# Physical Marker: Revealing Invisible Hyperlinks Hidden in Printed Trademarks

**Yuliang Xue, Lei Tan, Guobiao Li, Zhenxing Qian*, Sheng Li, Xinpeng Zhang**

School of Computer Science, Fudan University, ShangHai, China
ylxue21@m.fudan.edu.cn, 23110240157@m.fudan.edu.cn, 20210240200@fudan.edu.cn, zxqian@fudan.edu.cn, lisheng@fudan.edu.cn, zhangxinpeng@fudan.edu.cn

## Abstract

Embedding links in brand logos is a promising technology, which allows consumers to access the online information of products by capturing physical logo images. Previous physical data hiding methods primarily embed data within cover media in a global manner, making them ineffective for processing brand logos in vector graphics format with a transparent background. To address this issue, we propose in this paper a novel physical deep hiding scheme for invisibly embedding links in printed trademarks. Specifically, the encoder embeds links only into the area of the brand logo under the constraints of a mask, which is generated from the transparency information of the logo image. A background variation distortion is introduced into the distortion layer that approximate practical logo print-camera environments, such that the decoder could be learnt to retrieve the link from the camera-captured logo with various backgrounds. A feature prompt subspace modulator is further proposed and employed in the encoder to enhance the invisibility of the encoded logo pattern and in the decoder to boost hyperlink extraction accuracy. Various experiments have been conducted to demonstrate the advantage of our proposed method for embedding links in printed brand logos, which provides reliable extraction accuracy under both simulated and real scenarios.

## Introduction

In interconnected world, businesses face the challenge of effectively integrating their offline and online marketing efforts to create a unified and impact brand experience. The convergence of physical logos and digital platforms has become a powerful strategy to bridge the gap between the physical and digital realms, helping businesses enhance brand awareness, attract consumers, and drive sales.

Image print-camera resilient (PCR) watermarking offers promising way to bridge the gap between offline and online channels and create a seamless and dynamic brand experience. It embeds hyperlinks into physical printed images and allows consumers to extract the links from phone-captured version (Tancik, Mildenhall, and Ng 2020; Jia et al. 2022). As shown in the top row of Fig. 1, previous methods primarily focused on embedding links into entire images, including backgrounds. While these approaches enable link extraction from captured images, the presence of logos with backgrounds printed on products will impact aesthetics in real applications. To enhance visual appeal, transparent background vector logos are commonly utilized for printing in real-world scenarios, as shown in the bottom row of Fig. 1. However, previous approaches can not accurately extract the links due to the absence of original backgrounds leading to the loss of embedded messages. Additionally, the performance of previous methods would further compromised due to the diverse backgrounds, making it challenging to locate the region with embedded data and to accurately extract the links. Moreover, due to the homogeneous nature of logos, these regions lack variations to hide links both robustly and imperceptibly, rendering previous methods potentially inadequate. To sum up, achieving invisible and background-agnostic link embedding within printed logos, along with accurately extracting links from captured images, presents a more challenging task.

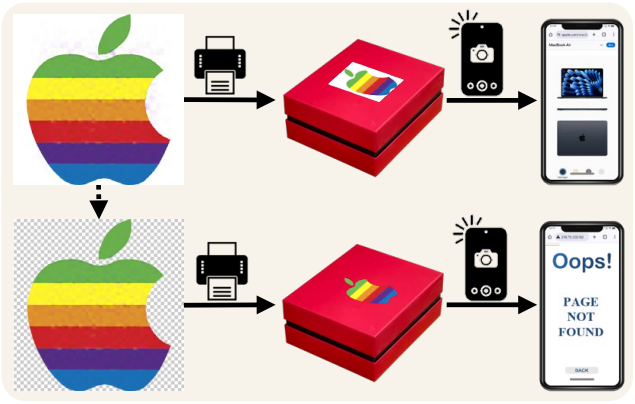



Figure 1: Examples of extracting links from photos of a logo printed on a red packaging box under two different printing ways. The top row involves printing with background, while the bottom row pertains to printing without background.

To address the above issues, we propose a novel physical deep hiding scheme for embedding links in printed trademarks. Specifically, the scheme follows the classic encoder-distortion-layer-decoder framework, where the encoder hides the link only within the area of the brand logo

*Corresponding author.

under the guidance of the mask generated from the transparency of the logo image. Furthermore, we introduce a background variation distortion within the well-designed distortion layer to simulate the transformations during the practical logo print-camera (PC) scenario. Then, we jointly train the encoder and decoder to adapt to the customized distortion layer. This allows the decoder to recover the links hidden by the encoder in a background-agnostic manner within a real PC environment. As the feature of logo images usually lie in a low-rank signal subspace, by properly learning and generating the basis vectors, the encoded image can keep most original features and suppress noise which is irrelevant to the generated basis set. Based on this, we propose a feature prompt subspace modulator (FPSM) consisting of two components enhance the effectiveness of our approach. One component is the Retrieve and Interaction Module (RIM), responsible for generating a feature prompt based on input characteristics and interacting with the input. The other component is the Feature Subspace Modulator (FSM), which modulates the reconstruction of encoded image features based on the output subspace of RIM. In the encoder, the FPSM assists the encoder in perceiving logo features to embed messages and eliminating redundancies by modulating reconstructed features. In the decoder, RIM is used to generate prompt decoded images by distinguishing features in the embedded message area from those in the non-embedded message area, thus facilitating precise message extraction.

Our main contributions are summarized below:

- We propose a novel invisible information hiding scheme for embedding brand links in printed trademarks, where a logo mask is considered for local data encoding and a background variation distortion is designed for stable link extraction from camera-captured logo images with different backgrounds.
- We propose a feature prompt subspace modulator to enhance the capabilities of both the encoder and decoder. It facilitates the encoder to produce the encoded logo with higher visual quality, while its component, RIM, aids the decoder in automatically identifying the concealed data region for precise link extraction.
- Comprehensive experiments demonstrate the effectiveness of the proposed method for embedding links in printed brand logos, which provides dependable robustness in both simulated and real-world scenarios.

## Related Works

**Image print-camera resilient watermarking.** Image PCR watermarking has been extensively researched. Traditional methods employed image processing techniques to embed watermarks in either the spatial (Kim, Lee, and Seo 2006; Pramila, Keskinarkaus, and Seppänen 2012; Chou and Li 1995) or frequency domain (Gourrame et al. 2019; Liang and Wang 2019; Fang et al. 2018). However,those schemes cannot guarantee the robustness and visual quality of watermarks in real PC environments due to the presence of more complex and random noise. With the development of deep learning, many methods proposed to use convolution neural network to achieve watermark embedding and extraction. HiDDeN (Zhu et al. 2018) is a first end-to-end network framework for robust image watermarking. It used a noise layer for robustness but had limited real-world resilience. To address this limitation, StegaStamp (Tancik, Mildenhall, and Ng 2020) introduced an advanced distortion layer to mimic diverse distortions encountered during printing and camera capture processes. Following extensive training stages, it demonstrated outstanding performance in both embedding and decoding. Later, Liu et al. (Liu et al. 2023a) introduced WRAP, which leveraged CycleGAN (Zhu et al. 2017) to train a distortion layer, aiming to enhance resilience against film-coating attacks. To improve visual quality of encoded images, Jia et al. (Jia et al. 2022) introduced Learning Invisible Markers (LIM), which hide messages within a square subspace of the cover image.

Although those methods perform well, they encounter challenges when hiding links within printed logos. The absence of a fixed embedding region for logo vector graphics and the variations in real-world backgrounds present difficulties for previous methods. Additionally, the relatively uniform color distribution in logo images complicates the task of ensuring high visual quality for encoded images. Therefore, we propose a solution to address this limitation.

**Prompt Learning.** In natural language processing, prompting (Liu et al. 2023b) initially refer to inserting directives into input sentences. Recent works (Li and Liang 2021; Potlapalli et al. 2024) suggests leveraging prompting techniques to handle various downstream tasks or domains with a combination of transformers without the need to optimize all parameters. In this paper, we introduce a set of feature prompts before encoding feature recovery to guide the modulation of encoded image features and the localization of decoding regions.

## Method

### Framework Overview

Fig. 2 shows the architecture of our proposed method, which includes a message encoder $E$, a distortion layer $N$ integrating various simulated image transformations, and a message decoder $D$. The inputs of encoder $E$ include a cover bitmap $I_c$ and a mask $I_m$ representing the logo region, generated from the logo vector graphics, along with the bit messages $M$ of length $L$ generated from the hyperlink. The output of $E$ is an encoded image $I_e$ with the same shape as $I_c$. Then, $I_e$ undergoes a distortion layer to generate the distorted image $I_n$. The noise layer includes simulations of diverse backgrounds, perspective transformation, various distortions arising from the PC processes and JPEG compression distortions. After that, the decoder utilizes the RIM to generate the prompt-decoded image,identifying the region with the embedded message and extracting the messages $M^{'}$ from the distorted images to recover the hyperlink.

### Modules and Training Strategy

**Encoder.** The encoder is trained to embed the message into the image while minimizing perceptual differences be-

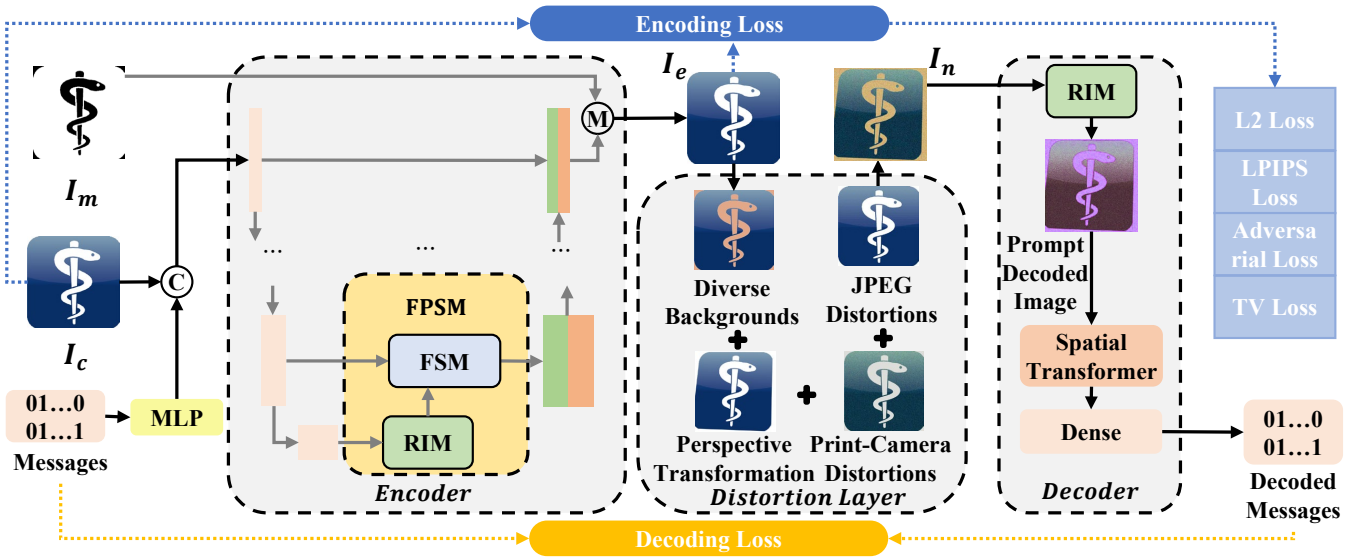



Figure 2: The framework of our proposed method. The encoder embeds messages within logo regions constrained by masks. The FPSM modulates the encoding features to eliminate redundancy and imperceptibly embeds messages into the encoded image features. Then, the encoded image undergoes processing through our proposed distortion layer, which simulates real PC environments, resulting in distorted images. Finally, the decoder employs the RIM to generate the prompt decoded image, locating the region with the embedded message and extracting the messages from the distorted images.

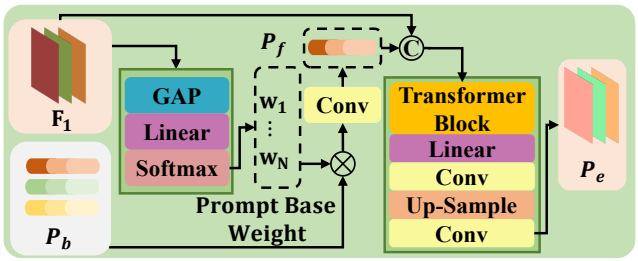
tween the input $I_c$ and encoded images $I_e$. Initially, the bit message $M$ undergoes processing through fully connected layers and upsampling, resulting in a tensor with dimensions matching $I_c$. This tensor is then concatenated with $I_c$ and fed into an enhanced U-Net (Ronneberger, Fischer, and Brox 2015) style encoder. The encoder directly generates the image instead of a residual image, which aids in convergence. This image is then masked by $I_m$ to produce $I_e$.

To retain the majority of the original features of $I_c$ and suppress noise during the generation of $I_e$, we introduce a specially designed Feature Prompt Subspace Modulator (FPSM) into the encoder. It consists of two key components: a Retrieve and Interaction Module (RIM) and a Feature Subspace Modulator (FSM).

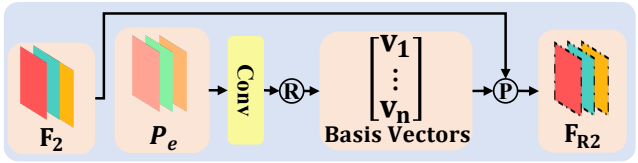
**RIM** introduces a prompt base $\mathbf{P}_b \in \mathbb{R}^{N\times H\times W\times C}$ comprising learnable parameters to extract deep-level feature $\mathbf{F}_1 \in \mathbb{R}^{H\times W\times C}$ insights and retrieve the feature prompt $\mathbf{P}_f \in \mathbb{R}^{H\times W\times C}$, which subsequently interacts with deep-level features to produce the encoding prompt $\mathbf{P}_e \in \mathbb{R}^{\hat{H}\times\hat{W}\times\hat{C}}$.

As shown in Fig. 3a, to dynamically generate feature prompt $\mathbf{P}_f$, RIM applies global average pooling (GAP) on the deep-level features $\mathbf{F}_1$. Subsequently, a linear layer and softmax operation are applied to produce prompt weights $w \in \mathbb{R}^N$. These weights are used to adjust the prompt base $\mathbf{P}_b$ before passing through a convolution layer. This process can be formulated as:

$$\mathbf{P}_f = Conv_{3\times3}\left(\sum_{c=1}^{N} w_i \mathbf{P}_b\right) \qquad (1)$$
$$w_i = Softmax(Linear(GAP(\mathbf{F}_1)))$$




(a) The architecture of Retrieve and Interaction Module




(b) The architecture of Feature Subspace Modulator

Figure 3: The architecture of Feature Prompt Subspace Modulator.

Subsequently, $\mathbf{P}_f$ is concatenated with $\mathbf{F}_1$ along the channel dimension. Following this, a transformer block (Zamir et al. 2022) is employed to facilitate interactions within the concatenated output. Then, convolutional upsampling operations are performed to produce the encoding prompt $\mathbf{P}_e$. As the feature of logo images lie in a low-rank signal subspace, by properly learning and generating the basis vectors, the encoded image can keep most original features and suppress noise which is irrelevant to the generated basis set.

**FSM** modulates the shallow-level feature $\mathbf{F}_2 \in \mathbb{R}^{\hat{H}\times\hat{W}\times\hat{C}}$ according to the subspace of the encoding prompt to suppress redundant noisy features, with the goal of enhancing smoothness in the final encoded image $I_e$.

As shown in Fig. 3b, a matrix $V = [v_1, v_2, ..., v_k]$ composed of the subspace basis vectors is generated from $\mathbf{P}_e$:

$$V = f_\theta\left(\mathbf{P}_e\right), \tag{2}$$

where $f_\theta : \mathbb{R}^{H\times W\times C} \rightarrow \mathbb{R}^{J\times K}$ is a function parameterized by $\theta$ achieved by a simple convolution network. Given the matrix $V \in \mathbb{R}^{J\times K}$ whose columns are basis vectors of a $K$-dimensional subspace $v \subset \mathbb{R}^J$, we project $\mathbf{F}_2$ onto $v$ by orthogonal linear projection. The orthogonal projection matrix $\mathbf{P}$ can be calculated from $V$ as:

$$\mathbf{P} = V\left(V^T V\right)^{-1} V^T, \tag{3}$$

where $\left(V^T V\right)^{-1}$ is a normalization term ensuring that the basis vectors are orthogonal to each other. Using the generated projection matrix, $\mathbf{F}_2$ can be reconstructed in the subspace of $\mathbf{P}_e$ by

$$\mathbf{F}_{R2} = \mathbf{P}\mathbf{P}_e \tag{4}$$

The resulting reconstruct feature $\mathbf{F}_{R2}$ is concatenated with $\mathbf{F}_2$ as the subsequent upsampling feature.

To minimal differences between $I_c$ and $I_e$, we introduce the pixel-level loss $\mathcal{L}EN1$, which computes the L2 loss between $I_c$ and $I_e$. Additionally, we incorporate the LPIPS perceptual loss (Zhang et al. 2018) as $\mathcal{L}EN2$. Besides, we introduce the adversarial loss $\mathcal{L}_{EN3}$, which aims at generating indistinguishable $I_e$ by the additional discriminator $DIS$.

$$\mathcal{L}_{EN3} = log(1 - DIS(I_e)) \tag{5}$$

The discriminator is a simple binary classification convolutional neural network designed to distinguish whether an image contains messages or not. It minimizes the loss $\mathcal{L}_{DIS}$:

$$\mathcal{L}_{DIS} = log(DIS(I_e)) + log(1 - DIS(I_c)) \tag{6}$$

To enhance image smoothness, we employ Total Variation loss (Mahendran and Vedaldi 2015). The smoothness loss can be formulated as:

$$\mathcal{L}_{EN4} = \sum_{i,j}\left(\left(I_{ei,j+1} - I_{eij}\right)^2 + \left(I_{ei+1,j} - I_{eij}\right)^2\right)^{\frac{\beta}{2}} \tag{7}$$

Typically, as the $\beta$ value increases, the image becomes smoother. However, for images with intricate textures, this may result in a loss of clarity. Nevertheless, for relatively flat logo images, increasing the $\beta$ value is advantageous for optimizing the outcomes. In our work, we set $\beta$ to 2.

**Distortion Layer.** The distortion layer plays a crucial role in our method to enhance the resistance to PC environments and different logo backgrounds. It includes background diversity distortion, perspective transformation distortion, PC distortions, JPEG compression distortion. To draw attention to the logo, businesses often position it against a simple background. We randomly alter the background, transforming it into a solid color background and introducing background distortion. To account for distortions caused by different shooting angles, we introduce perspective transformation distortion by manipulating the four corner vertices of the image. Environmental lighting during the capture and printing process can lead to color distortions. Additionally, camera motion or focus can introduce distortions. To simulate PC distortion, we sequentially apply effects such as blur, random noise, brightness adjustment, contrast adjustment, color transformation. JPEG distortions is used to simulate distortions that may occur during saving photos.

**Decoder.** The decoder serves as a network to recover the message from the encoded image. To enable lightweight localization of the embedded message regions, we integrate the RIM module into the decoder to produce the prompt encoded image as shown in Fig. 2. To enhance robustness against slight viewpoint changes when generating the prompt encoded image with localization information, we use a spatial transformer network (Jaderberg et al. 2015). Following a series of convolutional and dense layers, along with a sigmoid layer, the decoded message is obtained. The decoder network employs cross-entropy loss to supervise the extraction of the message $M^{'}{}_1$ and $M^{'}{}_2$ from both the encoded image and the distorted image:

$$\mathcal{L}_{DE} = MSE(M, M^{'}{}_1) + MSE(M, M^{'}{}_2) \tag{8}$$

**Training.** During training, we employed a two-stage training strategy to improve the performance of the encoder and decoder in a stable manner. In the first stage, we exclusively trained the decoder by minimizing $\mathcal{L}_{DE}$ while gradually increasing the level of distortion. This approach ensured that the decoder could accurately extract the embedded message even under high distortion levels. Once the decoder became proficient in extracting messages under high distortion, we proceeded to the second stage, which involved joint training of the encoder $E$, decoder $D$, and discriminator $DIS$. The training loss is formulated as:

$$\begin{aligned}\mathcal{L} =& \lambda_1\mathcal{L}_{EN1} + \lambda_2\mathcal{L}_{EN2} + \lambda_3\mathcal{L}_{EN3} \\ &+ \lambda_4\mathcal{L}_{EN4} + \lambda_5\mathcal{L}_{DE}.\end{aligned} \tag{9}$$

# Experiments

## Experiment Settings

**Datasets.** We began by standardizing the bitmap images from the Large Logo Dataset (Sage et al. 2017). This involved fixing the longer side and extending the shorter side to create square bitmaps, filling the extended areas with white, and resizing them to 400×400 pixels. Simultaneously, a grayscale threshold of 200 was applied to adjust the transparency of the white pixels to 0, generating the masks accordingly. These masks are adopted to segment the logo region, which is placed on the various background image to generate the composite image. For experiments in the digital environment, 98,418 images generated from Large Logo Dataset were used for training. To further assess the robustness and generalization capabilities of the method, 417 images from the METU Trademark Dataset (Tursun and Sinan 2015) were dedicated to testing. Moreover, we randomly select 20 images from test set for real PCR tests.

**Implementation Details.** The entire framework is implemented in PyTorch and executed on NVIDIA GeForce RTX

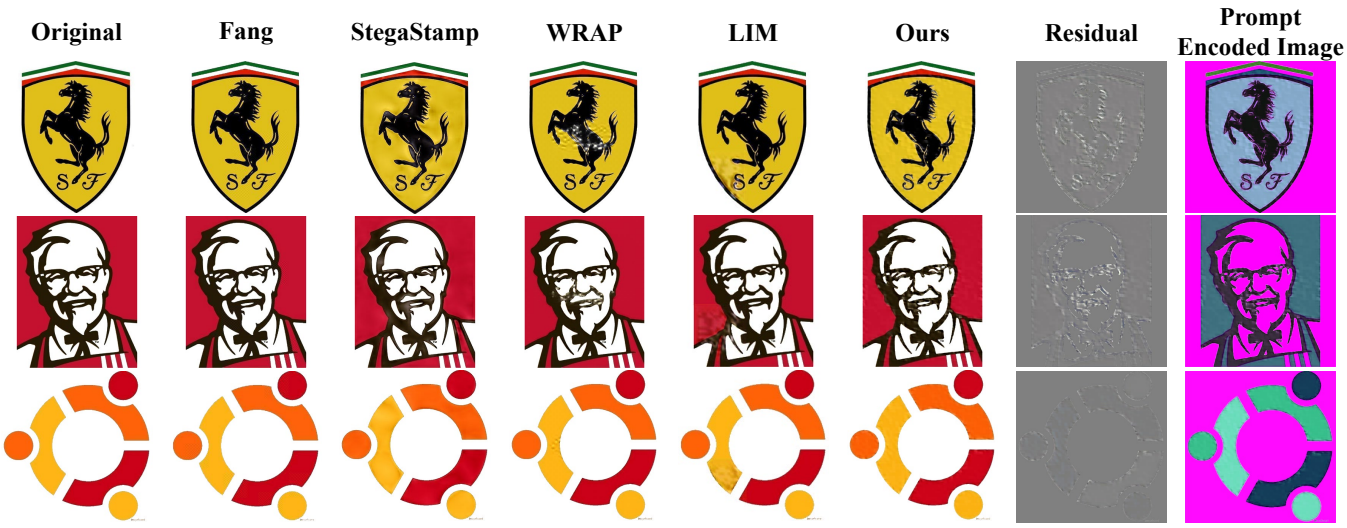



Figure 4: Examples of encoded images generated by five methods, the residual image of our method, and the visualization of the prompt encoded image.

2080 Ti. We employ the Adam optimizer for our model. During training, the size of input bitmap is $400 \times 400$ and the message is randomly generated with a length of 100 bits. The subspace dimensionality $K$ is set to 16. The parameter setting in distortion layer are as follows:

- Blur kernel size: $[3, 7]$
- Gaussian noise: $\sigma \sim U[0, 0.2]$
- Brightness adjustment: $[-0.3, 0.3]$
- Contrast adjustment: $[0.5, 1.5]$
- JPEG compression quality factor: $[50, 100]$

For the loss function in the Eq. (9), we choose $\lambda_1 = 2$, $\lambda_2 = 1.5$, $\lambda_3 = 0.5$, $\lambda_4 = 1.5$, $\lambda_5 = 2.5$. The batch size in the training is set to 16 and the model are trained for 30 epochs with an initial learning rate = 0.0001.

**Baselines.** Our baselines for comparison are Fang (Fang et al. 2018), StegaStamp (Tancik, Mildenhall, and Ng 2020), WRAP (Liu et al. 2023a), and LIM (Jia et al. 2022). All the methods except Fang are deep-learning-based. Despite our attempts to experiment with LIM, we could not replicate their reported best performance under conditions matching our method. Hence, we directly compared using their pre-trained models. In the experiments, Fang embeds messages with a length of 30 bits, while LIM uses a 16x16 matrix QR (quick response) code, and the other methods use 100 bits. For a fairer comparison, after embedding the messages, the non-logo region of encoded images were uniformly set to white using a mask for further testing, as in real-world scenarios this portion would have a transparency of 0. In addition, all tests in real PC environment were consistently carried out with marked locating and manually adjusted cropping to attain the best results.

**Evaluation Metrics.** We assess the visual quality of the encoded images using peak signal-to-noise ratio (PSNR) (Almohammad and Ghinea 2010) and structural similarity (SSIM) (Wang et al. 2004). The performance of decoder is evaluated by the average extraction accuracy (ACC), which indicates the percentage of correctly extracted messages. It's worth noting that for the LIM, ACC represents the percentage of correctly extracted matrices of QR codes.

| Methods | Fang | StegaStamp | WRAP | LIM | Ours |
|---|---|---|---|---|---|
| PSNR(dB) | 34.28 | 30.54 | 29.54 | **35.36** | 31.59 |
| SSIM | 0.946 | 0.834 | 0.904 | **0.953** | 0.907 |
| ACC(%) | 66.79 | 92.88 | 88.88 | 78.18 | **99.83** |

Table 1: Visual comparison of encoded images and accuracy comparison of message extraction between our method and the comparison methods.

| ACC(%) | JPEG | | Contrast | | Gaussian Blur | |
|---|---|---|---|---|---|---|
| Parameters | 50 | 75 | +50 | -50 | 1 | 2 |
| Fang | 63.07 | 66.07 | 57.74 | 66.11 | 62.35 | 61.72 |
| StegaStamp | 92.14 | 92.55 | 78.25 | 88.19 | 92.98 | 92.96 |
| WRAP | 88.91 | 88.96 | 88.29 | 88.78 | 88.61 | 88.52 |
| LIM | 77.39 | 77.79 | 73.20 | 77.85 | 78.24 | 78.20 |
| Ours | **99.68** | **99.73** | **97.66** | **99.68** | **99.78** | **99.76** |

Table 2: Comparison of extraction accuracy under different digital distortions.

## Experimental Results

**Visual Quality and Performance of Decoder.** Table 1 displays the average objective metrics of visual quality in encoding images and decoding performance of comparative methods and ours. Our model performs similarly to other models in terms of PSNR and SSIM, indicating that our model can produce images of reasonably good quality. Although our PSNR and SSIM is a little inferior to Fang and LIM, our model has the highest ACC and and em-


(a) The photos captured under different shooting distance


(b) The photos captured under different shooting horizontal angles

Figure 5: The captured photos and their corrected results

beds the watermark in the complete image instead of only in the subimage like Fang and LIM. As shown in Fig. 4, our method embeds messages into the edge regions of the image, enhancing visual quality. Furthermore, the decoder can automatically locate and extract the embedded message, contributing to the top accuracy achievement.

**Robustness Comparison in Digital Environment.** To assess robust performance in a digital setting, we initially introduced various noise types directly into the encoded image without print processing, as detailed in Table 2. When examining JPEG compression impact, we tested compression quality factors set at 50 and 75. Fang displayed a 3% accuracy decrease at a quality factor of 50, while other methods exhibited only minor reductions in extraction accuracy.

In contrast variation trials, we evaluated extraction accuracy across the five methods under 50% increased and decreased image contrast conditions. Except for the WRAP method, all approaches showed significant ACC drops when a 50% contrast boost, with our method demonstrating the smallest decline. A comparison between our embedding pattern and WRAP revealed that WRAP prioritizes robustness over image quality by embedding messages mainly in the image center, resulting in visible white patches.

Regarding resistance to Gaussian blur noise, we assessed ACC for the five methods at $\sigma = 1$ and $\sigma = 2$. Fang displayed the weakest robustness against Gaussian blur, while the other methods maintained stable ACC levels.

In summary, our method demonstrates robustness against digital noise.

**Robustness Comparison in Real Print-Camera Environment.** In this section, we compared the performance of the five methods in a real PC environment. We utilized BCH codes (Bose and Ray-Chaudhuri 1960) to perform error correction on the link transformed into 42 bits. With an estimated accuracy of 88% achieved on 100-bit messages, the link can be successfully restored. To obtain more convincing results, we evaluated the ACC metric using two different brands of smartphones under various shooting angles, distances, and background conditions.

| Devices | Methods | 15cm | 25cm | 40cm |
|---|---|---|---|---|
| vivo X100 Pro | Fang | 65.08 | 70.16 | 52.38 |
| | StegaStamp | 81.80 | 76.80 | 76.40 |
| | WRAP | 66.79 | 86.68 | 86.60 |
| | LIM | 76.87 | 71.73 | 73.87 |
| | Ours | **90.19** | **97.60** | **95.00** |
| iPhone 11 Pro | Fang | 69.21 | 63.17 | 60.00 |
| | StegaStamp | 78.80 | 79.40 | 78.99 |
| | WRAP | 67.75 | 85.20 | 83.20 |
| | LIM | 73.53 | 74.98 | 73.84 |
| | Ours | **90.99** | **94.60** | **96.20** |

Table 3: Comparison of extraction accuracy with different shooting distances.

| Devices | Methods | $\pm 15^\circ$ | $\pm 30^\circ$ | $\pm 45^\circ$ |
|---|---|---|---|---|
| vivo X100 Pro | Fang | 50.79 | 53.02 | 55.14 |
| | StegaStamp | 78.60 | 80.79 | 82.2 |
| | WRAP | 85.60 | 85.39 | 76.00 |
| | LIM | 74.30 | 80.96 | 74.66 |
| | Ours | **93.39** | **93.19** | **90.39** |
| iPhone 11 Pro | Fang | 50.79 | 59.37 | 59.68 |
| | StegaStamp | 80.40 | 80.99 | 80.60 |
| | WRAP | 79.60 | 76.59 | 72.59 |
| | LIM | 73.59 | 76.13 | 76.11 |
| | Ours | **93.59** | **91.79** | **90.19** |

Table 4: Comparison of extraction accuracy with different shooting angles.

**Different Shooting Distances.** During the test of how shooting distance impacts the accuracy of the methods, the printed logo size was $13cm \times 13cm$. The camera was maintained at a fixed angle ($0^\circ$) and height perpendicular to the image plane, with the distance gradually increased from 15cm to 40cm. Three distance settings were tested. The captured results were manually located and adjusted to create a square region as input for the each method, as shown in Fig. 5a.

The experimental results detailed in Table 3 reveal that despite a notable decrease in accuracy for our method at a testing distance of 15cm, it remained above 90%. This drop can be attributed to the inherent blurring effect of smartphones during close-up shots. When capturing the image with the vivo X100 Pro from a distance of 25cm, the accuracy surged to as high as 97.6%. To sum up, our method showcases robustness across different shooting distance conditions.

**Different Shooting Angles.** During the test of how shooting distance impacts the accuracy of the methods, the printed logo size was $13cm \times 13cm$. We kept the camera at a con-

Figure 6: Example of the actual background texture for real background testing.

sistent distance of 25cm from the image and systematically varied the horizontal angle between the camera and the image. Three angle settings were tested, ranging from $15°$ to $45°$. As illustrated in Fig. 5b, the captured results were manually located and adjusted.

The results in Table 4 demonstrate that as the shooting angle increased, the extraction complexity also heightened, leading to a decrease in the accuracy of our method. However, even at a $45°$ angle, our method sustained an extraction accuracy surpassing $90\%$. In contrast, Fang displayed an increase in accuracy with the angle, possibly due to the natural distribution of embedded messages on both sides of the logo. Conversely, WRAP embedded messages centrally within images, resulting in a reduction in accuracy as the angle increased. In conclusion, our method exhibits significant robustness across varying shooting angles.

**Different Backgrounds.** When evaluating the influence of different backgrounds on method accuracy, the printed logo size was $4cm \times 4cm$. We varied the backgrounds to red, green, and blue to assess method performance against plain-colored backgrounds. To enhance realism, we designed the logo as a crystal label affixed to a packaging box with a complex textured background, as depicted in Fig. 6. During the tests, the camera was consistently positioned $15cm$ away from the logos.

The results in Table 5 indicate that the accuracy of our method fluctuates by no more than $5\%$ tested against plain-colored backgrounds. Even tested against real backgrounds, our method maintains the highest accuracy level at around $90\%$. In conclusion, our method demonstrates robustness across diverse backgrounds, thanks to the inclusion of diverse backgrounds noise layer during training, making it well-suited for real-world applications. In summary, our method exhibits robustness across various backgrounds, attributed to the incorporation of noise layers during training, rendering it highly suitable for real-world applications.

## Ablation Study

**Importance of the mask.** Masks can be considered as an advanced crop attack, helping encoder embed messages repeatedly within the logo region. As the results shown in Table 6, models trained with mask assistance can achieve higher accuracy under identical conditions.

**Importance of the random background noise layer.** The random background noise layer is designed to train the decoder to accurately extract messages from the logo with varied backgrounds. The results in Table 6 demonstrate that incorporating random noise during training improves the decoder's ability to extract messages across various backgrounds, albeit at the cost of a notable decline in the visual quality of the encoded images.

| Devices | Methods | Red | Green | Blue | Real |
|---|---|---|---|---|---|
| vivo X100 | Fang | 49.21 | 51.59 | 50.97 | 51.43 |
| | StegaStamp | 78.24 | 82.00 | 76.12 | 72.28 |
| | WRAP | 82.00 | 86.92 | 84.50 | 57.00 |
| | LIM | 66.71 | 69.50 | 67.10 | 75.71 |
| | Ours | **94.66** | **97.37** | **95.99** | **90.24** |
| iPhone 11 Pro | Fang | 50.22 | 58.73 | 57.67 | 52.38 |
| | StegaStamp | 81.57 | 83.79 | 76.52 | 73.83 |
| | WRAP | 83.19 | 88.34 | 85.27 | 60.55 |
| | LIM | 69.97 | 72.57 | 65.19 | 75.80 |
| | Ours | **95.99** | **98.24** | **94.25** | **89.72** |

Table 5: Comparison of extraction accuracy with different backgrounds.

| Mask | Noise Layer (Background) | FPSM | ACC | PSNR | SSIM |
|---|---|---|---|---|---|
| ✗ | ✗ | ✗ | 84.60 | 30.54 | 0.834 |
| ✓ | ✗ | ✗ | 94.82 | 30.49 | 0.830 |
| ✓ | ✓ | ✗ | 98.07 | 27.18 | 0.809 |
| ✓ | ✓ | ✓ | **98.95** | **31.59** | **0.907** |

Table 6: The ablation study of mask, noise layer and FPSM.

**Importance of the FPSM.** The purpose of FPSM is to invisibly embed messages into the logo while maintaining its homogeneous characteristics. Additionally, RIM in FPSM also serves to enable the decoder to identify the embedded message area, thereby enhancing extraction precision. As the results shown in Table 6, the visual quality of the encoded images significantly improved with the addition of FPSM. Furthermore, there was an enhancement in the robustness of message extraction under different backgrounds.

## Conclusion

In this paper, we first introduce the existing PCR watermarking technologies and analyze the limitations they face when embedding links in actual logo images. To address these challenges, we propose an end-to-end network framework that incorporates a unique noise layer specifically designed to mitigate the impact of different noises present in real PC environments, thereby enhancing the robustness of our method. The well designed FPSM introduced in the encoder and decoder helps improve the visual quality of encoded images and extraction accuracy. Compared to existing methods, our framework demonstrates good performance in simulated and real-world scenarios, showcasing better practical applicability. We will continue to enhance this work in the future by introduce more efficient background segmentation techniques and simulate diverse noises to ensure our method maintains robustness across various conditions.

## Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grants U20B2051, U22B2047, 62450067, 62072114.

## References

Almohammad, A.; and Ghinea, G. 2010. Stego image quality and the reliability of PSNR. In *2010 2nd International Conference on Image Processing Theory, Tools and Applications*, 215–220. IEEE.

Bose, R.; and Ray-Chaudhuri, D. 1960. On a class of error correcting binary group codes. *Information and Control*, 68–79.

Chou, C.-H.; and Li, Y.-C. 1995. A perceptually tuned subband image coder based on the measure of just-noticeable-distortion profile. *IEEE Transactions on circuits and systems for video technology*, 5(6): 467–476.

Fang, H.; Zhang, W.; Zhou, H.; Cui, H.; and Yu, N. 2018. Screen-shooting resilient watermarking. *IEEE Transactions on Information Forensics and Security*, 14(6): 1403–1418.

Gourrame, K.; Douzi, H.; Harba, R.; Riad, R.; Ros, F.; Amar, M.; and Elhajji, M. 2019. A zero-bit Fourier image watermarking for print-cam process. *Multimedia Tools and Applications*, 78(2): 2621–2638.

Jaderberg, M.; Simonyan, K.; Zisserman, A.; et al. 2015. Spatial transformer networks. *Advances in neural information processing systems*, 28.

Jia, J.; Gao, Z.; Zhu, D.; Min, X.; Zhai, G.; and Yang, X. 2022. Learning invisible markers for hidden codes in offline-to-online photography. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2273–2282.

Kim, W.-g.; Lee, S. H.; and Seo, Y.-s. 2006. Image fingerprinting scheme for print-and-capture model. In *Advances in Multimedia Information Processing-PCM 2006: 7th Pacific Rim Conference on Multimedia, Hangzhou, China, November 2-4, 2006. Proceedings 7*, 106–113. Springer.

Li, X. L.; and Liang, P. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Liang, S.; and Wang, X. 2019. Robust Image Watermarking in the Print-Cam Process. In *2019 IEEE 19th International Conference on Communication Technology (ICCT)*.

Liu, G.; Si, Y.; Qian, Z.; Zhang, X.; Li, S.; and Peng, W. 2023a. WRAP: Watermarking Approach Robust Against Film-coating upon Printed Photographs. In *Proceedings of the 31st ACM International Conference on Multimedia*, 7274–7282.

Liu, P.; Yuan, W.; Fu, J.; Jiang, Z.; Hayashi, H.; and Neubig, G. 2023b. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9): 1–35.

Mahendran, A.; and Vedaldi, A. 2015. Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5188–5196.

Potlapalli, V.; Zamir, S. W.; Khan, S. H.; and Shahbaz Khan, F. 2024. Promptir: Prompting for all-in-one image restoration. *Advances in Neural Information Processing Systems*, 36.

Pramila, A.; Keskinarkaus, A.; and Seppänen, T. 2012. Toward an interactive poster using digital watermarking and a mobile phone camera. *Signal, Image and Video Processing*, 6(2): 211–222.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, 234–241. Springer.

Sage, A.; Agustsson, E.; Timofte, R.; and Van Gool, L. 2017. LLD - Large Logo Dataset - version 0.1. https://data.vision.ee.ethz.ch/cvl/lld.

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.

Tursun, O.; and Sinan, K. 2015. A challenging big dataset for benchmarking trademark retrieval. In *IAPR Conference on Machine Vision and Applications*, 28.

Wang, Z.; Bovik, A.; Sheikh, H.; and Simoncelli, E. 2004. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 600–612.

Zamir, S. W.; Arora, A.; Khan, S.; Hayat, M.; Khan, F. S.; and Yang, M.-H. 2022. Restormer: Efficient transformer for high-resolution image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5728–5739.

Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; and Wang, O. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 586–595.

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

Zhu, J.-Y.; Park, T.; Isola, P.; and Efros, A. A. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, 2223–2232.