ARIW-Framework: Adaptive Robust Iterative Watermarking Framework

Shaowu Wu[1], **Liting Zeng**[1], **Wei Lu**[1*]
[1]MoE Key Laboratory of Information Technology
School of Computer Science and Engineering
Sun Yat-sen University
Guangzhou, China
wushw25@mail2.sysu.edu.cn, zenglt9@mail2.sysu.edu.cn, luwei3@mail.sysu.edu.cn

Abstract

With the rapid rise of large models, copyright protection for generated image content has become a critical security challenge. Although deep learning watermarking techniques offer an effective solution for digital image copyright protection, they still face limitations in terms of visual quality, robustness and generalization. To address these issues, this paper proposes an adaptive robust iterative watermarking framework (ARIW-Framework) that achieves high-quality watermarked images while maintaining exceptional robustness and generalization performance. Specifically, we introduce an iterative approach to optimize the encoder for generating robust residuals. The encoder incorporates noise layers and a decoder to compute robustness weights for residuals under various noise attacks. By employing a parallel optimization strategy, the framework enhances robustness against multiple types of noise attacks. Furthermore, we leverage image gradients to determine the embedding strength at each pixel location, significantly improving the visual quality of the watermarked images. Extensive experiments demonstrate that the proposed method achieves superior visual quality while exhibiting remarkable robustness and generalization against noise attacks.

Code — https://github.com/wushaowu2014/ARIW
Datasets — https://github.com/wushaowu2014/ARIW
Extended version — https://arxiv.org/abs/2505.13101

Introduction

Deep learning-based digital image watermarking techniques have made significant progress in recent years (Hosny et al. 2024; Hu et al. 2024; Wang et al. 2024a; Jang et al. 2024; Fu et al. 2024). They enable the embedding of watermarks into arbitrary images in an imperceptible way, achieving tasks such as copyright protection and traceability. With the rapid development of large models, the need for copyright protection and traceability of generated content has become particularly critical (Dathathri et al. 2024; Zhang et al. 2024; Wang et al. 2024d; Lyu, Chen, and Fu 2023; Wang et al. 2024c; Wu, Liao, and Ou 2023; Fernandez et al. 2023; Wang et al. 2024b). In response, various countries have issued corresponding standards requiring generative content to include identification mechanisms (Golda et al. 2024; Ye et al. 2024; Wang et al. 2024d). However, current deep learning-based digital image watermarking techniques still face challenges in simultaneously optimizing visual quality, robustness, and generalization. Consequently, advancing deep learning watermarking technologies has become an urgent task.

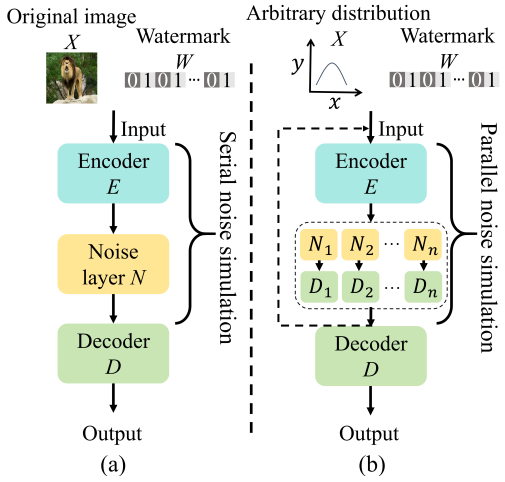

Figure 1: (a) Existing deep learning watermarking framework improves the robustness by employing serial noise simulation. (b) Proposed framework improves the robustness by employing parallel noise simulation.

In general, training a deep learning watermarking framework is a process of adversarial optimization. On the one hand, the network is trained to generate high-visibility watermarked images, ensuring that the watermarked images are visually indistinguishable from the original ones. However, the network must resist various noise attacks, such as JPEG compression, scaling, and Gaussian noise (Wan et al. 2022; Ma et al. 2023). Achieving high visual quality can be done by modifying fewer pixels (with lower intensity) or embedding less watermark information. In contrast, enhancing robustness typically requires modifying more pixels (with higher intensity) or embedding more watermark information. During the optimization process, the critical challenge lies in designing the network architecture and loss function to balance visual quality and robustness. Currently, most

*Corresponding author.

deep learning watermarking frameworks adopt an encoder-noise layer-decoder (E-N-D) structure (Zhu et al. 2018; Tancik, Mildenhall, and Ng 2020; Jia, Fang, and Zhang 2021; Ma et al. 2022; Huang et al. 2023; Fang et al. 2023; Ma et al. 2025a; Wu, Lu, and Luo 2025), as shown in Figure 1 (a). The encoder embeds the watermark into the original image to generate the watermarked image while ensuring its imperceptibility. The noise layer enhances the robustness by simulating the distortion process of watermarked images, enabling resistance to various noise attacks. The decoder extracts the watermark from the attacked watermarked images. In this framework, joint training with aggregated loss functions is typically used to optimize both visual quality and robustness (Jia, Fang, and Zhang 2021; Ge et al. 2023; Wang, Wu, and Wang 2023; Qin et al. 2024). Additionally, to address the issue of gradient truncation caused by rounding operations in the noise layer, a two-stage training strategy can be used: first, optimizing the encoder parameters and freezing them, and then optimizing the decoder parameters to enable effective gradient backpropagation (Liu et al. 2019; Yin et al. 2023; Ma et al. 2025b). Although existing deep learning watermarking methods have addressed many underlying challenges, limitations persist in visual quality and robustness. Robustness largely depends on the design of the noise layer, which in most networks simply concatenates multiple noises to simulate composite attacks (Ge et al. 2023; Huang et al. 2023; Hosny et al. 2024; Hu et al. 2024). However, this approach struggles to reflect the characteristics of specific individual noise types and is insufficient for adapting to complex real-world scenarios. Meanwhile, robustness optimization often compromises visual quality, making it difficult to simultaneously optimize multiple objectives in the trained network. In summary, establishing an end-to-end deep learning watermarking framework remains an urgent and unresolved challenge.

In this work, we propose an adaptive robust iterative watermarking framework (ARIW-Framework) that addresses the multi-objective optimization challenges. Specifically, our network learns to generate a robust residual, which is added to the original image to produce the watermarked image. The resulting watermarked image is visually imperceptible from the original image. To ensure high visual quality, we use the gradient of the original image to determine the embedding strength at each pixel location. This ensures that more watermark is embedded in complex regions while less is embedded in smooth regions. To enhance robustness against various types of noise attacks, we adopt a parallel design within the encoder, concatenating multiple noise types to achieve robustness optimization for individual noise attacks, as shown in Figure 1 (b). This design enables the encoder to generate watermarked images with both high visual quality and robustness. Additionally, our framework focuses solely on optimizing the robust residual, independent of the original image. In other words, our framework does not impose restrictions on the specific optimization target, any objective can be optimized to approach the domain of the optimal robust residual. In summary, the contributions can be summarized as bellow:

- We propose an adaptive robust iterative watermarking framework capable of sampling any spatial distribution as the iterative target, without being constrained by the original image.
- We design an encoder with linearly additive residuals, which simulates distortions caused by various attack types in parallel, enabling the training of highly robust residuals.
- We introduce the use of original image gradients to determine the adaptive embedding strength at each pixel, enabling watermark embedding in complex regions while avoiding smooth regions.
- Experimental results demonstrate that our method achieves outstanding robustness against signal processing and geometric attacks while maintaining high visual quality of the watermarked images.

Method

Pipeline Overview

Our proposed method addresses copyright protection and traceability for both original and AI-generated images. As illustrated in Figure 2, the watermarking framework consists of five stages: *Stage-1* Preprocessing of the original watermark, *Stage-2* Calculation of the original image gradient, *Stage-3* Iteration from the initial state (i.e., the process of finding the optimal residual), *Stage-4* Calculation of robust residual weights (i.e., determining robust residual weights for each type of attack), *Stage-5* Watermark extraction. Unlike existing watermarking methods, the proposed framework integrates the noise layer N directly into the encoder to generate highly robust residuals. The resulting residuals are added to the original image to produce the watermarked image, which is then passed to the decoder for watermark extraction without requiring additional distortion simulation on the watermarked image. The subsequent sections will provide a detailed explanation of our approach.

Watermark Preprocessing

The embedded watermark in this work is a sequence composed of 0s and 1s. Before feeding it into the network, a preprocessing step is required. This preprocessing primarily includes linear transformation and upsampling operations (Tan et al. 2024). Specifically, the linear transformation projects watermark vectors of fixed length into a higher-dimensional vector space to enhance their representational capacity. Meanwhile, the upsampling operation maps these vectors to higher-dimensional tensor spaces, enabling resolution adaptation to feature tensors of arbitrary sizes while simultaneously strengthening watermark representation capability to some extent. Let the original watermark be $W \in \{0,1\}^L$ and the original image be $X \in \mathbb{R}^{m\times n\times c}$. To enable successful upsampling of the watermark, W needs to undergo a linear transformation, mapping it to a feature $W_1 \in \mathbb{R}^{L_1}$:

$$f : W \in \{0,1\}^L \rightarrow W_1 \in \{0,1\}^{L_1} \tag{1}$$

subsequently, through reshape and upsampling operations, W_1 is mapped to a spatial representation $W_2 \in \mathbb{R}^{m\times n\times c}$:

$$f : W_1 \in \mathbb{R}^{L_1} \rightarrow W_2 \in \mathbb{R}^{m\times n\times c} \tag{2}$$

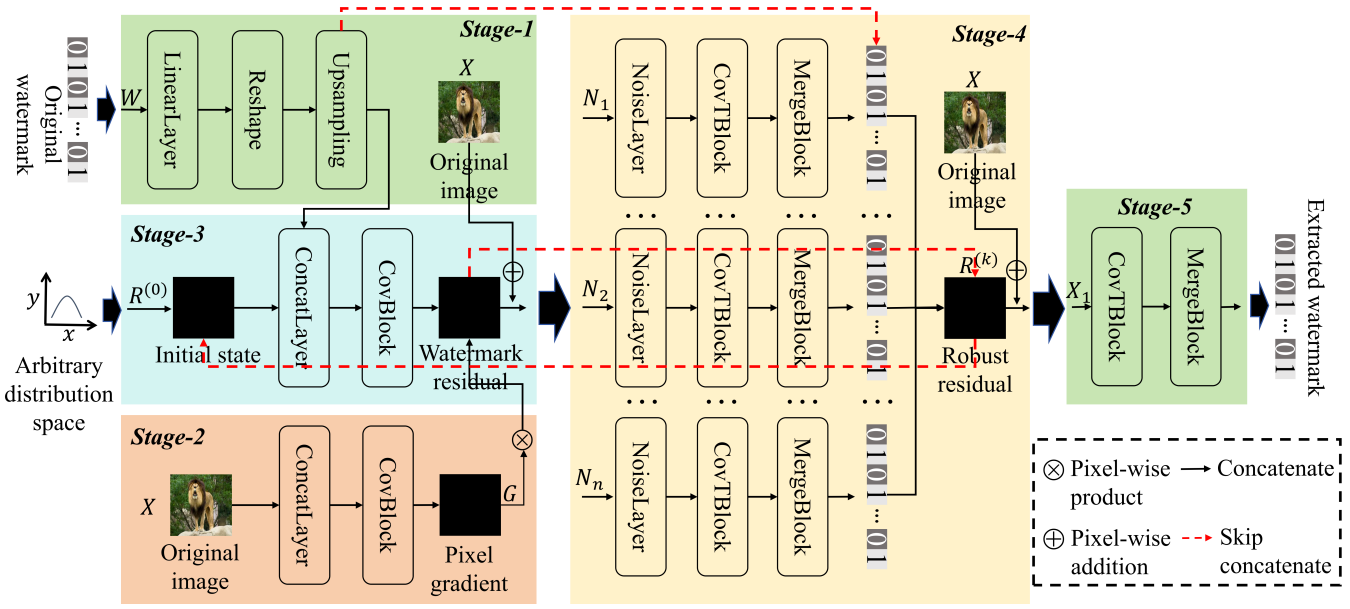

Figure 2: The pipeline of our framework.

here, the upsampling factor is $d = m \times n \times c/L_1$. The resulting W_2 has the same resolution as the original image X, facilitating seamless concatenation and establishing the relationship between the watermark and the feature map.

Gradient Calculation

Gradient computation is primarily used to determine the watermark embedding strength for each pixel. Since modifications in smooth regions of an image are more likely to cause noticeable visual changes, the watermark embedding should focus on complex regions while avoiding smooth areas to ensure high visual quality of the watermarked image. We use the gradient magnitude at each pixel position to measure the complexity of the region: the larger the absolute gradient value, the higher the texture complexity, and vice versa. Denoting the image gradient as $G \in \mathbb{R}^{m \times n \times c}$, then after the encoder E generates the watermark residual R, the adaptive watermark residual can be expressed as:

$$R = G * R \tag{3}$$

To efficiently compute the image gradient, we leverage TensorFlow's built-in automatic differentiation mechanism (Griewank and Walther 2008; Baydin et al. 2018). Using the encoder E as the computational process, we record the forward propagation and then utilize backpropagation through the computational tape to obtain the gradient values.

Residual Iterative Optimization

This section introduces the proposed robust watermark iterative framework, detailing the design of the encoder, robust weight calculation, decoder, and loss functions. For clarity, we denote the encoder, noise layer, and decoder as E, N and D, respectively. Ideally, the watermarked image X_1 generated by the encoder E, is related to the original image X by a residual R, expressed as:

$$X_1 = E(X) = X + R \tag{4}$$

The fundamental problem is to find a function E such that the space of the original image X can be mapped to the space of the watermarked image X_1:

$$E : X \in \mathbb{R}^{m \times n \times c} \rightarrow X_1 \in \mathbb{R}^{m \times n \times c} \tag{5}$$

such that

$$\phi_i(X, X_1) > \beta_i \tag{6}$$

where ϕ_i is a conditional restriction function, such as watermark accuracy, average peak signal-to-noise ratio (PSNR), structural similarity (SSIM) (Wang et al. 2004), etc., and β_i is a real number. This problem is equivalent to finding a function E that maps the zero space to the residual space R:

$$E : 0 \in \mathbb{R}^{m \times n \times c} \rightarrow X_1 - X = R \in \mathbb{R}^{m \times n \times c} \tag{7}$$

the task is to enable the encoder E to determine the optimal residual R such that $\phi_i(X, X + R) > \beta_i$. Recognizing that Equation (7) maps from the zero space to the residual space R, a natural question arises: can it map from any arbitrary distribution space to the residual space? The answer is affirmative. Transforming Equation (7):

$$E : 0 + \theta \in \mathbb{R}^{m \times n \times c} \rightarrow R + \theta \in \mathbb{R}^{m \times n \times c} \tag{8}$$

suggests that finding R involves an intermediate distribution θ, which is further optimized into the residual space R. To find the optimal residual R such that $X_1 = X + R$ satisfies the constraints $\phi_i(X, X + R) > \beta_i$, we propose the following iterative optimization format:

$$R^{(k+1)} = \alpha * \sum_{i}^{N_n} \omega_i R_i^{(k)}, k = 1, 2, 3, ... \tag{9}$$

where

$$R_i^{(k)} = G * E(R_i^{(k-1)}, W), k = 1, 2, 3, ... \quad (10)$$

ω_i represents the residual weights of the i-th attack type, α denotes the embedding strength, and G is the gradient. The extracted watermark W' is given by

$$W' = D(N(X + R^{(k)})) \quad (11)$$

Using Equation (9), starting from an initial $R^{(0)}$, the iterative method enables finding the optimal solution. Based on Equation (9), we design an encoder E, noise layer N, and decoder D to guide the network in finding the optimal residual R that satisfies the constraints. The following sections will provide a detailed explanation of these components.

Encoder The encoder E serves as a mapping from an arbitrary space to the residual space, primarily for generating watermark residuals and robust residuals. To achieve high visual quality for the watermarked images, the watermark residual should be as minimal as possible. Simultaneously, to enhance the robustness of the watermark residual, it is necessary to simulate image distortions caused by various attack types and refine the pixel values of the residuals, thereby obtaining robust residuals. To generate watermark residuals, we use multiple convolutional layers (He et al. 2016a,b) to extract residual features and establish relationships between each feature map and the watermark W_2. This design facilitates the localization of watermark positions during decoding and enables fast extraction of watermark information. Specifically, for each feature map x_i in the encoder:

$$f_E : x_i \in \mathbb{R}^{m_i \times n_i \times c_i} \rightarrow x_i' \in \mathbb{R}^{m_i \times n_i \times c_i + 3} \quad (12)$$

where $m_i \times n_i \times c_i$ is the resolution of the feature map x_i of the current layer i.

To ensure the robustness of the watermark residuals, we incorporate robust weight calculations for various attack types into the encoder, as shown in the *Stage-4* of Figure 2. This process involves simulating image distortions caused by each attack type and extracting watermark information, enabling the computation of the cross-entropy loss between the extracted watermark and the original watermark. This loss serves as the robust weight for the current attack type and is also treated as a local loss for joint optimization within the overall framework. Specifically, for each attack type, the distortion simulation is as follows:

$$f_N : X + R^{(k)} \in \mathbb{R}^{m \times n \times c} \rightarrow X' \in \mathbb{R}^{m \times n \times c} \quad (13)$$

where X' is the watermarked image subjected to attack type N_i. The attacked image X' is fed into the decoder to extract the watermark $W' = D(X')$(the structure of D will be described later). The robust weight for the current attack type N_i then calculated based on this extraction:

$$\omega_i = [softmax(\omega)]_i = \frac{\exp(\omega_i)}{\sum_j \exp(\omega_j)} = \frac{\exp(Acc_i)}{\sum_j \exp(Acc_j)} \quad (14)$$

where Acc_j is the accuracy or cross entropy of the original watermark W and the extracted watermark W'.

Decoder The decoder is responsible for extracting the watermark from the watermarked image. To facilitate rapid localization of the watermark positions and efficient extraction of watermark information, the decoder employs a multilayer deconvolution (Noh, Hong, and Han 2015) structure, with the number of channels symmetrically aligned with the encoder. This design ensures that the decoder can effectively return to the feature space of the encoder during feature extraction, enabling the retrieval of the watermark information embedded in the channels. Specifically, for each feature map x_i in the decoder:

$$f_D : x_i \in \mathbb{R}^{m_i \times n_i \times c_i} \rightarrow W_i' \in \mathbb{R}^{m_i \times n_i \times 3} \quad (15)$$

where W_i' represents the watermark information decoupled from the current feature map x_i.

Since each feature map in decoder D can decouple watermark information, our framework incorporates an aggregation layer at the final decoder stage to fuse these watermark components, thereby significantly enhancing robustness. The aggregation layer employs dual operations: (1) channel-wise summation and (2) channel-wise multiplication:

$$W'_{sum} = \sum_{i=1}^{l} W_i' \quad \text{and} \quad W'_{prod} = \prod_{i=1}^{l} W_i' \quad (16)$$

where l denotes the number of convolutional layers in the decoder D. The aggregated features W'_{sum} and W'_{prod} are then concatenated and fed into a final dense layer (whose output dimension matches the original watermark size), followed by a Sigmoid activation function to produce the extracted watermark. This aggregation layer effectively suppresses anomalous watermark components, ensuring stable and robust watermark generation.

Loss Function The proposed method employs an end-to-end joint training approach to simultaneously optimize the encoder and decoder. The loss function primarily consists of two components: image loss and watermark loss.

Image loss includes mean squared error (MSE) (Goodfellow 2016) loss $\mathcal{L}_1$ and PSNR (Wang et al. 2004) loss $\mathcal{L}_2$, both of which measure the visual quality between the original image X and the watermarked image X'. The $\mathcal{L}_1$ is expressed as:

$$\begin{aligned} \mathcal{L}_1 &= \frac{1}{m \times n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (X'(i,j) - X(i,j))^2 \\ &= \frac{1}{m \times n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} R^2 \end{aligned} \quad (17)$$

the $\mathcal{L}_2$ is expressed as:

$$\mathcal{L}_2 = 1/PSNR(X, X') \quad (18)$$

Watermark loss is optimized using cross-entropy (Mao, Mohri, and Zhong 2023) and includes global and local watermark losses. The global watermark loss measures the

cross-entropy between the original watermark W and the final extracted watermark W':

$$\mathcal{L}_3 = \frac{1}{L}\sum_{j=0}^{L-1} -w_j \log(w'_j) - (1-w_j)\log(1-w'_j) \quad (19)$$

where L is the length of the watermark W, w'_j represents the watermark extracted by the decoder D at the j-th position, and its value is 0 or 1.

The local watermark loss evaluates the cross-entropy between the original watermark W and the extracted watermark W'_{N_i} during robust weight calculation:

$$\begin{aligned}\mathcal{L}_4 &= \sum_{i=1}^{n} \mathcal{L}_{N_i} \\ &= \frac{1}{L}\sum_{i=1}^{n}\sum_{j=0}^{L-1} -w_{i,j}\log(w'_{i,j}) - (1-w_{i,j})\log(1-w'_{i,j})\end{aligned} \quad (20)$$

where $w'_{i,j}$ represents the watermark bit extracted at the j-th position under the i-th type noise N_i, whose value is 0 or 1. $\mathcal{L}_{N_i}$ serves as the weight ω_i of each robust residual R_{N_i}, which is dynamically optimized with the number of iterations.

Therefore, the final loss function of the network is:

$$\mathcal{L}_{Total} = \lambda_1\mathcal{L}_1 + \lambda_2\mathcal{L}_2 + \lambda_3\mathcal{L}_3 + \lambda_4\mathcal{L}_4 \quad (21)$$

where λ_1, λ_2, λ_3 and λ_4 are the corresponding loss weights, whose initial values are $(1.5, 1.0, 1.0, 1.0)$. These weights can be adjusted during training to balance the contributions of each loss component.

Experiment

Experimental Setup

Dataset The datasets include Mirflickr (D1) (Huiskes and Lew 2008), BOSSBase (D3) (Bas, Filler, and Pevnỳ 2011) and COCO (D2) (Lin et al. 2014). For the training set, we randomly selected 2,000 color images from the Mirflickr database. For the test set, we randomly selected 100 images from each of the Mirflickr, BOSSBase and COCO databases. Additionally, all input images require resizing to 400×400 resolution to satisfy the method's specifications.

Evaluation Metric The evaluation methods primarily focus on the visual quality and robustness. For visual quality, we use the PSNR and SSIM (Wang et al. 2004) as metrics, where higher values indicate better visual quality of the watermarked images. For robustness, we measure the average bit accuracy, which evaluates the accuracy of watermark extraction. Additionally, tests are conducted on multiple datasets to validate the generalization.

Implementation Detail The experiments are conducted on a platform running the Windows 10 operating system, equipped with an Intel(R) Xeon(R) Gold 6161 CPU @ 2.20GHz and 128 GB of memory. The implementation is carried out using Python 3.6. For training, the hyperparameter settings are as follows: the optimizer used is Adam, with a learning rate of 0.0001, a batch size of 1, and 140,000 iterations. The watermark length is 100, the convolution kernel size is 3×3, the stride is 1, and the embedding strength is 1.0.

Dataset	Metric	0.2	0.4	0.6	0.8	1.0	1.2	1.4	1.6	1.8	2.0
D1	PSNR↑	50.48	48.14	45.86	43.79	42.08	40.75	39.49	38.48	37.56	36.71
	SSIM↑	1.0	1.0	0.99	0.99	0.98	0.98	0.98	0.97	0.97	0.96
D2	PSNR↑	50.26	47.82	45.45	43.43	41.76	40.33	39.10	38.06	37.16	36.32
	SSIM↑	1.0	1.0	0.99	0.99	0.99	0.98	0.98	0.98	0.97	0.97
D3	PSNR↑	50.31	47.98	45.66	43.67	42.02	40.59	39.40	38.39	37.43	36.58
	SSIM↑	1.0	1.0	0.99	0.99	0.99	0.98	0.98	0.97	0.97	0.96

Table 1: Visual quality with different embedding strengths.

Dataset	α	Identity	JPEG	Gaussian noise	Gaussian filter	Crop	Cropout	Dropout	Scaling
D1	1.0	100.0	99.66	99.96	99.99	99.71	99.98	99.99	99.94
	0.8	99.98	99.16	99.93	99.96	99.81	99.98	100.0	99.82
	0.6	99.95	97.13	99.71	99.98	99.12	99.93	99.97	99.33
	0.4	99.83	91.39	98.81	99.73	97.70	99.84	99.77	97.63
	0.2	97.87	75.78	91.17	97.87	93.29	97.93	97.69	91.44
D2	1.0	100.0	99.54	99.98	99.98	99.88	99.98	99.99	99.94
	0.8	99.98	98.87	99.94	99.98	99.57	100.0	99.99	99.76
	0.6	99.88	96.51	99.67	99.88	98.74	99.91	99.91	99.20
	0.4	99.49	90.79	98.32	99.39	97.42	99.35	99.42	97.21
	0.2	96.39	76.68	90.37	96.09	91.25	96.08	95.99	89.38
D3	1.0	100.0	99.80	99.99	100.0	99.97	100.0	100.0	100.0
	0.8	100.0	99.32	99.99	100.0	99.90	100.0	100.0	99.98
	0.6	99.99	97.32	99.91	100.0	99.80	100.0	100.0	99.81
	0.4	99.93	91.59	99.19	99.94	98.85	99.90	99.91	99.19
	0.2	98.87	77.20	93.52	98.91	95.71	98.94	98.88	95.53

Table 2: Robustness with different embedding strengths (%).

Visual Quality

This section presents the visual effects of watermarked images generated by the proposed method under varying embedding strengths. As shown in Table 1, our method achieves high-quality visual results across multiple datasets, demonstrating strong generalization capabilities. Notably, at an embedding strength of α=1.0, the visual quality achieves PSNR >41dB and SSIM>0.98. When increasing to α=2.0, the metrics remain excellent with PSNR>36dB and SSIM>0.96. These results demonstrate that our model maintains superior image generation quality even under high embedding strengths.

Robustness

To evaluate robustness, we tested the watermark extraction accuracy under various noise attacks, including Identity (no distortion), JPEG compression (quality factor QF=50), Gaussian noise (variance σ=0.02), Gaussian filtering (variance σ=0.02, kernel size k=7), Dropout (rate p=0.9), Cropout (rate p=0.9), Crop (rate p=0.03), and Scaling (scaling factor p=0.5) (Fang et al. 2023; Ma et al. 2025a). To verify that the proposed method can produce high-visual-quality watermarked images while maintaining excellent robustness, we primarily assessed robustness at embedding strengths $\alpha \leq 1.0$.

As shown in Table 1 and Table 2, the proposed method demonstrates strong robustness across different types of at-

Dataset	Method	$m \times n$	L	PSNR↑	SSIM↑	Identity	JPEG	Gaussian noise	Gaussian filter	Crop	Cropout	Dropout	Scaling	Average
D1	HiDDeN	128×128	30	32.559	0.937	93.75	91.83	97.76	88.03	75.37	91.53	97.50	78.23	89.25
	StegaStamp	400×400	100	29.302	0.890	99.93	99.89	99.84	99.92	99.61	99.32	99.85	99.84	99.78
	MBRS	256×256	256	41.99	0.988	100.0	99.36	81.12	55.47	49.78	94.22	100.0	49.93	78.74
	CIN	128×128	30	41.437	0.980	100.0	99.67	100.0	100.0	99.93	100.0	98.30	98.93	99.60
	ARWGAN	128×128	30	22.647	0.877	99.16	90.27	94.50	99.13	95.80	98.87	99.23	95.83	96.60
	Document	400×400	100	32.133	0.991	99.13	99.16	98.83	95.45	52.85	99.22	99.21	98.53	92.80
	De-END	128×128	64	48.750	0.951	98.48	70.41	91.34	99.90	–	97.56	100.0	–	92.95
	ST-DCN*	224×224	196	35.973	0.992	100.0	99.93(40)	98.09(0.04)	100.0	96.75(0.2)	96.87(0.5)	95.44(0.7)	99.38(0.4)	98.31
	Ours	400×400	100	42.084	0.987	100.0	99.66	99.96	99.99	99.71	99.98	99.99	99.94	**99.90**
D2	HiDDeN	128×128	30	32.893	0.947	98.33	92.73	97.93	89.93	74.22	92.23	98.23	79.13	90.34
	StegaStamp	400×400	100	30.068	0.910	99.88	99.82	99.88	99.88	99.62	99.18	99.79	99.90	99.74
	MBRS	256×256	256	42.008	0.988	100.0	99.62	79.57	100.0	49.61	93.96	100.0	50.36	84.14
	CIN	128×128	30	41.909	0.983	100.0	99.77	100.0	100.0	99.97	100.0	97.40	99.27	99.55
	ARWGAN	128×128	30	31.736	0.959	99.20	94.86	96.13	99.20	96.23	99.23	99.06	96.30	97.53
	Document	400×400	100	32.299	0.993	99.96	99.95	99.44	99.89	53.19	99.42	99.51	99.42	93.85
	De-END	128×128	64	49.013	0.990	98.41	70.36	96.58	99.98	–	97.75	100.0	–	93.85
	ST-DCN*	224×224	196	35.973	0.992	100.0	99.93(40)	98.09(0.04)	100.0	96.75(0.2)	96.87(0.5)	95.44(0.7)	99.38(0.4)	98.31
	Ours	400×400	100	41.762	0.988	100.0	99.54	99.98	99.98	99.88	99.98	99.99	99.94	**99.91**
D3	HiDDeN	128×128	30	35.940	0.970	97.00	87.27	98.76	88.77	75.70	92.77	98.47	77.53	89.53
	StegaStamp	400×400	100	30.068	0.910	99.99	99.97	99.92	99.97	99.75	99.60	99.93	99.91	99.88
	MBRS	256×256	256	42.975	0.988	100.0	99.87	78.99	99.30	49.24	94.42	100.0	50.18	84.00
	CIN	128×128	30	43.219	0.984	100.0	99.60	100.0	100.0	100.0	100.0	96.40	99.30	99.41
	ARWGAN	128×128	30	31.736	0.959	95.87	85.93	93.09	99.30	97.20	99.17	98.993	97.70	95.91
	Document	400×400	100	32.299	0.993	99.96	99.98	99.95	99.89	53.19	99.97	99.99	99.96	94.11
	De-END	128×128	64	49.126	0.981	100.0	65.18	98.37	100.0	–	98.32	100.0	–	93.65
	ST-DCN*	224×224	196	35.973	0.992	100.0	99.93(40)	98.09(0.04)	100.0	96.75(0.2)	96.87(0.5)	95.44(0.7)	99.38(0.4)	98.31
	Ours	400×400	100	42.023	0.987	100.0	99.80	99.99	100.0	99.97	100.0	100.0	100.0	**99.97**

Table 3: Visual quality and robustness of different watermarking methods (%).

tacks. When the embedding strength α=1.0, the accuracy under all noise attacks exceeds 99%, with a visual quality of PSNR over 41dB and SSIM above 0.98. Even at a lower embedding strength of α=0.4, the method achieves over 90% accuracy under all noise attacks, with a PSNR exceeding 47dB and SSIM above 0.99. These results indicate that our method maintains excellent robustness while generating high-visual-quality watermarked images, offering a flexible range of generation schemes to meet diverse practical application requirements.

Comparative Experiment

To highlight the superiority of the proposed method, we selected state-of-the-art and representative deep learning watermarking methods for comparison, including HiDDeN (Zhu et al. 2018), StegaStamp (Tancik, Mildenhall, and Ng 2020), MBRS (Jia, Fang, and Zhang 2021), CIN (Ma et al. 2022), ARWGAN (Huang et al. 2023), Document (Ge et al. 2023), De-END (Fang et al. 2023) and ST-DCN (Ma et al. 2025a). We utilized the publicly available pre-trained models provided by these methods and express our gratitude for their open-source contributions.

Table 3 presents the visual quality and robustness against noise attacks for watermarked images generated by different methods. In the table, entries marked with an asterisk indicate results derived from the original papers. As shown, our method achieves superior robustness when the embedding strength $\alpha = 1.0$, with an overall average accuracy exceeding 99.90%, outperforming existing methods. In terms of visual quality, our method achieves higher PSNR and SSIM values compared to HiDDeN, StegaStamp, and ARWGAN, while being slightly lower than MBRS.

These experimental results demonstrate the exceptional overall performance of our proposed method, primarily attributed to the design of the encoder structure. By employing a parallel architecture, the robustness against each type of noise attack can be independently optimized. Additionally, the adaptive embedding strategy further ensures the visual quality of the generated watermarked images. In summary, our method achieves simultaneous optimization of both visual quality and robustness.

Ablation Experiment

Initial Iteration State Figure 3 (a) illustrates the convergence of visual quality when four different sampling spaces are used as the initial iterative optimization targets, while Table 4 presents the corresponding robustness. Experimental results demonstrate that our network can optimize from any initial state, achieving both high visual quality and robustness upon convergence. As theoretically analyzed in Equation (8), the adopted iterative residual optimization enables the network to progressively approach the neighborhood of optimal residuals from arbitrary initial states through loss minimization, thereby satisfying all constraints including visual quality and robustness requirements.

Image Gradient Figure 3 (b) illustrates the convergence of visual quality with and without gradient G as the adap-

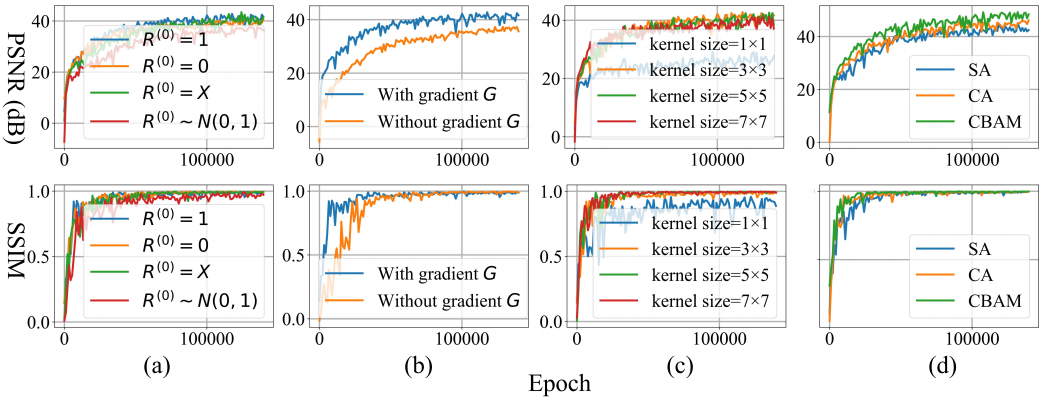

Figure 3: Convergence of PSNR and SSIM corresponding to different ablation modules.

Ablation module	Dataset	Identity	JPEG	Gaussian noise	Gaussian filter	Crop	Cropout	Dropout	Scaling
$X^{(0)}$=1	D1	100.0	99.66	99.96	99.99	99.71	99.98	99.99	99.94
	D2	100.0	99.54	99.98	99.98	99.88	99.98	99.99	99.94
	D3	100.0	99.80	99.99	100.0	99.97	100.0	100.0	100.0
$X^{(0)}$=0	D1	100.0	98.64	99.99	100.0	86.21	100.0	100.0	99.84
	D2	100.0	99.09	99.98	99.98	88.13	99.99	100.0	99.77
	D3	100.0	99.40	100.0	100.0	93.48	100.0	100.0	99.98
$X^{(0)}$=X	D1	100.0	99.85	100.0	100.0	98.02	100.0	100.0	99.98
	D2	100.0	99.85	99.99	100.0	97.46	100.0	100.0	99.98
	D3	100.0	99.87	100.0	100.0	99.28	100.0	100.0	100.0
$X^{(0)}\sim N(0,1)$	D1	100.0	99.50	99.99	100.0	99.59	100.0	100.0	99.44
	D2	100.0	99.53	99.98	99.99	99.59	100.0	100.0	99.60
	D3	100.0	99.76	100.0	100.0	99.91	100.0	100.0	99.93
Without G	D1	100.0	99.88	99.99	99.97	91.13	100.0	100.0	99.86
	D2	100.0	99.93	100.0	100.0	90.09	99.98	100.0	99.74
	D3	100.0	99.96	100.0	100.0	91.91	100.0	100.0	99.97
1×1	D1	99.63	98.80	99.67	99.77	95.38	99.60	99.81	99.19
	D2	99.89	99.42	99.86	99.90	95.94	99.87	99.84	99.57
	D3	99.98	99.86	99.94	99.98	97.47	99.97	99.97	99.94
5×5	D1	100.0	99.53	100.0	100.0	96.74	100.0	100.0	99.90
	D2	100.0	99.49	99.98	99.99	97.15	100.0	100.0	99.86
	D3	100.0	99.81	99.99	100.0	98.85	100.0	100.0	99.98
7×7	D1	100.0	99.52	99.95	100.0	89.27	99.99	100.0	99.86
	D2	99.97	99.43	99.97	99.99	88.64	99.98	99.96	99.82
	D3	100.0	99.74	99.98	100.0	91.10	100.0	100.0	99.97
SA	D1	99.99	93.33	99.93	99.99	98.91	99.99	99.99	99.37
	D2	99.93	94.02	99.78	99.88	98.57	99.94	99.96	99.24
	D3	100.0	96.33	99.98	100.0	99.63	100.0	100.0	99.94
CA	D1	100.0	94.08	99.92	100.0	99.10	100.0	99.99	99.86
	D2	99.97	92.84	99.85	99.93	99.15	99.96	99.96	99.79
	D3	99.99	94.40	99.99	100.0	99.59	99.98	100.0	99.94
CBAM	D1	99.62	77.88	98.45	99.64	94.71	99.59	99.57	98.11
	D2	99.25	79.00	98.28	99.38	95.42	99.24	99.21	97.85
	D3	99.93	76.56	99.61	99.92	98.84	99.94	99.91	99.73

Table 4: Robustness of different ablation modules (%).

tive embedding strength, while Table 4 presents the corresponding robustness. The results demonstrate that leveraging image gradients as adaptive embedding strength significantly enhances both visual quality and robustness. The underlying rationale is straightforward: without adaptive embedding, the amount of watermark is uniformly distributed across all spatial locations of an image, making it difficult to avoid embedding in smooth regions. In contrast, adaptive embedding effectively addresses this issue by concentrating watermark insertion in perceptually complex regions.

Receptive Field In addition, we evaluated the impact of the receptive field size of the network parameters. Figure 3 (c) and Table 4 demonstrate the visual quality convergence and robustness, respectively. The results indicate that a receptive field size of 3×3 achieves the optimal performance. A smaller receptive field fails to adequately optimize both visual quality and robustness. In contrast, a larger receptive field increases the number of trainable parameters, making the network more prone to overfitting.

Attention Mechanism Finally, we evaluated the performance of different attention mechanisms, including channel attention (CA), spatial attention (SA), and convolutional block attention module (CBAM) (Woo et al. 2018). Given that the encoder primarily generates a robust residual R, the attention mechanisms were applied specifically to R. The results are presented in Figure 3 (d) and Table 4. As observed, incorporating attention mechanisms can improve the visual quality. However, robustness against certain types of attacks may be compromised. Therefore, in our proposed method, excellent performance can still be achieved without the use of attention mechanisms.

Conclusion

To address the challenges of visual quality, robustness, and generalization in deep learning watermarking methods, we propose an adaptive robust iterative watermarking framework. Specifically, we develop a robust iterative watermarking scheme and design an encoder structure to generate watermarked images with strong robustness. Additionally, we leverage image gradients to determine the embedding strength at each pixel, further enhancing the visual quality of the watermarked images. Extensive experiments demonstrate the robustness of our method against various noise attacks and its generalization capability across datasets, while maintaining high imperceptibility in the generated watermarked images. Furthermore, ablation studies validate the effectiveness of our network design. In summary, our watermarking framework significantly improves visual quality and robustness, charting a promising path for future advancements.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No.62441237, No.62261160653), and the Guangdong Provincial Key Laboratory of Information Security Technology (No.2023B1212060026).

References

Bas, P.; Filler, T.; and Pevnỳ, T. 2011. " Break our steganographic system": the ins and outs of organizing BOSS. In *International Workshop on Information Hiding*, 59–70. Springer.

Baydin, A. G.; Pearlmutter, B. A.; Radul, A. A.; and Siskind, J. M. 2018. Automatic differentiation in machine learning: a survey. *Journal of Machine Learning Research*, 18(153): 1–43.

Dathathri, S.; See, A.; Ghaisas, S.; Huang, P.-S.; McAdam, R.; Welbl, J.; Bachani, V.; Kaskasoli, A.; Stanforth, R.; Matejovicova, T.; et al. 2024. Scalable watermarking for identifying large language model outputs. *Nature*, 634(8035): 818–823.

Fang, H.; Jia, Z.; Qiu, Y.; Zhang, J.; Zhang, W.; and Chang, E.-C. 2023. De-END: Decoder-Driven Watermarking Network. *IEEE Transactions on Multimedia*, 25: 7571–7581.

Fernandez, P.; Couairon, G.; Jégou, H.; Douze, M.; and Furon, T. 2023. The stable signature: Rooting watermarks in latent diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 22466–22477.

Fu, L.; Liao, X.; Guo, J.; Dong, L.; and Qin, Z. 2024. WaveRecovery: Screen-shooting Watermarking based on Wavelet and Recovery. *IEEE Transactions on Circuits and Systems for Video Technology*.

Ge, S.; Xia, Z.; Fei, J.; Tong, Y.; Weng, J.; and Li, M. 2023. A robust document image watermarking scheme using deep neural network. *Multimedia Tools and Applications*, 82(25): 38589–38612.

Golda, A.; Mekonen, K.; Pandey, A.; Singh, A.; Hassija, V.; Chamola, V.; and Sikdar, B. 2024. Privacy and Security Concerns in Generative AI: A Comprehensive Survey. *IEEE Access*.

Goodfellow, I. 2016. *Deep Learning*. MIT press.

Griewank, A.; and Walther, A. 2008. *Evaluating derivatives: principles and techniques of algorithmic differentiation*. SIAM.

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

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016b. Identity mappings in deep residual networks. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, 630–645. Springer.

Hosny, K. M.; Magdi, A.; ElKomy, O.; and Hamza, H. M. 2024. Digital image watermarking using deep learning: A survey. *Computer Science Review*, 53: 100662.

Hu, K.; Wang, M.; Ma, X.; Chen, J.; Wang, X.; and Wang, X. 2024. Learning-based image steganography and watermarking: A survey. *Expert Systems with Applications*, 123715.

Huang, J.; Luo, T.; Li, L.; Yang, G.; Xu, H.; and Chang, C.-C. 2023. ARWGAN: Attention-guided robust image watermarking model based on GAN. *IEEE Transactions on Instrumentation and Measurement*, 72: 1–17.

Huiskes, M. J.; and Lew, M. S. 2008. The mir flickr retrieval evaluation. In *Proceedings of the 1st ACM International Conference on Multimedia Information Retrieval*, 39–43.

Jang, Y.; Lee, D. I.; Jang, M.; Kim, J. W.; Yang, F.; and Kim, S. 2024. WateRF: Robust Watermarks in Radiance Fields for Protection of Copyrights. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12087–12097.

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.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, 740–755. Springer.

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*, 1509–1517.

Lyu, L.; Chen, C.; and Fu, J. 2023. A Pathway Towards Responsible AI Generated Content. In *IJCAI*, 7033–7038.

Ma, L.; Fang, H.; Wei, T.; Yang, Z.; Ma, Z.; Zhang, W.; and Yu, N. 2025a. A Geometric Distortion Immunized Deep Watermarking Framework with Robustness Generalizability. In *European Conference on Computer Vision*, 268–285. Springer.

Ma, R.; Guo, M.; Hou, Y.; Yang, F.; Li, Y.; Jia, H.; and Xie, X. 2022. Towards blind watermarking: Combining invertible and non-invertible mechanisms. In *Proceedings of the 30th ACM International Conference on Multimedia*, 1532–1542.

Ma, Z.; Fang, H.; Yang, X.; Chen, K.; and Zhang, W. 2025b. RoPaSS: Robust Watermarking for Partial Screen-shooting Scenarios. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Ma, Z.; Zhu, Y.; Luo, G.; Liu, X.; Schaefer, G.; and Fang, H. 2023. Robust steganography without embedding based on secure container synthesis and iterative message recovery.

Mao, A.; Mohri, M.; and Zhong, Y. 2023. Cross-entropy loss functions: Theoretical analysis and applications. In *International Conference on Machine Learning*, 23803–23828. PMLR.

Noh, H.; Hong, S.; and Han, B. 2015. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, 1520–1528.

Qin, C.; Li, X.; Zhang, Z.; Li, F.; Zhang, X.; and Feng, G. 2024. Print-Camera Resistant Image Watermarking With Deep Noise Simulation and Constrained Learning. *IEEE Transactions on Multimedia*, 26: 2164–2177.

Tan, C.; Zhao, Y.; Wei, S.; Gu, G.; Liu, P.; and Wei, Y. 2024. Rethinking the Up-Sampling Operations in CNN-based Generative Network for Generalizable Deepfake Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 28130–28139.

Tancik, M.; Mildenhall, B.; and Ng, R. 2020. Stegastamp: Invisible hyperlinks in physical photographs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2117–2126.

Wan, W.; Wang, J.; Zhang, Y.; Li, J.; Yu, H.; and Sun, J. 2022. A comprehensive survey on robust image watermarking. *Neurocomputing*, 488: 226–247.

Wang, B.; Wu, Y.; and Wang, G. 2023. Adaptor: Improving the Robustness and Imperceptibility of Watermarking by the Adaptive Strength Factor. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(11): 6260–6272.

Wang, G.; Ma, Z.; Liu, C.; Yang, X.; Fang, H.; Zhang, W.; and Yu, N. 2024a. MuST: Robust Image Watermarking for Multi-Source Tracing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 5364–5371.

Wang, K.; Wu, S.; Yin, X.; Lu, W.; Luo, X.; and Yang, R. 2024b. Robust image watermarking with synchronization using template enhanced-extracted network. *IEEE Transactions on Circuits and Systems for Video Technology*.

Wang, T.; Huang, M.; Cheng, H.; Zhang, X.; and Shen, Z. 2024c. LampMark: Proactive Deepfake Detection via Training-Free Landmark Perceptual Watermarks. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 10515–10524.

Wang, T.; Zhang, Y.; Qi, S.; Zhao, R.; Xia, Z.; and Weng, J. 2024d. Security and privacy on generative data in aigc: A survey. *ACM Computing Surveys*, 57(4): 1–34.

Wang, Z.; Bovik, A. C.; Sheikh, H. R.; and Simoncelli, E. P. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612.

Woo, S.; Park, J.; Lee, J.-Y.; and Kweon, I. S. 2018. Cbam: Convolutional block attention module. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 3–19.

Wu, S.; Lu, W.; and Luo, X. 2025. Robust Watermarking Based on Multi-layer Watermark Feature Fusion. *IEEE Transactions on Multimedia*, 1–14.

Wu, X.; Liao, X.; and Ou, B. 2023. Sepmark: Deep separable watermarking for unified source tracing and deepfake detection. In *Proceedings of the 31st ACM International Conference on Multimedia*, 1190–1201.

Ye, X.; Yan, Y.; Li, J.; and Jiang, B. 2024. Privacy and personal data risk governance for generative artificial intelligence: A Chinese perspective. *Telecommunications Policy*, 48(10): 102851.

Yin, X.; Wu, S.; Wang, K.; Lu, W.; Zhou, Y.; and Huang, J. 2023. Anti-rounding image steganography with separable fine-tuned network. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(11): 7066–7079.

Zhang, X.; Li, R.; Yu, J.; Xu, Y.; Li, W.; and Zhang, J. 2024. Editguard: Versatile image watermarking for tamper localization and copyright protection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11964–11974.

Zhu, J.; Kaplan, R.; Johnson, J.; and Fei-Fei, L. 2018. HiDDeN: Hiding Data With Deep Networks. In *European Conference on Computer Vision*.