



ARTICLE

# Functa Hiding: Steganography via Modulated Implicit Representations

Qiya Wang<sup>1,2</sup>, Jia Liu<sup>1,2,\*</sup>, Yuwei Lu<sup>1,2</sup>, Yujie Liu<sup>1,2</sup> and Peng Luo<sup>1,2</sup>

<sup>1</sup>Engineering University of PAP, College of Cryptography Engineering, Xi'an, China

<sup>2</sup>Engineering University of PAP, Key Laboratory of Network and Information Security of PAP, Xi'an, China

\*Corresponding Author: Jia Liu. Email: liujial022@gmail.com

Received: 05 March 2026; Accepted: 09 June 2026; Published: 23 July 2026

**ABSTRACT:** Implicit Neural Representation (INR) is a technique that models continuous signals using neural networks, replacing traditional discrete grid representations with a coordinate-to-value mapping function. As a data carrier, INR is gradually being adopted as the target for steganographic processing. However, existing INR-based steganographic schemes typically require modifying network structures (e.g., weights, nodes) and retraining to obtain stego INRs, leading to high time consumption and the need for re-training when replacing cover images. To address this issue, this paper proposes StegaMIR (Steganography via Modulated Implicit Representations), an image steganographic scheme based on modulated implicit representations. It adopts the Functa (Fast Implicit Neural Representation Framework) framework for fast data implicit representation, which represents data as a base network and a corresponding modulation vector. To embed messages, a fixed message extractor is introduced to constrain the image reconstructed by the modulation vector and the base network, yielding stego modulation, which is then combined with the base network to construct the stego INR. The sender samples a stego image from it for transmission, while the receiver recovers the secret message using the fixed message extractor, thereby completing covert communication. Experiments demonstrate that StegaMIR can efficiently construct an INR-based steganographic model for images, achieving a message extraction accuracy of over 97% with an average embedding time of 8 s per image, representing a 95.6% reduction compared to existing methods.

**KEYWORDS:** Implicit neural representation; INR-based steganography; fixed message extractor; information hiding; meta-learning

## 1 Introduction

Steganography is a technique for embedding secret information into multimedia carriers to achieve covert communication. In recent years, the application of deep neural networks to image steganography has become a major research focus [1–3]. These methods primarily utilize the encoding capability of neural networks to encode the secret message and the original image into a stego carrier, and then use a decoder network to recover the secret message from the stego carrier. While these methods enhance the concealment and image quality of steganography via neural networks' learning ability, their high model complexity causes high training costs and insufficient robustness against attacks. Particularly, the encoder-decoder networks are often only capable of processing a specific type of sample, lacking generalizability, which greatly limits the further application of deep steganography.

To address the poor generalizability of deep steganography, researchers have begun exploring Implicit Neural Representation (INR) for steganography. INR [4] is a method that uses neural networks to encode data (e.g., images, 3D shapes) into continuous functions. For instance, mapping 2D pixel coordinates to

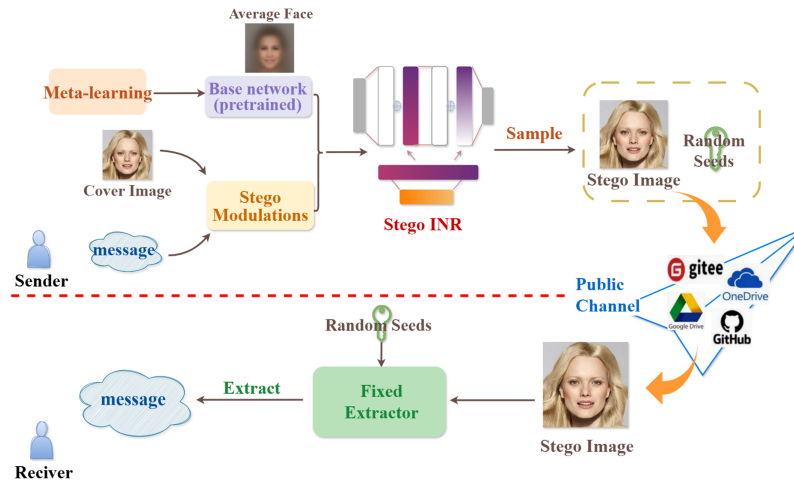
Red-Green-Blue (RGB) values in image representation. Existing INR-based steganography schemes mainly fall into two categories: One compressing the network structure [5], pruning redundant neurons to form an INR for the secret image and then reactivating them to construct an INR for the cover image. The other modifies the original network structure [6,7] typically training an INR for the secret data first and then expanding or pruning neurons to derive an INR for the cover data. However, these methods inherently require modifying the INR network structure. Despite their advantages in concealment and stego image quality, they share two key limitations: first, each steganographic model requires training two separate INRs (for secret and cover data), effectively doubling the training effort; second, all INR networks are initialized randomly and trained from scratch, leading to considerable time consumption.

We find that fast construction of INR for multimedia is critical to addressing the aforementioned problems. Dupont et al. [8] proposed the Functa scheme, which decomposes data's INR into a pre-trained base network and a rapidly obtainable modulation vector; their combination enables fast INR construction. Zhang et al. [9] presented the attention-driven localized INR (ANR), integrating a localized attention layer with a global Multi-Layer Perceptron (MLP) and using feature tokens R-Tokens (Representation Tokens) to replace traditional MLP weight parameters, enhancing high-frequency signal processing. Chen et al. [10] proposed NeRV-Enc/Dec (Neural Representation Video Encoder/Decoder): the former accelerates video INR encoding via a Transformer hypernetwork for direct weight generation. Additionally, schemes [11–13] enhanced INR efficiency via low-complexity architectures, joint optimization with regularization, and direct continuous signal reconstruction.

This paper adopts the Functa scheme for fast implicit neural representation, and on this basis, proposes a fast INR-based image steganography scheme via modulation, termed StegaMIR. As demonstrated in Fig. 1, our scheme first obtains the pre-trained base network from the Functa (Fast Implicit Neural Representation Framework) framework, and then stego modulations are derived under constraints from pre-defined secret messages cover images. Specifically, a fixed message extractor is introduced during training to constrain the image reconstructed by the modulation and the base network, enabling the fixed message extractor to recover the secret message from it. After training, combining the stego modulation with the base network yields the stego INR of the image data. The sender samples a stego image from this representation and transmits it alongside a random seed over a public channel. And the receiver initializes the fixed message extractor using the random seed and recovers the secret message from the stego image, completing covert communication. An alternative is to publicly transmit the stego INR network corresponding to the stego image; the receiver can then sample the stego image independently for secret message extraction. The motivation of this work is to address the above limitations by designing a fast, retraining-free steganographic scheme that can embed messages into any cover image within seconds using a pre-trained base network and a fixed message extractor.

To conclude, the main contributions of this paper can be summarized as follows:

1. Using a fast and efficient implicit neural representation method for steganography, reducing the per-image embedding time from 180 s (Luo et al. [6]) to 8 s—a 95.6% reduction and improving training efficiency.
2. A fixed message extractor is adopted to resolve the challenge of its secure concealed transmission, which reduces the risk of exposing the steganographic activities and lowering resource consumption during transmission.
3. A joint optimization framework integrating image reconstruction loss and message extraction loss is utilized to generate a stego modulation that well balances image reconstruction quality and message extraction accuracy.



**Figure 1:** Practical application diagram of StegaMIR.

## 2 Related Work

### 2.1 Traditional Steganography

Steganography conceals the existence of secret information for its transmission, with the core objective of ensuring the imperceptibility of the secret message. Traditional steganography is primarily categorized into three types: spatial domain, frequency domain, and compressed domain steganography. Spatial domain steganography embeds information by directly modifying the Least Significant Bits (LSB) [14,15] or sample values [16] of carrier pixels, leveraging human sensory robustness to minor perturbations. Chowdhuri et al. [17] proposed a key-controlled Discrete Cosine Transform (DCT) weighted-matrix scheme for highly compressed YCbCr images. Frequency domain steganography applies mathematical transforms [18] to the carrier, embeds information in transformed frequency domain coefficients, and exploits human sensory sensitivity differences to high/low-frequency components for message imperceptibility. Compressed domain steganography embeds secret information directly into compressed data, e.g., Joint Photographic Experts Group (JPEG), Moving Picture Experts Group Layer-3 (MP3), improving steganographic efficiency and carrier robustness. Though all three techniques hide information via carrier modification, they face growing threats with the continuous development of deep learning and steganalysis techniques.

### 2.2 Deep Steganography

Recent advances in neural network-based image feature extraction have propelled deep learning techniques [19] into image steganography. These methods leverage neural networks' powerful encoding capabilities to embed secret messages and cover images into stego carriers, while employing decoders to recover hidden information. Ref. [1] first introduced deep neural networks as encoders for image steganography, but it faced challenges in image quality and security. Subsequent works [2,3,20,21] have enhanced decoder robustness and steganographic capacity through innovations including noise layers, invertible architectures, flow-based frameworks, and fixed networks with controllable noise. Recently, Ref. [22] proposed a robust color image watermarking scheme using quaternion Charlier moments, Discrete Wavelet Transform (DWT), QR Decomposition and Singular Value Decomposition (QR-SVD), and chaotic mapping.

However, these approaches universally depend on real natural cover images for constructing steganographic data. Despite substantial progress in capacity and robustness, deep steganography confronts three

persistent challenges: transmission of message extractors elevates security risks; encoder-decoder networks lack multimodal generalizability; and extensive training requirements compromise time efficiency.

### 2.3 INR-Based Steganography

INR-based steganography refers to a class of steganographic schemes that utilize INR technology to convert multimedia data into functional data, and then employ function modification (including pruning and expansion) or function generation methods to construct a stego implicit representation function. These schemes typically exploit parameter redundancy for information hiding. While Han et al. [23] proposed an INR-based multimodal framework, it embeds implicit data in explicit images, remaining essentially traditional. Luo et al. [6] first proposed INR function steganography, representing secrets as INRs then expanding them as stego carriers, thereby unifying data formats. Yang et al. [24] concatenated multiple INRs with scrambling, though output degradation limited practical concealment. Subsequent work by Dong introduced neuron pruning [5] and multi-image hiding [25], respectively achieving simultaneous secret-stego representation and unified functional formatting. Recently, Ref. [26] enabled cross-modal hiding via key-based neuron selection.

However, the aforementioned schemes all require separately performing implicit neural representation for both the secret image and the carrier image before obtaining the steganographic model, which consumes considerable training resources and time. Therefore, we designed an image steganography scheme based on modulated implicit representations, utilizing INR to rapidly construct the steganographic model for images.

## 3 Preliminaries

### 3.1 Data Points as INRS

An INR is a function  $f_\gamma$  that maps spatial coordinates to feature values (e.g., RGB pixel values for images) parameterized by  $\gamma$ , by minimizing the mean squared error to fit the data points. Specifically, given a set of coordinates  $\{x_j\}_{j \in J}$  and corresponding features  $\{t_j\}_{j \in J}$  (where  $J$  is the index set of all pixel positions for an image), the goal is to minimize the following loss function:

$$\min_{\gamma} L(f_\gamma, \{x_j, t_j\}_{j \in J}) = \min_{\gamma} \sum_{j \in J} \|f_\gamma(x_j) - t_j\|_2^2 \quad (1)$$

Here,  $\gamma$  represents the parameters of the neural network,  $x_j$  is a spatial coordinate, and  $t_j$  is the corresponding feature value. The loss function is the Mean Squared Error (MSE), summed over all coordinate points. This paper uses the Sinusoidal Representation Network (SIREN) [4] as the foundational architecture for the INR of images. SIREN is a type of MLP, with each layer parameterized as:

$$x \mapsto \sin(\omega_0(Wx + b)) \quad (2)$$

where  $W$  and  $b$  are trainable parameters,  $\omega_0$  is a fixed hyperparameter, and  $x$  is the pixel coordinate. Eq. (2) represents the mapping from coordinate  $x$  to the output value  $y$ .

### 3.2 Fast Implicit Representation via Modulation

Dupont et al. [8] proposed Functa, a fast implicit representation framework that treats each data point as an individual function  $f_\gamma$  rather than a complete INR model. Using a meta-learning strategy, their approach separates the representation process into inner and outer loops. It pretrains a base network that captures generic image features (e.g., edges, textures). Meanwhile, a continuous low-dimensional modulation vector is randomly initialized and trained in an inner loop to fit the instance-specific characteristics of a single image.



Modulation training can be completed within seconds. Therefore, based on a pre-trained base network, representing a single image only requires seconds to obtain its corresponding modulation, which can then be combined with the base network to quickly derive an implicit neural representation  $f_\gamma$  of the image.

Building upon the SIREN-based INR, it introduced a modulation vector  $s$  as a bias term for each layer, designed to capture differences between data points. All data points share the base network weights  $W$  and  $b$ , differing only in their modulation vectors. Eq. (3) shows the parameterized form of the  $j$ -th network layer after incorporating modulation:

$$x \mapsto \sin(\omega_0(W^{(j)}x + b^{(j)} + s^{(j)})) \quad (3)$$

To further compress the representation, Functa introduces a latent vector  $\phi$  (hereafter referred to as the modulation), which is mapped to the modulation vectors  $s^{(j)}$  for all layers via a linear layer:

$$s^{(j)} = W_\phi^{(j)}\phi + b_\phi^{(j)} \quad (4)$$

where  $W_\phi$  and  $b_\phi$  are the weight and bias for mapping the modulation vector. Thus, for each data point, only its corresponding low-dimensional vector  $\phi$  needs to be stored. Therefore, the modulation  $\phi$  can be understood as an efficient representation in Functa. Its parameterized form can be expressed as:

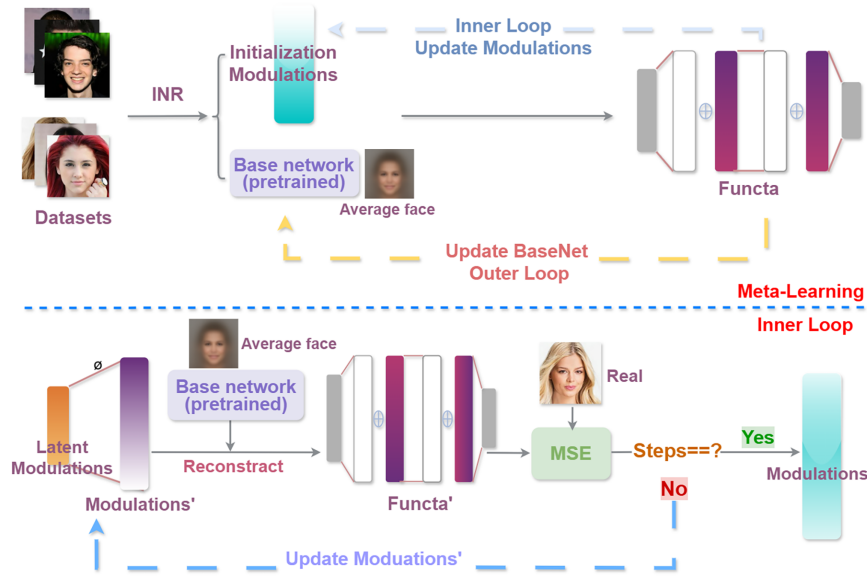
$$x \mapsto \sin(\omega_0(W^{(j)}x + b^{(j)} + W_\phi^{(j)}\phi + b_\phi^{(j)})) \quad (5)$$

$$= f_{W,b,W_\phi,b_\phi}(x, \phi) \quad (6)$$

$$= f_{\gamma_{\text{base}}}(x, \phi) \quad (7)$$

where  $f_{\gamma_{\text{base}}}$  is the shared base network,  $\gamma_{\text{base}}$  represents the parameters of the base network.

The base network and modulation signals are learned through an iterative bi-level optimization process (Fig. 2). Training proceeds in batches, with each batch undergoing both outer and inner loop optimization.



**Figure 2:** Training of the base network and the modulation.

In the outer loop, the base network parameters (such as  $W, b, W_\phi, b_\phi$ ) are updated while keeping modulation vectors fixed, minimizing the expected inner-loop loss:

$$\min_{\gamma} E_{D \sim p(D)} [L_{\text{inner}}(\phi^*(\gamma), D)] \quad (8)$$

where  $D$  represents a data point (a set of coordinate-feature pairs),  $L_{\text{inner}}$  is the inner loop loss, and  $E_{D \sim p(D)}$  is the expectation. Once the outer loop is completed, the process moves to the inner loop. At this stage, the modulation is optimized for each data point while the base network parameters remain fixed:

$$\phi^*(\gamma) = \arg \min_{\phi} L_{\text{inner}}(\phi; D; \gamma) \quad (9)$$

This alternating optimization continues across batches until convergence, yielding the trained base network and corresponding modulation signals for the entire dataset.

#### 4 Proposed Method

Adopting the Functa formulation as the backbone for efficient implicit neural representation, this paper designs a scheme for rapidly constructing steganographic models by introducing a fixed message extractor into the modulation optimization loop. Fig. 3 illustrates the overall architecture of StegaMIR.

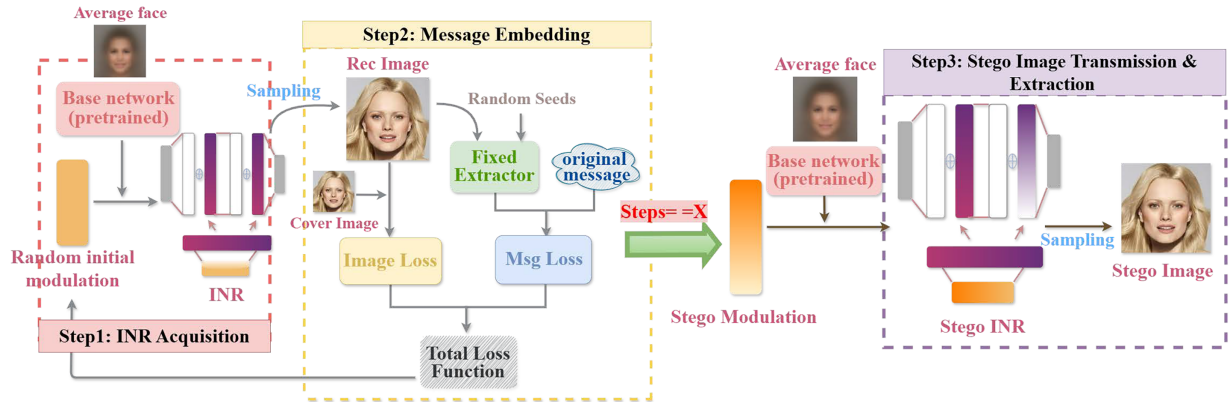


Figure 3: Flowchart of the StegaMIR.

For quick reference, the key mathematical symbols used in this section are summarized below.

| Symbol                 | Description                                 |
|------------------------|---------------------------------------------|
| $\phi$                 | Modulation vector                           |
| $\lambda_{\text{msg}}$ | Weight for message loss                     |
| $\alpha$               | Weight balancing sign loss and weighted MSE |
| $L_{\text{rec}}$       | Reconstruction loss (MSE)                   |
| $L_{\text{msg}}$       | Message loss                                |
| $L_{\text{sign}}$      | Sign loss component                         |
| $L_{\text{weighted}}$  | Weighted MSE component                      |
| $L_{L_2}$              | L2 regularization penalty                   |

The base network follows the Functa framework and is implemented as a 5-layer MLP with a hidden width of 256, using SIREN activation functions with  $\omega_0 = 30$ . The modulation vector  $\phi$  has a dimension of 64, 128, 256, or 512 in our experiments. The overall scheme is divided into three parts:

**Step 1 INR acquisition:** the INR of the cover image is obtained:

$$x_{\text{cov}} \mapsto \sin(\omega_0(W^{(j)}x + b^{(j)} + W_{\phi}^{(j)} + b_{\phi}^{(j)})) \quad (10)$$

where  $x_{\text{cov}}$  represents the pixel coordinates of the cover image.

**Step 2 Message embedding (inner loop):** during the inner loop, a fixed message extractor  $E_{\text{fix}}$ , initialized with a random seed  $S_{\text{ran}}$ , is used to extract messages from the modulation. The extracted result is compared with the predefined secret message SeM (a binary bit string of arbitrary length) using a loss function, enabling effective message embedding. Concurrently, the MSE loss constrains the difference between the reconstructed image and the original image, ensuring the quality of the stego image. The inner loop terminates after a predefined number of steps, yielding the optimized stego modulation  $\phi_{\text{stego}}$ :

$$\phi_{\text{stego}} = \text{Adam}_{\phi} [L_{\text{rec}} + \lambda_{\text{msg}} \cdot L_{\text{msg}}] \quad (11)$$

where  $\text{Adam}_{\phi}$  is the optimizer used for message embedding,  $L_{\text{rec}}$  and  $L_{\text{msg}}$  are the reconstruction image loss and message loss, respectively, and  $\lambda_{\text{msg}}$  is the weight of the message loss.

**Step 3 Stego image transmission & extraction:** the stego modulation is applied to the pretrained base network to obtain the INR network representing the stego image  $I_{\text{stego}}$ , completing the efficient INR-based steganographic model. The sender samples the stego image from this representation for transmission:

$$I_{\text{stego}} = f_{\gamma_{\text{base}}}(x; \phi_{\text{stego}}) \quad (12)$$

After receiving the image, the receiver uses the  $S_{\text{ran}}$  to initialize the fixed message extractor and recover the secret message:

$$\text{SeM} = E_{\text{fix}}(I_{\text{stego}}) \quad (13)$$

The following sections detail the designs of the loss function and fixed message extractor.

#### 4.1 Loss Function

The loss function achieves dual-objective optimization through a weighted combination of the reconstruction and message extraction losses, constrained by the mean squared error:

$$L_{\text{total}} = L_{\text{rec}} + L_{L_2} + \lambda_{\text{msg}} * L_{\text{msg}} \quad (14)$$

Here,  $L_{L_2}$  is the  $L_2$ -norm penalty on the modulation added in the Functa to prevent overfitting. A message loss weight  $\lambda_{\text{msg}}$  dynamically balances the trade-off between visual fidelity and message extraction accuracy during inner-loop optimization.

This concurrent optimization strategy prevents the degradation of either objective that might occur in sequential training approaches.

The image reconstruction loss is specifically defined as:

$$L_{\text{rec}} = \text{MSE}(I_{\text{rec}}, I_{\text{tar}}) = \frac{1}{N} \sum_{i=1}^N (I_{\text{rec}_i} - I_{\text{tar}_i})^2 \quad (15)$$

where  $I_{\text{rec}}$  is the reconstructed image,  $I_{\text{tar}}$  is the target image,  $N$  is the number of pixels in the image, and  $i$  denotes the  $i$ -th pixel.

The message loss integrates two complementary components: a sign loss that enhances robustness through bit-wise sign comparison, mitigating noise interference; and a weighted mean squared error that refines numerical accuracy for improved extraction precision. Specifically:

$$L_{\text{sign}} = \frac{1}{M} \sum_{k=1}^M \max(0, 1 - \text{Msg}_{E_k} \cdot \text{Msg}_{O_k}) \quad (16)$$

$$L_{\text{weighted}} = \frac{1}{M} \sum_{k=1}^M (\text{Msg}_{E_k} - \text{Msg}_{O_k})^2 \quad (17)$$

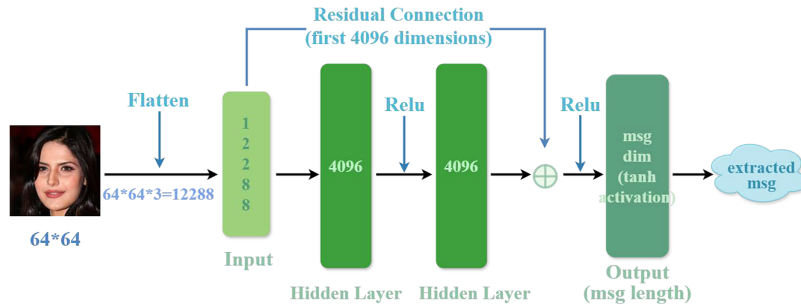
$$L_{\text{msg}} = \alpha \cdot L_{\text{sign}} + (1 - \alpha) \cdot L_{\text{weighted}} \quad (18)$$

where  $\alpha$  is the weight balancing the sign loss and the weighted loss,  $\text{Msg}_E$  is the extracted message,  $\text{Msg}_O$  is the original message,  $M$  is the number of bits in the message, and  $k$  denotes the  $k$ -th bit.

$\lambda_{\text{msg}}$  balances image quality and extraction accuracy. A larger value (e.g., 0.5) prioritizes accuracy at the cost of fidelity; a smaller value (e.g., 0.01) preserves quality but may reduce accuracy. Based on our ablation study (Section 5.9), we set  $\lambda_{\text{msg}} = 0.05$  for a balanced trade-off.  $\alpha$  balances the sign loss and weighted MSE. A higher  $\alpha$  (e.g., 0.9) emphasizes robustness against noise; a lower  $\alpha$  (e.g., 0.5) prioritizes numerical accuracy for faster convergence. We empirically set  $\alpha = 0.7$  to prioritize robustness while maintaining stable optimization.

#### 4.2 Fixed Message Extractor

We employ a fixed message extractor [21,27] to eliminate transmission overhead and prevent steganalysis suspicion associated with transmitting trainable parameters. The input to the extractor is the flattened pixel values of the entire stego image (resized to  $64 \times 64 = 12,288$  dimensions for RGB images), not intermediate features. The extractor uses a permanently frozen neural network with a three-layer fully connected architecture, comprising two 4096-dimensions hidden layers and an output layer that matching the secret message length. All parameters are initialized via a fixed random seed (set to 17 in our experiments). Once initialized, the parameters are permanently frozen and do not participate in any training process, ensuring the consistency and reliable message extraction. In practical applications, the random seed serves as a secret key shared between the sender and receiver, rather than a public constant. The detailed structure is shown in Fig. 4.



**Figure 4:** Fixed message extractor network architecture.

#### 4.3 Algorithm Overview

The overall encoding process of StegaMIR is summarized in Algorithm 1.

**Algorithm 1:** StegaMIR encoding

**Require:** Cover image  $I_{\text{cov}}$ , secret message  $SeM$ , pre-trained base network  $f_{\text{base}}$ , fixed extractor  $E_{\text{fix}}$ , message loss weight  $\lambda_{\text{msg}}$

**Ensure:** Stego modulation  $\phi_{\text{stego}}$ , stego image  $I_{\text{stego}}$

- 1: Initialize modulation vector  $\phi$  randomly
- 2: Generate coordinate grid  $x$  for image resolution
- 3: for step = 1 to  $N_{\text{steps}}$  (e.g., 100)
- 4:    $I_{\text{rec}} \leftarrow f_{\text{base}}(x; \phi)$
- 5:    $Msg_{\text{ext}} \leftarrow E_{\text{fix}}(\text{flatten}(I_{\text{rec}}))$
- 6:    $L_{\text{rec}} \leftarrow \text{MSE}(I_{\text{rec}}, I_{\text{cov}})$
- 7:    $L_{\text{msg}} \leftarrow \alpha \cdot L_{\text{sign}} + (1 - \alpha) \cdot L_{\text{weighted}}$
- 8:    $L_{\text{total}} \leftarrow L_{\text{rec}} + \lambda_{\text{msg}} \cdot L_{\text{msg}}$
- 9:    $\phi \leftarrow \text{Adam}(L_{\text{total}}, \phi)$
- 10: end for
- 11:  $\phi_{\text{stego}} \leftarrow \phi$
- 12:  $I_{\text{stego}} \leftarrow f_{\text{base}}(x; \phi_{\text{stego}})$
- 13: **return**  $\phi_{\text{stego}}, I_{\text{stego}}$

Decoding follows Eq. (13): the receiver initializes  $E_{\text{fix}}$  with the shared random seed and extracts  $SeM \leftarrow E_{\text{fix}}(\text{flatten}(I_{\text{rec}}))$ .

## 5 Experiments

Our experiments were conducted using the JAX 0.5.0 (Python 3.12) deep learning framework on an NVIDIA GeForce RTX 2070 GPU (CUDA 12.7), with CelebA-HQ [28] (64 \* 64) for training and validation. INR optimization used Stochastic Gradient Descent (SGD) (lr = 5e-1), while message embedding employed the Adam optimizer. Empirical results demonstrated that training for 100 iteration steps per image yielded well-performing stego modulations. During training, the message loss weight was set to 0.1, and the fixed message extractor's random seed was initialized to 17. Additional hyperparameters: the L2 regularization weight is 0.0001, the sign loss weight  $\alpha$  in Eq. (18) is 0.7, and image coordinates are normalized to [0, 1] with pixel-center alignment.

### 5.1 Evaluation Metrics

We evaluated steganographic performance through accuracy, fidelity, efficiency, robustness, undetectability and capacity. For the purpose of evaluating how the choice of message loss weights affects steganographic outcomes, ablation experiments pertaining to the associated hyperparameters were additionally designed and implemented. The following metrics were established for quantitative evaluation: Accuracy (Acc) for message extraction to restore fidelity, Peak Signal-to-Noise Ratio (PSNR) [29] for pixel-level similarity, and Learned Perceptual Image Patch Similarity (LPIPS) for perceptual quality.

### 5.2 Accuracy

We evaluate the message extraction accuracy of our model by computing the discrepancy between the extracted message and the original payload. To assess the performance of the proposed scheme, we conduct a comparative study against state-of-the-art steganographic methods, including deep learning-based approaches such as HiDDeN (Hiding Data With Deep Networks) [2], FNNS (Fixed Neural Network Steganography) [21], Diffusion-Stego (Diffusion Model-Based Steganography) [30], etc., and implicit neural

representation-based methods. The comparative results are summarized in Table 1, which reports both the secret recovery accuracy (for binary message hiding methods) and the cover image quality (for all methods).

**Table 1:** Accuracy and recoverability performance of different steganographic schemes.

| Category                          | Methods              | Image Size               | Recoverability |           | Cover PSNR (dB) |
|-----------------------------------|----------------------|--------------------------|----------------|-----------|-----------------|
|                                   |                      |                          | Acc (%)        | PSNR (dB) |                 |
| Deep Learning-Based Steganography | HiDDen [2]           | 16 × 16 (COCO)           | 99             | —         | 44.63           |
|                                   | FNNS [21]            | 64 × 64 (CelebA)         | 100            | —         | 36.06           |
|                                   | Diffusion-Stego [30] | 64 × 64 (CelebA)         | 98             | —         | NR              |
|                                   | StegaDDPM [31]       | 64 × 64 (CelebA)         | 100            | —         | NR              |
|                                   | SteganoGAN [32]      | 64 × 64 (COCO)           | 78–87          | —         | 25.98           |
| INR-Based Steganography           | StegaINR4MIH [5]     | 64 × 64 (CelebA-HQ)      | —              | 42–45     | 44.03           |
|                                   | Luo et al. [6]       | 64 × 64 (CelebA-HQ)      | —              | 32.50     | 24.80           |
|                                   | Dong et al. [33]     | 64 × 64 (NeRF Synthetic) | 100            | —         | 26.43           |
|                                   | GINR-Stega [34]      | 64 × 64 (CelebA)         | 100            | —         | 32.50           |
|                                   | Zhong et al. [35]    | 64 × 64 (CelebA-HQ)      | 100            | —         | 26.86           |
| <b>OURS</b>                       |                      | 64 × 64 (CelebA-HQ)      | <b>98</b>      | <b>—</b>  | <b>28.70</b>    |

Note: “—” indicates not applicable; “NR” indicates not reported. For binary message methods, recoverability PSNR is not meaningful.

Our method achieves 97% extraction accuracy, which is competitive with existing binary steganography methods. In terms of cover image quality, our method achieves 28.70 dB PSNR, which is comparable to or better than several existing methods.

### 5.3 Efficiency

The core competitiveness of the StegaMIR lies in its ability to rapidly construct stego implicit representation models for new cover images. Table 2 presents a comparison of training times with other steganographic methods. It is important to distinguish between model training time (one-time cost) and time per new carrier.

**Table 2:** Computational time, model parameters, and complexity comparison of different steganographic schemes.

| Category                          | Methods           | Training Times | Embedding            | Params (M)  | FLOPs (G)  |
|-----------------------------------|-------------------|----------------|----------------------|-------------|------------|
|                                   |                   |                | Time on New Carriers |             |            |
| Deep Learning-based Steganography | HiDDen [2]        | 40–80 h        | ~1 s                 | 12.4        | 4.2        |
|                                   | FNNS [21]         | 10 s–3 min     | 10 s–3 min           | 0.5         | 1.8        |
|                                   | StegaDDPM [31]    | 4–9 day        | 2–10 s               | —           | —          |
|                                   | SteganoGAN [32]   | 5–64 h         | ~1 s                 | —           | —          |
| INR-based Steganography           | Luo et al. [6]    | ~0.5 h         | 1–3 min              | ~8.6        | ~3.1       |
|                                   | Dong et al. [33]  | 20–30 h        | 20–30 h              | ~36.3       | ~15.0      |
|                                   | GINR-Stega [34]   | 75 h           | 3–27 s               | 12.5        | ~2.1       |
|                                   | Zhong et al. [35] | 24–80 h        | ~1 s                 | 27.4        | 3.5        |
| <b>OURS</b>                       |                   | <b>35 h</b>    | <b>8 s</b>           | <b>25.0</b> | <b>0.4</b> |

Note: “—” indicates that the corresponding value is not reported in the original paper. For our method, Training Time refers to the one-time pre-training of the base network (35 h), and Embedding Time refers to optimizing the modulation vector only (8 s per image).

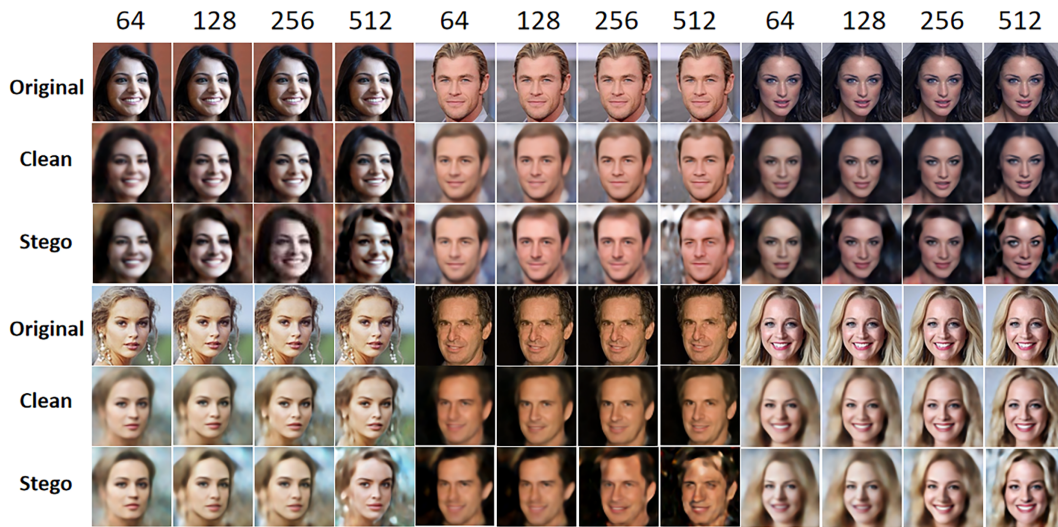


As shown in Table 2, methods like HiDDeN [2] and Zhong et al. [35] require substantial end-to-end training (5–80 h) before use for steganography. Methods like FNNS [21] and Luo et al. [6] have shorter training times but require full retraining or significantly longer processing (10 s–20 h) for each new cover image. Crucially, unlike existing steganographic methods that require end-to-end training of a dedicated steganographic model on large datasets, our approach decouples the process: the base network is pre-trained once, and secret messages are directly embedded into the modulation representation of any cover image without additional model training.

In summary, our method eliminates the need for end-to-end training of a steganographic model for each new carrier, offering a clear efficiency advantage for scenarios involving frequent carrier updates.

#### 5.4 Fidelity

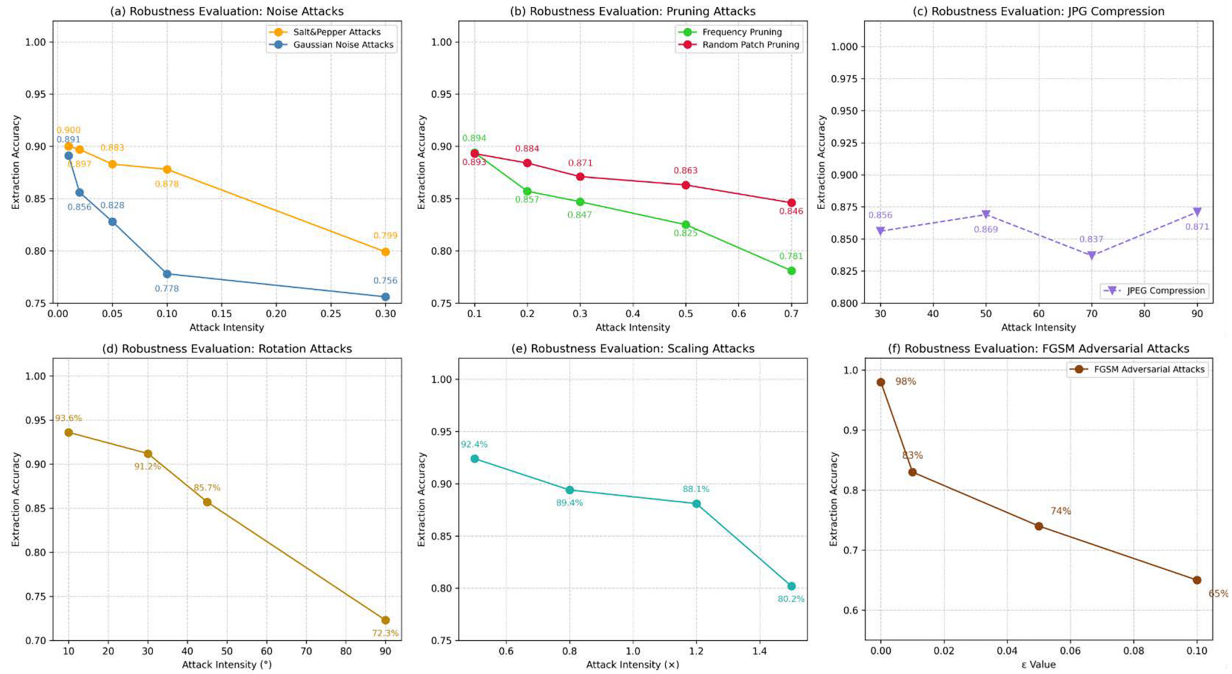
We evaluate model fidelity by reconstructing both clean and stego images using modulation dimensions of 64, 128, 256, and 512. Fig. 5 presents comparative results across six image groups, each displaying original images (first row), clean modulation reconstructions (second row), and stego modulation reconstructions (third row) across the four dimensions. Visual assessment shows minimal perceptual divergence between stego and clean images at 128 or 256 modulation dimensions, thereby confirming high fidelity.



**Figure 5:** Comparison of original images (first row), clean modulation reconstructions (second row), and stego modulation reconstructions (third row) for modulation dimensions  $\phi = 64, 128, 256, 512$ .

#### 5.5 Robustness

We evaluate our model's resilience through four attack categories: 1) noise attacks including Gaussian noise and salt-and-pepper noise; 2) JPEG compression; 3) network attacks via random and frequency pruning; 4) geometric attacks (rotation and scaling) and 5) adversarial attacks using Fast Gradient Sign Method (FGSM). Fig. 6 shows that after being subjected to attacks, our model can still maintain message extraction accuracy above the baseline level, except for random pruning which achieved 77% accuracy. Under FGSM adversarial attacks, the accuracy drops to 78% at  $\epsilon = 0.01$  and 61% at  $\epsilon = 0.05$ .



**Figure 6:** Message extraction accuracy under various attacks: (a) salt-and-pepper noise, (b) Gaussian noise, (c) frequency pruning, (d) JPEG compression, (e) rotation & scaling, and (f) FGSM adversarial attacks.

We analyze that random block pruning drastically reduces message extraction accuracy by discarding local pixel blocks—destroying INR’s required coordinate-feature mapping and interfering with the extractor’s global feature reliance. In contrast, frequency pruning preserves global structures, while noise attacks and JPEG compression only introduce partial distortion—maintaining model robustness.

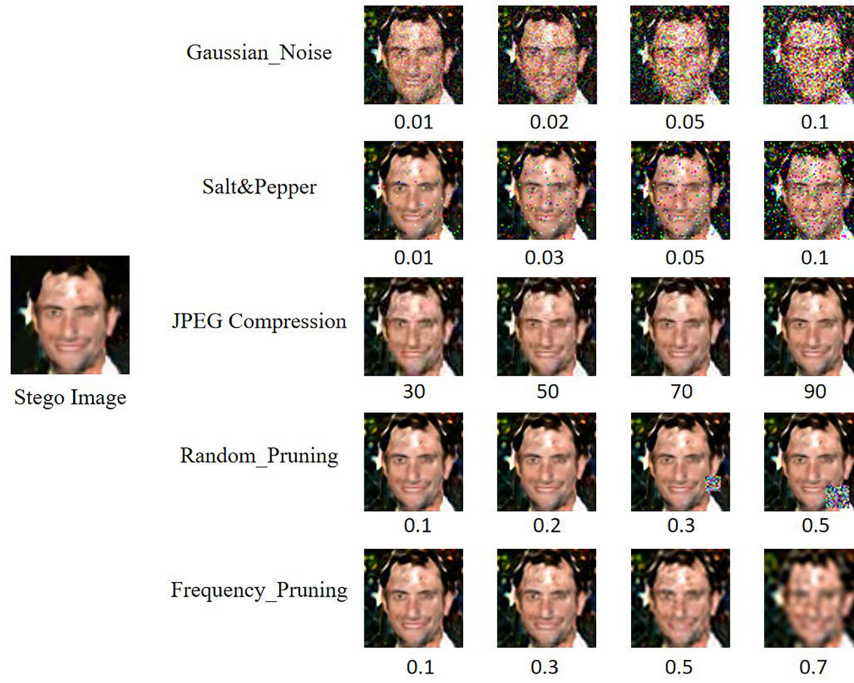
Fig. 7 presents stego images following the five attack types. While noise attacks substantially degrade visual quality, JPEG compression and pruning remain perceptually negligible at lower intensities, with artifacts becoming pronounced only under heightened attack strength.

### 5.6 Undetectability

To evaluate the undetectability of our stego images, we first apply the Spatial Rich Model (SRM) [36] with an ensemble classifier—a standard steganalysis benchmark. Using 100 clean/stego image pairs ( $\phi = 128$ , 256-bit payloads), we extract SRM features and perform 5-fold cross-validation.

As shown in Table 3, the detection accuracy is 52.3%, close to random guessing, confirming that our stego images are statistically indistinguishable from clean ones.

Beyond SRM, we preliminarily evaluated our stego images using a pre-trained Convolutional Neural Network (CNN)-based detector, SiaStegNet [37]. On 100 stego images, it achieved a detection accuracy of approximately 60%, only 10% above random guessing (50%). While slightly higher than SRM’s 52.3%, this result suggests our method maintains a degree of resistance to CNN-based detectors. We attribute this to our embedding strategy: perturbations are distributed globally by optimizing a low-dimensional modulation vector, rather than making localized pixel-wise modifications. A comprehensive evaluation against detectors like XuNet [38] and Spatial Residual Network (SRNet) [39] is a key direction for our future work.



**Figure 7:** Visual examples of stego images after different attacks at varying intensities.

**Table 3:** Detection accuracy of SRM-based steganalysis.

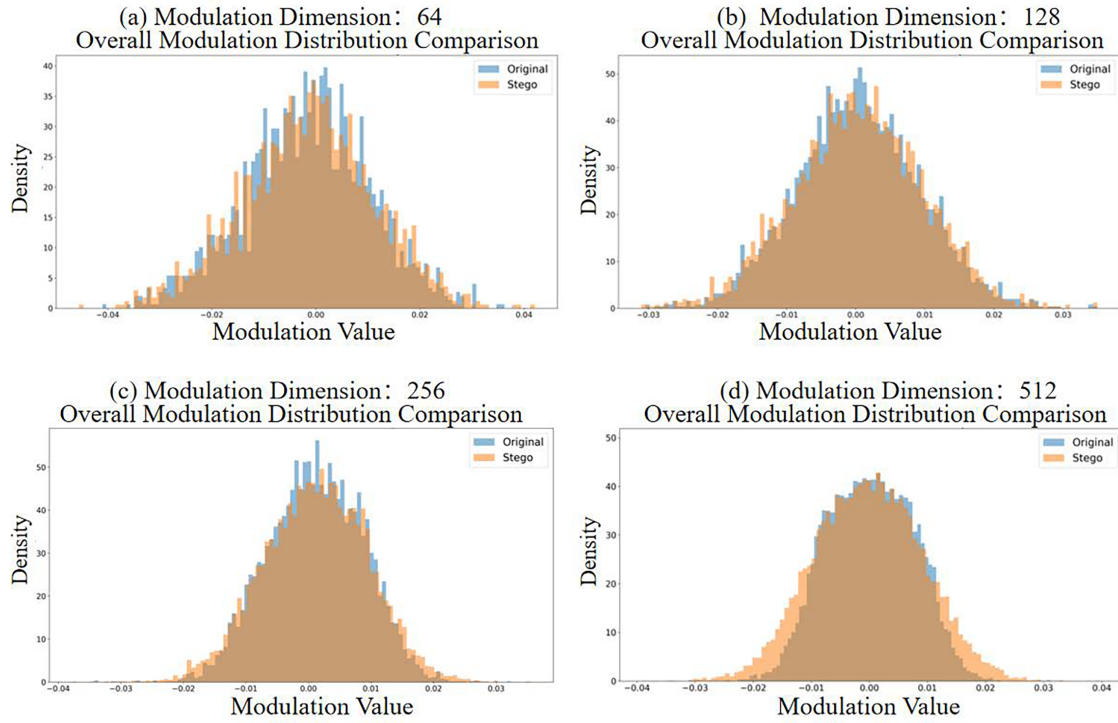
| Method                     | Detection Accuracy |
|----------------------------|--------------------|
| SRM + Ensemble Classifier  | 52.3%              |
| Random Guessing (Baseline) | 50%                |

To understand the origin of this resilience, we analyze the weight distributions of clean and stego modulations across four dimensions ( $\phi = 64, 128, 256, 512$ ), with 100 images per group. As illustrated in Fig. 8, the distributions largely overlap for  $\phi = 64, 128, 256$ , indicating minimal parameter-level perturbations. Together, these results demonstrate both external undetectability and internal consistency of our scheme.

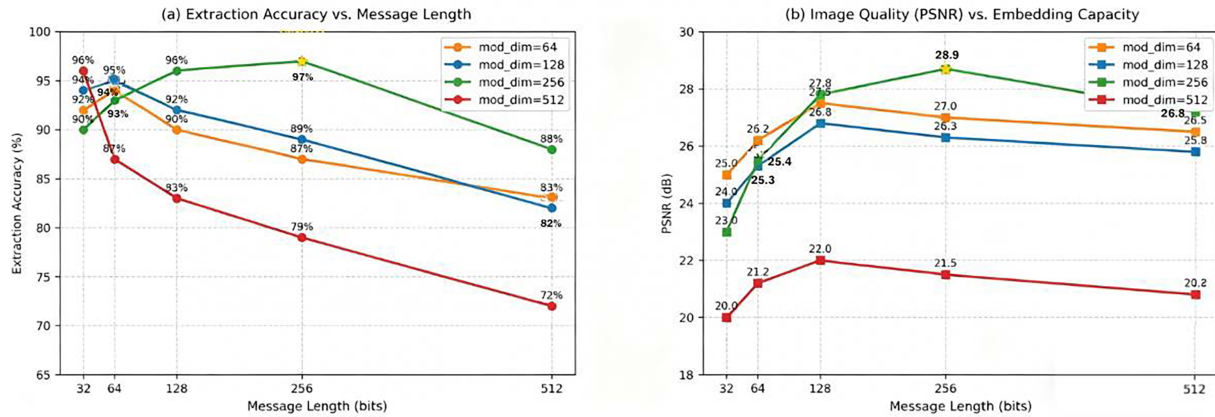
### 5.7 Capacity

We evaluate steganographic capacity by embedding messages of varying bit lengths using four modulation dimensions. Here, the modulation vector dimension determines the representational capacity of the INR, while the message length determines the embedding task difficulty.

As shown in Fig. 9a, for a fixed modulation dimension, extraction accuracy gradually declines as message length increases. Larger consistently achieves higher accuracy at the same message length, indicating that increased representational capacity benefits message embedding. Fig. 9b illustrates the trade-off between image quality and embedding capacity. As message length increases, PSNR decreases, reflecting the inherent compromise between hiding more information and maintaining visual fidelity. Among the tested dimensions, offer the best balance between capacity, accuracy, and image quality.



**Figure 8:** Modulation weight distribution comparison between clean (blue) and stego (yellow) INRs: (a) 64; (b) 128; (c) 256. (d) 512.



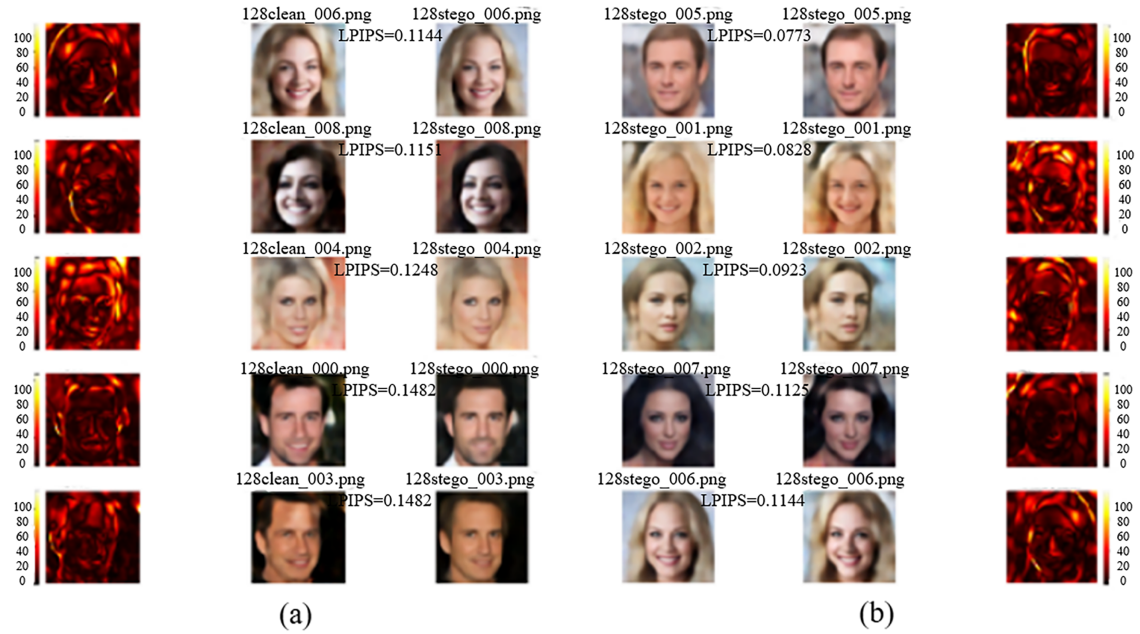
**Figure 9:** Accuracy and visual samples under varying message lengths and modulation dimensions. (a) Extraction accuracy under different message lengths; (b) PSNR under different message lengths.

### 5.8 Distortion

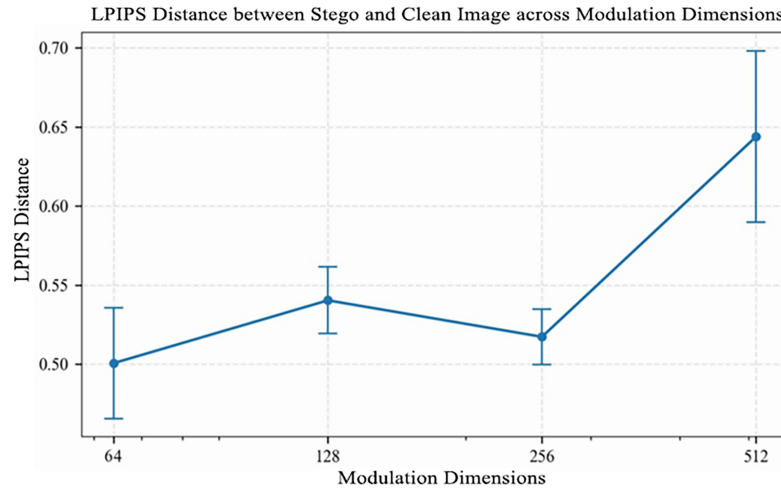
To evaluate the image quality before and after steganography, we employ the LPIPS metric—for which lower values indicate superior quality—by comparing images reconstructed from clean modulations and stego modulations under the 128-dimensional setting. Fig. 10 reveals case-dependent variations.

Fig. 11 quantifies the progressive increase in LPIPS values between clean and stego images with higher modulation dimensions. Corroborating Fig. 5's visual evidence, this confirms that elevated dimensions amplify perceptual discrepancies.





**Figure 10:** LPIPS comparison between clean and stego images under  $\phi = 128$ . (a) Minor distortions suggesting stronger embedding; (b) nearly identical appearance due to embedding in smoother regions.



**Figure 11:** LPIPS difference vs. modulation dimension, showing that higher dimensions amplify perceptual discrepancies.

### 5.9 Ablation

Using a message loss weight of 0.05 as baseline, we further investigated its impact on embedding performance by testing weights of 0.01, 0.1, and 0.5. As shown in Fig. 12, the 0.01–0.05 range maintains optimal extraction accuracy.

### 5.10 Overall Performance Comparison

To provide an overall comparison with existing methods, we summarize the key performance metrics of representative steganographic schemes in Table 4. As shown in the table, our method achieves 97% accuracy and 28.70 dB PSNR, which are competitive with existing approaches.

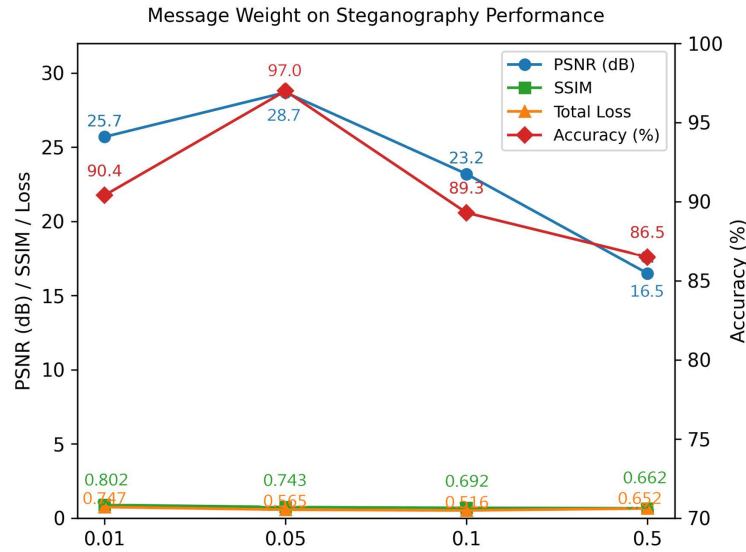


Figure 12: Impact of different message loss weights on experimental results.

Table 4: Overall performance comparison with state-of-the-art methods.

| Category                          | Method            | Acc (%)   | Cover PSNR (dB) | Embedding Time | Capacity (bpp) |
|-----------------------------------|-------------------|-----------|-----------------|----------------|----------------|
| Deep Learning-Based Steganography | HiDDen [2]        | 99        | 44.63           | 40–80 h        | 0.203          |
|                                   | FNNS [21]         | 100       | 36.06           | 10–180 s       | 1.000          |
|                                   | SteganoGAN [32]   | 87–78     | 25.98           | 1 s            | 0.400          |
| INR-Based Steganography           | Luo et al. [6]    | 100       | 24.80           | 6–180 s        | —              |
|                                   | Dong et al. [33]  | 100       | 26.43           | 20–30 h        | 6.00           |
|                                   | GINR-Stega [34]   | 100       | 32.50           | 3–27 s         | 1.00           |
|                                   | Zhong et al. [35] | 100       | 26.86           | 1 s            | 0.20           |
| <b>Ours</b>                       |                   | <b>98</b> | <b>28.70</b>    | <b>8 s</b>     | <b>0.50</b>    |

Note: Luo et al. [6] hides images rather than binary bit strings, so capacity is not reported. “—” indicates not applicable.

In terms of embedding time, our method (8 s) is significantly faster than FNNS [21] (10–180 s), Luo et al. [6] (6–180 s), and Dong et al. [33] (20–30 h). Although some methods report higher capacity, they require much longer embedding time or suffer from lower cover image quality. In addition, our method maintains strong robustness and good undetectability. Overall, StegaMIR offers a balanced trade-off across multiple performance dimensions.

## 6 Conclusion

This paper proposes StegaMIR, a fast image steganography scheme based on modulated INRs. By adopting the paradigm from the Functa framework, which decomposes INRs into a pre-trained base network and a modulation vector, our scheme enables rapid implicit encoding of image data. Meanwhile, a fixed message extractor is introduced to constrain the image reconstructed by the modulation and base network,

generating a stego modulation. This enables the efficient construction of a stego INR for target images. StegaMIR eliminates the requirement for retraining stego images and addresses the long training latency issue of existing INR-based steganography methods. Experiments demonstrate that StegaMIR achieves 97% message extraction accuracy, an average embedding time of 8 s per new carrier (a 95.6% reduction compared to existing INR-based methods), and maintains robust performance against common steganalysis attacks.

Due to the limitation of the Functa framework (the base network training code is not publicly available), our current experiments are restricted to the CelebA-HQ dataset at  $64 \times 64$  resolution. In future work, we will explore ways to train the base network independently and extend our steganographic scheme to more diverse datasets and even cross-modal scenarios. Nevertheless, the existing experiments have already demonstrated that our method significantly improves embedding efficiency.

Additionally, while our method demonstrates strong resistance to traditional statistical steganalysis (e.g., SRM with 52.3% detection accuracy), preliminary evaluations against CNN-based detectors show a detection accuracy of approximately 60%, indicating room for improvement. Enhancing robustness against deep learning-based steganalysis will be a key focus of our future research.

**Acknowledgement:** The authors acknowledge the support from the General Program of the National Natural Science Foundation of China. To ensure linguistic accuracy and adherence to academic conventions, the English translation of the manuscript text and abstract was initially completed by the authors, followed by AI-assisted polishing using the DeepSeek and DouBao large language models. The authors have subsequently conducted thorough manual review, verification, and optimization of the entire content, and take full responsibility for the final accuracy and integrity of the manuscript.

**Funding Statement:** This research was funded by the National Natural Science Foundation of China, grant numbers 62272478, 61872384 and 62102451. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

**Author Contributions:** The authors confirm contribution to the paper as follows: Conceptualization, Qiya Wang and Jia Liu; methodology, Qiya Wang; software, Qiya Wang; validation, Qiya Wang, Yuwei Lu and Yujie Liu; formal analysis, Qiya Wang; investigation, Qiya Wang and Peng Luo; resources, Jia Liu; data curation, Qiya Wang; writing—original draft preparation, Qiya Wang; writing—review and editing, Jia Liu; visualization, Qiya Wang; supervision, Jia Liu; project administration, Jia Liu; funding acquisition, Jia Liu. All authors reviewed and approved the final version of the manuscript.

**Availability of Data and Materials:** The data that support the findings of this study are available within the article, and the archived version of the code supporting the findings of this study is openly available in GitHub at <https://github.com/twinlj77/StegaMIR/>. Additional data are available from the Corresponding Author, Jia Liu, upon reasonable request.

**Ethics Approval:** This study involves no human/animal subjects, no ethics approval is required.

**Conflicts of Interest:** The authors declare no conflicts of interest. The funders played no role in this research.

## References

1. Baluja S. Hiding images within images. *IEEE Trans Pattern Anal Mach Intell.* 2019;42(7):1685–97. doi:10.1109/TPAMI.2019.2901877.
2. Zhu J, Kaplan R, Johnson J, Fei-Fei L. HiDDeN: hiding data with deep networks. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Cham, Switzerland: Springer; 2018. p. 657–72. doi:10.1007/978-3-030-01267-0\_40.



3. Jing J, Deng X, Xu M, Wang J, Guan Z. HiNet: deep image hiding by invertible network. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2021 Oct 10–17; Montreal, QC, Canada. Piscataway, NJ, USA: IEEE; 2021. p. 4733–42. doi:10.1109/ICCV48922.2021.00469.
4. Sitzmann V, Martel J, Bergman A, Lindell D, Wetzstein G. Implicit neural representations with periodic activation functions. *Adv Neural Inf Process Syst*. 2020;33:7462–73.
5. Dong W, Liu J, Chen L, Sun W, Pan X, Ke Y. Implicit neural representation steganography by neuron pruning. *Multimed Syst*. 2024;30(5):266. doi:10.1007/s00530-024-01476-9.
6. Luo P, Liu J, Ke Y, Zhang M, Mu D. Hiding functions within functions: steganography by implicit neural representations. *Tsinghua Sci Technol*. 2026;31(2):1058–74. doi:10.26599/TST.2025.9010030.
7. Chen L, Song C, Liu J, Sun W, Dong W, Di F. IW-NeRF: using implicit watermarks to protect the copyright of neural radiation fields. *Appl Sci*. 2024;14(14):6184. doi:10.3390/app14146184.
8. Dupont E, Kim H, Eslami S, Rezende D, Rosenbaum D. From data to functa: your data point is a function and you can treat it like one. *arXiv:2201.12204*. 2022.
9. Zhang S, Liu K, Gu J, Cai X, Wang Z, Bu J, et al. Attention beats linear for fast implicit neural representation generation. In: European Conference on Computer Vision. Cham, Switzerland: Springer; 2024. p. 1–18. doi:10.1007/978-3-031-73668-1\_1.
10. Chen H, Xie S, Lim SN, Shrivastava A. Fast encoding and decoding for implicit video representation. In: European Conference on Computer Vision. Cham, Switzerland: Springer; 2024. p. 402–18. doi:10.1007/978-3-031-72933-1\_23.
11. Xu H, Zhang X, Wu X. Fast point cloud geometry compression with context-based residual coding and INR-based refinement. In: European Conference on Computer Vision. Cham, Switzerland: Springer; 2024. p. 270–88. doi:10.1007/978-3-031-73113-6\_16.
12. Rao L, Zhang X, Huang Y, Tan T, Tong T. Implicit neural representation-based MRI reconstruction method with sensitivity map constraints. *arXiv:2506.06043*. 2025.
13. Peng L, Wu A, Li W, Xia P, Dai X, Zhang X, et al. Pixel to gaussian: ultra-fast continuous super-resolution with 2D gaussian modeling. *arXiv:2503.06617*. 2025.
14. Chan CK, Cheng LM. Hiding data in images by simple LSB substitution. *Pattern Recognit*. 2004;37(3):469–74. doi:10.1016/j.patcog.2003.08.007.
15. Tamimi AA, Abdalla AM, Al-Allaf O. Hiding an image inside another image using variable-rate steganography. *Int J Adv Comput Sci Appl*. 2013;4(10):18–21. doi:10.14569/IJACSA.2013.041004.
16. Wu DC, Tsai WH. A steganographic method for images by pixel-value differencing. *Pattern Recognit Lett*. 2003;24(9–10):1613–26. doi:10.1016/S0167-8655(02)00402-6.
17. Chowdhuri P, Jana B, Giri D. Secured steganographic scheme for highly compressed color image using weighted matrix through DCT. *Int J Comput Appl*. 2021;43(1):38–49. doi:10.1080/1206212x.2018.1505024.
18. Westfeld A. F5—a steganographic algorithm: high capacity despite better steganalysis. In: International Workshop on Information Hiding; 2001. Berlin, Germany. Springer; 2001. p. 289–302. doi:10.1007/3-540-45496-9\_21.
19. Li L, Hou M, Chen K, Liu J. Survey on interpretability research of deep learning. *J Comput Appl*. 2022;42(12):3639–50. (In Chinese). doi:10.11772/j.issn.1001-9081.2021091649.
20. Xu Y, Mou C, Hu Y, Xie J, Zhang J. Robust invertible image steganography. In: Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2022 Jun 18–24; New Orleans, LA, USA. Piscataway, NJ, USA: IEEE; 2022. p. 7875–84. doi:10.1109/CVPR52688.2022.00772.
21. Kishore V, Chen X, Wang Y, Li B, Weinberger KQ. Fixed neural network steganography: train the images, not the network. In: Proceedings of the 10th International Conference on Learning Representations; 2022 Apr 20–29; Virtual.
22. Tahiri MA, Karmouni H, Sayyouri M, Qjidaa H, Ahmad M, Hammad M, et al. An improved reversible watermarking scheme using embedding optimization and quaternion moments. *Sci Rep*. 2024;14(1):18485. doi:10.1038/s41598-024-69511-3.

23. Han G, Lee DJ, Hur J, Choi J, Kim J. Deep cross-modal steganography using neural representations. In: Proceedings of the 2023 IEEE International Conference on Image Processing; 2023 Oct 8–11; Kuala Lumpur, Malaysia. p. 1205–9. doi:10.1109/ICIP49359.2023.10222113.
24. Yang S, Song S, Yoo CD, Kim J. Flexible cross-modal steganography via implicit representations. arXiv:2312.05496. 2023.
25. Dong W, Liu J, Chen L, Sun W, Pan X, Ke Y. StegaINR4MIH: steganography by implicit neural representation for multi-image hiding. J Electron Imaging. 2024;33(6):063017. doi:10.1117/1.JEI.33.6.063017.
26. Song Q, Luo Z, Huang X, Li S, Wan R. Unified steganography via implicit neural representation. arXiv:2312.05496. 2023.
27. Luo Z, Li S, Li G, Qian Z, Zhang X. Securing fixed neural network steganography. In: Proceedings of the 31st ACM International Conference on Multimedia; 2023 Oct 29–Nov 3; Ottawa, ON, Canada. New York, NY, USA: ACM; 2023. p. 7943–51. doi:10.1145/3581783.3611920.
28. Karras T, Aila T, Laine S, Lehtinen J. Progressive growing of GANs for improved quality, stability, and variation. arXiv:1710.10196. 2017.
29. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP. Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process. 2004;13(4):600–12. doi:10.1109/TIP.2003.819861.
30. Kim D, Shin C, Choi J, Jung D, Yoon S. Diffusion-Stego: training-free diffusion generative steganography via message projection. arXiv:2305.18726. 2023.
31. Peng Y, Hu D, Wang Y, Chen K, Pei G, Zhang W. StegaDDPM: generative image steganography based on denoising diffusion probabilistic model. In: Proceedings of the 31st ACM International Conference on Multimedia; 2023 Oct 29–Nov 3; Ottawa, ON, Canada. New York, NY, USA: ACM; 2023. p. 7143–51. doi:10.1145/3581783.3612514.
32. Zhang KA, Cuesta-Infante A, Xu L, Veeramachaneni K. SteganoGAN: high capacity image steganography with GANs. arXiv:1901.03892. 2019.
33. Dong W, Liu J, Sun W, Chen L, Pan X, Ke Y. Model steganography scheme based on implicit neural representation. Sci Technol Eng. 2024;24(25):10857–65. doi:10.12404/j.issn.1671-1815.2400315.
34. Zhong Y, Ke Y, Liu M, Liu J. Image steganography based on generative implicit neural representation. J Electron Imaging. 2024;33(6):063043. doi:10.1117/1.JEI.33.6.063043.
35. Zhong Y, Liu J, Luo P, Ke Y, Zhang M. Generative implicit steganography via message mapping. Appl Sci. 2025;15(20):11041. doi:10.3390/app152011041.
36. Fridrich J, Kodovsky J. Rich models for steganalysis of digital images. IEEE Trans Inf Forensics Secur. 2012;7(3):868–82. doi:10.1109/tifs.2012.2190402.
37. You W, Zhang H, Zhao X. A siamese CNN for image steganalysis. IEEE Trans Inf Forensics Secur. 2020;16:291–306. doi:10.1109/tifs.2020.3013204.
38. Xu G, Wu HZ, Shi YQ. Structural design of convolutional neural networks for steganalysis. IEEE Signal Process Lett. 2016;23(5):708–12. doi:10.1109/lsp.2016.2548421.
39. Boroumand M, Chen M, Fridrich J. Deep residual network for steganalysis of digital images. IEEE Trans Inf Forensics Secur. 2018;14(5):1181–93. doi:10.1109/tifs.2018.2871749.