Flow-Based Robust Watermarking with Invertible Noise Layer for Black-Box Distortions

Han Fang[1], Yupeng Qiu[1], Kejiang Chen[2*], Jiyi Zhang[1], Weiming Zhang[2], Ee-Chien Chang[1*]
[1] National University of Singapore
[2] University of Science and Technology of China
fanghan@nus.edu.sg, qiu_yupeng@u.nus.edu, chenkj@ustc.edu.cn, jiyizhang@u.nus.edu, zhangwm@ustc.edu.cn, changec@comp.nus.edu.sg

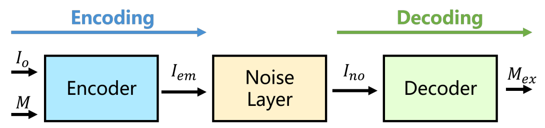
Abstract

Deep learning-based digital watermarking frameworks have been widely studied recently. Most existing methods adopt an "encoder-noise layer-decoder"-based architecture where the embedding and extraction processes are accomplished separately by the encoder and the decoder. However, one potential drawback of such a framework is that the encoder and the decoder may not be well coupled, resulting in the fact that the encoder may embed some redundant features into the host image thus influencing the invisibility and robustness of the whole algorithm. To address this limitation, this paper proposes a flow-based robust watermarking framework. The basic component of such framework is an invertible up-down-sampling neural block that can realize the embedding and extraction simultaneously. As a consequence, the encoded feature could keep high consistency with the feature that the decoder needed, which effectively avoids the embedding of redundant features. In addition, to ensure the robustness of black-box distortion, an invertible noise layer (INL) is designed to simulate the distortion and is served as a noise layer in the training stage. Benefiting from its reversibility, INL is also applied as a preprocessing before extraction to eliminate the distortion, which further improves the robustness of the algorithm. Extensive experiments demonstrate the superiority of the proposed framework in terms of visual quality and robustness. Compared with the state-of-the-art architecture, the visual quality (measured by PSNR) of the proposed framework improves by 2dB and the extraction accuracy after JPEG compression (QF=50) improves by more than 4%. Besides, the robustness against black-box distortions can be greatly achieved with more than 95% extraction accuracy.

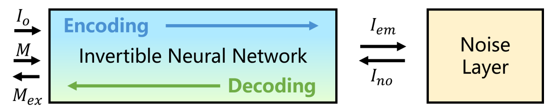
Introduction

Robust digital watermarking is widely used for copyright protection and leaking source tracing. By embedding invisible watermark signals, the watermarked carrier is endowed with authorized information. When a copyright dispute arises from an illegal copy, we can extract the watermark for copyright authentication. The most important property of robust watermarking is robustness, which greatly affects the applicability of the algorithm. Robustness refers to the ability to resist distortion, i.e., whether the watermark can be losslessly recovered when the watermarked signal has been distorted. In pursuit of robustness, traditional algorithms often embed watermarks in transform domain coefficients, commonly in the discrete cosine transform (DCT) domain (Ko et al. 2020), discrete wavelet transform (DWT) domain (Daren et al. 2001) and discrete Fourier transform (DFT) domain (Urvoy, Goudia, and Autrusseau 2014).

[*]Corresponding Authors.



(a) The architecture of END-based framework.



(b) The architecture of flow-based framework.

Figure 1: The network comparison of END and flow-based framework.

With the success of deep learning in many applications, deep-learning-based watermarking frameworks (Mun et al. 2017, 2019; Ahmadi et al. 2020) have recently been proposed. The main backbone of such frameworks is an autoencoder-like architecture which contains an **e**ncoder, a **n**oise layer, and a **d**ecoder (END), as shown in Fig. 1 (a). The encoder aims to embed the watermark message into the host image. The noise layer distorts the watermarked image with several differentiable distortions. The decoder tries to extract the watermark message from the distorted images. Based on the joint training of these three components, robustness can be well guaranteed.

Although the whole framework could be trained end-to-end in an automatic way, it suffers from the potential drawback that the encoder may not be well coupled with the decoder. The consequent result is that the encoder may embed redundant features that are useless for decoding, thus limiting the performance of the framework. In addition, most of the existing noise layers are designed for white-box distortions (differentiable/non-differentiable), such as Gaussian noise, JPEG compression, print-shooting, etc. These distortions have been studied extensively, so that they can be simulated by one or several differentiable operations. The simulated operation could effectively serve as a noise layer to ensure the corresponding robustness. However, in real life, there are not only white-box distortions but also many black-box distortions, such as style transferring in social media applications. The mechanisms of these distortions are often unknown. When facing such black-box distortions, traditional noise layers may not be effective in guaranteeing targeted robustness.

To address these two limitations, we propose a flow-based watermarking framework with an invertible noise layer. For the unsatisfactory coupling between encoder and decoder, inspired by invertible neural network (INN), we propose a flow-based architecture which consists of multiple invertible "up-down-sampling" blocks. The proposed architecture can effectively achieve forward-encoding and backward-decoding functions with the same network, as shown in Fig. 1 (b). Thus, both encoding and decoding share the same parameters, which ensures tight coupling between the encoder and the decoder.

For black-box distortions, we propose to use another INN to simulate such distortions and treat it as a noise layer. The resulting invertible noise layer effectively ensures robustness to black-box distortions. In addition, benefiting from the reversibility of the invertible noise layer, we can utilize the backward process as a denoising operation before extraction which could further improve the robustness. Consequently, the proposed framework achieves state-of-the-art performance on invisibility, white-box robustness and black-box robustness.

The main contributions of this paper are summarized as follows:

1) In dealing with the unsatisfactory coupling of encoder and decoder in existing END-based watermarking architecture, we proposed a flow-based robust watermarking framework, which guarantees a tight coupling of encoder and decoder where the same parameters are utilized for both encoding and decoding.
2) We propose an invertible noise layer, which can be used both for forward-training and backward-denoising, thus effectively ensuring robustness to black-box distortion.
3) Extensive experiments indicate the superior performance of the proposed scheme compared with the state-of-the-art DNN-based watermarking schemes on visual quality, and robustness for both white-box distortion and black-box distortion.

Related Work

Steganography&Watermarking

Steganography (Chen et al. 2018) is a technique for hiding information in a carrier to enable covert communication. Most traditional image steganography focuses on improving the performance of embedding efficiency and anti-detection, such as HUGO (Filler and Fridrich 2010), UNIWARD (Holub, Fridrich, and Denemark 2014)and HILL (Li et al. 2014). In recent years, many DNN-based steganography schemes have also been proposed, among which, the flow-based schemes (Jing et al. 2021; Xu et al. 2022) achieve the best performance thanks to the reversibility of INN. Different from steganography, in which the most important property is concealment, digital watermarking is mostly designed for realizing copyright protection, and its most important property is robustness to different distortions. Recently, many end-to-end deep-learning-based robust watermarking frameworks have been proposed. Zhu*et al.* (Zhu et al. 2018) first proposed the END architecture and successfully guaranteed several image processing robustness by setting different noise layers. Based on END, Tancik *et al.* (Tancik, Mildenhall, and Ng 2019) proposed to simulate the print-shooting process with several differentiable operation and generate the corresponding noise layer. As a result, the print-shooting robustness was effectively guaranteed. Wengrowski *et al.* (Wengrowski and Dana 2019) proposed to use a network named CDTF to approximate the screen-shooting distortion and set the well-trained CDTF as the noise layer to satisfy the screen-shooting robustness. Jia *et al.* (Jia, Fang, and Zhang 2021) proposed a method for JPEG compression distortion called MBRS whereby robustness is achieved by alternatively training the network with "real JPEG" and "simulated JPEG" noise. These methods are one-stage methods that attempt to simulate the distortion in the noise layer. In addition to the one-stage method, Liu *et al.* (Liu et al. 2019) proposed a two-stage method for black-box distortion, they first initialized an encoder and a decoder and further fine-tuned the decoder with black-box distorted images to achieve stronger robustness. However, the biggest drawback of all these schemes is that tight coupling between the encoder and the decoder cannot be effectively ensured, which may affect the robustness. Since steganography is fundamentally different from digital watermarking, we do compare with steganography methods.

Normalizing Flow-based Model

Normalizing flow-based model can directly compute the likelihoods, which is widely used for generation tasks. The structure of the flow-based model is invertible neural networks, which could effectively achieve the forward and backward mapping with the same network and the same parameters. Based on the reversibility of INN, the normalization mapping could be well satisfied. Previous research, such as NICE (Dinh, Krueger, and Bengio 2014) and RealNVP (Dinh, Sohl-Dickstein, and Bengio 2016), has discovered the powerful generation ability of flow-based models. (Gilbert et al. 2017) gives a great explanation of reversibility. Based on NICE and RealNVP, Glow (Kingma and Dhariwal 2018) and i-RevNet (Jacobsen, Smeulders, and Oyallon 2018) further update the specific architecture of INN for density estimation and achieve better generation performance. The reversibility of the flow-based method is naturally well suited to the needs of robust watermarking, as it can effectively achieve full coupling of encoder and decoder.

Proposed Framework

Overview

The main purpose of the proposed method is to design a robust watermarking framework that could be used not only

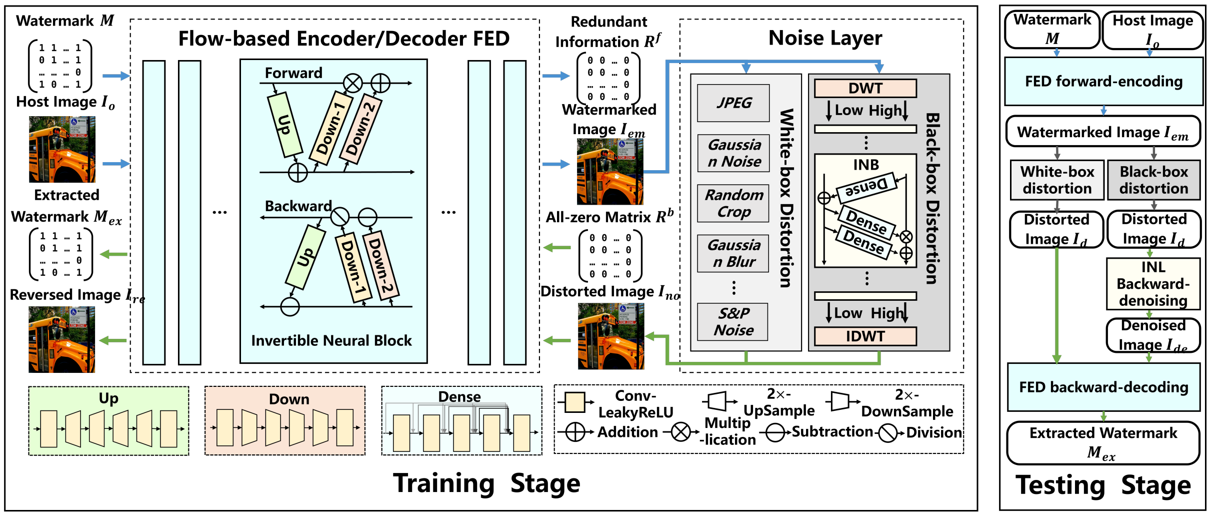


Figure 2: The framework of the proposed method. The main architecture of the algorithm is a flow-based encoder and decoder, which consists of several invertible neural blocks and can be used for a forward-encoding process and a backward-decoding process. Between the encoder and decoder, a noise layer which contains the white-box and black-box distortions is performed to distort the watermarked image into a distorted version for decoder training.

for white-box distortion but also for black-box distortion, as shown in Fig. 2. The main component to be trained is a flow-based encoder/decoder (FED) with several invertible neural blocks. Due to the structural reversibility of FED, it can implement both forward-encoding and backward-decoding processes with the same parameters. The forward-encoding process hides the watermark information M in the host image I_o to obtain the watermarked image I_{em} and the redundant R^f. Then the noise layer effectively distorts the watermarked image I_{em} and provides the distorted image I_d for decoder training. The backward-decoding process takes the distorted image I_d which is processed by the noise layer and the all-zero matrix R^b as input, and decodes them to obtain the extracted watermark signal M_{ex}. For white-box distortion, we use the existing differentiable noise layer for training. As for black-box distortion, we will adopt another INN to simulate such distortions and use the well-trained model as the noise layer named invertible noise layer (INL), details of which will be illustrated in Section . In testing stage, when facing the black-box distortion, we first use the backward process of INL to denoise the distorted image I_d into a denoised version I_{de}, and feed I_{de} into the FED for further decoding.

Flow-based Encoder/Decoder

As aforementioned, the flow-based network is naturally suitable for watermarking tasks. There are two basic components of flow-based models, the forward-encoding function denoted as f_θ and the corresponding inverse function f_θ^{-1} with the same parameter θ. In forward-encoding process, it receives the watermark matrix $M \in \mathcal{R}^{h\times w\times 1}$ and the host image $I_o \in \mathcal{R}^{H\times W\times 3}$ as inputs and then outputs the watermarked image $I_{em} \in \mathcal{R}^{H\times W\times 3}$ and the redundant information $R^f \in \mathcal{R}^{h\times w\times 1}$. For backward-decoding process, the all-zero matrix $R^b \in \mathcal{R}^{h\times w\times 1}$ and the distorted image $I_d \in \mathcal{R}^{H\times W\times 3}$ are fed into the FED for decoding to obtain the extracted watermark $M_{ex} \in \mathcal{R}^{h\times w\times 1}$ and the recovered image $I_{re} \in \mathcal{R}^{H\times W\times 3}$.

FED are combined with n invertible neural blocks. Fig. 3 shows the structure of the i^{th} invertible neural block, which consists of an up-sub-network U_i, and two down-sub-networks D_i^1 and D_i^2. U_i aims to up-sample $m_i \in \mathcal{R}^{h\times w\times 1}$ to the same size as the image $I_o \in \mathcal{R}^{H\times W\times 3}$. D_i^1 and D_i^2 aim to down-sample $x_{i+1} \in \mathcal{R}^{H\times W\times 3}$ to the same size as $m_i \in \mathcal{R}^{h\times w\times 1}$. For the i^{th} invertible neural block in the forward-encoding process, the inputs are m_i and x_i, and the outputs m_{i+1} and x_{i+1} can be formulated as follows:

$$\begin{aligned} x_{i+1} &= x_i + U_i\left(m_i\right) \\ m_{i+1} &= m_i \otimes \exp\left(D_i^1\left(x_{i+1}\right)\right) + D_i^2\left(x_{i+1}\right) \end{aligned} \tag{1}$$

where $\otimes$ indicates the dot product operation. After the last invertible neural network, we can obtain m_{n+1} and x_{n+1} which correspond to the final outputs, i.e., redundant information R^f and the watermarked image I_{em}.

For the backward-decoding process, the information flows are from the $(i+1)^{th}$ invertible neural network to the i^{th} invertible neural network, as shown in Fig. 3. Specifically, for the first invertible neural network, the input is an all-zero matrix R^b and the distorted image I_d generated by the noise layer, and the outputs of the first last invertible neural network in backward-decoding are r_n and x'_n. For the i^{th} invertible neural network, the inputs are r_{i+1} and x'_{i+1}, and the outputs are r_i and x'_i which could be calculated with:

$$\begin{aligned} r_i &= \left(r_{i+1} - D_i^2\left(x'_{i+1}\right)\right) \otimes \exp\left(-D_i^1\left(x'_{i+1}\right)\right) \\ x'_i &= x'_{i+1} - U_i\left(r_i\right) \end{aligned} \tag{2}$$

After the process of the last invertible neural network in the backward-decoding process, the output M_{ex}, i.e., r_1, is obtained as the extracted watermark. It should be noted that $R^b \in \mathcal{R}^{h\times w\times 1}$ is an all-zero matrix, so for decoding, no prior information other than the distorted image I_d is

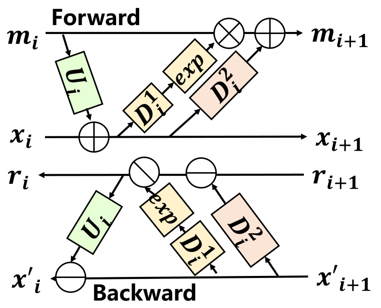


Figure 3: The backbone of the i^{th} invertible neural block, which consists of one up-sub-network U_i and two down-sub-networks D_i^1 and D_i^2.

needed, which ensures the blind extraction. In this paper, the basic components of each U_i and D_i consist of 6 "Conv-LeakyReLU" blocks, as shown in Fig. 2.

Noise Layer

The noise layer is the key to ensure the robustness. In this paper, we use different noise layers for white-box and black-box distortion, respectively. For the white-box distortion, we directly use the existing differentiable noise layer (e.g., JPEGSS (Shin and Song 2017) for JPEG compression). For black-box distortion (e.g., style transferring), we propose a novel way to achieve the corresponding robustness by using INN to simulate the distortion and serving the well-trained model as the noise layer. The details can be described as follows:

Training Data of INL To train the INL (denoted as N_{θ_d} with parameter θ_d), we first generate the training data by querying the black-box distortion process. Specifically, we feed the original image I_+ and obtain the corresponding distorted version I_-. Then the image pairs $\{I_+, I_-\}$ are used as the training data for INL.

Architecture of INL The architecture of INL is shown in Fig. 4. The whole INL is composed of k invertible noise blocks with the same architecture, which is constructed as follows. For the i^{th} block in the forward process, the inputs are y_i^h and y_i^l, and the corresponding outputs are y_{i+1}^h and y_{i+1}^l, which can be formulated as follows,

$$\begin{aligned} y_{i+1}^l &= y_i^l + \varphi_i\left(y_i^h\right) \\ y_{i+1}^h &= y_i^h \otimes \exp\left(\rho_i\left(y_{i+1}^l\right)\right) + \omega_i\left(y_{i+1}^l\right) \end{aligned} \tag{3}$$

where φ, ρ and ω can be arbitrary functions and we choose dense block in (Jing et al. 2021) which is proven to ensure good representation ability. For the first block, the inputs of which $\{y_1^h \in \mathcal{R}^{H/2\times W/2\times 9}, y_1^l \in \mathcal{R}^{H/2\times W/2\times 3}\}$ are the high frequency component and low frequency component of image $I_+ \in \mathcal{R}^{H\times W\times 3}$ after DWT respectively. Each y_i^h and y_i^l maintains the same size as y_1^h and y_1^l. After the final block, the inverse DWT will be performed on the output y_{k+1}^h and y_{k+1}^l to generate the final image I_+^d. Then the backward process will be conducted with the input I_- and output the recovered image I_-^{de}.

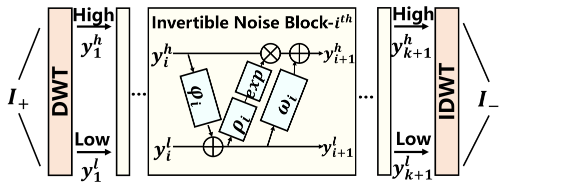


Figure 4: The backbone of the invertible noise layer, which consists of several invertible noise blocks. Each invertible noise block is combined with three modules ϕ, ρ, and ω, which are arbitrary functions.

Training Loss of INL It is required that the distorted image I_+^d should be as similar as possible to the ground-truth I_- in the forward process, and when feeding ground-truth I_- in the backward, the resulting I_-^{de} should also be similar to the original image I_+. Here, we use the term $\mathcal{L}_{dis}$ to minimize the average distance among each pair of training data, which can be formulated by:

$$\mathcal{L}_{dis}\left(\theta_d\right) = MSE\left(I_+^d, I_-\right) + MSE\left(I_-^{de}, I_+\right) \tag{4}$$

where I_+^d is equivalent to $N_{\theta_d}(I_+)$ with N_{θ_d} indicating the forward process of INL, and I_-^{de} is equivalent to $N_{\theta_d}^{-1}(I_-)$ with $N_{\theta_d}^{-1}$ indicating the backward process of INL. θ_d indicates the parameters of the INL and *MSE* indicates the mean square error. After training INL, it will be fixed as a noise layer to train the former FED.

Loss Function

The total loss function is composed of two different losses: the image loss to ensure invisibility, and the message loss to ensure robustness.

Image Loss The forward-encoding process aims to embed the watermark M into the host image I_o to generate the watermarked image I_{em}. To achieve invisibility, the watermarked image is required to be close to the host image. To achieve this goal, the image loss $\mathcal{L}_{image}$ is defined as follows:

$$\mathcal{L}_{image}\left(\theta\right) = MSE\left(I_o, I_{em}\right) \tag{5}$$

where I_{em} is equivalent to $f_\theta(I_o, M)$, with θ indicating the parameter of the proposed FED.

Message Loss The backward-decoding process aims to losslessly extract the watermark from the distorted image I_d. Toward this goal, we define the message loss $\mathcal{L}_{message}$ as follows:

$$\mathcal{L}_{message}\left(\theta\right) = MSE\left(M, M_{ex}\right) \tag{6}$$

where M_{ex} is equivalent to $f_\theta^{-1}(I_d, R^b)$ with f_θ^{-1} indicating the backward process. R^b is the all-zero matrix with the same size as M.

Total loss The total loss function $\mathcal{L}_{total}$ is a weighted sum of image loss $\mathcal{L}_{image}$, message loss $\mathcal{L}_{message}$, as follows,

$$\mathcal{L}_{total} = \lambda_1\mathcal{L}_{image} + \lambda_2\mathcal{L}_{message} \tag{7}$$

Here, λ_1 and λ_2 are weights to balance these two losses. It should be noted that we do not set any restriction on the

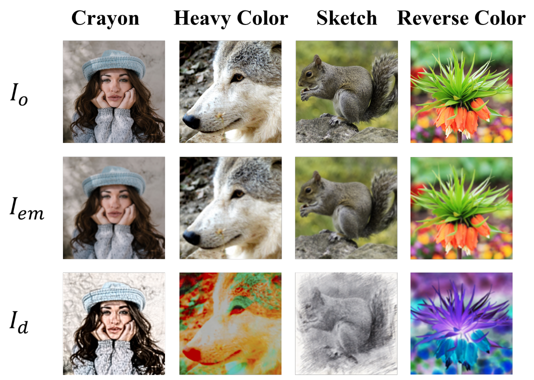


Figure 5: The tested four black-box distortions.

forward redundant information R^f and the recovered image I_{re}, because these two variables are not important in the whole process of watermarking.

Experimental Results

Experimental Settings

Datasets and Settings In this paper, the DIV2K (Agustsson and Timofte 2017) training dataset is used for training. The testing dataset we choose is the classical USC-SIPI (Viterbi 1977) image dataset. The width W and height H of the image are set to 128, and the length of the watermark message is set to 64 bits, that is, h and w are set as 8. The parameters of λ_1 and λ_2 are fixed as 1 and 10, respectively. The number of invertible neural blocks in FED n is set as 8 and the number of invertible noise blocks k is set as 8. The framework is implemented by PyTorch (Collobert, Kavukcuoglu, and Farabet 2011) and is run on one NVIDIA RTX 3090ti. [1] For parameter optimization of each network, we utilize Adam (Kingma and Ba 2015) with a learning rate of 1e-4 as default hyperparameters.

Benchmarks To verify the invisibility and robustness of the proposed method, we compare it with several state-of-the-art (SOTA) watermarking methods, including three DNN-based methods: HiDDeN (Zhu et al. 2018), TSDL (Liu et al. 2019) and MBRS (Jia, Fang, and Zhang 2021). To test the robustness, we choose 7 white-box distortions ("Cropout", "Dropout", "Gaussian Noise", "Salt&Pepper Noise", "Gaussian Blur", "Median Blur" and "JPEG Compression") and 4 black-box style transferring distortions ("Crayon", "Heavy Color", "Reverse Color", "Sketch", as shown in Fig. 5)[2]. For each distortion, we train a specific watermarking network for better illustration. It should be noted that for black-box distortion training, we use INL combined with "Gaussian noise" pre-trained model for better convergence. For a fair comparison, all the DNN-based methods are re-trained with the same dataset and same noise layer. All experiments are carried out with the image of size 128×128 and watermark of size 8×8 bits.

[1]Source code: https://github.com/QQiuyp/FIN.

[2]https://pc.meitu.com/image

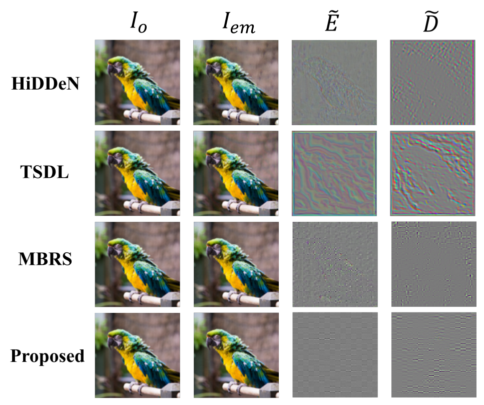


Figure 6: The coupling of different architectures.

Evaluations To measure the invisibility of the watermarked image, we adopt peak signal-to-noise ratio (PSNR) as the metric, and the larger value indicates the better invisibility. For robustness, we directly utilize the extraction bit accuracy (ACC) as the evaluation metric, and the larger ACC indicates better robustness.

Coupling Measurement

As aforementioned, the existing END architecture may suffer from the potential drawback that the encoder and the decoder are not well coupled. However, the proposed FED could effectively ensure the tight coupling of the encoder and the decoder. In this section, we will give one typical example to visually show the features of the encoder and the decoder trained with JPEG compression for different architectures, as shown in Fig. 6.

As seen in Fig. 6, the first column is the original images, the second column indicates the watermarked images, the third and fourth columns represent the normalized features that encoder embedded $\widetilde{E}$ and the normalized features that decoder needed $\widetilde{D}$. For $\widetilde{D}$, we use the gradient map that is generated by backpropagating the MSE-loss of the extracted watermark and the original watermark message on the original image to represent. We can see that for the three methods compared, $\widetilde{E}$ is not highly consistent with $\widetilde{D}$. The encoder may embed more redundant features into the host image. But for the proposed method, $\widetilde{E}$ and $\widetilde{D}$ show the same characteristics, which greatly shows that the encoder and decoder are well coupled in the proposed framework.

Invisibility and Robustness Measurement

White-box Distortions As introduced before, 7 different kinds of white-box distortions are tested in this paper. The noise layer we used for training is same as the settings of MBRS. The specific results of invisibility and robustness are shown in Table 1. "Cropout" refers to the distortion that crops a certain ratio of images and replaces the cropped region with zeros. In experiments, the test crop ratio ranges from 0.1 to 0.5, as shown in Table 1. We can see that for "Cropout" distortion, the proposed scheme maintains the best invisibility. As for robustness, the extraction accuracy is 2% higher than the compared schemes when the crop ratio

Method	Cropout (%)						Dropout (%)					
	VQ(dB)↑	r=0.1	0.2	0.3	0.4	0.5	VQ(dB) ↑	r=0.6	0.5	0.4	0.3	0.2
HiDDeN	40.62	95.6	94.7	88.8	76.9	61.7	42.59	82.7	86.7	87.1	89.6	90.2
TSDL	47.48	98.7	98.5	96.9	93.8	93.2	53.59	90.4	92.3	93.5	95.2	98.5
MBRS	48.05	99.7	99.2	97.2	90.4	83.5	57.25	90.6	92.6	94.2	94.7	96.3
Proposed	**50.61**	**100**	**99.3**	**97.9**	**96.5**	**93.8**	**58.63**	**100**	**100**	**100**	**100**	**100**
Method	S&P Noise (%)						JPEG Compression (%)					
	VQ(dB)↑	r=0.01	0.02	0.03	0.04	0.05	VQ(dB)↑	QF=50	60	70	80	90
HiDDeN	46.04	95.1	93.8	93.5	92.9	90.4	33.29	91.2	92.9	93.3	93.5	94.4
TSDL	51.16	97.3	95.6	93.5	92.7	91.5	39.39	91.5	94.0	94.2	94.4	94.7
MBRS	51.79	98.1	98.7	98.3	97.6	96.7	45.16	94.9	96.7	97.7	97.7	98.4
Proposed	**51.97**	**100**	**100**	**100**	**100**	**100**	**47.21**	**99.71**	**100**	**100**	**100**	**100**
Method	Gaussian Noise (%)				Gaussian Blur (%)				Median Blur (%)			
	VQ(dB)↑	σ=0.01	0.03	0.05	VQ(dB)↑	σ=0.5	1	2	VQ(dB)↑	w=3	5	7
HiDDeN	36.25	89.6	84.0	79.2	46.21	95.2	94.3	84.4	37.07	86.3	83.7	79.7
TSDL	39.46	92.1	88.3	82.9	45.97	99.8	98.5	93.2	38.64	99.4	97.2	95.1
MBRS	39.70	99.9	98.1	94.2	47.91	98.3	97.7	87.8	40.98	99.4	98.9	97.3
Proposed	**40.05**	**100**	**99.9**	**98.8**	**48.09**	**99.8**	**99.6**	**98.1**	**41.47**	**100**	**100**	**100**

Table 1: Benchmark comparisons on invisibility and robustness against different white-box distortions.

is 0.4. Similar results could be found in "Dropout" distortion, with 1dB higher of PSNR values, the proposed scheme achieves the best performance of robustness.

In terms of noise distortion, the proposed scheme also maintains the best performance in both invisibility and robustness. For salt&pepper noise, when facing the noise with the ratio of 0.05, the proposed scheme guarantees an extraction accuracy superiority of at least 3%. As for Gaussian noise, the robustness is more advantageous in the case of large variance. Under the premise of the highest PSNR value, the extraction accuracy when noise variance is 0.05 achieves 98.8%, which is at least 4% higher than the compared schemes.

Superior performance could also be found when facing blurring distortions. For Gaussian blurring, the proposed scheme maintains the highest extraction accuracy with the highest PSNR value. As for median blurring, when facing the strongest distortion with 7×7 filter window, the proposed scheme can still maintain an extraction accuracy of 100% which is 2% higher than the other schemes.

As for JPEG compression, we use the JPEGSS (Shin and Song 2017) noise layer for training and test the performance with quality factor from 50 to 90. Similar to the other distortions, the robustness against JPEG compression of the proposed scheme is also the best. With the advantage of more than 2dB PSNR, the proposed method guarantees an increase in extraction accuracy of nearly 5% when QF=50. This result greatly indicates the robustness against JPEG compression.

Since these methods are trained with the same dataset, the same noise layer and the same training strategy, such a significant performance improvement greatly indicates the advantage of the network structure. Utilizing invertible neural network could effectively ensure the good coupling of the encoder and the decoder, which further improves the robustness and invisibility.

Black-box Distortions In this paper, we choose four different kinds of black-box style transferring attack to evaluate the performance of the proposed scheme. For HiDDeN and MBRS, since they are not designed for black-box distortions, we directly use the combined noise to conduct the experiments. For TSDL, we finetune the decoder as the paper illustrated with the tested black-box distortions. For the compared method, the PSNR values of each distortion are the same. The PSNR values of the watermarked image with HiDDeN, MBRS and TSDL are 31.82dB, 39.32dB and 44.62dB respectively. The PSNR values of the proposed scheme with "Crayon", "Heavy Color", "Reverse Color", "Sketch" distortions are 52.04dB, 47.08dB, 50.57dB and 46.11dB respectively. From the perspective of visual quality, the proposed scheme maintains the largest PSNR value, which indicates the best invisibility of the framework. Under such visual quality, we test the black-box distortions with different strengths (light '+', medium '++', heavy '+++').

Examples of the black-box attacks are shown in Fig. 5. We can see that these attacks have significantly influenced the original appearance of the watermarked images, especially for "heavy color" and "reverse color". However, even under such a stronger distortion, the proposed scheme still maintains a high level of extraction accuracy, as shown in Table 2. For "crayon" and "reverse color" distortions, the extraction accuracy achieves more than 99% even when the strength is heavy. For all the distortions, the proposed method effectively maintains the best PSNR value and the highest extraction accuracy, which indicates the excellent robustness of the

Method	Caryon (%)			Heavy Color (%)		
	+	++	+++	+	++	+++
HiDDeN	86.0	80.2	75.4	82.9	74.8	70.0
TSDL	92.0	90.8	90.8	83.7	79.5	77.5
MBRS	98.8	98.4	98.1	99.7	98.6	95.9
Proposed	**99.8**	**99.6**	**99.5**	**99.8**	**98.6**	**96.8**
Method	Reverse Color (%)			Sketch (%)		
	+	++	+++	+	++	+++
HiDDeN	52.6	53.0	55.3	92.9	79.8	58.1
TSDL	86.4	93.5	94.8	58.7	61.0	65.2
MBRS	64.6	65.5	65.8	93.9	86.0	79.7
Proposed	**99.9**	**99.9**	**99.9**	**99.2**	**98.1**	**95.1**

Table 2: Benchmark comparisons on invisibility and robustness against different black-box distortions.

Distortion	Heavy Color		Sketch	
	PSNR(dB)	ACC(%)	PSNR(dB)	ACC(%)
G	42.17	94.4	44.62	92.5
INL	35.23	67.5	45.86	94.9
$G + INL$	**47.08**	**96.8**	**46.11**	**95.1**

Table 3: Comparisons of different noise layer components with black-box distortions.

proposed scheme.

Ablation Study

Importance of Pre-trained Model for INL For black-box distortions, we use INL and a pre-trained model of Gaussian noise for better convergence. To show the effectiveness of INL and the pre-trained model, we conduct the following experiments. We tested the performance of the network that is trained only with Gaussian noise (denoted as G), only with INL (denoted as INL) and with both (denoted as $G + INL$), respectively. Each model will be trained for 400 epochs. Then we record the corresponding PSNR and extraction accuracy as shown in Table 3.

It can be seen that only containing G may result in strong robustness, but the invisibility is not good enough. As for training only with INL, the network may not achieve good performance, we believe it is mainly because the distortion of INL is complex, it may not easy to converge to a general solution. But when combining INL with G, it gives a pre-defined direction, so fine-tuning will be easier to conduct, and the performance with $G + INL$ will be the best.

Comparison of INL with CNN-Based Noise Layer As aforementioned, we choose to simulate the black-box distortion with INN as the noise layer. However, such a process can also be achieved by traditional CNN. In this section, we will show and discuss the influence of the architecture of the noise layer simulation network. We use the traditional "ResNet" (He et al. 2016) based architecture which contains 9 "Res-Blocks" to conduct the comparison. Specifically, we train one "ResNet" to simulate the forward noising process

Distortion	Heavy Color		Sketch	
	PSNR(dB)	ACC(%)	PSNR(dB)	ACC(%)
Res^f	42.72	90.8	42.81	79.3
Res^b	42.72	82.5	42.81	76.6
INL^f	**47.08**	95.6	**46.11**	91.8
INL^b	**47.08**	**96.8**	**46.11**	**95.1**

Table 4: Comparisons of traditional CNN-based noise layer and INL.

(denoted as Res^f) and another "ResNet" to simulate the backward denoising process (denoted as Res^b). Then we use Res^f as the noise layer for training. At the extraction stage, we test the performance of the network trained with Res^f and the forward process of INL (denoted as INL^f). Since we utilize the backward process of INL (denoted as INL^b) as denoising, we also test the accuracy on the distorted images with and without denoising. The experimental results are shown in Table 4.

Compared with traditional CNN, the proposed INL will achieve the better performance in invisibility and robustness, which is 3dB larger in PSNR and 14% higher in extraction accuracy. We summarize the reason for this as the invertible nature of INNs significantly improves the generalizability of the resultant model, so training with INL will lead to better robustness. It should be noted that utilizing Res^b to denoise the distorted image will even degrade the performance. We conclude the reason as the features of Res^f and Res^b are not same. Therefore, Res^b not only denoises the distortion but also erase the watermark feature. However, when applying backward process of INL as denoising, the extraction accuracy will be slightly higher than directly feeding the distorted images. But the increase in extraction accuracy is related to the specific distortion. For "heavy color", it can improve 1.5%, but for "sketch", it can improve 3.3%. We attribute the reason to the precision of the INL simulation.

Conclusion

To address the potential drawback of the existing END-based watermarking framework where the encoder may not be well coupled with the decoder, this paper proposes a flow-based architecture which can tightly couple the encoder and the decoder with the same parameters. As a consequence, the encoded feature could keep high consistency with the feature that decoder needed, which greatly avoids the embedding of redundant features. In addition, to guarantee the robustness for black-box distortion, we also design an INN-based noise layer named INL which can simulate the black-box distortion. The forward process of INL could serve as a noise layer for training and the backward process of INL could be utilized for denoising before extraction. Based on this, the black-box distortion could be effectively guaranteed. Extensive experimental results show that our method can achieve strong robustness against not only white-box distortions but also black-box distortions, which significantly outperforms other SOTA methods both in invisibility and robustness.

Acknowledgments

This research/project is supported by the National Research Foundation, Singapore under its Strategic Capability Research Centres Funding Initiative and Huawei International. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore.

References

Agustsson, E.; and Timofte, R. 2017. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 126–135.

Ahmadi, M.; Norouzi, A.; Karimi, N.; Samavi, S.; and Emami, A. 2020. ReDMark: Framework for residual diffusion watermarking based on deep networks. *Expert Systems with Applications*, 146: 113157.

Chen, K.; Zhou, H.; Zhou, W.; Zhang, W.; and Yu, N. 2018. Defining cost functions for adaptive JPEG steganography at the microscale. *IEEE Transactions on Information Forensics and Security*, 14(4): 1052–1066.

Collobert, R.; Kavukcuoglu, K.; and Farabet, C. 2011. Torch7: A matlab-like environment for machine learning. In *BigLearn, NIPS workshop*, CONF.

Daren, H.; Jiufen, L.; Jiwu, H.; and Hongmei, L. 2001. A DWT-based image watermarking algorithm. In *IEEE International Conference on Multimedia and Expo, 2001. ICME 2001.*, 80–80. IEEE Computer Society.

Dinh, L.; Krueger, D.; and Bengio, Y. 2014. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*.

Dinh, L.; Sohl-Dickstein, J.; and Bengio, S. 2016. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*.

Filler, T.; and Fridrich, J. 2010. Gibbs construction in steganography. *IEEE Transactions on Information Forensics and Security*, 5(4): 705–720.

Gilbert, A. C.; Zhang, Y.; Lee, K.; Zhang, Y.; and Lee, H. 2017. Towards understanding the invertibility of convolutional neural networks. *arXiv preprint arXiv:1705.08664*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

Holub, V.; Fridrich, J.; and Denemark, T. 2014. Universal distortion function for steganography in an arbitrary domain. *EURASIP Journal on Information Security*, 2014(1): 1–13.

Jacobsen, J.-H.; Smeulders, A.; and Oyallon, E. 2018. i-revnet: Deep invertible networks. *arXiv preprint arXiv:1802.07088*.

Jia, Z.; Fang, H.; and Zhang, W. 2021. Mbrs: Enhancing robustness of dnn-based watermarking by mini-batch of real and simulated jpeg compression. In *Proceedings of the 29th ACM International Conference on Multimedia*, 41–49.

Jing, J.; Deng, X.; Xu, M.; Wang, J.; and Guan, Z. 2021. HiNet: deep image hiding by invertible network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4733–4742.

Kingma, D. P.; and Ba, J. 2015. 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*.

Kingma, D. P.; and Dhariwal, P. 2018. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31.

Ko, H.-J.; Huang, C.-T.; Horng, G.; and Shiuh-Jeng, W. 2020. Robust and blind image watermarking in DCT domain using inter-block coefficient correlation. *Information Sciences*, 517: 128–147.

Li, B.; Wang, M.; Huang, J.; and Li, X. 2014. A new cost function for spatial image steganography. In *2014 IEEE International Conference on Image Processing (ICIP)*, 4206–4210. IEEE.

Liu, Y.; Guo, M.; Zhang, J.; Zhu, Y.; and Xie, X. 2019. 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*, 1509–1517. ACM.

Mun, S.-M.; Nam, S.-H.; Jang, H.; Kim, D.; and Lee, H.-K. 2019. Finding robust domain from attacks: A learning framework for blind watermarking. *Neurocomputing*, 337: 191–202.

Mun, S.-M.; Nam, S.-H.; Jang, H.-U.; Kim, D.; and Lee, H.-K. 2017. A robust blind watermarking using convolutional neural network. *arXiv preprint arXiv:1704.03248*.

Shin, R.; and Song, D. 2017. Jpeg-resistant adversarial images. In *NIPS 2017 Workshop on Machine Learning and Computer Security*, volume 1, 8.

Tancik, M.; Mildenhall, B.; and Ng, R. 2019. StegaStamp: Invisible Hyperlinks in Physical Photographs. *arXiv preprint arXiv:1904.05343*.

Urvoy, M.; Goudia, D.; and Autrusseau, F. 2014. Perceptual DFT watermarking with improved detection and robustness to geometrical distortions. *IEEE Transactions on Information Forensics and Security*, 9(7): 1108–1119.

Viterbi, U. 1977. The USC-SIPI Image Database. http://sipi.usc.edu/database/. Accessed: Mar. 2023.

Wengrowski, E.; and Dana, K. 2019. 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*, 1515–1524. Computer Vision Foundation / IEEE.

Xu, Y.; Mou, C.; Hu, Y.; Xie, J.; and Zhang, J. 2022. Robust Invertible Image Steganography. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7875–7884.

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