al.* [36]	92.1%	90.2%	83.8%	62.1%
Proposed	**92.2%**	**90.4%**	**92.1%**	92.0%

size $3 \times 3, 5 \times 5$ and 7×7. The testing accuracy are shown in Table VII.

TABLE VII: The extraction accuracy with medium blur distortion.

Window	3×3	5×5	7×7
Zhu *et. al.* [13]	82.7%	73.1%	72.5%
Liu *et. al.* [15]	52.1%	52.3%	50.4%
Kang *et. al.* [12]	82.4%	59.3%	49.4%
Ma *et. al.* [36]	87.3%	69.7%	52.7%
Proposed	**95.3%**	**94.5%**	**94.0%**

As can be observed in Table VII, the extraction performance of proposed framework is much better than the compared scheme. And it is worth noting that medium blur will totally destroy the watermark signal embedded with [15], since the accuracy is no higher than 53%. Besides, for [36], median blur will greatly influence the extraction performance too. Since the median blur will greatly affect the statistical features of the image, so the extraction of [36] cannot perform well. As for Zhu *et. al.* [13] and Kang *et. al.* [12] , the performance of the proposed method is better in all the filtering windows. Besides, all the extraction accuracy of the proposed scheme is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3149641, IEEE Transactions on Multimedia

larger than 94%, which indicates the great performance against medium blur distortion.

g) Resize Distortion: To resist the resize distortion, we randomly resize the watermarked image to 0.5 to 2 times the original size to generate the adversarial training dataset. And we test the robustness with the resize ratio 0.5, 0.75, 1.25, 1.5 and 2. The corresponding results are shown in Table VIII.

TABLE VIII: The extraction accuracy with resize distortion.

Ratio	0.5	0.75	1.25	1.5	2
Zhu *et. al.* [13]	81.2%	84.8%	84.2%	83.9%	84.0%
Liu *et. al.* [15]	84.4%	88.5%	89.3%	89.7%	89.6%
Kang *et. al.* [12]	80.1%	83.4%	85.0%	84.3%	84.5%
Ma *et. al.* [36]	87.7%	88.3%	89.4%	89.8%	89.9%
Proposed	**98.1%**	**98.7%**	**99.1%**	**98.8%**	**98.8%**

From Table VIII we can see that the extraction accuracy with the proposed scheme is higher than the compared schemes in all the resizing ratios. The accuracy of the proposed method reaches at least 98% which indicates the outstanding performance against the resize attack.

2) Robustness against non-differentiable distortions:

a) JPEG Compression Distortion: In phase-3, we randomly select the value from 50 to 90 as the quality factor of the JPEG compression to generate the dataset. And in test experiment, we test the quality factor (QF) from 50-90. The corresponding results are shown in Table IX.

TABLE IX: The extraction accuracy with JPEG compression distortion.

QF	50	60	70	80	90
Zhu *et. al.* [13]	68.5%	70.6%	70.8%	73.3%	76.6%
Liu *et. al.* [15]	78.1%	80.8%	82.7%	85.0%	87.7%
Kang *et. al.* [12]	83.3%	83.8%	84.0%	84.3%	84.2%
Ma *et. al.* [36]	90.0%	90.2%	90.4%	91.4%	92.2%
Proposed	**91.5%**	**92.5%**	**93.7%**	**94.3%**	**95.0%**

As shown in Table IX, the extraction accuracy of proposed framework is higher than the compared schemes in all the QFs of JPEG compression. We believe the improvement is highly attribute to the mask-guided frequency enhancement operation, which ensures the preservation of the encoded feature in JPEG compression. Zhu *et. al.* [13] is announced to be robust to JPEG compression However, we find when facing the real JPEG compression, such framework performs bad. Besides, we find the extraction difference of Kang *et. al.* [12] changes little with different quality factor, it is mainly because such method is based on DFT, which is resilient to JPEG compression operation. Same conclusion can be found in Ma *et. al.* [36], the JPEG compression get little influence on the embedding features of [36], so the extraction are bigger than 90%.

b) Robustness Against Style Transferring: In this paper, we select four kinds of different black-box style transfer operations ("crayon", "oil painting", "star light" and "color pencil") to train the specific decoder. The experiment results are shown in Table X.

TABLE X: The extraction accuracy with different black-box style transfer schemes.

Style transfer	Crayon	Oil painting	Star light	Color pencil
Liu *et. al.* [15]	88.1%	92.0%	87.8%	85.6%
Kang *et. al.* [12]	59.9%	65.1%	73.4%	62.5%
Ma *et. al.* [36]	55.9%	48.0%	48.8%	**89.4%**
Proposed	**91.4%**	**92.3%**	**94.6%**	76.2%

From Table X we can observe that when facing the style transfer of "crayon", "oil painting", "star light", the extraction accuracy can up to 90%, however, when transferring to "color pencil" style, the accuracy becomes lower. The reason can be concluded that the "color pencil" style transferring operation only transforms the internal part of the image, and crops the external part out, and such operation will heavily influence the extraction process of the proposed scheme.

But for [15], the "color pencil" style transfer affect little on the performance. It is mainly because of the message duplicating ways. Since in [15], each bit message is duplicated $H \times W$ times and further concatenate with the hidden layer, which result to that the robustness against cropping distortion is better. So the performance of "color pencil" style transferring is better than the proposed method. But such duplicating scheme cannot be adapted to large embedding capacity and strong distortions. As for other style transfer distortions, the proposed scheme maintains a higher extraction accuracy. Besides, we find the style transferring heavily influence the performance of [12] which resulted to a low extraction accuracy. And for Ma *et. al.* [36], the operation of "Crayon", "Oil painting" and "Star light" style transferring cause great influence on extraction accuracy, but "Color pencil" affect little. It's mainly because "Color pencil" may not greatly change the pixel distribution in local area, so the extraction of [36] will not greatly influenced.

c) Robustness Against Screen-shooting Distortion: The results of robustness against screen-shooting process is shown in Table XI. In this paper, we randomly choose 1000 images to embed the watermark and conduct the screen shooting process on these images with the default screen "AOC-G2770PF" and phone "Huawei P30 Pro". The images used for generating the training dataset in phase-3 is captured randomly at 20-30cm and further perspective corrected and cropped to its original size. Then we test the robustness with shooting distance at 20-60cm respectively.

TABLE XI: The extraction accuracy with different screen-shooting distance.

Distance	20cm	30cm	40cm	50cm	60cm
Liu *et. al.* [15]	53.7%	54.2%	52.3%	54.3%	51.0%
Kang *et. al.* [12]	76.7%	74.6%	68.9%	62.8%	61.9%
Ma *et. al.* [36]	88.5%	87.9%	77.5%	77.1%	**83.9%**
Proposed	**92.2%**	**94.1%**	**78.4%**	**86.9%**	83.7%

It is clear to see from Table XI that for the test distance of 20-30cm, the extraction accuracy is above 90%, where at 50-60cm, the accuracy is lower than 90% and with the distance

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3149641, IEEE Transactions on Multimedia

Fig. 5: The robustness tests on non-differentiable distortions. We show the visual quality of original image I_o, the enhanced image I_{en} and the distorted image I_{no} influenced by five different types of non-differentiable distortions: style transfer, instant message app transmission and screen-shooting.

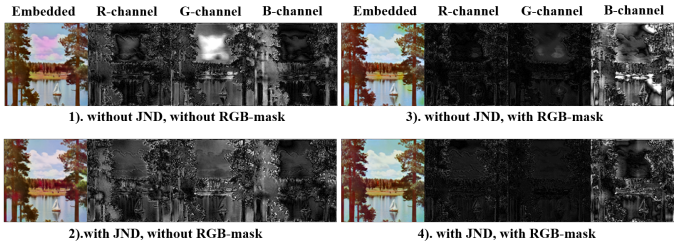

Fig. 6: The encoded image of epoch 30 with different training loss: 1). without JND, without RGB-mask 2).with JND, without RGB-mask 3). without JND, with RGB-mask 4). with JND, with RGB-mask.

arise, the accuracy decreases. But it should be noted that the when shooting at 40cm, the accuracy becomes extremely low. The reason is that when shooting at 40cm, there are obvious moiré patterns occur in the captured image, which is not appeared in training dataset and greatly influence the extraction process, so the performance will be bad. Besides, we believe that if we enlarge the training dataset, the accuracy will be higher. But for [15], the accuracy stays in a low range of 50%, which means the decoder cannot effectively learn enough features to realize the extraction. That is to say, the robustness against screen-shooting distortion for [15] is weaker than the proposed scheme. Since [12] is designed for print-shooting process, so the performance against screen-shooting distortion is not good enough. As for [36], it performs better than [15] and [12], but for the distance of 20-50cm, the proposed scheme is still better than [36].

d) Robustness Against Instant Message (IM) APP transmission: As for instant message APP transmission, the image before and after transmission will undergo a series image processing operations such as resizing and lossy compression. And for different APP, the specific procedure will be different. In this paper, we use "Wechat" as the default APP to generate the training dataset and test the performance on "QQ", "Facebook", "Twitter" and "Instagram". The results are shown in Table XII.

As can be seen in Table XII, the performance against

TABLE XII: The extraction accuracy with different instant messaging application's transmission.

IM APP	QQ	Wechat	Twitter	Facebook	Instagram
Liu *et. al.* [15]	81.7%	80.3%	84.7%	84.8%	80.3%
Kang *et. al.* [12]	85.4%	85.2%	85.7%	85.7%	85.7%
Ma *et. al.* [36]	92.7%	57.6%	53.5%	48.8%	89.1%
Proposed	**91.4%**	**91.6%**	**93.6%**	**93.5%**	**92.8%**

Fig. 7: The comparison of mask-guided RI and non-mask-guided RI.

IM APP transmission of the proposed method is better than [15] in the view of "QQ", "Wechat", "Twitter", 'Facebook" and "'Instagram". Besides, although the training dataset is generated with "Wechat", the decoder is useful in the other 4 IM APPs. From the results we can conclude that the decoder training with "Wechat" transmission can be successful in other IM APPs. It can be seen that the distortion of "Twitter", "Facebook" and "Instagram" transmission seems got the same influence in [12], which resulted to the same extraction accuracy.

C. Robustness against unknown distortions

In order to illustrate the generalization of the proposed scheme, that is, the robustness against unknown distortions, we conduct the corresponding experiments. We use the pre-trained decoder of one distortion to extract the watermark from the images that are distorted by other distortions. For example, we use the decoder trained with JPEG compression attack to extract the Gaussian noised images. So that for JPEG-

decoder, such noise is unknown. We use the mismatch way to represent the unknown black-box distortions and illustrated the robustness of the proposed scheme. The corresponding results are shown in Table XIII.

TABLE XIII: The extraction accuracy with different pre-trained decoder.

Pre-trained Decoder	Gaussian Noise (σ = 0.005)	Salt&Pepper Density=0.03	Medium Blur window = 3	JPEG QF = 70
Gaussian Noise	91.8%	91.5%	90.4%	89.3%
Salt&Pepper	78.3%	97.7%	94.4%	83.8%
Medium Blur	71.5%	72.0%	94.5%	83.9%
JPEG	90.5%	87.1%	94.1%	93.7%

As can be seen in Table XIII, the distortion we choose to pre-trained the decoder are "Gaussian Nosie", "Salt&Pepper Noise", "Medium Blur" and "JPEG Compression". And the distortion used for testing are "Gaussian Nosie" with σ=0.005, "Salt&Pepper Noise" with $Density = 0.03$, "Medium Blur" with $window = 3$ and "JPEG Compression" with $QF = 70$. From Table XIII we can see that for all the distortions, extracting with the corresponding pre-trained decoder will ensure the highest accuracy. For example, when facing "Salt&Pepper Noise", extract with pre-trained "Salt&Pepper-decoder" will achieve 97% accuracy while with "Medium Blur-decoder", it can only achieve 72%. This indicates that if we want to get the best extraction performance, training with the target distortion is needed.

Nevertheless, we find that some pre-trained decoder have certain generalization ability, such as "Gaussian Noise-decoder". The extraction accuracy with "Gaussian Noise-decoder" maintains high level for all distortions we test. So maybe using multiple combined noise to train decoder is a good way to improve generalization ability for unknown distortions.

V. Ablation Study

A. The influence of JND-mask loss

In this section, we mainly show and discuss the importance of JND-mask loss. We conduct phase-1 with four different image loss: 1). without JND, without RGB-mask 2).with JND, without RGB-mask 3). without JND, with RGB-mask 4). with JND, with RGB-mask, where JND refers to the JND-guided-mask calculated by Eq. (1), and RGB-mask represents the weight for each channel $(\delta_r, \delta_g, \delta_b)$. The results of encoded image with epoch 30 are shown in Fig. 6.

As can be seen from Fig. 6, training with JND-mask loss will significantly improve the image quality at the same iteration epoch. Comparing the encoded image trained with loss 1) and 2) we can find that with JND guided loss, the encoder learns to embed the watermark signal into the region with complex texture instead of the smooth region, so that there will be less visual distortion. As for RGB-mask loss, we can see from the loss 1) and 3) that the encoder tends to embed the watermark signal into the blue channel more instead of embedding uniformly in three channels. Besides, compared 4) with 1),2) and 3), we can observe that training with both JND-guided and RGB-mask loss, the visual quality of encoded image is the best, so we can draw the conclusion that the designed JND-mask loss can effectively improve the visual quality.

B. The influence of β

To better adjust the enhancement process, we set an enhancement factor β in phase-2. In this section, we will show and dicuss the influence of β with PSNR value and extraction accuracy after JPEG compression (QF=50). After phase-1 training, we embed the watermark with the pre-trained encoder at 30^{th} epoch and enhanced such images with different β. The range of β is selected from 11 to 20. And then we utilize the enhanced images to train phase-3 with JPEG compression (QF=50). The corresponding PSNR values and extraction accuracy are shown in Table XIV.

TABLE XIV: The visual quality and extraction accuracy with different enhancement factor β.

β	11	13	15	17	19
PSNR(dB)	36.25	35.06	34.00	33.04	32.16
Accuracy	84.9%	88.2%	90.5%	92.6%	94.1%

It can be illustrated from Table XIV that the bigger β will result to poorer visual quality, but will maintain stronger robustness against distortions. β can serve as a parameter to adjust the visual quality and robustness. We can choose the appropriate β to complete the image embedding process according to different requirements.

C. The influence of mask-guided frequency enhancement

In this section, we will show the importance of the enhancement process from the aspect of robustness and visual quality. Specifically, after embedding the watermark by the pre-trained encoder in phase-1, we conduct three different operations to complete the phase-2 process and generate the dataset for phase-3 training. The three operations are: 1). no enhancement; 2) frequency enhancement without mask guided; 3) mask-guided frequency enhancement. And the distortion we used in phase-3 is JPEG compression (QF=70). The corresponding results are shown in Fig. 7 and Table XV.

We can see from the watermarked image that the visual quality generated with mask-guided RI is better than the other two RIs. And from the RI image, it can be seen that after mask guided, the modification is more concentrate on blue channel and the high weighted region, which is same as the constraint in phase-1. So that applying mask-guided frequency enhancement is good for obtaining high visual quality.

For fair comparison, we have controlled the value of each RI so that the PSNR of the watermarked image generated by it is at the same level of 32.5 ± 0.5dB. And the corresponding extraction accuracy trained with each RI is shown in Table XV.

Table XV indicates that the mask-guided enhancement can effectively enhance the robustness of the algorithm. The extraction accuracy corresponded to mask-guided RI is almost 5% higher than that without enhancement, and 3% higher than

TABLE XV: The extraction accuracy with different RI.

RI	no enhancement	no mask-guided	mask-guided
PSNR(dB)	32.70	32.84	32.59
Accuracy	88.9%	90.2%	93.7%

that with only frequency enhancement. We conclude the reason as the mask effectively adjusts the weight of RI, so that under the same visual quality constraints, the weight of the region that carries more information is larger. So the decoder can obtain more information after distortion to be trained, which resulted to higher extraction accuracy.

VI. Limitations

Although the proposed framework can well adapt to both differentiable and non-differentiable distortions, there are still some limitations should be improved in future work.

1) Such work is weak to desynchronization attack such as crop and rotate, since the watermark is reshaped and flattened in the preprocessing, so 1-bit message is corresponded to a specific block of the images. Once such block is cropped or transformed, the extraction will greatly be influenced.

2) Training the proposed scheme is time-consuming. Since in phase-3, the training dataset should be generated by real distortions, so it takes much time in obtain the dataset. Our future work may focus on how to optimize the performance of the proposed framework from the aspects of above two weaknesses.

VII. Conclusion

In this paper, we propose a triple-phase watermark framework for practical distortions, which consists of an end-to-end noise-free training phase, a mask-guided frequency enhancement phase and an adversarial training-based decoder enhancement phase. By utilizing the mask-guided frequency enhancement operation, the encoded feature is greatly enhanced so that it can be preserved during the distortions and can be further recovered by the decoder. Extensive experiments demonstrate the effectiveness of the designed structure in the view of robustness against differentiable and non-differentiable distortions. Besides, we show and discuss our understanding of the encoding features as training processed, which we hope will benefit the further research.

References

[1] X. Kang, R. Yang, and J. Huang, "Geometric invariant audio watermarking based on an LCM feature," *IEEE Trans. Multimedia*, vol. 13, no. 2, pp. 181–190, 2011.

[2] X. Zhang, F. Peng, and M. Long, "Robust coverless image steganography based on dct and lda topic classification," *IEEE Transactions on Multimedia*, vol. 20, no. 12, pp. 3223–3238, 2018.

[3] Z. Chen, L. Li, H. Peng, Y. Liu, and Y. Yang, "A novel digital watermarking based on general non-negative matrix factorization," *IEEE Transactions on Multimedia*, vol. 20, no. 8, pp. 1973–1986, 2018.

[4] Y. Huang, B. Niu, H. Guan, and S. Zhang, "Enhancing image watermarking with adaptive embedding parameter and psnr guarantee," *IEEE Transactions on Multimedia*, vol. 21, no. 10, pp. 2447–2460, 2019.

[5] M. Andalibi and D. M. Chandler, "Digital image watermarking via adaptive logo texturization," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5060–5073, 2015.

[6] C. Chang and J. Shen, "Features classification forest: A novel development that is adaptable to robust blind watermarking techniques," *IEEE Trans. Image Process.*, vol. 26, no. 8, pp. 3921–3935, 2017.

[7] B. Mathon, F. Cayre, P. Bas, and B. Macq, "Optimal transport for secure spread-spectrum watermarking of still images," *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1694–1705, 2014.

[8] H. Fang, W. Zhang, H. Zhou, H. Cui, and N. Yu, "Screen-shooting resilient watermarking," *IEEE Trans. Inf. Forensics Secur.*, vol. 14, no. 6, pp. 1403–1418, 2019.

[9] A. Pramila, A. Keskinarkaus, V. Takala, and T. Seppänen, "Extracting watermarks from printouts captured with wide angles using computational photography," *Multim. Tools Appl.*, vol. 76, no. 15, pp. 16 063–16 084, 2017.

[10] D. Gugelmann, D. Sommer, V. Lenders, M. Happe, and L. Vanbever, "Screen watermarking for data theft investigation and attribution," in *10th International Conference on Cyber Conflict, CyCon 2018, Tallinn, Estonia, May 29 - June 1, 2018*. IEEE, 2018, pp. 391–408.

[11] A. Pramila, A. Keskinarkaus, and T. Seppänen, "Toward an interactive poster using digital watermarking and a mobile phone camera," *Signal, Image and Video Processing*, vol. 6, no. 2, pp. 211–222, 2012.

[12] X. Kang, J. Huang, and W. Zeng, "Efficient general print-scanning resilient data hiding based on uniform log-polar mapping," *IEEE Trans. Inf. Forensics Secur.*, vol. 5, no. 1, pp. 1–12, 2010.

[13] J. Zhu, R. Kaplan, J. Johnson, and L. Fei-Fei, "Hidden: Hiding data with deep networks," in *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XV*, ser. Lecture Notes in Computer Science, vol. 11219. Springer, 2018, pp. 682–697.

[14] M. Tancik, B. Mildenhall, and R. Ng, "Stegastamp: Invisible hyperlinks in physical photographs," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. IEEE, 2020, pp. 2114–2123.

[15] Y. Liu, M. Guo, J. Zhang, Y. Zhu, and X. Xie, "A novel two-stage separable deep learning framework for practical blind watermarking," in *Proceedings of the 27th ACM International Conference on Multimedia, MM 2019, Nice, France, October 21-25, 2019*. ACM, 2019, pp. 1509–1517.

[16] E. Wengrowski and K. Dana, "Light field messaging with deep photographic steganography," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, 2019, pp. 1515–1524.

[17] R. G. Van Schyndel, A. Z. Tirkel, and C. F. Osborne, "A digital watermark," in *Proceedings 1994 International Conference on Image Processing, Austin, Texas, USA, November 13-16, 1994*. IEEE Computer Society, 1994, pp. 86–90.

[18] T. Zong, Y. Xiang, I. Natgunanathan, S. Guo, W. Zhou, and G. Beliakov, "Robust histogram shape-based method for image watermarking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 5, pp. 717–729, 2015.

[19] G. Hua, Y. Xiang, and L. Y. Zhang, "Informed histogram-based watermarking," *IEEE Signal Process. Lett.*, vol. 27, pp. 236–240, 2020.

[20] T. Nakamura, A. Katayama, M. Yamamuro, and N. Sonehara, "Fast watermark detection scheme for camera-equipped cellular phone," in *Proceedings of the 3rd international conference on Mobile and ubiquitous multimedia*. ACM, 2004, pp. 101–108.

[21] H. Hu and T. Lee, "Frame-synchronized blind speech watermarking via improved adaptive mean modulation and perceptual-based additive modulation in DWT domain," *Digit. Signal Process.*, vol. 87, pp. 75–85, 2019.

[22] Y. Gao, J. Wang, and Y. Shi, "Dynamic multi-watermarking and detecting in DWT domain," *J. Real Time Image Process.*, vol. 16, no. 3, pp. 565–576, 2019.

[23] W. Ding, Y. Ming, Z. Cao, and C.-T. Lin, "A generalized deep neural network approach for digital watermarking analysis," *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2021.

[24] B. Chen, Y. Wu, G. Coatrieux, X. Chen, and Y. Zheng, "Jsnet: A simulation network of jpeg lossy compression and restoration for robust image watermarking against jpeg attack," *Computer Vision and Image Understanding*, vol. 197, p. 103015, 2020.

[25] S. Mellimi, V. Rajput, I. A. Ansari, and C. W. Ahn, "A fast and efficient image watermarking scheme based on deep neural network," *Pattern Recognition Letters*, vol. 151, pp. 222–228, 2021.

[26] M. Ahmadi, A. Norouzi, N. Karimi, S. Samavi, and A. Emami, "Redmark: Framework for residual diffusion watermarking based on deep networks," *Expert Systems with Applications*, vol. 146, p. 113157, 2020.

[27] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015 - 18th International*

Conference Munich, Germany, October 5 - 9, 2015, Proceedings, Part III, ser. Lecture Notes in Computer Science, vol. 9351. Springer, 2015, pp. 234–241.

[28] C.-H. Chou and Y.-C. Li, "A perceptually tuned subband image coder based on the measure of just-noticeable-distortion profile," *IEEE Transactions on circuits and systems for video technology*, vol. 5, no. 6, pp. 467–476, 1995.

[29] C. Ware, *Information visualization: perception for design*. Elsevier, 2012.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 2016, pp. 770–778.

[31] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. IEEE Computer Society, 2017, pp. 5967–5976.

[32] T. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," in *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, ser. Lecture Notes in Computer Science, D. J. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., vol. 8693. Springer, 2014, pp. 740–755.

[33] R. Collobert, K. Kavukcuoglu, and C. Farabet, "Torch7: A matlab-like environment for machine learning," 2011.

[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[35] The USC-SIPI Image Database. Accessed: Sep. 2019. [Online]. Available: http://sipi.usc.edu/database/.

[36] Z. Ma, W. Zhang, H. Fang, X. Dong, L. Geng, and N. Yu, "Local geometric distortions resilient watermarking scheme based on symmetry," *IEEE Transactions on Circuits and Systems for Video Technology*, 2021.

Han Fang received his B.S. degree in 2016 from Nanjing University of Aeronautics and Astronautics (NUAA) and a Ph.D degree in 2021 from University of Science and Technology of China (USTC). Currently, he is a research fellow at National University of Singapore. His research interests include image watermarking, information hiding and adversarial machine learning.

Zhaoyang Jia Zhaoyang Jia has been studying as an undergraduate at University of Science and Technology of China (USTC) since 2018. Currently, he is also an intern at Microsoft Research Asia. His research intersts include digital watermarking, multimedia computing and deep learning.

Hang Zhou received his B.S. degree in 2015 from Shanghai University (SHU) and a Ph.D. degree in 2020 from the University of Science and Technology of China (USTC). Currently, he is a postdoctoral researcher at Simon Fraser University. His research interests include computer graphics, multimedia security and deep learning.

Zehua Ma received his B.S. degrees in information security from the University of Science and Technology of China (USTC) in 2018. He is currently pursuing the Ph.D. degree in information security in USTC. His research interests include image watermarking, information hiding, and image processing.

Weiming Zhang received his M.S. degree and Ph.D. degree in 2002 and 2005 respectively from the Zhengzhou Information Science and Technology Institute, P.R. China. Currently, he is a professor with the School of Information Science and Technology, University of Science and Technology of China. His research interests include information hiding and multimedia security.