Learning Robust Image Watermarking with Lossless Cover Recovery

Jiale Chen[1,2,3], Wei Wang[1,2,3,*], Chongyang Shi[1], Li Dong[4,5,*], Xiping Hu[1,2,3,*]
[1]Beijing Institute of Technology, [2]Shenzhen MSU-BIT University,
[3]Guangdong-Hong Kong-Macao Joint Laboratory for Emotional Intelligence and Pervasive Computing,
[4]Ningbo University, [5]Guangdong Provincial Key Laboratory of Intelligent Information Processing

Abstract

*Watermarking as a traceable authentication technology has been widely applied in image copyright protection. However, most existing watermarking methods embed watermarks by adding irremovable perturbations to the cover image, causing permanent distortion. To address this issue, we propose a novel watermarking approach termed **C**over-**R**ecoverable Water**Mark** (CRMark). CRMark can losslessly recover the cover image and watermark in lossless channels and enables robust watermark extraction in lossy channels. CRMark leverages an integer Invertible Watermarking Network (iIWN) to achieve a lossless invertible mapping between the cover-image-watermark pair and the stego image. During the training phase, CRMark employs an encoder-noise-layer-decoder architecture to enhance its robustness against distortions. In the inference phase, CRMark first maps the cover-image-watermark pair into an overflowed stego image and a latent variable. Subsequently, the overflowed pixels and the latent variable are losslessly compressed into an auxiliary bitstream, which is then embedded into the clipped stego image using reversible data hiding. During extraction, in lossy channels, the noised stego image can directly undergo inverse mapping via iIWN to extract the watermark. In lossless channels, the latent variable and overflowed stego image are first recovered using reversible data hiding, followed by watermark extraction through iIWN. Extensive experimental results demonstrate that CRMark can be perfectly recovered in lossless channels while remaining robust to common distortions. Code is available at* `https://github.com/chenoly/CRMark`.

1. Introduction

With the rapid growth of digital content and internet technology, copyright protection issues have become increasingly significant. Digital watermarking embeds identifying information into media to verify authenticity and ownership, supporting secure digital content transmission [28].

*Corresponding author.

Most robust image watermarking methods introduce non-erasable perturbations to the cover image, resulting in irreversible damage. To address this, robust reversible watermarking (RRW) was born. Unlike watermark removal methods [41], RRW can perfectly recover the cover image in lossless channels and robustly watermark extraction in lossy channels. This enables RRW to detect arbitrary attacks, making it especially suitable for applications requiring both data integrity and copyright verification [31]. Currently, most RRW methods predominantly adopt a two-stage embedding strategy. For the first stage, the watermark bits are embedded into a cover image using a traditional robust watermarking method. For the second stage, to ensure reversibility, the auxiliary bitstream required for recovery of the cover image is embedded into the stego image using reversible data hiding [26]. This ensures the functional integration between robustness and reversibility. However, existing two-stage RRW methods often demand complex designs and significant computational time to balance robustness, reversibility, and capacity effectively.

In recent years, invertible neural networks (INNs) have emerged as a pivotal approach in robust image watermarking [11] and steganography [25], leveraging their unique capacity for invertible mapping between the cover and stego images, showing them highly effective for data hiding tasks. However, most INN-based methods are based on real-valued flows [9]. This leads to a problem: despite the cover image being discrete data. the stego image from INNs is real-valued and requires quantization to match discrete pixel values for storage. This quantization process inevitably causes information loss that hampers the perfect recovery of the original cover image upon inverse mapping. Thus, quantization-induced irreversibility limits the practical utility of INNs in robust reversible watermarking.

To tackle this challenge, a key question arises: *Can invertible neural networks be applied to achieve reversible watermarking?* The answer is affirmative. To establish an invertible mapping between the cover-image-watermark pair and stego image, we propose a novel two-stage watermarking method termed **C**over-**R**ecoverable Water**Mark** (CRMark), as shown in Fig. 1. We develop an integer

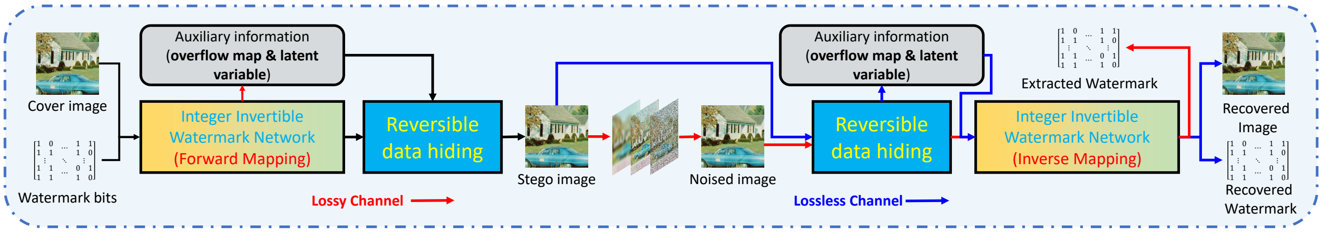

Figure 1. Overview of the proposed CRMark. CRMark employs a two-stage embedding approach similar to traditional robust reversible watermarking. In the first stage, a robust watermark is embedded using an integer invertible watermarking network. In the second stage, auxiliary information bits are integrated into the clipped stego image via reversible data hiding.

Invertible Watermark Network (iIWN) to ensure lossless mapping between the cover-image-watermark pair and the stego image. This innovative design not only guarantees reversibility but also effectively mitigates the irreversibility issues commonly associated with quantization, offering a reliable solution for applications requiring both robustness and lossless recovery.

In the proposed CRMark framework, the first stage involves mapping the cover image-watermark pair to an overflowed stego image and a latent variable using the iIWN. However, this stage presents two challenges: 1) the potential under/overflow of pixel values in the overflowed stego image; 2) the lossless storage of the latent variable. To tackle these issues, we introduce an under/overflow penalty loss during training to constrain the pixel values within the range of $[0, 255]$ as far as possible. In the second stage, we utilize arithmetic coding [35] to encode the latent variable and under/overflowed pixels into an auxiliary bitstream. Finally, this auxiliary bitstream is embedded into the clipped overflowed stego image using reversible data hiding, ensuring lossless recovery.

2. Related Works

Robust Reversible Watermarking: Reversible Data Hiding (RDH) refers to watermarking schemes that can losslessly recover the cover images and watermarks in a lossless channel. Early research primarily focused on embedding watermarks in the spatial domain, with classic methods such as Difference Expansion (DE) [32] and Histogram Shifting (HS) [26]. However, to address more complex lossy channels, robust reversible watermarking (RRW) is proposed. These approaches not only allow for the recovery of the cover image and extraction of the watermark in lossless channels but also ensure robust watermark extraction in lossy channels. Unlike traditional robust watermarking methods, RRW must balance both reversibility and robustness, presenting higher design challenges.

Mainstream RRW schemes primarily include histogram-shifting (HS) methods and two-stage embedding strategies. HS-based RRW schemes embed the watermark within robust features using histogram shifting [3, 7, 12, 16, 23, 27, 37]. The two-stage embedding strategy firstly embeds the watermark bits into the cover image using robust watermarking, followed by embedding auxiliary information into the stego image via RDH. This approach effectively resists common attacks such as JPEG compression and geometric distortions while balancing robustness and reversibility [6, 17, 21, 30, 31, 33, 34]. However, both HS and two-stage embedding schemes involve relatively complex designs.

In addition, several RRW methods based on deep learning models have been introduced [18, 24]. These models are typically trained within an encoder-noise layer-decoder framework, where watermark bits are robustly embedded using a CNN-based watermarking approach in the first stage. Due to the irreversibility of CNN-based models, in the second stage. The residual between the stego and cover images is treated as auxiliary information and embedded into the stego image to achieve reversibility. While CNN-based models enhance robustness to various distortions, this approach introduces a significant challenge related to the limited capacity of RDH in the second stage.

Real-Valued Flow-Based Data Hiding: Normalizing Flows offer a useful approach for density estimation and generative modeling, as illustrated by methods such as RealNVP [9], Glow [20], and NICE [8]. A primary advantage of flow-based models is their use of coupling layers [8], which enable invertible mappings between complex distributions and more tractable ones. The dual-input structure of these coupling layers makes flow-based models particularly effective for watermark embedding and extraction tasks [11, 22, 25, 36]. Despite the theoretical invertibility and promising results of normalizing flows in watermarking, existing flow-based watermarking approaches primarily depend on real-valued operations. Consequently, while cover images fed into the models are discrete, integer-valued data, the outputs are real-valued. Converting these outputs to discrete pixel values in digital image processing requires quantization, which can lead to information loss and compromise the recovery of the cover image. Recent studies have minimized the difference between the cover image and the stego image in real-valued flow-based steganography [22], but fully avoiding quantization-induced loss and achieving exact recovery remains challenging.

Integer Discrete Flows: Integer Discrete Flows (IDFs) [14] have been specifically designed to model discrete data effectively. These models adopt integer operations to keep data integer-valued in forward and reverse mappings and fully prevent quantization-induced information loss. Unlike real-valued flow models, IDFs ensure that inputs and outputs remain in integer form by employing integer discrete coupling layers throughout the process. This approach preserves data discreteness by carefully quantizing the network outputs to eliminate any quantization-induced information loss. In recent years, IDFs have garnered substantial attention in fields such as lossless data compression [5, 13, 14, 39, 40], generative modeling [15], achieving remarkable results in handling diverse discrete data types efficiently. In reversible data hiding tasks, IDFs naturally meet the requirements due to their lack of quantization error, inherent reversibility, and adaptability, showing great potential for practical applications.

3. Proposed Method

This section describes the proposed CRMark method in three parts. The first part introduces the iIWN, composed of integer coupling layers that enable invertible mappings of discrete data distributions. The second part outlines the CRMark training framework. As shown in Fig. 2, CRMark adopts an encoder-noise layer-decoder training framework similar to other deep learning-based robust watermarking method [11] and achieves enhanced distortion resistance via a differentiable noise layer. The third part details the CRMark inference process, illustrated in Fig. 3, which follows the common approach of traditional two-stage RRWs.

3.1. Integer Invertible Watermark Network

A flow-based invertible network can be regarded as an invertible function composed of L coupling layers as shown in Fig. 2, denoted as $\mathcal{F}_\theta = \mathcal{G}_\theta^1 \circ \mathcal{G}_\theta^2 \circ \cdots \circ \mathcal{G}_\theta^L$, with its inverse function $\mathcal{F}_\theta^{-1}$ sharing the same parameters θ. This coupling layer has become a fundamental building block for constructing such invertible functions and has been widely applied in image generation, etc. Due to the dual-input of coupling layers, they are particularly well-suited for watermarking tasks and have been extensively utilized in image steganography tasks.

In previous watermarking tasks [11], both the encoder and decoder were carefully modeled as an invertible function $\mathcal{F}_\theta(\cdot)$, with the embedding and extraction processes corresponding to the forward and inverse mapping of this function, respectively. Let $\mathbf{I}_{\text{org}} \in \{0, 1, \ldots, 255\}^{H\times W\times C}$ represent the original cover image, and $\mathbf{w} \in \{0, 1\}^M$ denote the watermark bits. The embedding process can be expressed as $[\mathbf{I}_{\text{stego}}, \mathbf{z}] = \mathcal{F}_\theta([\mathbf{I}_{\text{org}}, \mathbf{w}])$, while the extraction process can be accurately formulated as $[\mathbf{I}_{\text{rec}}, \hat{\mathbf{w}}] = \mathcal{F}_\theta^{-1}([\mathbf{I}_{\text{rec}}, \hat{\mathbf{z}}])$, where $\mathbf{I}_{\text{rec}}, \hat{\mathbf{w}}$ signify the reconstructed image and extracted watermark bits, respectively.

It is noteworthy that in previous flow-based watermarking methods [11], despite the input being integer discrete data, the output of the function is real-valued, *i.e.,* $\mathbf{I}_{\text{stego}}, \mathbf{z} \in \mathbb{R}$. To achieve lossless inverse mapping, no information loss is permissible. However, images often need to be quantized during storage, leading to information loss and hampering lossless inverse mapping. One possible solution is to encode the real-valued parts of the outputs into a bitstream and embed them in the stego image using RDH, but this poses significant challenges for reversible data hiding capacity. Consequently, invertible mapping data in the integer domain is a possible feasible solution. To this end, we designed the iIWN to handle discrete data distributions, thereby avoiding information loss caused by quantization.

Integer Coupling Layer: As depicted in Fig. 2, iIWN consisted of L integer coupling layers. For the i-th integer discrete coupling layer, assuming the outputs of the previous layer are $\mathbf{r}_i^1$ and $\mathbf{r}_i^2$, the forward mapping of the integer discrete coupling layer can be represented as:

$$\begin{aligned} \mathbf{s}_i^1 &= \mathbf{r}_i^1 + \lfloor \mathcal{U}_i(\mathbf{r}_i^2) \rceil, \\ \mathbf{s}_i^2 &= \mathbf{r}_i^2 \odot \lfloor \exp(\sigma(\mathcal{S}_i(\mathbf{s}_i^1))) \rceil + \lfloor \mathcal{Q}_i(\mathbf{s}_i^1) \rceil, \end{aligned} \tag{1}$$

where $\odot$ denotes element-wise multiplication, $\sigma(\cdot)$ is the sigmoid function, and $\lfloor \cdot \rceil$ represents the rounding operation. In contrast, the inverse mapping is given by:

$$\begin{aligned} \mathbf{r}_i^2 &= (\mathbf{s}_i^2 - \lfloor \mathcal{Q}_i(\mathbf{s}_i^1) \rceil) \oslash \lfloor \exp(\sigma(\mathcal{S}_i(\mathbf{s}_i^1))) \rceil, \\ \mathbf{r}_i^1 &= \mathbf{s}_i^1 - \lfloor \mathcal{U}_i(\mathbf{r}_i^2) \rceil, \end{aligned} \tag{2}$$

where $\oslash$ represents element-wise division, $\mathbf{s}_i^1$ and $\mathbf{s}_i^2$ are the outputs of the i-th layer. It is important to highlight that due to the dimensional discrepancy between $\mathbf{I}_{\text{org}}$ and $\mathbf{w}$, $\mathcal{Q}_i(\cdot)$ typically serves as a down-sampling transformation, whereas $\mathcal{U}_i(\cdot)$ generally acts as an up-sampling transformation and the network architecture described in Appendix A. Additionally, the rounding operators in Eqns. (1) and (2) ensure that the model can invertibly map discrete data distributions. However, the rounding operator has a zero gradient in $\mathbb{R}$, hindering network parameter updates during backpropagation. To overcome this challenge, we adopt the Straight-Through Estimator [4]. Given $\mathbf{y} = \lfloor \mathbf{x} \rceil$, the gradient with respect to $\mathbf{x}$ is approximated as $\nabla_{\mathbf{x}} \lfloor \mathbf{x} \rceil \approx \mathbf{I}$, which can be implemented as:

$$\mathbf{y} = \mathbf{x} + \texttt{stop\_gradient}(\lfloor \mathbf{x} \rceil - \mathbf{x}). \tag{3}$$

This effectively solves the gradient vanishing due to the rounding operator, enabling the effective updating of parameters. Furthermore, we adopt the stochastic round for training. As shown in Fig. 2, The quantized outputs are added with a uniformly distributed noise as $\mathbf{y} = \mathbf{x} + \texttt{stop\_gradient}(\lfloor \mathbf{x} \rceil - \mathbf{x}) + \mathbf{u}$, where $\mathbf{u} \sim U(-\mathbf{0.5}, \mathbf{0.5})$.

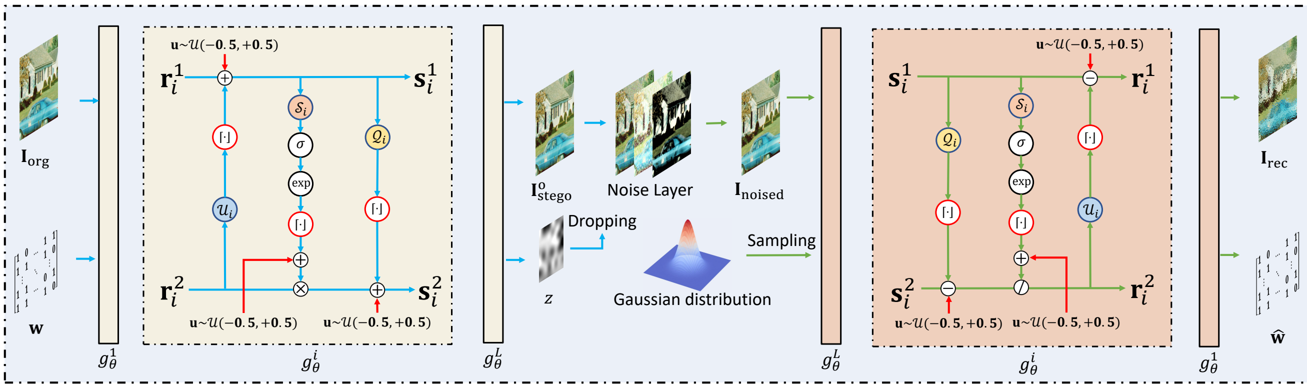

Figure 2. Overview of the training framework for the proposed CRMark. CRMark adopts the encoder-noise layer-decoder training structure common to other deep learning-based watermarking methods. Unlike traditional robust reversible watermarking approaches, CRMark enhances robustness against various distortions by learning.

3.2. Training

As shown in Fig. 2, CRMark employs the encoder-noise layer-decoder training framework: in the forward mapping, the cover image $\mathbf{I}_{\text{org}}$ and watermark $\mathbf{w}$ are mapped through $\mathcal{F}_\theta$ to obtain an overflowed stego image $\mathbf{I}^{\text{o}}_{\text{stego}}$ and a latent variable $\mathbf{z}$ as $\left[\mathbf{I}^{\text{o}}_{\text{stego}}, \mathbf{z}\right] = \mathcal{F}_\theta([\mathbf{I}_{\text{org}}, \mathbf{w}])$. Then, $\mathbf{I}^{\text{o}}_{\text{stego}}$ undergoes distortion through a noise layer $\mathcal{N}(\cdot)$, resulting in $\mathbf{I}_{\text{noised}} = \mathcal{N}(\mathbf{I}^{\text{o}}_{\text{stego}})$. In the reverse mapping, $\mathbf{I}_{\text{noised}}$ and a latent variable $\hat{\mathbf{z}}$ sampled from a Gaussian distribution are used with the inverse mapping $\mathcal{F}_\theta^{-1}$ to extract the watermark bits as $[\mathbf{I}_{\text{rec}}, \hat{\mathbf{w}}] = \mathcal{F}_\theta^{-1}([\mathbf{I}_{\text{noised}}, \hat{\mathbf{z}})]$, where $\mathbf{I}_{\text{rec}}$ is the recovered cover image.

Differentiable Noise Layer: The differentiable noise layer includes six common distortions: JPEG compression, Gaussian blur, Gaussian noise, salt-and-pepper noise, median filtering, and dropout. During training, one distortion is randomly selected each time the noise layer is invoked.

Loss Functions: The CRMark loss function consists of forward and reverse losses. The forward loss incorporates image loss, expressed as:

$$\mathcal{L}_s = \text{MSE}(\mathbf{I}^{\text{o}}_{\text{stego}}, \mathbf{I}_{\text{org}}), \;\; \mathcal{L}_l = \text{LPIPS}(\mathbf{I}^{\text{o}}_{\text{stego}}, \mathbf{I}_{\text{org}}), \tag{4}$$

where LPIPS$(\cdot)$ denotes the Learned Perceptual Image Patch Similarity [38]. Additionally, the regularization loss for the latent variable $\mathbf{z}$ is given by:

$$\mathcal{L}_z = \text{MSE}(\mathbf{z}, \mathbf{0}). \tag{5}$$

To address potential pixel overflow in $\mathbf{I}^{\text{o}}_{\text{stego}}$, we introduce an under/overflow penalty loss, defined as:

$$\mathcal{L}_p = \text{MSE}(\mathbf{I}^{+}_{\text{stego}}, \mathbf{0}) + \text{MSE}(\mathbf{I}^{-}_{\text{stego}}, \mathbf{0}), \tag{6}$$

where $\mathbf{I}^{+}_{\text{stego}} = \texttt{relu}(\mathbf{I}^{\text{o}}_{\text{stego}} - 255)$ and $\mathbf{I}^{-}_{\text{stego}} = \texttt{relu}(0 - \mathbf{I}^{\text{o}}_{\text{stego}})$. Here, $\texttt{relu}(\cdot)$ finds pixel values outside the range $[0, 255]$. In reverse mapping, the primary objective is to improve the extraction accuracy of the watermark. The watermark loss function is defined as follows:

$$\mathcal{L}_{\text{w}} = \text{MSE}(\hat{\mathbf{w}}, \mathbf{w}), \tag{7}$$

where $\hat{\mathbf{w}}$ denotes the extracted watermark and $\mathbf{w}$ represents the original watermark. Finally, the combined loss function is as follows:

$$\mathcal{L} = \lambda_s \cdot \mathcal{L}_{\text{s}} + \lambda_l \cdot \mathcal{L}_{\text{l}} + \lambda_z \cdot \mathcal{L}_{\text{z}} + \lambda_p \cdot \mathcal{L}_{\text{p}} + \lambda_w \cdot \mathcal{L}_{\text{w}}, \tag{8}$$

where λ_s, λ_l, λ_z, λ_p, and λ_w are the weighting coefficients that control the contribution of each loss functions. These coefficients ensure a balanced optimization process by adjusting the relative influence of each loss function.

3.3. Inference

The inference process of CRMark consists of two main parts. The first part focuses on cover-recoverable embedding for the watermark. The second part addresses the recovery of the cover image in a lossless channel and robust extraction in a lossy channel. The details are as follows.

Cover-Recoverable Embedding: As shown in Fig. 3, the embedding process follows a similar procedure to the training stage, *i.e.*, $[\mathbf{I}^{\text{o}}_{\text{stego}}, \mathbf{z}] = \mathcal{F}([\mathbf{I}_{\text{org}}, \mathbf{w}])$. Then, two key challenges must be addressed: 1) managing under/overflowed pixels in the stego image and 2) losslessly encoding the latent variable. To tackle these issues, the under/overflow map is encoded into a bitstream and defined as follows:

$$\mathbf{O}(i,j,c) = \begin{cases} \mathbf{I}^{\text{o}}_{\text{stego}}(i,j,c) - 255, & \text{if } \mathbf{I}^{\text{o}}_{\text{stego}}(i,j,c) > 255 \\ 0 - \mathbf{I}^{\text{o}}_{\text{stego}}(i,j,c), & \text{if } \mathbf{I}^{\text{o}}_{\text{stego}}(i,j,c) < 0 \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

This under/overflow map $\mathbf{O}$ records only pixel values exceeding the $[0, 255]$ range, with all other values set to zero.

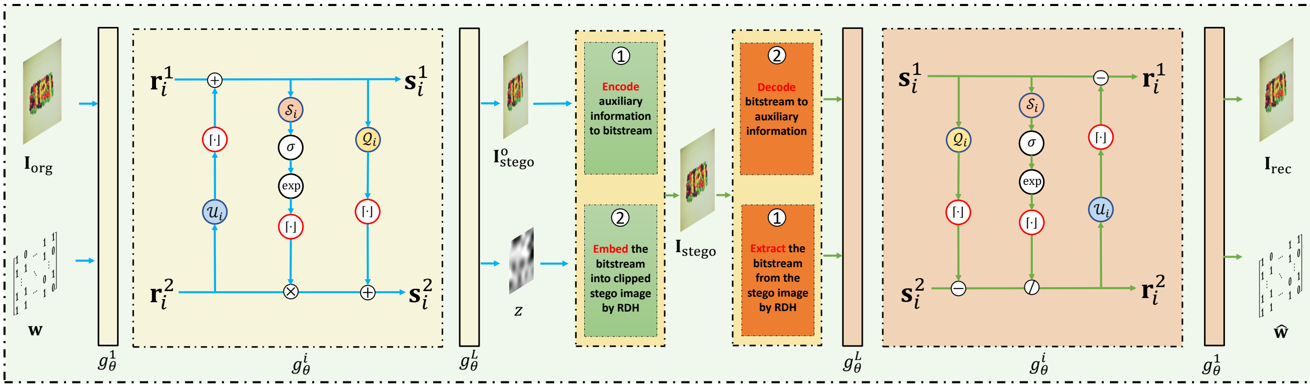

Figure 3. Overview of the inference framework for CRMark. CRMark employs a two-stage embedding strategy. In the first stage, the cover image-watermark pair is forward-mapped into an overflowed stego image and latent variables through an integer invertible watermark network. In the second stage, auxiliary bitstreams are embedded into the clipped stego image using reversible data hiding.

Subsequently, the under/overflowed stego image $\mathbf{I}^{\text{o}}_{\text{stego}}$ is clipped to fit within the valid range $[0, 255]$ as $\mathbf{I}_{\text{clip}} = \texttt{clip}(\mathbf{I}^{\text{o}}_{\text{stego}}, 0, 255)$, and $\mathbf{I}_{\text{clip}}$ will be used to store auxiliary information in the next stage. The resulting under/overflow map $\mathbf{O}$ and latent variable $\mathbf{z}$ are compressed into a bitstream using arithmetic coding [35], then embedded into $\mathbf{I}_{\text{clip}}$ to obtain the stego image $\mathbf{I}_{\text{stego}}$ by RDH.

Cover Recovery and Robust Extraction: Cover recovery and robust extraction are performed through different channels. In lossy channels, the main goal is to perform robust extraction. For a noised stego image $\mathbf{I}_{\text{noised}}$, the watermark is extracted using inverse mapping, as expressed by:

$$[\mathbf{I}_{\text{rec}}, \hat{\mathbf{w}}] = \mathcal{F}_\theta^{-1}([\mathbf{I}_{\text{noised}}, \hat{\mathbf{z}}]),$$

where $\hat{\mathbf{z}}$ is a latent variable sampled from a Gaussian distribution. In lossless channels, our primary goal is to recover the cover image and the watermark perfectly. First, we extract the auxiliary information from the stego image $\mathbf{I}_{\text{stego}}$ using RDH, and we can obtain the clipped stego image $\mathbf{I}_{\text{clip}}$. Then, we decode auxiliary information by arithmetic code to recover the overflow map $\mathbf{O}$ and latent variable $\mathbf{z}$. Subsequently, the overflowed stego image $\mathbf{I}^{\text{o}}_{\text{stego}}$ is reconstructed as:

$$\mathbf{I}^{\text{o}}_{\text{stego}} = \mathbf{I}_{\text{clip}} + \mathbf{I}^{255}_{\text{mask}} \odot \mathbf{O} - \mathbf{I}^{0}_{\text{mask}} \odot \mathbf{O},$$

where the masks $\mathbf{I}^{255}_{\text{mask}}$ and $\mathbf{I}^{0}_{\text{mask}}$ are defined as:

$$\mathbf{I}^{255}_{\text{mask}}(i,j,c) = \begin{cases} 1, & \text{if } \mathbf{O}(i,j,c) > 0, \\ 0, & \text{otherwise}, \end{cases}$$

$$\mathbf{I}^{0}_{\text{mask}}(i,j,c) = \begin{cases} 1, & \text{if } \mathbf{O}(i,j,c) < 0, \\ 0, & \text{otherwise}. \end{cases}$$

Finally, the original cover image and watermark can be recovered through inverse mapping, *i.e.*, $[\mathbf{I}_{\text{org}}, \mathbf{w}] = \mathcal{F}^{-1}([\mathbf{I}^{\text{o}}_{\text{stego}}, \mathbf{z}])$. This process can reliably ensure the lossless recovery of both the original cover image and the embedded watermark with complete fidelity.

4. Experiments

4.1. Experimental Setup

In our experiments, the training dataset is DIVK [10]. The test datasets include the classical USC-SIPI [1] and Kodak24 [2]. The experimental framework is implemented in PyTorch and runs on an NVIDIA 4090 Ti GPU. The model is optimized using the AdamW optimizer with a batch size of 8, a learning rate of $\gamma = 10^{-4}$, $\beta_1 = 0.5$, and $\beta_2 = 0.999$. The number of coupling layers is set to $L = 5$. To achieve lossless compression and auxiliary bitstream embedding, we employ arithmetic coding [35] and classical PEE [29]. More training details in Appendix E.

To evaluate the proposed method, we compare it with two categories of approaches. The first category includes state-of-the-art (SOTA) reversible robust watermarking (RRW) methods, such as STDM [31] and PZM [30]. The second category consists of learning-based irreversible robust image watermarking (IRW) methods, including the real-valued flow-based robust watermarking method FIN [11], the CNN-based robust image watermarking method MBRS [19], and a variant of CRMark where the quantization operator is removed, referred to as R-CRMark, which degenerates into a real-valued flow-based CRMark, noting that R-CRMark is irrecoverable. Moreover, CRMark does not retain auxiliary information, in contrast to CRMark-R, which does. It is important to note that FIN, R-CRMark, and CRMark are trained on the six distortions described in Section 3.2, using images of size 256×256 and watermark lengths of 64. In contrast, STDM and PZM, which are grayscale image-based methods, use test images of size 512×512 with watermark lengths of 256. The detailed experimental analysis is presented below.

4.2. Comparison with Existing Methods

This subsection primarily compares CRMark with two categories of methods: the RRW methods STDM [31] and

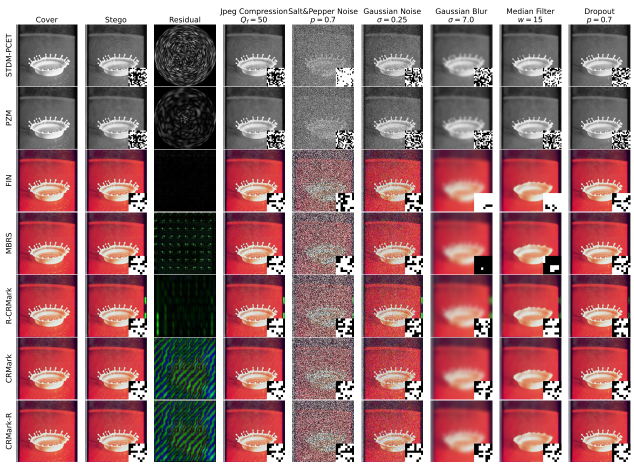

Figure 4. Visual comparison of the proposed CRMark†, CRMark-R†, R-CRMark†, STDM† [31], PZM† [30], FIN† [11], and MBRS† [19]. Here, † indicates cover-recoverable watermark methods, while † denotes non-recoverable ones.

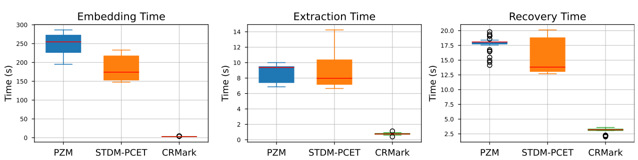

Figure 5. Comparison of embedding, extraction, and recovery time complexity for watermarking methods CRMark, STDM [31], and PZM [30], capable of lossless recovery.

PZM [30], and IRW methods MBRS [19], FIN [11], and R-CRMark. The experimental analysis is presented below.

Comparison with RRW methods: We compare CRMark with existing robust reversible watermarking methods, including STDM [31] and PZM [30]. The evaluation covers key aspects such as watermark imperceptibility, robustness, computational time complexity, and the length of auxiliary bitstream. The results for imperceptibility are shown in Fig. 4, we can observe that STDM and PZM embed watermarks using template-like perturbations, resulting in more noticeable distortions and weaker coupling with the cover image. In contrast, the stego image to CRMark demonstrates higher imperceptibility, with stronger coupling between the watermark and the cover image. Quantitative metrics presented in Tab. 1 reveal that CRMark attains a PSNR/SSIM of $40.95/0.9995$. It surpasses STDM by $4.41/0.0392$ and PZM by $2.78/0.0136$. In summary, whether evaluated subjectively through visual perception or objectively through quantitative visual quality metrics, the imperceptibility of CRMark stands out as superior to existing RRW methods.

Moreover, Tab. 2 compares the extraction accuracy of CRMark with STDM [31] and PZM [30] across six common distortions. Specifically, under Gaussian blur ($\sigma = 5.0$), CRMark and CRMark-R achieve impressive accuracies of 98.70% and 98.77%, respectively, far surpassing STDM's 53.76% and PZM's 61.92%. In Gaussian noise ($\sigma = 0.25$), the two methods record extraction accuracies of 88.87% and 89.16%, again substantially outperforming STDM and PZM. For salt-and-pepper (S&P) noise ($p = 0.7$), CRMark and CRMark-R deliver 95.53% and 95.88%, respectively, reflecting a peak improvement of approximately 46% over STDM and PZM. In dropout ($p = 0.7$) and median filtering ($w = 15$), both methods consistently maintain accuracies ranging from 92% to 97% or higher, markedly exceeding the performance of the other two approaches. Under JPEG compression, although CRMark and CRMark-R fall slightly below STDM and PZM's perfect 100.0% accuracy at $Q_f = 50$, they still sustain extraction accuracy above 99%. In conclusion, CRMark and CRMark-R perform exceptionally well under most distortions, with only a slight shortfall at high JPEG compression ratios. CRMark achieves maximum improvements of 46.40% in salt-and-pepper noise and 44.94% in Gaussian blur. These results suggest that CRMark exhibits superior robustness and performance over traditional RRW methods in handling most digital distortions.

Table 1. Comparison of SSIM and PSNR (mean ± standard deviation) for proposed CRMark†, CRMark-R†, R-CRMark†, STDM† [31], PZM† [30], FIN† [11], and MBRS† [19]. Here, † indicates cover-recoverable watermark methods, while † denotes non-recoverable ones.

Method	SSIM	PSNR
STDM† [31]	0.9603 ± 0.0195	36.54 ± 1.77
PZM† [30]	0.9859 ± 0.0078	38.17 ± 1.78
FIN† [11]	0.9988 ± 0.0015	39.74 ± 2.49
MBRS† [19]	**0.9996 ± 0.0002**	40.73 ± 1.67
R-CRMark†	0.9971 ± 0.0046	33.92 ± 5.01
CRMark†	0.9995 ± 0.0005	**40.95 ± 1.24**
CRMark-R†	0.9995 ± 0.0005	40.90 ± 1.23

We compare the efficiency of CRMark with PZM [30] and STDM [31] in embedding, extraction, and recovery times as shown in Fig. 5. CRMark averages 4.51 s for embedding, reducing PZM time of 248.65 s and STDM time of 184.26 s by $55.1\times$ and $40.9\times$, respectively. Extraction averages 1.50 s for CRMark, outperforming PZM time of 8.54 s and STDM time of 8.93 s by $5.7\times$ and $6.0\times$, respectively. Recovery takes 4.96 s for CRMark, beating PZM time of 17.71 s and STDM time of 15.89 s by $3.6\times$ and $3.2\times$, respectively. CRMark significantly enhances efficiency over PZM and STDM for time cost. Furthermore, we compare the auxiliary information bitstream lengths of CRMark with PZM [31] and STDM [31], as shown in Fig. 6. The average bitstream length of PZM is 61896.94 bits, ranging

Table 2. Extraction accuracy (%) of the proposed CRMark†, CRMark-R†, R-CRMark†, STDM† [31], PZM† [30], FIN† [11], and MBRS† [19] under various distortions. Here, † indicates cover-recoverable watermark methods, while † denotes non-recoverable ones. **Bold** highlights optimal performance among cover-recoverable watermark methods, and underlining indicates the global best performance.

Methods	JPEG(Q_f)			Gaussian Blur(σ)			Gaussian Noise(σ)		
	50	70	90	5.0	6.0	7.0	0.05	0.15	0.25
STDM† [31]	**100.0**	**100.0**	**100.0**	53.76	50.46	49.71	99.66	92.79	74.33
PZM† [30]	**100.0**	**100.0**	**100.0**	61.92	54.31	50.12	99.86	88.71	62.97
FIN† [11]	98.64	99.14	99.40	47.98	47.58	47.73	98.88	92.28	83.42
MBRS† [19]	97.68	99.95	100.0	87.05	74.65	62.95	99.99	97.58	90.18
R-CRMark†	93.33	96.06	96.84	81.18	73.05	66.64	95.30	92.30	85.00
CRMark†	99.27	100.0	100.0	98.70	97.11	93.87	100.0	97.59	88.87
CRMark-R†	99.31	**100.0**	**100.0**	**98.77**	**97.00**	**93.83**	**100.0**	**97.56**	**89.16**
Methods	S&P(p)			Dropout(p)			Median Filter(w)		
	0.3	0.5	0.7	0.3	0.5	0.7	11	13	15
STDM† [31]	56.46	49.81	49.48	97.98	87.24	64.01	77.18	69.82	63.76
PZM† [30]	49.71	49.61	50.09	99.98	97.72	71.45	79.81	73.71	68.18
FIN† [11]	72.01	62.73	57.41	98.04	94.99	88.38	53.93	50.35	49.50
MBRS† [19]	99.77	96.62	82.58	99.99	99.42	96.61	97.08	85.74	69.00
R-CRMark†	95.02	87.63	74.42	93.98	84.03	69.81	94.36	90.73	85.08
CRMark†	**99.86**	99.06	95.53	99.98	97.75	92.35	99.41	98.83	97.52
CRMark-R†	99.85	**99.28**	**95.88**	**99.99**	**97.68**	**92.40**	**99.39**	**98.77**	**97.61**

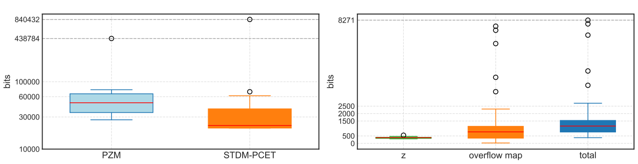

Figure 6. Comparison of auxiliary bitstream lengths for watermarking methods CRMark, STDM [31], and PZM [30]. The right plot shows the under/overflow map $\mathbf{O}$ bitstream length, $\mathbf{z}$ bitstream length, and total auxiliary bitstream length for the proposed CRMark, while the left plot shows results for STDM and PZM.

from a minimum of 27248 bits and a maximum of 438784 bits. STDM averages 56968.52 bits, ranging from 20480 bits to 840432 bits. CRMark consists of the latent variable $\mathbf{z}$ (mean 382.61 bits, min 384, max 8271) and the under/overflow map (mean 1558.58 bits, min 384, max 8271), achieving a total mean length of 1962.19 bits (min 384, max 8271). Compared to PZM, it reduces the length by $31.5\times$, and compared to STDM, it reduces the length by $29.0\times$. It is important to note that PZM and STDM are unable to achieve fully reversible embedding for all test images, while CRMark consistently succeeds in this task. These results highlight that CRMark outperforms both PZM and STDM in terms of achieving lossless recovery.

Comparison with IRW Methods: We compare CRMark with existing IRW methods, including FIN [11], MBRS [19], and R-CRMark, across watermark imperceptibility and robustness. The results for imperceptibility are shown in Fig. 4, we can observe that FIN exhibits low-frequency texture distortion with noticeable white noise, while MBRS presents a grid-like template that is visibly distinct. In contrast, CRMark and CRMark-R exhibit higher watermark imperceptibility with stronger coupling between the watermark and the cover image. Quantitative results in Tab. 1 indicate that CRMark achieves a PSNR/SSIM of $40.95/0.9995$, exceeding FIN by $1.21/0.0007$, MBRS by $0.22/-0.0001$, and R-CRMark by $7.03/0.0024$ relative to their respective values of $39.74/0.9988$, $40.73/0.9996$, and $33.92/0.9971$. CRMark-R closely follows with a PSNR/SSIM of $40.90/0.9995$. These results demonstrate that CRMark also outperforms existing IRW methods in watermark imperceptibility.

For robustness, Tab. 2 assesses the extraction accuracy of CRMark, FIN [11], MBRS [19], and R-CRMark under six digital distortions. Specifically, under JPEG compression ($Q_f = 50$), CRMark and CRMark-R both achieve extraction accuracies exceeding 99.00%, surpassing FIN, MBRS, and R-CRMark. For Gaussian blur ($\sigma = 7$), CRMark and CRMark-R maintain accuracies above 90% and 90%, far outperforming FIN at 47.73%, MBRS at 62.95%, and R-CRMark at 66.64%. With Gaussian noise ($\sigma = 0.25$), they reach 88.87% and 89.16%, still close to 90%, ahead of FIN and R-CRMark, though slightly below MBRS by just 1.31% and 1.02%. Under salt-and-pepper noise ($p = 0.7$), CRMark and CRMark-R attain 95.53% and 95.88%, clearly outperforming the other three methods. Similarly, in median filtering, they notably exceed the others. However, in dropout distortion, CRMark lags MBRS by about 4%, yet

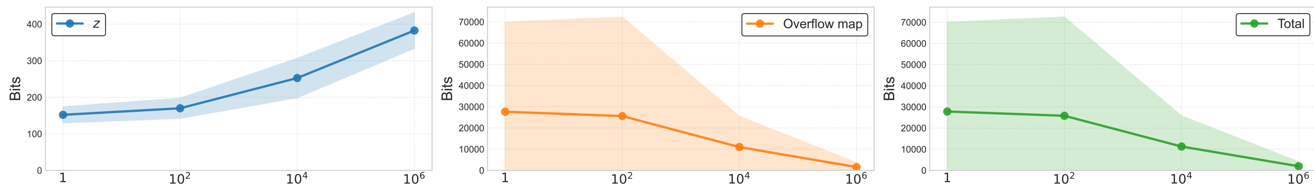

Figure 7. Effect of λ_p on bitstream length for $\lambda_p \in \{0, 10^2, 10^4, 10^6\}$. Length decreases by $14.18\times$ at $\lambda_p = 10^6$ compared to $\lambda_p = 0$.

still greatly surpasses the rest. Although CRMark is slightly weaker than MBRS in dropout distortion, it significantly improves over other methods for most distortions.

Overall, CRMark and CRMark-R outperform FIN, MBRS, and R-CRMark in watermark imperceptibility and robustness, consistently achieving high accuracy across most distortions. Remarkably, CRMark uniquely ensures the lossless recovery of original cover images and watermarks in lossless channels, unlike IRW methods.

4.3. Ablation Experiments

In Section 3.2, we introduced a penalty loss for the pixel under/overflow and a regularization loss for the latent variable $\mathbf{z}$. In this section, we conduct an ablation study to evaluate the effectiveness of these losses.

Under/Overflow Penalty Loss: To evaluate the effectiveness of penalty loss, we conduct ablation experiments by varying the penalty loss weight λ_p across $\lambda_p \in \{0, 10^2, 10^4, 10^6\}$. Here, $\lambda_p = 0$ serves as the baseline with no overflow penalty. These experiments evaluated the effect of λ_p on the encoded bit length of auxiliary information, comprising the latent variable $\mathbf{z}$ and the overflow map. The experimental results are shown in Fig. 7, we can observe that at $\lambda_p = 0$, the auxiliary bitstream meets longest, averaging 27828.87 bits. The overflow map compressed into 27655.90 bits, while $\mathbf{z}$ stay at 151.97 bits. As λ_p increases, the total bitstream length exhibits a decreasing trend. At $\lambda_p = 10^2$, it decreases to 25807.48 bits, a $1.08\times$ reduction from the baseline, with the overflow map dropping to 25616.58 bits ($1.08\times$ reduction) and $\mathbf{z}$ slightly rising to 169.90 bits ($1.12\times$ increase). This trend intensifies at $\lambda_p = 10^4$, where the total length falls to 11276.45 bits, a $2.47\times$ reduction, driven by the overflow map shrinking to 11002.81 bits ($2.51\times$ reduction), while $\mathbf{z}$ grows to 252.65 bits ($1.66\times$ increase). The most pronounced effect occurs at $\lambda_p = 10^6$, reducing the total length to 1962.19 bits, a substantial $14.18\times$ decrease, with the overflow map at 1558.58 bits ($17.74\times$ reduction), $\mathbf{z}$ at 382.61 bits ($2.52\times$ increase), and the bitstream length with minimal fluctuation. This indicates that the overflow penalty loss term can effectively suppress pixel overflow, thereby reducing the length of the auxiliary bitstream.

Regularization Loss for Latent variable z: We evaluated the impact of applying a regularization loss to the latent variable compared to omitting it. We found that even when using FLOAT64 precision, if the regularization loss is not applied, numerical overflow issues still occur during the iIWN inference process, resulting in approximately 9.67% of stego images failing to achieve lossless recovery through inverse mapping. This regularization loss efficiently solves this issue. Nevertheless, we still recommend using FLOAT64 precision during inference to ensure full lossless recovery of all test cover images.

In summary, these results demonstrate the effectiveness of penalty and regularization losses in enhancing the performance of CRMark. The penalty loss effectively reduces the auxiliary bitstream length, decreasing the bitstream length by $14.18\times$ from $\lambda_p = 0$ to $\lambda_p = 10^6$. Additionally, the regularization loss eliminates numerical overflow issues in iIWN inference, ensuring lossless recovery.

5. Conclusion

This paper introduces a novel watermarking approach called **C**over-**R**ecoverable Water**Mark** (CRMark) that addresses the challenge of permanent distortion imposed on cover images by traditional watermarking. CRMark achieves perfect lossless recovery in lossless channels while enabling robust watermark extraction in lossy channels, addressing a critical need for distortion-free and resilient watermarking solutions. CRMark employs a two-stage embedding process: the first stage utilizes a designed integer Invertible Watermark Network to map the cover-image-watermark-pair into an underoverflowed stego image and a latent variable $\mathbf{z}$, followed by a second stage where reversible data hiding embeds the encoded underoverflow map and latent variable bitstream into the clipped stego image, enabling lossless recovery. To validate the effectiveness of CRMark, we conducted experiments that demonstrated superior robustness and imperceptibility compared to existing cover-recoverable watermark methods, with an embedding time reduction of up to $55.1\times$ and an auxiliary bitstream length compression of up to $31.5\times$. Furthermore, we propose an innovative overflow penalty loss to decrease auxiliary bitstream length, with experiments confirming its remarkable effectiveness in significantly compressing this length by up to $14.18\times$ compared to the no-penalty baseline. Most importantly, CRMark successfully endows all cover images with lossless recovery, which existing RRW methods cannot, thereby enhancing the efficiency of CRMark and practical applicability.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (62171244), Guangdong Provincial Key Laboratory (Grant 2023B1212060076), and Innovation Team Project of Guangdong Province of China (No. 2024KCXTD017).

References

[1] The usc-sipi image database, 1977. Available: http://sipi.usc.edu/database/. 5

[2] Kodak Lossless True Color Image Suite, 1999. 5

[3] Lingling An, Xinbo Gao, Xuelong Li, Dacheng Tao, Cheng Deng, and Jie Li. Robust reversible watermarking via clustering and enhanced pixel-wise masking. *IEEE Transactions on image processing*, 21(8):3598–3611, 2012. 2

[4] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013. 3

[5] Rianne van den Berg, Alexey A Gritsenko, Mostafa Dehghani, Casper Kaae Sønderby, and Tim Salimans. Idf++: Analyzing and improving integer discrete flows for lossless compression. *arXiv preprint arXiv:2006.12459*, 2020. 3

[6] Dinu Coltuc and Jean-Marc Chassery. Distortion-free robust watermarking: a case study. In *Security, Steganography, and Watermarking of Multimedia Contents IX*, pages 585–592. SPIE, 2007. 2

[7] Christophe De Vleeschouwer, J-F Delaigle, and Benoit Macq. Circular interpretation of bijective transformations in lossless watermarking for media asset management. *IEEE Transactions on Multimedia*, 5(1):97–105, 2003. 2

[8] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. In *International Conference on Learning Representations (ICLR) Workshop*, 2015. 2

[9] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016. 1, 2

[10] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111:98–136, 2015. 5

[11] Han Fang, Yupeng Qiu, Kejiang Chen, Jiyi Zhang, Weiming Zhang, and Ee-Chien Chang. Flow-based robust watermarking with invertible noise layer for black-box distortions. In *Proceedings of the AAAI conference on artificial intelligence*, pages 5054–5061, 2023. 1, 2, 3, 5, 6, 7

[12] Xinbo Gao, Lingling An, Yuan Yuan, Dacheng Tao, and Xuelong Li. Lossless data embedding using generalized statistical quantity histogram. *IEEE Transactions on Circuits and Systems for Video Technology*, 21(8):1061–1070, 2011. 2

[13] Jonathan Ho, Evan Lohn, and Pieter Abbeel. Compression with flows via local bits-back coding. *Advances in Neural Information Processing Systems*, 32, 2019. 3

[14] Emiel Hoogeboom, Jorn Peters, Rianne Van Den Berg, and Max Welling. Integer discrete flows and lossless compression. *Advances in Neural Information Processing Systems*, 32, 2019. 3

[15] Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. *Advances in Neural Information Processing Systems*, 34:12454–12465, 2021. 3

[16] Donghui Hu, Dan Zhao, and Shuli Zheng. A new robust approach for reversible database watermarking with distortion control. *IEEE Transactions on Knowledge and Data Engineering*, 31(6):1024–1037, 2018. 2

[17] Runwen Hu and Shijun Xiang. Cover-lossless robust image watermarking against geometric deformations. *IEEE Transactions on Image Processing*, 30:318–331, 2020. 2

[18] Jie Huang, Fengyong Li, Lei Zhang, and Chuan Qin. Two-stage robust reversible image watermarking based on deep neural network. In *International Conference on Artificial Intelligence and Security*, pages 325–335. Springer, 2022. 2

[19] 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*, pages 41–49, 2021. 5, 6, 7

[20] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 10215–10224, 2018. 2

[21] Rajeev Kumar and Ki-Hyun Jung. Robust reversible data hiding scheme based on two-layer embedding strategy. *Information Sciences*, 512:96–107, 2020. 2

[22] Fengyong Li, Yang Sheng, Kui Wu, Chuan Qin, and Xinpeng Zhang. Lidinet: A lightweight deep invertible network for image-in-image steganography. *IEEE Transactions on Information Forensics and Security*, 2024. 2

[23] Xingyuan Liang and Shijun Xiang. Robust reversible audio watermarking based on high-order difference statistics. *Signal Processing*, 173:107584, 2020. 2

[24] Zhangyu Liu, Zhi Li, Long Zheng, and Dandan Li. Two-stage robust lossless dwi watermarking based on transformer networks in the wavelet domain. *Applied Sciences*, 13(12): 6886, 2023. 2

[25] 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*, pages 10816–10825, 2021. 1, 2

[26] Zhicheng Ni, Yun-Qing Shi, Nirwan Ansari, and Wei Su. Reversible data hiding. *IEEE Transactions on circuits and systems for video technology*, 16(3):354–362, 2006. 1, 2

[27] Zhicheng Ni, Yun Q Shi, Nirwan Ansari, Wei Su, Qibin Sun, and Xiao Lin. Robust lossless image data hiding designed for semi-fragile image authentication. *IEEE Transactions on circuits and systems for video technology*, 18(4):497–509, 2008. 2

[28] Christine I Podilchuk and Edward J Delp. Digital watermarking: algorithms and applications. *IEEE signal processing Magazine*, 18(4):33–46, 2001. 1

[29] Vasiliy Sachnev, Hyoung Joong Kim, Jeho Nam, Sundaram Suresh, and Yun Qing Shi. Reversible watermarking algorithm using sorting and prediction. *IEEE Transactions on Circuits and Systems for Video Technology*, 19(7):989–999, 2009. 5

[30] Yichao Tang, Shuai Wang, Chuntao Wang, Shijun Xiang, and Yiu-Ming Cheung. A highly robust reversible watermarking scheme using embedding optimization and rounded error compensation. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(4):1593–1609, 2022. 2, 5, 6, 7

[31] Yichao Tang, Chuntao Wang, Shijun Xiang, and Yiu-ming Cheung. A robust reversible watermarking scheme using attack-simulation-based adaptive normalization and embedding. *IEEE Transactions on Information Forensics and Security*, 2024. 1, 2, 5, 6, 7

[32] Jun Tian. Reversible data embedding using a difference expansion. *IEEE transactions on circuits and systems for video technology*, 13(8):890–896, 2003. 2

[33] Haodong Wang, Heng Yao, Chuan Qin, and Xinpeng Zhang. When robust reversible watermarking meets cropping attacks. *IEEE Transactions on Circuits and Systems for Video Technology*, 2024. 2

[34] Xiang Wang, Xiaolong Li, and Qingqi Pei. Independent embedding domain based two-stage robust reversible watermarking. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(8):2406–2417, 2019. 2

[35] Ian H Witten, Radford M Neal, and John G Cleary. Arithmetic coding for data compression. *Communications of the ACM*, 30(6):520–540, 1987. 2, 5

[36] 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*, pages 7875–7884, 2022. 2

[37] Xian-Ting Zeng, Ling-Di Ping, and Xue-Zeng Pan. A lossless robust data hiding scheme. *Pattern Recognition*, 43(4): 1656–1667, 2010. 2

[38] 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*, pages 586–595, 2018. 4

[39] Shifeng Zhang, Ning Kang, Tom Ryder, and Zhenguo Li. iflow: Numerically invertible flows for efficient lossless compression via a uniform coder. *Advances in Neural Information Processing Systems*, 34:5822–5833, 2021. 3

[40] Shifeng Zhang, Chen Zhang, Ning Kang, and Zhenguo Li. ivpf: Numerical invertible volume preserving flow for efficient lossless compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 620–629, 2021. 3

[41] Xuandong Zhao, Kexun Zhang, Zihao Su, Saastha Vasan, Ilya Grishchenko, Christopher Kruegel, Giovanni Vigna, Yu-Xiang Wang, and Lei Li. Invisible image watermarks are provably removable using generative ai. *Advances in Neural Information Processing Systems*, 37:8643–8672, 2025. 1