A Watermark Updating Framework for Multi-stage Image Content Distribution

Yanyan Liu[1], Bin Liu[1] *, Jie Zhang[2], Xiang Zhang[1], Zehua Ma[1], Nenghai Yu[1]
[1] University of Technology and Science of China. [2]CFAR and IHPC, ASTAR

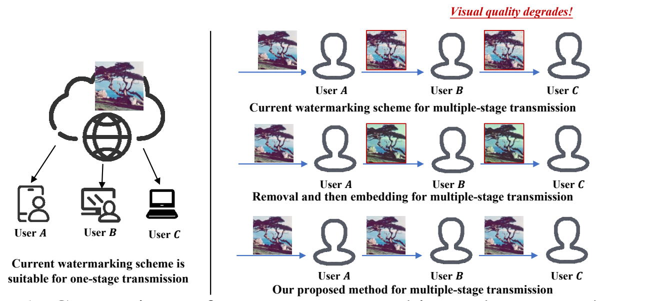
***Abstract*—Deep image watermarking embeds identification data into images to facilitate source tracking. However, existing schemes are primarily designed for single-stage transmission scenarios, and in practical multi-stage distribution requirements, current methods degrade image quality and reduce watermark extraction accuracy. In this paper, we introduces WATERUP, a deep watermark updating framework. WATERUP automatically updates watermark information as the image is transmitted, preserving image quality while accurately recording the transmission path for traceability. The core of WATERUP is a flow-based encoder-decoder (FED), which utilizes a forward and backward network to enable efficient watermark updating with minimal computational and storage demands. Experimental results show that WATERUP outperforms state-of-the-art methods, maintaining high visual quality with a PSNR exceeding 38 dB across multiple transmissions.**

***Index Terms*—Watermarking, Watermark Updating**

I. INTRODUCTION

Nowadays, the transmission of images has become increasingly convenient, creating opportunities for attackers. When images containing sensitive content are leaked or attacked, the resulting economic losses can be substantial. A study by IBM and the Ponemon Institute documented a 12% increase in data breach costs over the past five years [1]. While complete prevention of data breaches is unrealistic, a more practical approach is to quickly trace the source of breaches and minimize the associated losses. Digital watermarks, which embed invisible information into data, serve to protect copyright and track the source. They significantly improve the ability to respond to data leakage incidents. Current watermarking technologies [2], [3] are primarily designed for single-stage data distribution, where images are transmitted without intermediaries, as depicted in Fig. 1. However, in multi-stage scenarios, such as in government or enterprise workflows, images are routed through multiple intermediaries. Consequently, watermarks must be re-embedded with each image transmission. As illustrated in Fig. 1, repeated embedding degrades both data quality and usability and results in earlier watermarks being interpreted as noise during subsequent extractions. These challenges result in current solutions being unable to meet the traceability requirements in multi-stage distribution.

To meet traceability requirements in multi-stage distribution scenarios, we introduce the concept of **watermark updating**. Which involves automatically updating watermark information as digital content is transferred to another terminal, while preserving data quality. Specifically, the system first removes the existing watermark and then embeds new watermark information, including the identities of both the sending and receiving terminals. Traditional watermarking schemes, such as reversible watermarking [4], [5], allow the recovery of the original cover image without distortion, thereby supporting watermark updating. However, these traditional approaches, which rely on handcrafted features for embedding and extraction, exhibit weaknesses in both invisibility and robustness against complex distortions. Deep learning-based watermarking schemes [6], on the other hand, can fully leverage image features and manage more complex distortions. Nonetheless, the crucial need for watermark updating has not been adequately addressed in these deep-learning-based methods.


Visual quality degrades!
User A
User B
User C
Current watermarking scheme is suitable for one-stage transmission
Current watermarking scheme for multiple-stage transmission
Removal and then embedding for multiple-stage transmission
Our proposed method for multiple-stage transmission
Fig. 1: Comparison of current watermarking schemes and our proposed scheme in multi-stage distribution system.

Current deep learning technology most closely related to watermark update is watermark removal, which treats multiple watermarks as noise to eliminate their impact [7]. However, since watermarks are adaptively embedded and differ from common types of noise, such methods may lead to a lot of quality loss in the reconstructed images. As illustrated in Fig. 1, integrating these techniques with watermarking schemes for multi-stage distribution fails to achieve the desired outcomes. Moreover, implementing watermark update based on these solutions requires the integration of an additional watermark embedding model, which imposes extra computational and storage demands on regular office equipment. To address these challenges, we present WATERUP. WATERUP adopts a flow-based network architecture to facilitate the embedding and extraction of watermarks. Unlike traditional approaches, WATERUP innovatively integrates watermark embedding and removal into a single network. Specifically, the network's forward process performs watermark embedding, while the

* Corresponding author. This work is supported by the National Key Research and Development Program of China under Grant 2022YFB3103203.

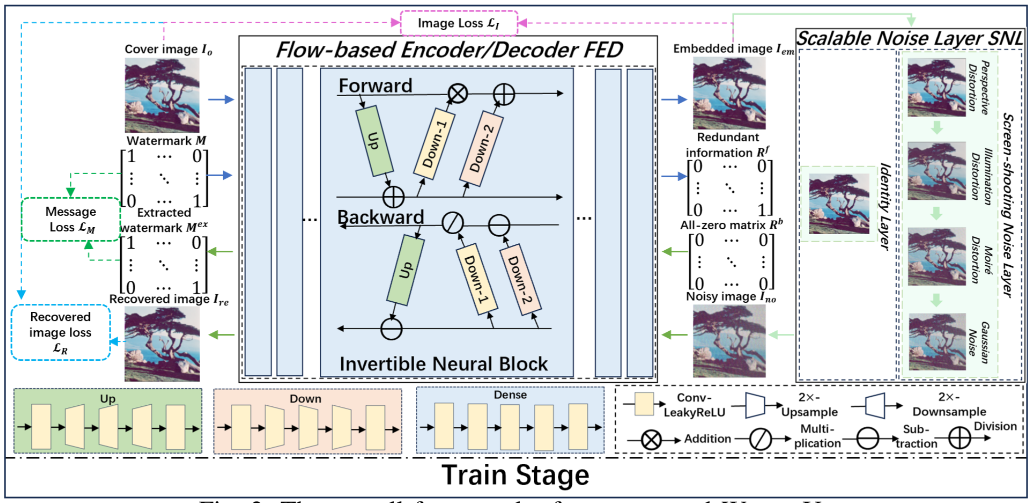


Fig. 2: The overall framework of our proposed WATERUP.

backward process handles watermark removal. This design enables WATERUP to leverage shared parameters, allowing the knowledge of watermark embedding to enhance the quality of original image restoration during watermark removal. Additionally, the bidirectional flow within a single model significantly reduces computational and storage demands. We conducted extensive experiments, demonstrating that WATERUP effectively meets the requirements for watermark updates and outperforms all existing solutions.

The contributions of this paper are as follows:

- To the best of our knowledge, this is the first work to propose watermark updating, specifically addressing the tracing needs of multi-stage distribution scenarios.
- We proposes a watermark updating framework called WATERUP. This framework is implemented using a flow-based model and a scalable noise layer, achieving high-performance watermark updating with strong scalability.
- Extensive experimental results demonstrate that the proposed scheme outperforms state-of-the-art deep learning-based watermarking methods in terms of visual quality and robustness. As a result, effective leakage traceability is supported in multi-stage distribution scenarios.

II. RELATED WORK

A. Single-stage Watermarking schemes

Current watermarking schemes prioritize both robustness and invisibility, achieving great performance in single-stage distribution scenarios. Recently, several deep learning-based watermarking schemes have been proposed. Zhu *et al.* [6] introduced the END architecture, which effectively ensures robustness against various image processing techniques by incorporating different noise layers. Jia *et al.* [2] achieved robustness against JPEG compression by alternately training the network with both “real JPEG” and “simulated JPEG” noise. And Fang *et al.* [3] proposed a noise layer named PIMoG which simulates the most impactful distortions in a differentiable manner. Additionally, Fang *et al.* [8] introduced a flow-based robust watermarking framework that ensures a tight coupling between the encoder and the decoder, resulting in high visual quality of watermarked images. However, since the aforementioned watermarking schemes cannot remove watermarks during extraction, they are unsuitable for multi-stage transmission scenarios.

B. Invisible Watermark Removal

Removal attacks aim to completely remove watermarks from watermarked images. A common approach to removing invisible watermarks is to treat the watermark as a type of noise added to the image. Attack methods based on this approach aim to preserve the visual quality of the attacked image. Quiring *et al.* [9] proposed a black-box attack method based on adversarial machine learning, which seeks to prevent watermark detection rather than removal. Additionally, Zhao *et al.* [10] presented a universal attack pattern and experimentally demonstrated that using the diffusion model yields the most effective attack results. Similarly, Li *et al.* [7] introduced a distance-guided conditional diffusion model for watermark removal, which reduces extraction accuracy to below 50% and reconstructs images that closely resemble the originals. However, in multi-stage distribution scenarios, where multiple instances of watermark removal are involved, the inevitable decline in visual quality cannot be overlooked.

III. THE PROPOSED WATERUP

A. Overview

The system overview of WATERUP is illustrated in Fig. 2. WATERUP consists of two primary components: a trainable flow-based encoder-decoder (FED) and a scalable noise layer (SNL). With each image transmission, WATERUP updates a new watermark containing the identity information of both parties involved in the transmission. This enables traceability in the event of information leaks.

The FED is composed of multiple invertible neural blocks and utilizes the same parameters for both forward encoding and backward decoding. In the forward process, the watermark message M is embedded to the cover image I_o, producing the embedded image I_{em} and a redundancy matrix R^f. Then SNL effectively distorts the watermarked image I_{em} and provides the noisy image I_{no} for decoder training. In the backward process, the FED takes the noisy image I_{no} and the all-zero matrix R^b as input, and decodes them to obtain the extracted message M^{ex} and the recovered image I_{re}. Leveraging the parameter-sharing mechanism, the image restoration process can effectively utilize the parameters knowledge from watermark embedding to achieve higher-quality image recovery.

Image information leakage often occurs through screen-shooting or direct forwarding from the local machine. To address this, we designed the SNL layer to simulate the image distortions caused by these two methods, thereby enhancing the robustness of watermark extraction. Specifically, the SNL consists of a Screen-Shooting Layer and an Identity Layer. The former simulates distortions encountered during screen capturing, while the latter simulates legitimate transmission without visual degradation. During training, one noise layer is randomly chosen from the SNL for each batch.

B. Flow-based Encoder and Decoder

As aforementioned, the FED consist of n invertible neural blocks. The structure of the i^{th} reversible neural block is shown in Fig. 3. Each block comprises an up-sub-network U_i and two down-sub-networks D_i^1 and D_i^2. The fundamental building block of each sub-network consists of six "ConvLeakyReLU" blocks, as detailed in Fig. 2. The aim of U_i is to up-sample $m_i \in R^{h\times w\times 1}$ to the same size as the image $I_o \in R^{H\times W\times 3}$. And the down-sub-networks aim to down-sample $x_{i+1} \in R^{H\times W\times 3}$ to the same size as m_i.

For the i^{th} invertible neural block in the forward-encoding process, the inputs are m_i and x_i, and the outputs are m_{i+1} and x_{i+1}. The details can be represented by follows:

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

where $\otimes$ indicates the dot product operation. After the last invertible neural block, we can obtain m_{n+1} as the redundant information R^f, and x_{n+1} as the watermarked image I_{em}.

In the backward-decoding process, the information flows are from the i^{th} invertible neural block to the $(i-1)^{th}$ invertible neural block, as shown in Fig. 3. For the first block in backward process, the inputs are an all-zero matrix $R^b \in R^{h\times w\times 1}$ and the noisy image I_{no}, and the outputs are m'_n and r_n. For the the i^{th} block, the process can be represented by the following equations:

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

After the last invertible neural block in the backward-decoding process, the output m'_1 is obtained as the extracted watermark M^{ex}, and r_1 is obtained as the recovered image I_{re}, which will be embedded with new message during watermark updating. It should be noted that an all-zero matrix is used here to ensure blind extraction, which implies that no prior information other than the I_{no} is needed for decoding.

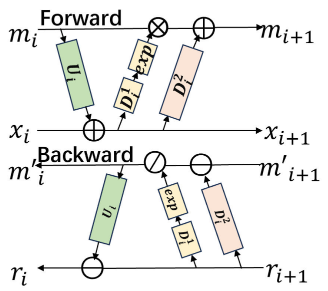


Fig. 3: The structure of the i^{th} invertible neural block.

C. Scalable Noise Layer

As previously mentioned, SNL includes several noise layers, one of which is the Identity Layer, which performs an identity transformation on the input image. For the Screen-shooting Layer, PIMoG [3] is adopted to simulate distortions occurring during the screen-shooting process. It encapsulates the primary distortions encountered into three main components: perspective distortion, illumination distortion, and moiré distortion. A differentiable formulation is proposed for each distortion type to approximate its effects. The complete noise layer comprises simulated perspective distortion, simulated illumination distortion, simulated moiré distortion, and an additional Gaussian noise layer that represents the remaining distortions. Robustness against other distortions can also be achieved by incorporating additional noise layers into the SNL.

D. Watermark Updating

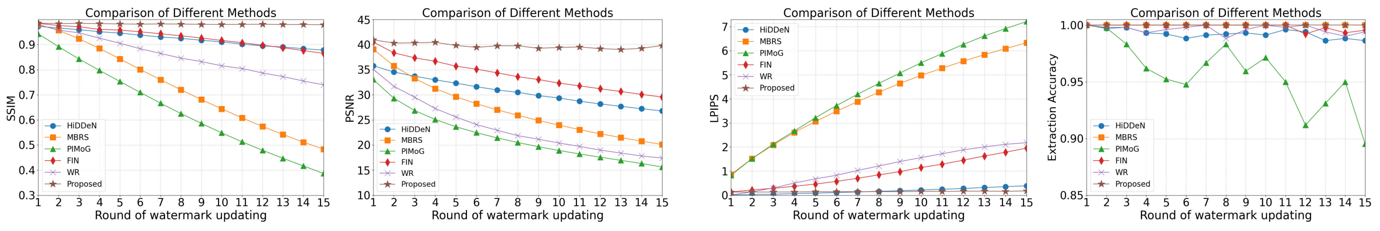
The concept of watermark updating is introduced to mitigate the impact of multiple watermarks on the quality of image. This process can be succinctly summarized as first removing the previous watermark information and then embedding a new watermark. In multi-stage distribution scenarios, images pass through several intermediary points within a system, forming a transmission chain. For the j^{th} point on the transmission chain, it receives image $I_{em_{j-1}}$ embedded with message M_{j-1} from the $(j-1)^{th}$ point and then send image I_{em_j} embedded with message M_j to the $(j+1)^{th}$ point. The process of updating watermark information can be formulated as follows:

$$\begin{aligned} I_{em_1} &= Encode(M_1, I_{co}), \\ I_{em_j} &= Encode(M_j, I_{co}) \\ &= Encode(M_j, Decode(I_{em_{j-1}})), \end{aligned} \tag{3}$$

while *Encode* and *Decode* are the forward-encoding process and backward-decoding process of the framework in Fig. 2.

E. Loss Function

As noted, two noise layers simulate different scenarios: the Identity Layer for watermark updating and the Screen-shooting Layer for leakage traceability. The loss functions for these scenarios differ. For traceability, the total loss function includes image loss and message loss to ensure invisibility and robustness. In watermark updating, the total loss also incorporates recovered image loss to maintain the quality of the watermark-free image.



(a) SSIM of different methods (b) PSNR of different methods (c) LPIPS of different methods (d) Accuracy of different methods

Fig. 4: Comparison among different methods under multiple rounds watermark updating.

1) Image Loss: The aim of forward-encoding process is to embed the watermark M into the cover image I_o to generate the watermarked image I_{em}. To achieve invisibility, the watermarked image is required to be close to the cover image, with the image loss defined as follows:

$$\mathcal{L}_{I_1} = MSE(I_o, I_{em}), \tag{4}$$

$$\mathcal{L}_{I_2} = LPIPS(I_o, I_{em}). \tag{5}$$

Here, We used two methods, *MSE* (Mean Squared Error) and *LPIPS* (Learned Perceptual Image Patch Similarity), to calculate the differences between cover images and watermarked images. The *LPIPS* function [11] measures perceptual image similarity by using neural networks to compare feature differences. λ_{I_1} and λ_{I_2} are the weights.

2) Message Loss: The backward-decoding process aims to extract watermark information M^{ex} from the noisy image I_{no}. To achieve robustness, the message loss $\mathcal{L}_M$ is defined as:

$$\mathcal{L}_I = MSE(M, M^{ex}). \tag{6}$$

3) Recovered Image Loss: Since the recovered image will be embedded with a new message during watermark updating, it should ideally be consistent with the original image. This can be achieved by inputting both the redundant information matrix and the watermarked image into FED during the backward-decoding process. However, the storage and transmission of redundant information matrix are impractical. Therefore, a restriction is imposed on the recovered image to ensure it closely approximates the cover image, as follows:

$$\mathcal{L}_R = LPIPS(I_o, I_{re}). \tag{7}$$

LPIPS measures the difference between cover image I_o and recovered image I_{re}, as the focus on higher-level features by perceptual loss aids in recovering the watermark-free image.

4) Total Loss: The total loss $\mathcal{L}_{total}$ is a sum of image loss, message loss and recovered image loss. For the purpose of traceability, the total loss is defined as follows:

$$\mathcal{L}_{total} = \lambda_{I_1}\mathcal{L}_{I_1} + \lambda_{I_2}\mathcal{L}_{I_2} + \lambda_M\mathcal{L}_M. \tag{8}$$

For watermark updating, the total loss is defined as:

$$\mathcal{L}_{total} = \lambda_{I_1}\mathcal{L}_{I_1} + \lambda_{I_2}\mathcal{L}_{I_2} + \lambda_M\mathcal{L}_M + \lambda_R\mathcal{L}_R, \tag{9}$$

where λ_{I_1}, λ_{I_2}, λ_M and λ_R are the weights.

IV. Experiments

A. Experimental Setup

1) Datasets: To train the entire network, the DIV2K dataset [12], which contains high-quality images for image super-resolution tasks, is selected as the training dataset. And the classical USC-SIPI image set [13], known for its diverse range of grayscale and color images, is used as the testing dataset.

2) Metrics: To evaluate the visual quality of the watermarked image, PSNR, SSIM, and LPIPS are utilized as the primary metrics. Higher values of PSNR and SSIM denote improved visual quality, while a higher LPIPS value signifies a reduction in visual quality. For assessing robustness, extraction bit accuracy is used as the metric, with higher accuracy indicating greater robustness.

3) The Baselines: The proposed scheme is compared with four traditional deep learning-based methods, MBRS [2], HiDDeN [6], PIMoG [3] and FIN [8]. For the strength factor in MBRS [2], we set it to 1 here. And for FIN [8], the model is trained with screen-shooting noise layer for better comparison. We also implement watermark updating by watermark removal attack [14] (WR) for comparative analysis.

4) Implementation Details: The framework is implemented by PyTorch and executed on NVIDIA RTX 3080ti [15], [16]. The width W and height H of the image are set to 128, respectively, and the length of the watermark message is set to 64 bits, i.e., h and w are both set as 8. For parameter optimization, Adam is applied. During training, the noise layer is randomly chosen from the SNL for each batch, and the corresponding loss function is used to compute the loss. To balance invisibility and robustness, the training process is divided into two phases, with each phase training for 500 epochs. In the first phase, the parameters λ_{I_1}, λ_{I_2}, λ_M and λ_R are set to 0.75, 0, 10 and 0.01, respectively. In the second phase, λ_{I_2} is adjusted to 1.75 and λ_R is adjusted to 0.5, while the remaining parameters are maintained unchanged.

B. Performance under Watermark Updating

In this section, we simulate the process of watermark updating and evaluate the visual quality and robustness of different methods. Specifically, the embedding-extraction process is repeated 15 times, with embedded messages randomly sampled for each round. For our method, new messages are embedded into the recovered images from the previous round of extraction. As for traditional methods, since watermark-free images cannot be produced, new messages are embedded

TABLE I: Extraction under different shooting distances.

Distance (*cm*)	30	40	50	60
Round-1	98.2%	97.2%	97.1%	97.0%
Round-2	98.0%	98.5%	98.1%	97.8%
Round-3	97.8%	97.5%	97.0%	98.1%
Round-4	97.7%	98.0%	97.3%	97.3%
Round-5	98.5%	97.5%	97.2%	97.0%
Round-6	97.4%	98.4%	97.1%	97.8%
Round-7	97.4%	97.5%	97.2%	97.9%

into the images output from the previous round of embedding. We also simulate watermark updating with HiDDeN [6] and watermark removal method based on diffusion model [7] for comparison, referred to as WR. For WR, messages are first embedded into the cover images and then removed to obtain watermark-free images, into which new messages will be embedded. For each round of watermark updating, PSNR, SSIM, and LPIPS are utilized to evaluate the quality of watermarked image. Additionally, the extraction accuracy is also measured. The results are shown in Fig. 4.

From Fig. 4a, 4b, and 4c, it can be observed that high visual quality is effectively maintained by our method, whereas a significant decline in visual quality is evident in other methods. Fig. 5 shows the watermarked images after 15*th* update by different methods, providing a more intuitive demonstration of the superior performance of the proposed method in terms of visual quality. Up to the 15th update, the watermarked images produced by the proposed method retained excellent visual quality. Indeed, even after 30 rounds of updates, the watermarked images continued to exhibit high quality, with SSIM maintained above 0.97 and PSNR above 38. This is due to the fact that the proposed method mitigates the impact of multiple watermarks by updating information and preserving the quality of the cover.

Fig. 4d shows the results of extraction accuracy. As anticipated, a decrease in extraction accuracy has been observed in other methods to varying degrees, except for MBRS [2] and the proposed method. Although MBRS demonstrates robustness to the impact of multiple watermarks, it exhibits poor performance in terms of visual quality. It can also be observed that WR performs worse than HiDDeN in Fig. 4b and 4c, which is attributed to the fact that the invisible watermark removal attack does not achieve the expected visual quality. Therefore, this method is not practical for watermark updating.

C. Robustness Against Screen-shooting

In this section, we will test robustness against screen-shooting. Specifically, randomly sampled messages are embedded into the images from the dataset [13], and the embedded messages are updated 7 times. Each time, the watermarked images are saved and subsequently displayed on the screen. Then we use mobile phones to capture these images under various conditions. For the captured images, detection and perspective correction of the watermarked image in each photo are first performed, followed by watermark extraction from the corrected images using the decoder.

TABLE II: Comparisons of different λ_{I_1}.

λ_{I_1}	0.25	0.5	0.75	1	1.25
$SSIM_1$	0.9615	0.9685	0.9853	0.9769	0.9739
$PSNR_1$	37.6785	38.9956	40.9490	39.0498	38.1861
$LPIPS_1$	0.1084	0.1095	0.1340	0.1248	0.1580
$Accuracy_1$	100%	100%	100%	100%	100%
$SSIM_{15}$	/	0.7976	0.9787	0.7037	0.8987
$PSNR_{15}$	/	25.3245	38.8124	24.445	30.8267
$LPIPS_{15}$	/	2.1888	0.1792	1.9522	0.8542
$Accuracy_{15}$	/	94.87%	100%	96.54%	99.33%

TABLE III: Comparisons of different λ_{I_2}.

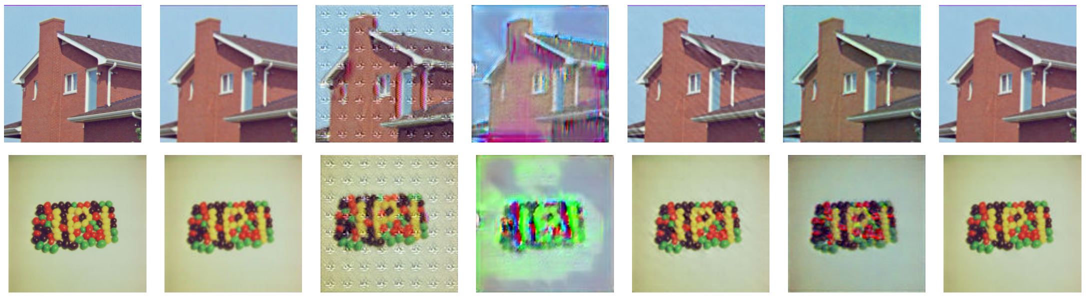
λ_{I_2}	0	0.75	1.75	2.75	3.5
$SSIM_1$	0.8987	0.97028	0.9853	0.9858	0.9870
$PSNR_1$	34.2170	38.0194	40.9490	40.9129	41.2677
$LPIPS_1$	2.5680	0.2551	0.1340	0.1193	0.1023
$Accuracy_1$	100%	100%	100%	100%	99.89%
$SSIM_{15}$	0.8981	0.9709	0.9787	0.9796	0.9808
$PSNR_{15}$	34.1204	37.7199	38.8124	39.1044	39.5334
$LPIPS_{15}$	2.4720	0.2799	0.1792	0.1513	0.1223
$Accuracy_{15}$	100%	100%	100%	99.78%	99.89%

We mainly test the robustness in the view of shooting distance. After displaying the watermarked images on the screen, we use mobile phone to capture them at various distances. The shooting distance varies from 30cm to 60cm. As shown in TABLE I, the proposed scheme maintains a high extraction accuracy exceeding 97% across all shooting distances. It is also observed that the extraction accuracy does not decrease with an increasing number of watermark updates, indicating that the proposed scheme can achieve traceability in the event of leakage due to screen capturing, even after multiple transmissions. These experiments are carried out under the device of "VA2430-FHD" and "iPhone13".

D. The Impact of Hyper-parameters

In this section, the impact of the weight parameters in loss function is evaluated. Experiments are conducted by varying one parameter at a time while the others are kept constant. The framework is retrained with new parameters, and the PSNR, SSIM, LPIPS, and extraction accuracy are measured for the 1st and 15th rounds of watermark updating.

From TABLE II and TABLE III, it can be observed that the approach combined with *MSE* and *LPIPS* to image loss is necessary. These two methods employ different calculation approaches. *MSE* calculates pixel-wise differences between images, which emphasizes precise pixel matching. *LPIPS* measures perceptual similarity by comparing images within the feature space of deep neural networks, aligning more closely with human visual perception and excelling in perceptual differences. Consequently, a higher weight is assigned to *LPIPS* to achieve improved visual quality. TABLE IV illustrates the trade-off between invisibility and robustness. When λ_M is too high, a high level of robustness may be achieved, with the expense of reduced visual quality. Similar results can be seen



(a) Original (b) HiDDeN (c) MBRS (d) PIMoG (e) FIN (f) WR (g) Proposed

Fig. 5: The visual example of watermarked images after 15 rounds of watermark updating.

TABLE IV: Comparisons of different λ_M.

λ_M	7.5	10	12.5	15
$SSIM_1$	0.9749	0.9853	0.9671	0.9645
$PSNR_1$	39.8040	40.9490	37.0977	37.6682
$LPIPS_1$	0.0976	0.1340	0.1891	0.1558
$Accuracy_1$	100%	100%	100%	100%
$SSIM_{15}$	0.9015	0.9787	0.9545	0.9301
$PSNR_{15}$	35.0513	38.8124	36.5443	36.3099
$LPIPS_{15}$	0.6843	0.1792	0.3286	0.4438
$Accuracy_{15}$	96.65%	100%	100%	100%

TABLE V: Comparisons of different λ_R.

λ_R	0	0.25	0.5	0.75	1
$SSIM_1$	0.9724	0.9811	0.9853	0.9823	0.9818
$PSNR_1$	38.3274	39.9105	40.9490	39.7030	39.7574
$LPIPS_1$	0.1812	0.1484	0.1340	0.1431	0.1446
$Accuracy_1$	100%	100%	100%	100%	100&
$SSIM_{15}$	0.9305	0.9754	0.9787	0.9789	0.9780
$PSNR_{15}$	35.7564	35.1862	38.8124	38.8779	38.8848
$LPIPS_{15}$	1.1724	0.1857	0.1792	0.1671	0.1646
$Accuracy_{15}$	90%	99.89%	100%	100%	100%

in TABLE III. It is evident that a balance must be achieved through the adjustment of these weight values.

From TABLE V, We can see that it is necessary to establish quality constraints for I_{re}. It is also observed that high λ_R may result in a decrease in the visual quality of watermarked images, as the model prioritizes minimizing the differences between recovered images and original images over the quality of watermarked images. Additionally, it has been found that during the backward-decoding process, the visual quality of recovered images is affected by the quality of the input watermarked images. Generally, a positive correlation is observed, as shown in TABLE V. Therefore, when determining the values of λ_{I_2} and λ_R, a higher value of λ_{I_2} is preferred to achieve high-quality watermarked images.

V. Conclusion

This paper introduced WaterUp, a deep learning-based watermark updating framework for multi-stage image content distribution. By leveraging a flow-based encoder-decoder, WaterUp effectively maintains image quality and watermark robustness across multiple transmission stages. Experimental results show that our approach significantly outperforms existing methods, offering superior visual integrity and traceability, and setting a new standard for watermarking in complex distribution scenarios.

References

[1] Kawser Ahmed, "Canada's cyber security in a globalized environment: Challenges and opportunities," *Routledge companion to global cyber-security strategy*, pp. 451–462, 2021.

[2] Zhaoyang Jia, Han Fang, and Weiming Zhang, "Mbrs: Enhancing robustness of dnn-based watermarking by mini-batch of real and simulated jpeg compression," in *ACM MM*, 2021, pp. 41–49.

[3] Han Fang, Zhaoyang Jia, Zehua Ma, Ee-Chien Chang, and Weiming Zhang, "Pimog: An effective screen-shooting noise-layer simulation for deep-learning-based watermarking network," in *ACM MM*, 2022, pp. 2267–2275.

[4] Manikandan Vazhora Malayil and Masilamani Vedhanayagam, "A novel image scaling based reversible watermarking scheme for secure medical image transmission," *ISA transactions*, vol. 108, pp. 269–281, 2021.

[5] 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*, vol. 19, pp. 4114–4129, 2024.

[6] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei, "Hidden: Hiding data with deep networks," in *ECCV*, 2018, pp. 657–672.

[7] Xinyu Li, "Diffwa: Diffusion models for watermark attack," in *ICIICS*. IEEE, 2023, pp. 1–8.

[8] 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 *AAAI*, 2023, vol. 37, pp. 5054–5061.

[9] Erwin Quiring and Konrad Rieck, "Adversarial machine learning against digital watermarking," in *EUSIPCO*. IEEE, 2018, pp. 519–523.

[10] 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," *NIPS*, vol. 37, pp. 8643–8672, 2024.

[11] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *CVPR*, June 2018.

[12] Eirikur Agustsson and Radu Timofte, "Ntire 2017 challenge on single image super-resolution: Dataset and study," in *CVPRW*, 2017, pp. 1122–1131.

[13] U. Viterbi, "The usc-sipi image database," http://sipi.usc.edu/database/, 1977, Accessed: JUly. 2024.

[14] Linfeng Geng, Weiming Zhang, Haozhe Chen, Han Fang, and Nenghai Yu, "Real-time attacks on robust watermarking tools in the wild by cnn," *Journal of Real-Time Image Processing*, vol. 17, pp. 631–641, 2020.

[15] Xiang Zhang, Jinyang Huang, Huan Yan, Yuanhao Feng, Peng Zhao, Guohang Zhuang, Zhi Liu, and Bin Liu, "Wiopen: A robust wi-fi-based open-set gesture recognition framework," *IEEE Transactions on Human-Machine Systems*, 2025.

[16] Xiang Zhang, Yan Lu, Huan Yan, Jinyang Huang, Yu Gu, Yusheng Ji, Zhi Liu, and Bin Liu, "Resup: Reliable label noise suppression for facial expression recognition," *IEEE Transactions on Affective Computing*, 2025.