2024 IEEE International Conference on Multimedia and Expo (ICME) | 979-8-3503-9015-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICME57554.2024.10687737

Enhanced Blind Watermarking Against Black-Box Noise: Leveraging CIN Framework

Rui Ma
School of Computer Science
Peking University
Beijing, China
rui_m@stu.pku.edu.cn

Mengxi Guo
Bytedance Inc.
Shanghai, China
nicolasguo@pku.edu.cn

Peidong Jia
School of Computer Science
Peking University
Beijing, China
peidongjia@stu.pku.edu.cn

Chenxuan Li
School of Computer Science
Peking University
Beijing, China
2301210279@stu.pku.edu.cn

Yi Hou
School of Computer Science
Peking University
Beijing, China
yihou@pku.edu.cn

Yuan Li
School of Computer Science
Peking University
Beijing, China
yuanli@pku.edu.cn

Xiaodong Xie
School of Computer Science
Peking University
Beijing, China
donxie@pku.edu.cn

Shanghang Zhang✉
School of Computer Science
Peking University
Beijing, China
shanghang@pku.edu.cn

***Abstract*—Blind watermarking is a technology for image copyright protection and digital fingerprinting. However, the introduction of non-differentiable noise makes it challenging to be trained end-to-end for black-box scenes. The phased training technique is used for coping with black-box noise, but it limits the performance since the encoder and decoder cannot be end-to-end optimized. This work proposes a blind watermarking framework CIN+ based on the CIN to address black-box noise. Combining the structural characteristics of an Invertible Neural Network (INN) with the two-stage strategy allows joint updates of the encoder and decoder when encountering non-differentiable noise. We utilize Noise and Gradient Propagation Gate (NGPG) modules to perform a batch-level optimization akin to the two-stage approach, allowing encoder parameters to remain unlocked, thereby enhancing the model's ability to resist black-box attacks. Additionally, a Pre-Extraction Module (PEM) is introduced to simplify the complexity and usability of CIN. Our experimental results reveal that CIN+ achieves a new state-of-the-art performance.**

***Index Terms*—Blind watermarking, Invertible neural network, Real scenes, Black-box noise**

I. Introduction

As image editing technologies continue to advance, the proliferation of novel image processing techniques has posed a greater challenge to classic digital watermarking algorithms. When images are tampered using black-box methods such as *style transfer* and *image filtering*, it becomes difficult to develop a robust algorithm based on typical watermarking techniques. Therefore, adversarial training emerges as a viable solution. Fig.1(a) illustrates two existing deep learning-based frameworks to address black-box noise. One approach is the two-stage training framework, such as [1], [2], where the first stage trains both the encoder and decoder in a noise-free scenario, and then fixes the encoder parameters to only train the decoder after the introduction of black-box noise. Another option is to replace the black-box noise with a simulator, which can be an algorithmic approximation [3], [4] or a neural network simulation [5]. The two-stage framework froze the pre-trained encoder and only updated the decoder. This strategy cannot find the global optimum in the same way as an end-to-end training framework. Utilizing an emulator instead of a non-differentiable quantization model is a common approach in anti-compression watermarking, as it allows for the use of a fixed and conveniently simulated framework. However, the effectiveness and overhead of using a deep learning network as a simulator should be carefully considered. The CIN method [6] based on Invertible Neural Networks (INN) has demonstrated remarkable excellence in watermarking tasks. Further, we have observed that the unique structural characteristics of invertible modules proficiently evade the issue of falling into local optima, a common pitfall when employing a two-stage method to tackle black-box noise.

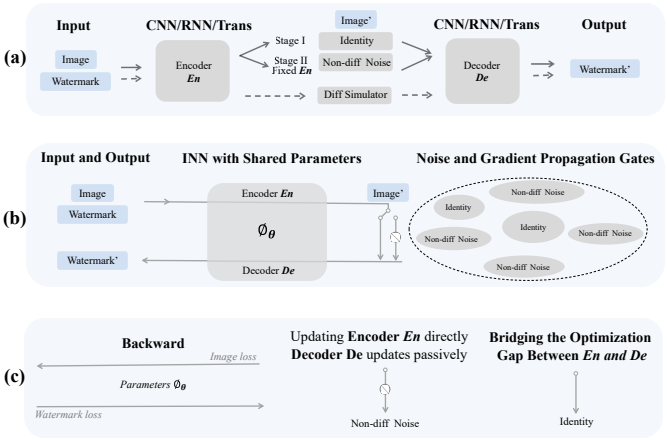

Fig. 1. The top panels show two classic blind watermarking frameworks against black-box noise; the middle is our proposed framework; and the bottom is our method's optimization strategy.

✉ is the corresponding author. This work is funded by the National Science and Technology Major Project of China (No. 2022ZD0117801).

We propose CIN+, an extended method based on CIN framework, aimed at addressing the challenge of end-to-end optimization for black-box noise. In Fig.1(b), while the shared parameters of INN enable training encoder and decoder with non-differentiable noise, a direct link between the two is absent. This detachment coupled with the constraints imposed by the optimizing function, results in an enhanced performance of the decoder, but at the expense of the encoder's performance.

Therefore, we referred to the ideology of a two-stage training approach [1]. Due to INN's parameter-sharing feature, fixing the encoder during noise-aware training was unfeasible. So, we merged noise-free and noise-aware aspects into a noise and gradient propagation gates (NGPG) module. By randomly sampling noise-free and noise-aware layers at the batch level, this module reinforces the model against black-box attacks. This can be seen as a batch-level two-stage method where the encoder parameters aren't fixed. Despite potential uncertainties in the encoder's optimization direction when sampling non-diff noise, introducing noise-free elements guides its path. Additionally, the sampling ratio between the non-diff noise and the *Identity* can be adjusted to balance robustness and imperceptibility. Thus, Fig.1(c), even in CIN+ with black-box attacks or non-differentiable noise, the optimization function updates both encoder and decoder parameters simultaneously.

The research conducted in CIN indicates that the invertible module demonstrates strong robustness against additive noise, while the non-invertible module exhibits greater resilience against compression-based attacks. Specifically, the non-invertible module compensates for the limitations of the invertible module under severe attacks on the watermarked image, owing to the network's symmetric and parameter-sharing structure. To enhance the overall performance of the model, CIN introduces additional discriminators. Training corresponding discriminators for different severe attacks not only increases computational overhead but also elevates the model's complexity and instability. Hence, we propose a Pre-Extraction Module (PEM) to replace the NIAN and NSM modules in CIN. This adaptation ensures robustness against additive noise via residual connections and resilience in compression-based attack scenarios through an Attention-based block.

The experiments indicate that in black-box noise scenarios—*Style Transfer*, *Image Filtering*, and *Superimpose*—the robustness of CIN+ has respectively improved by 27.98%, 20.12%, and 42.8% compared to CIN. Our contributions can be summarized as follows:

- We propose a blind watermarking framework for black-box attacks based on CIN, capable of end-to-end optimization of model parameters when introducing non-differentiable noise.
- We introduce an NGPG module that combines the structural characteristics of INN and a two-stage traning approach, enabling the optimization of encoder parameters even in a noise-aware stage. Additionally, we present a PEM module to simplify the complexity and usability of the CIN model.
- Extensive experimental results demonstrate that our framework achieves state-of-the-art performance in blind watermarking against black-box attacks.

II. RELEATED WORKS

INN based on normalized flow [7], [8] has seen success in image generation tasks. Recent works, such as [9]–[12], have demonstrated the potential of INN in delivering cutting-edge performance in tasks related to image steganography and information embedding. The INN framework for blind watermarking was initially introduced by CIN [6]. However, it tackles CIN's performance decline against strong attacks through an added decoder and noise discriminator. This elevates model complexity and training difficulty while tying the algorithm's robustness to the discriminator's accuracy. Moreover, the CIN framework lack the capability to withstand black-box attacks.

Compared to constructing an approximate substitute module for black-box attacks [3]–[5], A two-stage approach as presented in [1] constitutes a more efficient strategy to combat black-box attacks. However, this framework's performance is constrained as it fixes the parameters of the encoder in stage II (noise-aware). Hence, we've developed CIN+: a blind watermarking framework based on the CIN architecture. It optimizes end-to-end and effectively handles non-differentiable noise by combining INN-shared parameters and a two-stage framework to combat black-box noise.

III. PROPOSED METHOD

Our framework, comprising the Embedding and Extraction Module (EEM), Invertible Module (IM), Fusion and Alignment Module (FAM), Noise and Gradient propagation Gates (NGPG), and Pre-extraction Module (PEM) modules, handles dimension adjustments, watermark imperceptibility enhancement, gradient update management, and noise-related extraction concerns.

Embedding and Extraction Module We simplified the embedding and extraction process of the DEM module in CIN by employing a combination of *Linear* and *PixelShuffle* methods, which replaces the previous method of embedding separately through RGB channels.

$$\zeta_{eem} = \mathbf{\Gamma}_{haar}(\Theta_{shuffle}(\Theta_{conv}(\Theta_{fc}(W_m)))) \tag{1}$$

$$\zeta_{eem}^{-1} = \Theta_{fc}(\Theta_{mean}(\Gamma_{haar}^{-1}(\mathbf{O}(u_{wm}^{0})))) \tag{2}$$

Where, $\Theta_{fc}(\cdot)$ is the fully-connected layer that utilized to map the watermark length to a latent size of 256, $\Theta_{conv}(\cdot)$ is the 2D convolution, $\Theta_{shuffle}(\cdot)$ is the pixel shuffling, $\Theta_{mean}(\cdot)$ and $\Gamma_{haar}(\cdot)$ are the average and the Haar transform, respectively. The ζ_{eem} and ζ_{eem}^{-1} represent the process of embedding and extraction, respectively.

Invertible Module Unlike the CIN [6] method, our approach involves the addition of the ActN layer and invertible 1×1 convolution layer within the IM. IM consists of sixteen submodules, each containing an Invertible 1×1 Convolution

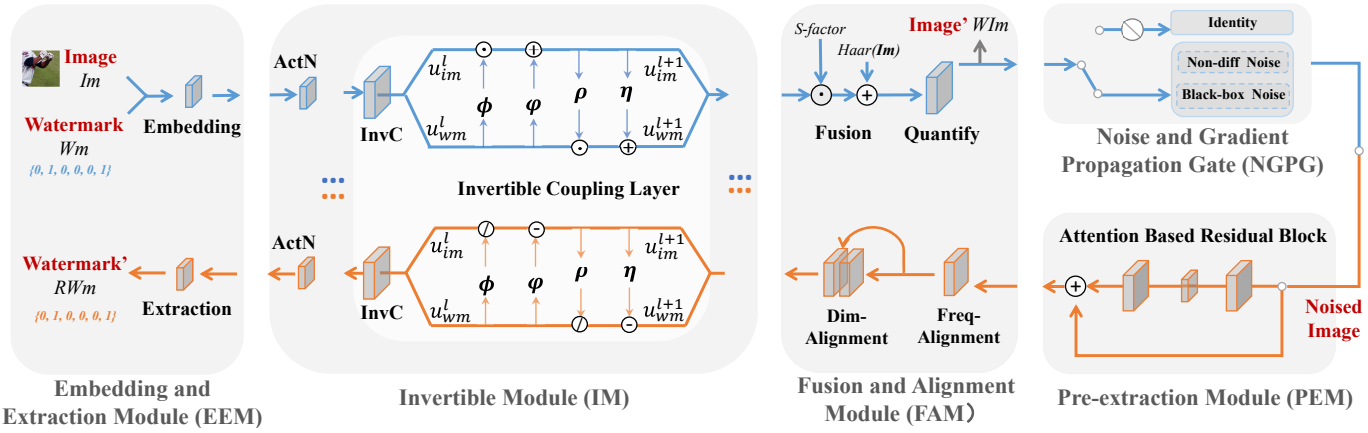

Fig. 2. Overall architecture. EEM is capable of diffusing and extracting watermarks. The diffused watermark is transformed by IM into an imperceptible format corresponding to the input image. The transformed watermarking and the input image are integrated via FAM. NGPG determines gradient pathways by sampling the different layers. When noise is introduced, PEM can still allow the IM to meet distribution symmetry.

Layer accompanied by an Affine Layer. The invertible module is formulated as:

$$\mathbf{O}(\mathbf{u}_{im}^1, \mathbf{u}_{wm}^1) = \mathbf{ActN}(\mathbf{u}_{im}^0, \mathbf{u}_{wm}^0) \tag{3}$$

$$\mathbf{O}(\mathbf{u}_{im}^l, \mathbf{u}_{wm}^l) = \mathbf{InvConv}(\mathbf{u}_{im}^l, \mathbf{u}_{wm}^l), l > 0 \tag{4}$$

$$\mathbf{O}(\mathbf{u}_{im}^{l+1}) = \mathbf{O}(\mathbf{u}_{im}^l) \odot \exp\left(\phi(\mathbf{O}(\mathbf{u}_{wm}^l))\right) + \varphi(\mathbf{O}(\mathbf{u}_{wm}^l)) \tag{5}$$

$$\mathbf{O}(\mathbf{u}_{wm}^{l+1}) = \mathbf{O}(\mathbf{u}_{wm}^l) \odot \exp\left(\rho\left(\mathbf{O}(\mathbf{u}_{im}^{l+1})\right)\right) + \eta\left(\mathbf{O}(\mathbf{u}_{im}^{l+1})\right) \tag{6}$$

Where, $\mathbf{O}(\cdot)$ is the l^{th} layer's tensor, $\mathbf{u}_{wm}^l$ and $\mathbf{u}_{wm}^l$ are the l^{th} watermarking and image, ActN is the ActNorm layer, InvConv is the invertible 1×1 convolution layer, $\exp(\cdot)$ is exponential operator, $\phi(\cdot)$, $\varphi(\cdot)$, $\rho(\cdot)$, and $\eta(\cdot)$ are DenseNet, $\odot$ is the Hadamard product. The extraction of the watermark can be achieved by reversing the affine transformation, as defined in the equations above.

Fusion and Alignment Module Haar transformation and tensor replication are employed separately to achieve frequency alignment and dimension alignment of the input for the IM module. The equations are provided below:

$$WI_m = \Gamma_{haar}^{-1}(\mathbf{O}(u_{wm}^l) \times S + \mathbf{O}(u_{im}^0))) \tag{7}$$

$$\mathbf{O}(\mathbf{u}_{im}^l, \mathbf{u}_{wm}^l) = \Theta_{copy}(\Gamma_{haar}(\mathbf{O}(u_{im}^{pem}))) \tag{8}$$

where $\Gamma_{haar}^{-1}(\cdot)$ is inverse Haar transform, WI_m is watermarked image, u_{wm}^{pem} is the output of the PEM. S is strength factor.

Noise and Gradient propagation Gates During training, we balance robustness and imperceptibility by setting the sampling ratio between black-box noise and the 'Identity' layer at 2:1, a decision made based on empirical observation. The rationale behind this is that the gradient of the decoder exhibits a higher value with noise compared to the 'Identity' layer without noise.

$$u_{im}^{noise} = \{\mathbf{Sp}(N_{iden}, N_{n-diff}, N_{n-diff}) | N_{sp}(WI_m)\} \tag{9}$$

Where, $\mathbf{Sp}(\cdot)$ and $N_{sp}(\cdot)$ denote randomly sampling a layer and applying the layer, respectively. N_{iden} and N_{n-diff} are the *Identity* layer and non-differentiable noise, respectively.

Pre-extraction Module PEM serves as an effective preprocessing method that mitigates noise during watermark extraction, significantly enhancing experimental outcomes. The network architecture for the PEM module is depicted in Figure 1 of supplementary material. The module consists of the ConvBNRelu layer, attention layer, and residual connectivity. Further details can be found in the supplementary.

$$u_{im}^{pem} = \Phi(\mathbf{O}(u_{im}^{noise})) + \mathbf{O}(u_{im}^{noise}) \tag{10}$$

Where, u_{im}^{noise} is noised image, $\Phi(\cdot)$ denotes attention based networks.

Loss Functions Employing L_2 loss to guide the watermarked image WI_m to be visually alike to the reference image I_m.

$$\mathcal{L}_{WIm} = ||I_m - f_{CIN+}(\theta, I_m, W_m))||_2^2 \tag{11}$$

Calculate the L_2 distance for each pair of input watermark W_m and the extracted watermark RW_m.

$$\mathcal{L}_{RWm} = ||Wm - f_{CIN+}^{-1}(\theta, N_{sp}(WI_m))||_2^2 \tag{12}$$

where, $f_{CIN+}(\cdot)$ and $f_{CIN+}^{-1}(\cdot)$ represent the watermark embedding and extraction process, respectively.

The compact loss $\mathcal{L}_{total}$ with the corresponding weight λ_{WIm}, λ_{RWm} is

$$\mathcal{L}_{total} = \lambda_{WIm}\mathcal{L}_{WIm} + \lambda_{RWm}\mathcal{L}_{RWm} \tag{13}$$

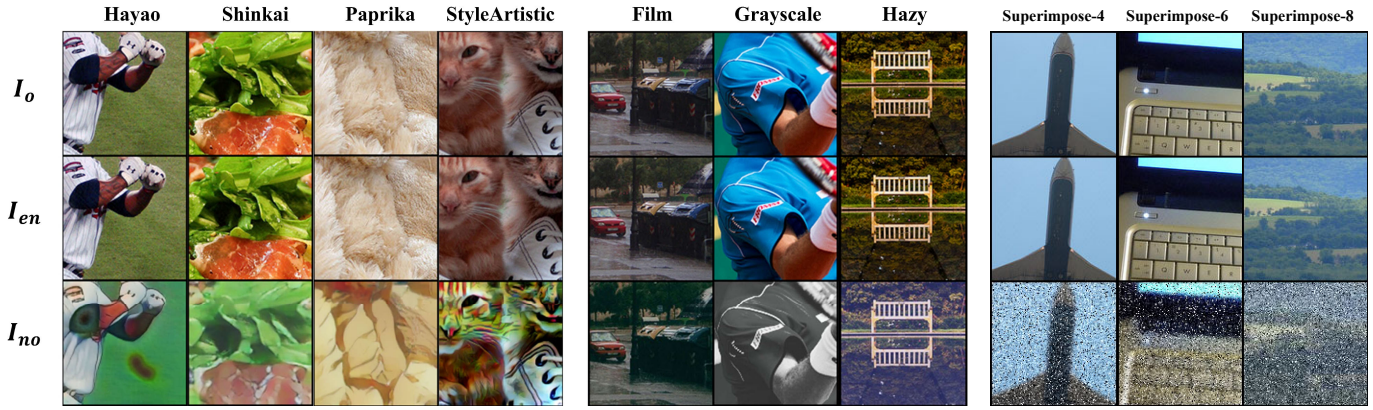

Fig. 3. Top to bottom, input images, watermarked images, and noised images are listed. The experimental results for various noises are given from left to right. On the left panel is a black-box noise with *Style Transfer*, on the middle panel is an *Image Filtering*, while on the right panel is an *Superimpose*.

IV. EXPERIMENTS

A. Experimental Setup

Datasets We use the COCO2017 [13] dataset for training and testing. The training set consists of 20,000 images, while the test set consists of 100 images for the Transport Compression for real applications experiment and 1,000 images for the other experiments. In addition, test experiments were also conducted on the DIV2K [14] dataset. In all experiments, the image size is scaled to 128×128.

Metrics The evaluation metrics for imperceptibility of the watermarked images are the Peak Signal-to-Noise Ratio (PSNR) and Structure Similarity Index Measure (SSIM) in the RGB color space. The robustness of the watermark is determined by the accuracy (ACC) of the extracted watermark. In addition, we also conducted experiments using the visual evaluation metrics LPIPS [15].

Training The framework is implemented in PyTorch and executed on an NVIDIA RTX 3080. The black-box noise experiments are trained for 10 epochs, while the non-differentiable compressed noise studies are trained for 20 epochs. The Adam optimizer is used with a learning rate of 0.0001 and a loss decay technique is applied. The loss weights λ_{WIm} and λ_{RWm} are set to 1 and 1 by default, but may be adjusted based on the strength of the applied noise.

Noise To ensure the reproducibility of the experiment, specific parameter settings have been provided. For *Style Transfer* and *Image Filtering*, the values of parameters remain the default settings of the API. For *JPEG*, the quality factor Q is set to 50. *JPEG2000* is configured in the rates mode with quality parameters Q set at [0.5, 0.75]. *WebP* has its quality factor Q set to 50, *H.264* has its quality parameter *qp* set at 30, and *MBT2018* has its quality parameter q set at 5. For *Superimpose*, we provide detailed instructions in the supplementary material.

B. Black-box Noise Experiment

The black box noise observed in the experiment can be divided into three categories based on the application scenarios: *Style Transfer* using deep learning models, *Image Filtering* using image processing algorithms or neural networks, and *Superimpose* using common image editing methods. It is essential to note that we do not have knowledge of the exact implementation of noise during the training process, and both noises that participate in this stage are in the form of non-differentiable terms.

Style Transfer The *Style Transfer* is implemented using the Baidu PaddleHub API[1], with transfer types used in the experiment including anime (Hayao, Shinkai, Paprika) and artistic (StyleArtistic). The experiment results can be seen in the left panel of Fig.3. The noise image I_{no} shows that the stylization attack approach significantly reduces most of the high-frequency texture details, while still maintaining some of the image structure.

Image Filtering The *Image Filtering* is implemented using TencentCloud SDK[2], with noise used in the experiment including *Film*, *Gray*, *Old tune*, *Fashion*, *Sunset*, and *Hazy*. The results of the experiment are represented in the right panel of Fig.3. These filters are widely used in a number of photo editing software programs such as Meituxiuxiu, Snapseed, and Gallery. Generally, they manipulate the hue and color of images, although the implementation and parameters of these filters can differ among software packages.

Superimpose The combination attack of multiple noises is a common and easily implementable image editing method. For a superimpose scenario, the types, intensities, and quantities of noise inside the black box are unknown. It's a black-box image processing module that can be called via API or offline. This also leads to the inability to backpropagate gradients when incorporating it into an end-to-end network for joint training. The black box outputs an image subjected to N types of noise attacks (parameters 4, 6, 8 for *Superimpose* in Table I). The details in supplements.

Intuitively, the stylization model has a greater impact and

[1]https://www.paddlepaddle.org.cn/
[2]https://cloud.tencent.com/

TABLE I

EXPERIMENTAL RESULTS AGAINST BLACK-BOX NOISE. THE MESSAGE LENGTH L_{msg} IS 64. THE FOLLOWING EXPERIMENT DIVIDES THE BLACK-BOX NOISE INTO TWO CATEGORIES: *Style Transfer*, WHICH IS BASED ON AN UNIDENTIFIED DEEP LEARNING MODEL, AND *Image Filtering*, WHICH IS BASED ON A DEEP NETWORK OR IMAGE PROCESSING METHOD. PSNR_{w2c} AND PSNR_{w2n} INDICATE THE PSNR VALUES BETWEEN THE WATERMARKED IMAGE AND THE ORIGINAL INPUT IMAGE, AND THE WATERMARKED IMAGE AND THE NOISED IMAGE, RESPECTIVELY.

Block-box Noise	**Style Transfer**				**Image Filtering**						**Superimpose**		
	hayao	*shinkai*	*paprika*	*StylArt*	*Film*	*Gray*	*Old tune*	*Fashion*	*Sunset*	*Hazy*	*4 types*	*6 types*	*8 types*
PSNR_{w2n} (dB) ↑	15.23	16.62	16.54	12.74	21.68	14.73	22.13	23.32	20.06	14.87	18.05	16.25	13.27
SSIM_{w2c} ↑	0.893	0.920	0.896	0.930	0.991	0.990	0.994	0.993	0.991	0.994	0.941	0.922	0.890
PSNR_{w2c} (dB) ↑	32.34	33.37	32.71	35.35	43.26	42.92	45.30	44.69	43.55	43.58	38.21	36.92	35.14
ACC (%) ↑	**93.06**	**92.04**	**85.30**	**96.15**	**99.14**	**97.44**	**98.98**	**97.02**	**99.68**	**98.01**	**98.29**	**97.51**	**97.02**

change on the image, while the *Image Filtering* model preserves more of the image's texture. As shown in Table. I, the PSNR of the stylization model is lower than the *Image Filtering* model, indicating that the intensity of the noise in the former is higher. At the same time, even if the ACC of the *Image Filtering* model is above 97%, then the PSNR can be maintained at around 42 dB. To achieve better watermark robustness, the PSNR must be kept at around 33 dB for the stylization model.

TABLE II

THE COMPARATIVE RESULTS REGARDING BLACK-BOX NOISE, PRESENTING THE AVERAGE ACCURACY AND IMPERCEPTIBILITY (ACC%↑/PSNR_{w2c} *dB*↑) UNDER THREE DISTINCT TYPES OF NOISE.

Noise	**Style Transfer**	**Image Filtering**	**Superimpose**
Liu [1]	88.74%/33.20*dB*	94.10/43.51	73.29/36.04
Fang [2]	89.57%/33.62*dB*	96.31/43.73	94.54/36.28
CIN [6]	63.65%/33.54*dB*	78.23/43.61	54.80/36.52
CIN+	91.63%/33.44*dB*	98.37/43.88	97.60/36.75

TABLE III

THE VISUAL QUALITY OF THE WATERMARKED IMAGE (ACC >98%). Q REPRESENTS THE QUALITY FACTOR OF COMPRESSION. EACH DATASET EXPERIMENT USES 1K TEST IMAGES. THE FINAL RESULT IS THE AVERAGE OF THREE RUNS.

Dataset	**Compression Noise**	**SSIM↑**	**LPIPS ↑**
COCO2017	Jpeg (Q=50)	0.9702	0.0113
DIV2K	Jpeg (Q=50)	0.9643	0.0154

In Table III, the results of the compression-resistant experiments on the COCO2017 and DIV2K datasets are presented. Additionally, we have included the experimental outcomes for the visual assessment metric SSIM and LPIPS.

C. Non-differentiable Compression Experiment

Compression Algorithms To evaluate the robustness of watermarking, studies are conducted using the following five different compression techniques: *Jpeg*, *Jpeg2000* [16], *Webp* [17], *H264* [18], and *mbt2018* [19]. As shown in Table. IV, the PSNR is uniformly set to 33±0.5 dB to compare the

TABLE IV

EXPERIMENT WITH VARIOUS NON-DIFFERENTIABLE COMPRESSION TECHNIQUES. THE WATERMARK LENGTH L_{msg} IS 64. THE WATERMARKED IMAGE'S PSNR IS CONSISTENTLY SET TO 33±0.5 DB.

Non-diff Noise	***Jpeg***	***Jp2***	***Webp***	***H264***	***mbt2018***
PSNR_{w2n} (dB) ↑	27.18	30.33	28.66	30.42	24.05
SSIM_{w2c} ↑	0.931	0.907	0.922	0.902	0.938
ACC (%) ↑	**99.92**	**98.88**	**99.84**	**98.68**	**99.91**

TABLE V

COMPARATIVE EXPERIMENT: THE ACCURACY(ACC% ↑) OF THE WATERMARKING AGAINST COMPRESSION WHEN TRANSMITTED IN SOCIAL SOFTWARE. THE PSNR IS CONTINUALLY ADJUSTED TO ROUGHLY 32.5±0.5 DB BY VARYING THE STRENGTH FACTOR S. THE WATERMARK IS 64 BITS LONG. RED REPRESENTS THE TOP ACCURACY VALUE, BLUE TAKES THE SECOND PLACE.

Applications	***QQ***	***Wechat***	***Twitter***	***Facebook***	***Instagram***
Liu [1]	81.7	80.3	84.7	84.8	80.3
Kang [20]	85.4	85.2	85.7	85.7	85.7
Ma [21]	92.7	57.6	53.5	48.8	89.1
Fang [2]	91.4	91.6	93.6	93.5	92.8
CIN [6]	98.37	96.40	97.37	99.37	98.89
CIN+	99.98	99.98	99.37	99.98	99.85

robustness of different compression algorithms. The experiments showed that even without a differentiable compression simulation module (using approximate differentiability or deep network simulation), our framework can achieve a watermark accuracy of higher than 98% under various compression methods.

Transport Compression for Real Applications We conduct comparative studies in two groups utilizing our framework on five popular social software, as shown in Table. V. For fairness in the comparison experiments, the PSNR_{w2c} is set to 32.5±0.5 dB, and the watermark is 64 bits in length. Part of the experimental results in Table. V are cited from [2]. CIN+ is trained in conjunction with actual compression algorithms like WebP and H.264, empowering the watermark to more effectively resist the compression losses common in popular social media platforms. Meanwhile, CIN solely relies on differentiable simulation-based *JPEG* for joint training,

TABLE VI
EXPERIMENTAL RESULTS OF EMBEDDING VARIOUS WATERMARK LENGTHS L_{msg}. THE *Jpeg* COMPRESSION IS EMPLOYED AS THE TESTED NOISE IN THE EXPERIMENTS. THE WATERMARKED IMAGE'S PSNR IS CONSISTENTLY SET TO 36±0.5 DB.

L_{msg} (*Bits*)	**30**	**64**	**128**	**192**	**256**
PSNR_{w2n} (dB) ↑	27.59	27.53	27.55	27.43	27.46
SSIM_{w2c} ↑	0.938	0.957	0.964	0.957	0.956
ACC (%) ↑	**99.97**	**99.54**	**99.66**	**99.18**	**98.07**

TABLE VII
ABLATION EXPERIMENTS. THE AVERAGE ACC AND PSNR UTILIZING *Image Filtering* AND *Jpeg* NOISE, RESPECTIVELY, ARE DISPLAYED. THE BASELINE DENOTES THE FRAMEWORK'S USE OF SIMPLY EEM, IM, AND FAM. THE L_{msg} IS 64.

Modules			ACC (%) ↑		PNSR (dB) ↑	
Baseline	NGPG	PEM	*Im Filtering*	*Jpeg*	*Im Filtering*	*Jpeg*
✓			76.99	63.18	34.01	30.13
✓	✓		98.30	98.61	38.15	37.96
✓	✓	✓	98.56	98.66	43.67	38.37

resulting in limitations for the watermark encoder and decoder to fully learn real compression distortion behaviors. Table. V indicates that our framework is robust in various real-world application scenarios.

D. Ablation Study

Various Watermarking Lengths Table. VI gives the experimental results for watermarking lengths L_{msg}={30, 64, 128, 192, 256}. The PSNR is set to 36±0.5 dB, and the experimental results show our framework has excellent robustness under various watermark lengths.

Ablation Modules We conduct ablation studies in real-world scenarios, using filtering and compression noise, and training of the same number of epochs. As shown in Table VII, the NGPG and PEM modules are critical for the effectiveness of our CIN+ framework.

V. CONCLUSION

We introduce a novel framework designed for blind watermarking in the presence of black-box noise. Building upon the CIN framework, our approach leverages the parameter-sharing feature of INN and employs a two-stage strategy to combat black-box noise, enabling simultaneous updates of encoder and decoder parameters even in the presence of non-differentiable noise. Additionally, we introduce the NGPG to bridge the gaps between watermark encoder and decoder. Furthermore, we have developed a PEM module to simplify the complexity arising from using discriminators and two decoders within CIN. Through comprehensive experiments, our approach achieves new state-of-the-art performance.

REFERENCES

[1] Yang Liu, Mengxi Guo, Jian Zhang, Yuesheng Zhu, and Xiaodong Xie, "A novel two-stage separable deep learning framework for practical blind watermarking," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 1509–1517.

[2] Han Fang, Zhaoyang Jia, Hang Zhou, Zehua Ma, and Weiming Zhang, "Encoded feature enhancement in watermarking network for distortion in real scenes," *IEEE Transactions on Multimedia*, 2022.

[3] J. Zhu, R. Kaplan, J. Johnson, and L. Fei-Fei, "Hidden: Hiding data with deep networks," 2018.

[4] Zhaoyang Jia, Han Fang, and Weiming Zhang, "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*, 2021, pp. 41–49.

[5] Stk Jan, J. Messou, Y. C. Lin, J. B. Huang, and G. Wang, "Connecting the digital and physical world: Improving the robustness of adversarial attacks," 2019, pp. 962–969.

[6] Rui Ma, Mengxi Guo, Yi Hou, Fan Yang, Yuan Li, Huizhu Jia, and Xiaodong Xie, "Towards blind watermarking: Combining invertible and non-invertible mechanisms," in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 1532–1542.

[7] Laurent Dinh, David Krueger, and Yoshua Bengio, "Nice: Non-linear independent components estimation," *arXiv preprint arXiv:1410.8516*, 2014.

[8] Durk P Kingma and Prafulla Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions," *Advances in neural information processing systems*, vol. 31, 2018.

[9] Mengxi Guo, Shijie Zhao, Yue Li, Junlin Li, Li Zhang, and Yue Wang, "Invertible single image rescaling via steganography," in *2022 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2022, pp. 1–6.

[10] Shao-Ping Lu, Rong Wang, Tao Zhong, and Paul L Rosin, "Large-capacity image steganography based on invertible neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10816–10825.

[11] Youmin Xu, Chong Mou, Yujie Hu, Jingfen Xie, and Jian Zhang, "Robust invertible image steganography," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 7875–7884.

[12] Chong Mou, Youmin Xu, Jiechong Song, Chen Zhao, Bernard Ghanem, and Jian Zhang, "Large-capacity and flexible video steganography via invertible neural network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 22606–22615.

[13] David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I*, vol. 8689, Springer, 2014.

[14] Eirikur Agustsson and Radu Timofte, "Ntire 2017 challenge on single image super-resolution: Dataset and study," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 126–135.

[15] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 586–595.

[16] Majid Rabbani, "Book review: Jpeg2000: Image compression fundamentals, standards and practice," 2002.

[17] Google, "Web picture format," https://chromium.googlesource.com/webm/libweb, 2010.

[18] Loren Merritt and Rahul Vanam, "x264: A high performance h. 264/avc encoder," *online] http://neuron2. net/library/avc/overview_x264_v8_5. pdf*, 2006.

[19] David Minnen, Johannes Ballé, and George D Toderici, "Joint autoregressive and hierarchical priors for learned image compression," *Advances in neural information processing systems*, vol. 31, 2018.

[20] Xiangui Kang, Jiwu Huang, and Wenjun Zeng, "Efficient general print-scanning resilient data hiding based on uniform log-polar mapping," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 1, pp. 1–12, 2010.

[21] Zehua Ma, Weiming Zhang, Han Fang, Xiaoyi Dong, Linfeng Geng, and Nenghai Yu, "Local geometric distortions resilient watermarking scheme based on symmetry," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 12, pp. 4826–4839, 2021.