



High-accuracy image steganography with invertible neural network and generative adversarial network

Ke Wang^{ID}, Yani Zhu^{ID*}, Qi Chang^{ID}, Junyu Wang, Ye Yao^{ID}

School of Cyberspace, Hangzhou Dianzi University, Hangzhou 310018, China

ARTICLE INFO

Keywords:

Image steganography
Invertible neural network
Generative adversarial network
Message extraction accuracy

ABSTRACT

Image steganography conceals secret messages imperceptibly within cover images. However, many existing deep learning-based image steganography methods have limitations in visual quality, payload size, and security. Specifically, they often require error correction codes for complete message extraction. In this paper, we propose a novel image steganography network architecture based on Invertible Neural Network (INN) and Generative Adversarial Network (GAN). Leveraging the reversibility of INN and the connection with the hidden network, we design three extraction networks based on DenseNet, shared weights, and unshared weights. These are respectively combined with the hidden network and discriminator network to create new network structures, effectively improving invisibility and message extraction accuracy. Furthermore, the discriminator participates in adversarial training by comparing cover and stego images in a patch-to-patch manner, thereby enhancing visual quality and security. Extensive experiments demonstrate the effectiveness of our proposed method across various aspects, including image quality, payload, extraction accuracy, and security, particularly achieving close to 100% message extraction accuracy without requiring error correction codes.

1. Introduction

In recent years, due to rapid advancements in science and technology, the internet has become increasingly integral to people's lives. While the internet's openness encourages users to readily share personal information, data transmitted over the network is vulnerable to interception, alteration, or theft by hackers, resulting in an increasing number of privacy breaches. To ensure the security of data transmission, various techniques have been proposed. Among them, data hiding techniques [1,2] are recognized as an effective method, capable of embedding information into carriers such as text, images, or audio to achieve covert communication. Due to the unique attributes of images, steganography methods have received significant attention [3]. They can protect privacy, ensure information security, facilitate secret communication and identity authentication when necessary. As the capacity of hidden data within images increases, the images' imperceptibility and security decrease. Many researchers aim to develop high-performance methods that allow steganographically altered images to retain the original content's characteristics to the fullest extent, resisting steganalysis and improving message extraction accuracy.

Image steganography methods are typically classified into two categories: traditional steganography [4–8] and deep learning-based steganography [9–21]. Traditional image steganography techniques

manipulate image pixels or frequency coefficients with fine adjustments to hide secret messages within the cover image, achieving high imperceptibility. However, these traditional techniques are limited by the trade-off between embeddable data and security. With the continuous evolution of deep learning, researchers increasingly favor deep learning-based steganography methods. Unlike traditional methods, which rely on manual feature extraction, deep learning enables automatic learning of image features through neural networks, enhancing the accuracy and efficiency of steganographic messages extraction. Steganography embedding patterns and rules learned from large datasets confer greater robustness to the generated models, enabling them to withstand various image processing operations and attacks such as compression, rotation, and cropping. Despite the many advantages of deep learning-based image steganography, current state-of-the-art methods still have limitations. On one hand, the imperceptibility of stego images produced by these methods is insufficient, and some approaches may lead to visually perceptible distortions. On the other hand, due to inevitable errors in neural networks, the accuracy of message extraction needs improvement, necessitating the introduction of more error correction mechanisms to ensure effective data transmission. However, lengthy error-correcting codes result in a

* Corresponding author.

E-mail addresses: 222270063@hdu.edu.cn (K. Wang), zyn@hdu.edu.cn (Y. Zhu), qichang@hdu.edu.cn (Q. Chang), 211270020@hdu.edu.cn (J. Wang), yaoye@hdu.edu.cn (Y. Yao).

<https://doi.org/10.1016/j.sigpro.2025.109988>

Received 19 May 2024; Received in revised form 19 January 2025; Accepted 2 March 2025

Available online 10 March 2025

0165-1684/© 2025 Elsevier B.V. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

lower effective payload, requiring multiple transmissions to convey all secret messages. This increases the risk of exposure during transmission and consumes more communication bandwidth for transmitting secret information.

To address the aforementioned issues, we propose a high-accuracy image steganography method based on INN and GAN. In comparison with other deep learning-based schemes, our image steganography approach utilizes the characteristic of INN employed as the hidden network to maintain information integrity, and is achieved through studying the relationship between the hidden and extraction networks. Simultaneously, based on the inverse process of INN, three different extraction networks are respectively designed based on DenseNet, shared weights, and unshared weights. Each type of extraction network is combined with the hidden network to form a new structure, exploiting unique advantages and performance characteristics. Furthermore, to enhance the invisibility of the generated stego images, a discriminator network is incorporated into each network structure for adversarial training. The discriminator can distinguish between cover and stego images, providing additional gradient information to improve the effectiveness of hidden and extraction networks. Experimental results demonstrate that our proposed image steganography method exhibits high invisibility and generalization performance. It also achieves extraction accuracy close to 100% for all three network structures. In practical applications, perfect messages extraction can be achieved with minimal or even no error correction codes.

The main contributions of this paper are summarized as follows:

(1) We propose a novel network architecture that combines the advantages of INN and GAN. INN preserves input information, ensuring robustness to small perturbations in the input, while GAN enhances the visual quality of the stego images generated by the network through adversarial training, thereby improving the invisibility and security of the method.

(2) Leveraging the relationship with the hidden network, we propose three different extraction networks based on DenseNet, shared weights, and unshared weights. These networks are integrated with the hidden network and the discriminator network to create new network structures, which are trained end-to-end.

(3) Extensive experiments on various image datasets demonstrate the superiority of the proposed method over other GAN-based algorithms in terms of invisibility, security, and reversibility. Specifically, all three network structures achieve close to 100% accuracy in extracting hidden messages. Furthermore, our method maintains stability when encountering different types of steganalysis.

The remainder of this paper is organized as follows. Section 2 reviews related works on image steganography categorized by different deep learning methods. The architecture and details of our method, as well as the loss functions used for training, are presented in Section 3. Subsequently, the experimental results validating the performance of the proposed method are exhibited in Section 4. Finally, Section 5 concludes this paper.

2. Related works

Presently, deep learning-based image steganography methods can be primarily classified into three types: methods based on modified matrix [10–13], methods based on adversarial cover generation [14–16], and methods based on generative adversarial network [17–20]. In this section, we systematically review relevant works corresponding to different categories and discuss the strengths and weaknesses of each method.

2.1. Modified matrix-based methods

Image steganography methods based on modified matrix combine deep learning with traditional adaptive steganography. The modified matrix dimensions match those of the cover image, with each value representing the impact of a corresponding pixel modification on image quality and security. This facilitates embedding information into image regions causing minimal impact and considered more secure. Unlike traditional adaptive image steganography methods, which manually define cost functions to calculate the modified matrix, deep learning-based methods use neural networks to produce the matrix through training. Adversarial training results in the modified matrix generated by deep neural networks exhibiting increased robustness. After calculating the modified matrix, syndrome-trellis codes (STC) encoding [22] are used to embed information into the cover image. Tang et al. first proposed a method called ASDL-GAN [10] in 2017, which employs the adversarial concept of GAN to produce a modified matrix such that the information becomes difficult to detect after being embedded in the image. Since the training process of neural networks requires all operations to be differentiable to ensure that gradients can propagate through every layer of the network, Tang et al. trained a ternary embedding simulator (TES) sub-network to approximate the embedding process. Building upon ASDL-GAN, Yang et al. proposed the UT-GAN method [11], which replaces the TES network with two tanh activation functions, thereby eliminating the time required for pre-training the TES network. In addition, they employed an U-Net [23] network to generate the modified matrix, achieving better performance and shorter training times. More recently, Huang et al. introduced StegGMAN [13], which uses multiple discriminators to provide more gradient information to the generator, thus enhancing security. Due to the use of STC coding for embedding information, modified matrix-based methods exhibit good imperceptibility. However, they have limited embedding capacity, with most methods only being able to embed data below 0.4 bits per pixel (bpp), hence they are not suitable for scenarios that require hiding large amounts of data.

2.2. Adversarial cover generation-based methods

Adversarial cover generation methods produce adversarial noise and add it to the original cover to create an embeddable adversarial cover. These methods usually involve two networks trained using GAN: an adversarial cover generator and a stego image discriminator. The adversarial cover generator generates the adversarial cover, while the discriminator distinguishes whether the image contains hidden secret information. By embedding information on an adversarial cover, stego images become more difficult for steganalyzers to detect. In [14], Zhou et al. used a fully convolutional network (FCN) [24] to generate adversarial noise, creating an adversarial cover. They employed the steganalysis approach YeNet [25] as a discriminator to distinguish between cover and stego images. In contrast, Yang et al. [15] employed Siamese [26] architecture as a generator for producing adversarial covers. This ensures that the adversarial cover and the original cover share similar noise residual characteristics, thereby enhancing the approach's security. Additionally, Alejandro et al. [16] used a genetic algorithm to select different adversarial cover generator network structures based on various data hiding methods. Adversarial cover generation-based methods also have drawbacks. Firstly, the adversarial noise added to the original cover causes distortion, resulting in a greater visual discrepancy between the stego and original cover images. Secondly, the security of these methods largely depends on the steganalysis [27] capabilities of the discriminator, requiring the preprocessing step of training.

2.3. Generative adversarial network-based methods

Image steganography methods based on GAN utilize deep neural networks for direct embedding and extraction of information. They consist of an hidden network, an extraction network, and a discrimination network. The hidden network inserts information into the cover image, the extraction network retrieves hidden information from the stego image, and the discrimination network enhances the quality of the generated stego images. Compared to other methods, this category maintains high security while accommodating a greater capacity for information hiding. In 2018, Zhu et al. introduced the HiDeN methodology [17], which employs an autoencoder structure to embed and extract information. The hidden network takes the cover image and the secret information as input, producing a stage image as output. The extraction network takes the stage image as input and outputs the embedded secret information. The discriminator takes both the stage and the cover images as input and produces a binary image classification result. However, HiDeN replicates the secret information multiple times to form a secret message block for network input, resulting in difficulty accommodating large information hiding capacities. To address this issue, Zhang et al. proposed the SteganoGAN method [18] in 2019. SteganoGAN inputs the secret information into each convolution layer within the hidden network, thus possessing a significant embedding capacity. Furthermore, it enhances imperceptibility by implementing skip connections in both the hidden and extraction networks. Building on SteganoGAN, Tan et al. introduced the CHAT-GAN method [19], which incorporates a channel attention mechanism and employs XuNet [28] as the discriminator to further enhance security. Unlike the aforementioned methods, which rely solely on adversarial interactions between hidden and discriminating networks to improve security, Yuan et al. proposed the ADF-IS method [20]. This approach generates adversarial noise through the adversarial network UAN [29] and adds it to the stego image to further enhance security levels. Due to limitations inherent in neural networks, methods based on GAN may embed some information in parts of the cover image that are unsuitable for modification. This compromises the invisibility and security of the stego image, making them unsuitable for applications requiring high-quality image standards.

Due to the flexibility of neural network structures, deep learning-based image steganography algorithms often achieve higher embedding capacity, and the use of GANs can further enhance their security. Currently, many GAN-based image steganography algorithms, e.g., HiDeN [17], SteganoGAN [18], and CHAT-GAN [19], only employ single-scale convolution in network construction, making it difficult to learn features of varying sizes, which in turn limits the ability to select optimal embedding locations. Additionally, these methods typically add embedding perturbations directly to the original carrier, overlooking the fact that different pixels have varying impacts on image quality. This leads to greater distortion between the steganographic and cover images. Moreover, the inevitable errors in neural networks reduce the accuracy of information extraction. To address these issues, our paper introduces INNs as both the hidden and extraction networks in GANs. INNs not only possess bidirectional mapping capabilities but also ensure precise mapping of input data, minimizing the impact on image quality during information embedding. The proposed method dynamically combines INNs based on the relationship between the hidden and extraction networks to achieve optimal performance.

3. Proposed method

In this section, we propose a novel end-to-end network architecture that incorporates the advantages of INN and GAN. The entire GAN structure is composed of three main modules: the hidden network, the extraction network, and the discriminator. The hidden network is responsible for embedding secret information into the image, the extraction network is used to retrieve the hidden information from the

Table 1

The main notations used in this paper.

Parameter	Definition
c	The number of channels in the image.
h, w	The height and width of the image.
z	A parameter for adjusting the amount of messages.
n	The number of invertible blocks.
$DWT(\cdot)$	The DWT process.
$IWT(\cdot)$	The inverse DWT process.
X	The cover image.
X_D	Image X in DWT domain.
S	The stego image.
S_D	Stego image S in DWT domain.
M	The reshaped messages.
M'	The extracted messages.

stego image, and the discriminator distinguishes between stego images and original images to improve the effectiveness and concealment of the steganography. Sections 3.1 and 3.2 respectively introduce the specific network structure designs for the hidden and extraction stages in image steganography, including the hidden network, sub-networks, discriminator network, and three different extraction networks. The loss functions during training of network models will be described in Section 3.3. The main notations used in this paper and their definition are listed in Table 1.

3.1. Hidden stage

The overall architecture of the hidden stage is illustrated in Fig. 1. Firstly, the cover image X undergoes preprocessing to transform from spatial domain to DWT domain image X_D . Subsequently, the image X_D and the secret messages M are jointly input into the hidden network, which comprises numerous invertible blocks, to obtain the DWT stego image S_D . Finally, the image S_D is post-processed to become the spatial stego image S , which is input into the discriminator network along with the original cover image X . Through adversarial training, the stego image generated by the hidden network becomes more similar to the cover image and harder to distinguish. Detailed explanations of different parts of the hidden stage will be provided following.

3.1.1. Image preprocessing and postprocessing

In our approach, the input to the hidden network is the preprocessed image in the DWT domain, which offers excellent multi-scale and local feature representation capabilities. The preprocessing method employs the Haar wavelet transform, which decomposes the pixel values of the image into different frequency subbands, facilitating more efficient information embedding during the steganography process. The Haar wavelet transform decomposes the image into four sub-bands: LL, LH, HL, and HH, each representing different frequency information of the image. These subbands are then concatenated along the channel dimension for further processing. Specifically, for a cover image X and a stego image S both with dimensions $c \times h \times w$, after DWT transformation, the size of the subbands from images X and S becomes $c \times h/2 \times w/2$. The DWT domain cover image X_D and stego image S_D obtained by concatenating these subbands have dimensions $4c \times h/2 \times w/2$, where h and w represent the height and width of the image, and c denotes the number of channels. When the cover image is an RGB image, $c = 3$.

The postprocessing is the inverse process of preprocessing. For a DWT domain stego image S_D with dimensions $4c \times h/2 \times w/2$, it is initially split into multiple subbands LL, LH, HL, and HH according to channels. Next, these subbands are arranged in the spatial order based on the preprocessing. Finally, the spatial domain stego image S with dimensions $c \times h \times w$ is obtained through IWT transformation.

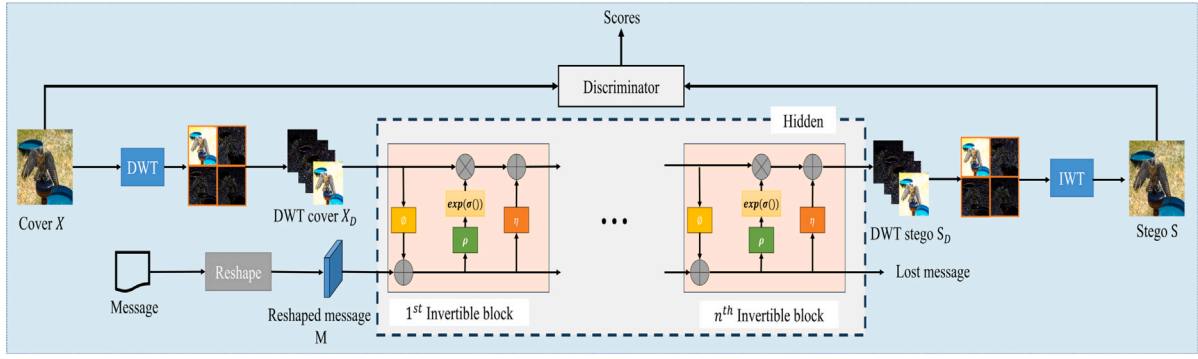


Fig. 1. The overall architecture of the hidden stage, where ϕ , ρ , and η are three sub-networks.

3.1.2. Hidden network

The hidden network consists of n invertible blocks, each comprising three sub-networks denoted as ϕ , ρ , and η . Each invertible block is computationally reversible and arranged in a serial manner, where the output of each block serves as the input to the next one, ensuring that the entire hidden network is invertible. The hidden network contains cover and message branches from top to bottom. Assuming C_i and C_{i+1} are the input and output of the cover branch, and M_i and M_{i+1} are the input and output of the message branch in the i th invertible block, the entire process can be described using Eq. (1):

$$\begin{aligned} C_{i+1} &= C_i \odot \exp(\sigma(\rho(M_i))) + \eta(M_i) \\ M_{i+1} &= M_i + \phi(C_i) \end{aligned} \quad (1)$$

where $\odot$ represents the Hadamard product and $\sigma(\cdot)$ is the activation function. In the first invertible block, C_1 represents the DWT domain cover image X_D , and M_1 represents the secret messages M . The output C_{n+1} of the last invertible block is the DWT domain stego image S_D . From the equation, it can be seen that in each invertible block, the cover branch C_i and the message branch M_i are sent into separate sub-networks for feature extraction, followed by interrelated calculations between the two. As the image data and secret data pass through multiple invertible blocks, their features are progressively processed and fused. Each invertible block ensures that the information is transmitted losslessly, allowing the secret data to be embedded into the image while maintaining the original image's visual quality without significant degradation. Ultimately, after the fusion process through n invertible blocks, the output is a stego image that, on the surface, appears identical to the original cover image. The number of invertible blocks n is influenced by various factors, including task complexity, image resolution, security requirements, training difficulty, and hardware resources. If the embedded and extracted information is large or has a complex structure, more invertible blocks are needed to capture image features and ensure precise information embedding or extraction. Image resolution also impacts the number of invertible blocks; high-resolution images typically contain more details, requiring additional blocks to process fine details and complete embedding and extraction tasks. Steganography's key factors (e.g., concealment and security) also dictate the number of blocks. To achieve higher concealment and security, the number of invertible blocks must increase to enhance the network's capacity. However, more invertible blocks lead to a more complex model architecture, which not only raises training difficulty but also increases computational demands on hardware resources. Therefore, our method must balance these factors when choosing the appropriate number of invertible blocks to ensure both accuracy and efficiency in the embedding and extraction processes.

Each sub-network of the invertible block utilizes the DenseNet structure to mitigate the problem of vanishing gradients and make full use of features at different levels. The specific structure of the three sub-networks is illustrated in Fig. 2, where 'Conv' represents convolution, 'ReLU' denotes the activation function, and 'C' indicates

Table 2

The input and output channel numbers of the convolutional layers in the invertible blocks, discriminator network, and extraction network.

Network	j th conv	The number of	
		Input channels	Output channels
ϕ	1	12	32
	2	12 + 32	32
	3	12 + 32 × 2	32
	4	12 + 32 × 3	32
	5	12 + 32 × 4	z
ρ, η	1	z	32
	2	$z + 32$	32
	3	$z + 2 \times 32$	32
	4	$z + 3 \times 32$	32
	5	$z + 4 \times 32$	12
Discriminator	1	3	32
	2	32	32
	3	32	32
	4	32	1
Extraction	1	12	32
	2	12 + 32	32
	3	12 + 32 × 2	32
	4	12 + 32 × 3	z

concatenation along the channel dimension. It can be observed from the figure that the input and feature maps of each layer are concatenated as part of the input to each subsequent convolution via skip connections. Different feature maps and their corresponding skip connections are distinguished by different colors. To seamlessly integrate secret messages with feature maps, the secret messages are adjusted to $M \in \{0, 1\}^{z \times h/2 \times w/2}$, where $h/2$ and $w/2$ represent the height and width of the input image X_D , respectively. Additionally, the parameter z modulates the quantity of embedded messages, with a larger z value indicating a greater volume of embedded data. The entire sub-network can be represented using the following equation:

$$x_{j+1} = \begin{cases} f^{(1)}(X_D) & \text{if } j = 1 \\ f^{(2)}([x_2; X_D]) & \text{if } j = 2 \\ f^{(3)}([x_2; x_3; X_D]) & \text{if } j = 3 \\ f^{(4)}([x_2; x_3; x_4; X_D]) & \text{if } j = 4 \\ f^{(5)}([x_2; x_3; x_4; x_5; X_D]) & \text{if } j = 5 \end{cases} \quad (2)$$

where $[\cdot]$ represents concatenation, $f^{(j)}$ denotes the j th convolution and x_{j+1} represents the output of $f^{(j)}$. In all three sub-networks ϕ , ρ , and η , the convolutional kernels used for feature extraction are set to 3×3 . The input and output channel numbers of different convolutions are also not the same, detailed parameters are recorded in Table 2.

3.1.3. Discriminator network

The discriminator primarily evaluates whether an image contains hidden data. Through adversarial training against the generator, the

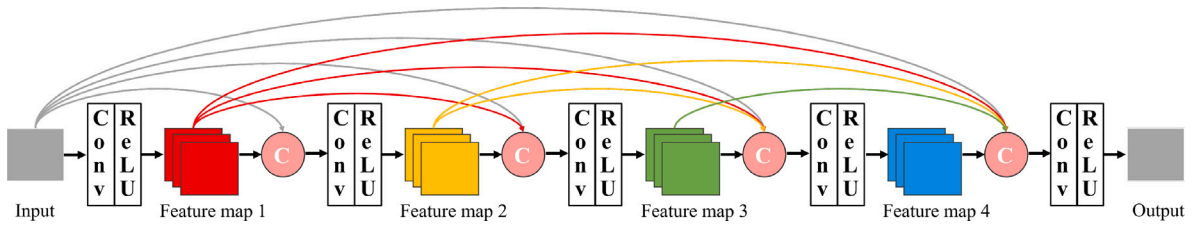


Fig. 2. The subnetwork structure of invertible blocks.

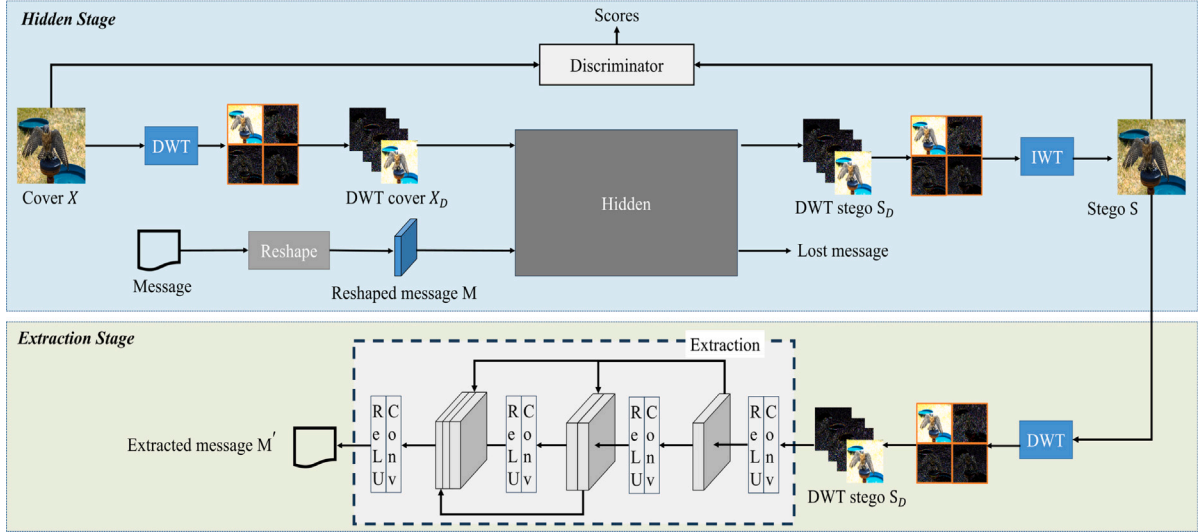


Fig. 3. Overall structure diagram of INN-GAN(D) using DenseNet as the extraction network.

discriminator pushes the generated image closer to resembling the cover, thus helping the generator achieve superior outcomes. In previous approaches, both SteganoGAN [18] and CHAT-GAN [19] employ a discriminator trained with WGAN loss [30] to produce scores for input images. While WGAN loss improves stability compared to traditional GAN loss, its effectiveness in terms of visual quality and security still requires improvement.

For further enhancement, our discriminator network comprises multiple convolutional layers, each consisting of multiple 3×3 convolutional kernels with a stride of 1. The discriminator we use no longer focuses solely on the global features of the entire image but instead independently evaluates local regions within the image. This approach allows for better detection of subtle changes caused by the embedding of secret data, especially in edge-rich or texture-dense areas. To further improve the stability and training performance of the discriminator network, we have added spectral normalization (SN) after each convolution layer, which helps generate more realistic and inconspicuous steganographic images. Additionally, we have incorporated spatial rich model (SRM) filtering into the discriminator [19], further enhancing its ability to detect stego images effectively. In Fig. 1, the “scores” represent the probability values output by the discriminator network when analyzing different blocks of the input image. Each probability value indicates whether the current image block is an unaltered original block (i.e., whether it does not contain any hidden information). If the discriminator network determines that an image block has not undergone steganographic processing, it outputs a high probability value (close to 1). Conversely, if the network believes that an image block has been subjected to steganographic processing, it outputs a low probability value (close to 0). The ultimate goal of the discriminator network is to accurately distinguish between normal image blocks and those containing hidden information through continuous learning, thereby enhancing the concealment and security of the steganographic method. Mean squared error (MSE) is utilized as the loss function

for adversarial training of the discriminator network, with detailed parameters shown in Table 2.

3.2. Extraction stage

The extraction stage is responsible for extracting the secret messages hidden in the stego image. Our proposed method employs blind extraction, meaning data extraction is based solely on the stego image, without knowledge of the original cover image. Thus, the input to the extraction stage is the stego image, and the output is the secret messages. In this paper, we design three different extraction network structures: one based on DenseNet, and the others based on shared weights and unshared weights. These extraction networks are trained end-to-end together with the hidden network and the discriminator network, forming combined network structures denoted as INN-GAN(D), INN-GAN(S), and INN-GAN(US), respectively. Below, we will introduce the different extraction networks in the three proposed network structures.

3.2.1. INN-GAN(D)

The overall structure of INN-GAN(D) with the DenseNet-based extraction network is illustrated in Fig. 3, where the details of the hidden network in the hidden stage are consistent with the description in Section 3.1.2. As secret messages are hidden in the DWT coefficients, the stego image needs to be converted from the spatial domain to the DWT domain before inputting it into the extraction network. Each convolutional kernel in the extraction network has a size of 3×3 with a stride of 1, ensuring consistency in input and output sizes for each convolution. A ReLU activation function is added after each convolution to enhance the extraction network’s fitting capability. Through skip connections, the shallow convolutional features are fed into the deep convolutional layers, significantly improving feature utilization efficiency. The detailed parameters of the extraction network are also listed in Table 2.

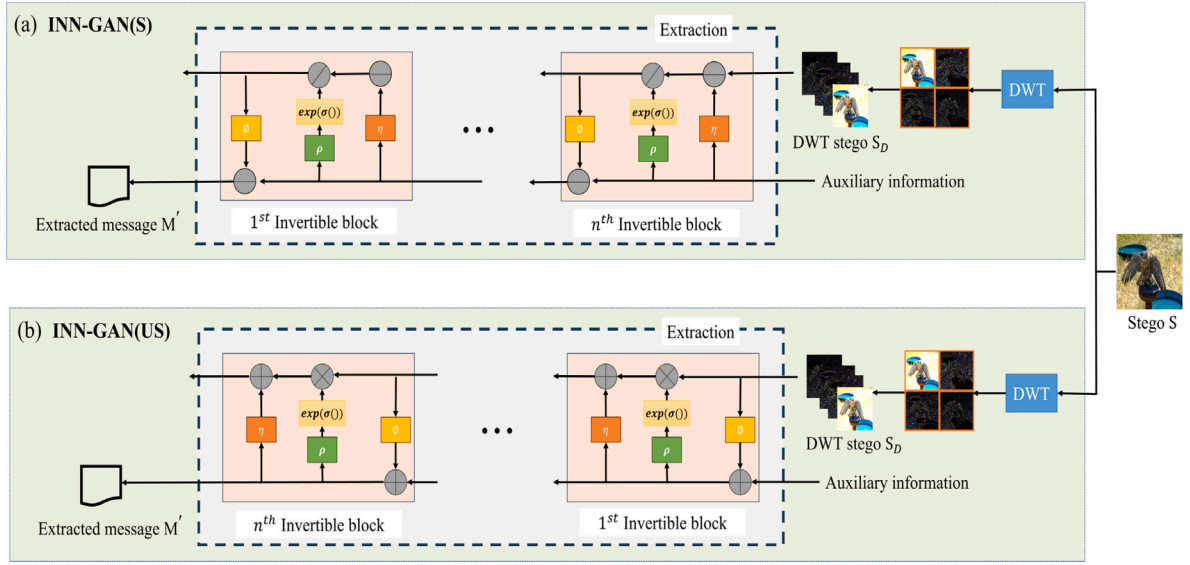


Fig. 4. The architecture of the extraction stages in the INN-GAN(S) and INN-GAN(US): (a) INN-GAN(S); (b) INN-GAN(US).

3.2.2. INN-GAN(S)

The architecture of the extraction stage in INN-GAN(S), featuring the shared weights-based extraction network, is depicted in Fig. 4(a). As can be seen, the extraction stage is the inverse process of the hidden stage shown in Fig. 1. INN-GAN(S) utilizes the reversibility of the INN, enabling sharing of parameters between the hidden and extraction networks. Specifically, the forward and backward processes of the INN are employed for data embedding and extraction, respectively. Like the hidden network, the extraction network also has two branches: the stego branch and the message branch. In the i th invertible block, assuming S_{i+1} and S_i are the input and output of the stego branch, and A_{i+1} and A_i are the input and output of the message branch. The entire extraction process can be represented by the following equation:

$$\begin{aligned} S_i &= (S_{i+1} - \eta(A_{i+1})) \odot \exp(-\sigma(\rho(A_{i+1}))) \\ A_i &= A_{i+1} - \phi(S_{i+1}) \end{aligned} \quad (3)$$

where S_{n+1} is the DWT domain stego image S_D , and A_1 is the extracted secret messages M' . From the equation and the structural diagram, it can be seen that the INN flows from 1st invertible block to n th invertible block in the hidden stage. In contrast, the shared-weights INN flows backwards from n th invertible block to 1st invertible block in the extraction stage. The proposed method uses the DWT domain stego image as the auxiliary information input to the extraction network. Since blind extraction only has access to the stego image, whereas the INN requires two inputs, this is done to enhance the extraction network's ability to recover the secret messages with the aid of auxiliary information.

3.2.3. INN-GAN(US)

As shown in Fig. 4(b), the architecture of the extraction stage in INN-GAN(US) differs from Fig. 4(a) in that it employs two separate INN for the information hiding and extraction processes, rather than using the same network's forward and backward processes. Similar to INN-GAN(S), INN-GAN(US) also comprises a stego branch and a message branch. In the i th invertible block, assuming S_i and S_{i+1} are the input and output of the stego branch, and A_i and A_{i+1} are the input and output of the message branch. The overall process can be described by Eq. (4):

$$\begin{aligned} S_{i+1} &= S_i \odot \exp(\sigma(\rho(A_i))) + \eta(A_i) \\ A_{i+1} &= A_i + \phi(S_i) \end{aligned} \quad (4)$$

where S_1 represents the DWT domain stego image S_D , and A_{n+1} is the extracted secret messages M' . The construction of the auxiliary information is consistent with that of INN-GAN(S).

3.3. Loss functions

Image steganography entails concealing secret messages within cover images. To enhance the algorithm's invisibility, we aim for the stego images to closely resemble the cover images. Since our proposed method involves both spatial domain and DWT domain images, we design spatial loss functions and frequency loss functions to minimize differences between cover and stego images. Additionally, accuracy loss functions are necessary for improved secret messages extraction. As GAN is employed for training, both adversarial loss and training discriminator network loss are necessary. The details of the different loss functions are as follows.

3.3.1. Pixel-wise loss

Pixel-wise loss measures the difference at the pixel level between the cover image X and the stego image S . Minimizing pixel-wise loss aids in generating stego images that closely resemble the cover image. The definition of pixel-wise loss L_P is as follows:

$$L_P = MSE(X, S) \quad (5)$$

where $MSE(X, S)$ represents the mean squared error between the two compared images X and S .

3.3.2. Frequency coefficient loss

Frequency coefficient loss calculates the difference between the frequency domain coefficients of the cover image and the stego image. When calculating the frequency domain coefficient loss, focusing on the LL sub-band minimizes modifications to the core information of the image, ensuring that the embedded secret information has minimal effect on the overall appearance of the image. Calculating the loss across the entire frequency domain would require processing all frequency bands, increasing computational complexity and training time. Since the LL sub-band carries most of the image's energy and information, computing the loss in this band alone is sufficient to maintain the perceptual quality of the image and the concealment of the steganographic information. The frequency coefficient loss L_F is defined as follows:

$$L_F = MSE(X_{D_{LL}}, S_{D_{LL}}) \quad (6)$$

where $X_{D_{LL}}$ and $S_{D_{LL}}$ represent the LL subbands of the DWT domain cover image X_D and the DWT domain stego image S_D , respectively.

3.3.3. Information extraction accuracy loss

Information extraction accuracy loss measures the difference between the embedded secret messages M and the extracted secret messages M' . Minimizing this loss enhances the proposed method's accuracy in extracting messages from stego images. The definition of information extraction accuracy loss L_E is as follows:

$$L_E = -M \log_2 M' - (1 - M) \log_2 (1 - M') \quad (7)$$

3.3.4. Adversarial loss

The purpose of the adversarial loss is to deceive the discriminator network into classifying the stego image as the cover image. This encourages the algorithm to generate stego images that can effectively fool the discriminator network. When the discriminator network is unable to distinguish between the cover image and the stego image, it indicates a high similarity between the two. The adversarial loss L_A is defined as follows:

$$L_A = \frac{1}{[h/t_1] \times [w/t_2]} \sum_{k_1=1}^{[h/t_1]} \sum_{k_2=1}^{[w/t_2]} (D(S)(k_1, k_2) - a(k_1, k_2))^2 \quad (8)$$

where $a \in \{1\}^{h \times w}$ is the label of the entire cover image, $D(S)(k_1, k_2)$ represents the output of the discriminator network for a $t_1 \times t_2$ stego image block, and $a(k_1, k_2)$ represents the label of a $t_1 \times t_2$ cover image block. The adversarial loss aims to make the output value of the discriminator network for the stego image as close as possible to the label of the cover image.

3.3.5. Total loss

The entire algorithm involves training the hidden network, the discriminator network, and the extraction network. Specifically, the hidden network and the extraction network use the same optimizer for training and the overall loss function L_T can be expressed as follows:

$$L_T = \lambda_P L_P + \lambda_F L_F + \lambda_E L_E + \lambda_A L_A \quad (9)$$

where λ_P , λ_F , λ_E , and λ_A respectively represent the weights of L_P , L_F , L_E , and L_A in the overall loss function L_T .

The discriminator network is trained with a separate loss function and is not involved in the embedding and extraction of messages. Its purpose is solely to provide gradient signals during training, enabling the algorithm to generate more realistic stego images. Once the model training is complete, the discriminator network can be discarded without affecting the algorithm's performance. The loss function L_D of the discriminator network is defined as follows:

$$L_D = \frac{1}{[h/t_1] \times [w/t_2]} \sum_{k_1=1}^{[h/t_1]} \sum_{k_2=1}^{[w/t_2]} [(D(X)(k_1, k_2) - a(k_1, k_2))^2 + (D(S)(k_1, k_2) - b(k_1, k_2))^2] \quad (10)$$

where $a \in \{1\}^{h \times w}$ and $b \in \{0\}^{h \times w}$ represent the labels of the entire cover image and the stego image, respectively.

4. Experiment results and analysis

In this section, the performance of the three proposed network structures, INN-GAN(D), INN-GAN(S), and INN-GAN(US), will be evaluated by comparing them with some state-of-the-art methods. The experimental settings including databases, model parameters, comparison methods, and evaluation metrics, are listed in Section 4.1. Section 4.2 analyzes the embedding strategy of the proposed method in the frequency domain. Then, image payload and quality analysis are discussed in Sections 4.3 and 4.4, respectively. Finally, Section 4.5 verifies the security performance through various steganalysis.

4.1. Experimental settings

4.1.1. Datasets and parameter settings

The experiments were conducted using the PyTorch deep learning framework, and model training was performed on GTX3090. The training set comprises 20,000 randomly selected images from the COCO dataset [31], while the test set includes 3000 images-1000 from the COCO dataset [31], 1000 from the VOC dataset [32], and 1000 from the ImageNet dataset [33]. None of the test set images were included in the training set, and both sets' images were resized to 256×256 . In this experiment, the number of invertible blocks n in the extraction network is set to 16, and the model is trained with a fixed steganographic embedding rate of 1 bit per pixel (bpp) throughout. Sub-networks ϕ , ρ , and η in each invertible block are constructed by 5-layer densely connected convolutional block with leaky ReLU activation. During training, the Adam optimizer is employed to update the model parameters, with an initial learning rate of $1e-4$. The batch size is set to 20 and the total number of iterations is 200K. Besides, the weights λ_P , λ_F , λ_E , and λ_A in the loss function L_T are set to 1, 1, 1, and 0.1, respectively.

4.1.2. Comparison methods and evaluation metrics

The proposed method will be compared with several other GAN-based image steganography methods, including ADV-EMB [9], SteganoGAN [18], and CHAT-GAN [19], through comparative experiments. These methods will use the same training and test datasets, and the hyperparameters used in their model training will be consistent with the corresponding papers.

For the evaluation of objective visual quality, PSNR, SSIM, and LP will be used as evaluation metrics to assess the invisibility of the stego images generated by the proposed method. PSNR primarily compares pixel differences between two images. Its calculation formula is as follows:

$$PSNR(X, S) = 10 \times \log_{10} \left(\frac{MAX^2}{MSE(X, S)} \right) \quad (11)$$

where MAX represents the maximum possible pixel value, which is 255 for RGB images.

Unlike PSNR, SSIM compares brightness, contrast, and structure between two images, which better corresponds to human perception. For images X and S , where μ_X and μ_S represent the mean values of pixels, σ_X and σ_S denote the variances of pixels, and σ_{XS} represents the covariance between the two images, the SSIM calculation formula is as follows:

$$SSIM(X, S) = \frac{(2\mu_X\mu_S + c_1)(2\sigma_{XS} + c_2)}{(\mu_X^2 + \mu_S^2 + c_1)(\sigma_X^2 + \sigma_S^2 + c_2)} \quad (12)$$

where c_1 and c_2 are constants to prevent division by zero.

LP is a recent evaluation metric proposed with the advancements in deep learning. It compares differences in feature maps of two images by feeding them into a pre-trained neural network. The calculation formula is as follows:

$$LP(X, S) = MSE(F(X), F(S)) \quad (13)$$

where F indicates the neural network, commonly used VGG [34].

Message extraction accuracy (ACC) is defined as the proportion of correctly extracted bits to the total embedded bits by a steganography algorithm. It is formulated as follows:

$$ACC = \frac{N_{\text{correct}}}{N_{\text{total}}} \quad (14)$$

where N_{correct} is the number of correctly extracted bits, and N_{total} is the total embedded bits.

To evaluate the algorithm's payload effectiveness, we utilize the evaluation metric RS-BPP, which represents the proportion of effective information, excluding the error correction code, to the total information when perfect extraction is ensured by adding Reed-Solomon (RS) error correction codes. The formula for RS-BPP is as follows:

$$RS-BPP = 1 - 2p \quad (15)$$

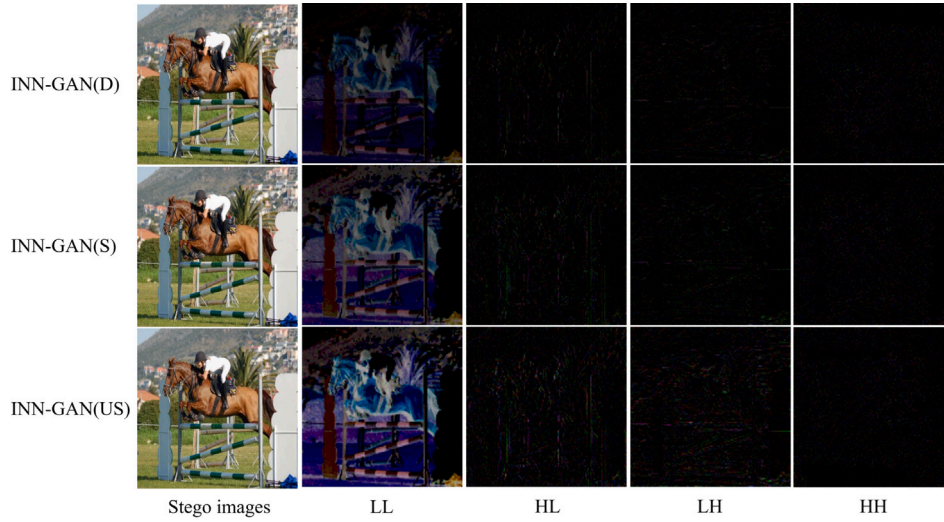


Fig. 5. Differences between the cover image X and the stego image S in different frequency bands at 1 bpp embedding rate. Each pair of columns corresponds to the stego images S and the residual signal of $10|X - S|$ in different frequency bands. From top to bottom are the proposed network structures INN-GAN(D), INN-GAN(S), and INN-GAN(US).

Table 3

The ACC of setting different subbands to zero (INN-GAN(S)).

The subband	LL	HL	LH	HH
ACC	0.634	0.767	0.783	0.714

where $p = 1 - ACC$ represents the probability of a bit being incorrectly extracted.

4.2. Frequency analysis

In this section, we analyze the embedding strategy of the proposed method in the DWT domain of images to determine where the hidden network embeds messages. In the experiment, INN-GAN(S) serves as the testing network structure. After embedding the secret messages into the cover image to generate stego images, the LL, LH, HL, and HH subbands of the stego images are respectively set to zero, and then inputted into the extraction network again to observe the impact of different subband losses on ACC. The experimental results are recorded in Table 3. From the table, it can be observed that setting HH to zero results in a lower ACC compared to HL and LH, indicating a preference for embedding messages in the high-frequency region. When LL is set to zero, the ACC of the method is the lowest, as the LL subband contains most of the information in the image, and setting it to zero causes most of the content in the entire image to be lost, resulting in lower ACC. Similar ACC values obtained when setting HL and LH sub-bands to 0 suggest a similar probability of message embedding in these two bands. To provide a more intuitive representation of the distribution of embedded information across different frequency bands, Fig. 5 illustrates the differences between the cover image and the stego image across various frequency bands at an embedding rate of 1 bpp. To make the observations clearer, the original differences have been amplified by a factor of 10. As shown in the figure, the LL frequency band contains a significant amount of additional information. In contrast, the HL and LH frequency bands carry much less information, while the HH frequency band carries the least. This phenomenon aligns with the experimental data in Table 3 and supports the summarized embedding characteristics of the frequency domain.

4.3. Payload analysis

When hiding messages in images, it is often necessary to use error correction codes to ensure the accurate extraction of hidden information from stego images. Effective payload refers to the number of bits

Table 4

RS-BPP of the comparison methods and the proposed method on different datasets when the size of the embedded information is fixed.

Method name	COCO	VOC	ImageNet
ADV-EMB [9]	0.924	0.932	0.930
SteganoGAN [18]	0.986	0.992	0.994
CHAT-GAN [19]	0.986	0.988	0.998
INN-GAN	≈ 1	≈ 1	≈ 1

in the hiding process excluding error correction codes, indicating how many bits of secret messages can be transmitted in a single embedding operation, showcasing the efficiency of the steganography algorithm. In this section, we use RS-BPP to evaluate the payload when the size of the embedded information is fixed. The experimental results are shown in Table 4, where INN-GAN represents INN-GAN(D), INN-GAN(S), and INN-GAN(US) since all achieve RS-BPP close to 1 (exceed 0.999). From the results in the table, it can be inferred that the proposed method has a higher payload compared to other methods when embedding the same amount of information. This also suggests that our method requires minimal error correction codes, allowing more space for increasing the capacity of messages embedding. As a result, the number of stego image transmissions in everyday communication can be reduced, thus lowering the risk of exposure during transmission.

4.4. Quality analysis

This section demonstrates the visual quality of the proposed method under three different network structures and compares them with other methods. The data embedding rate for all methods is fixed at 1 bpp. First, two test images randomly selected from the VOC dataset serve as cover images input into the three network models to obtain the corresponding stego images, and the results are visualized. As shown in Fig. 6, there is almost no discernible difference between the stego images generated by the three network structures and the original cover images to the human eye. To highlight differences, the absolute differences between the cover images and stego images are amplified by factors of 5, 10, and 20, respectively. Even after amplification by a factor of 10, the differences shown by the three network structures remain insignificant. Among them, INN-GAN(S) exhibits the highest visual quality. Additionally, the differences between the generated stego images and their corresponding cover images are more pronounced in the high-frequency region than in the low-frequency region. This

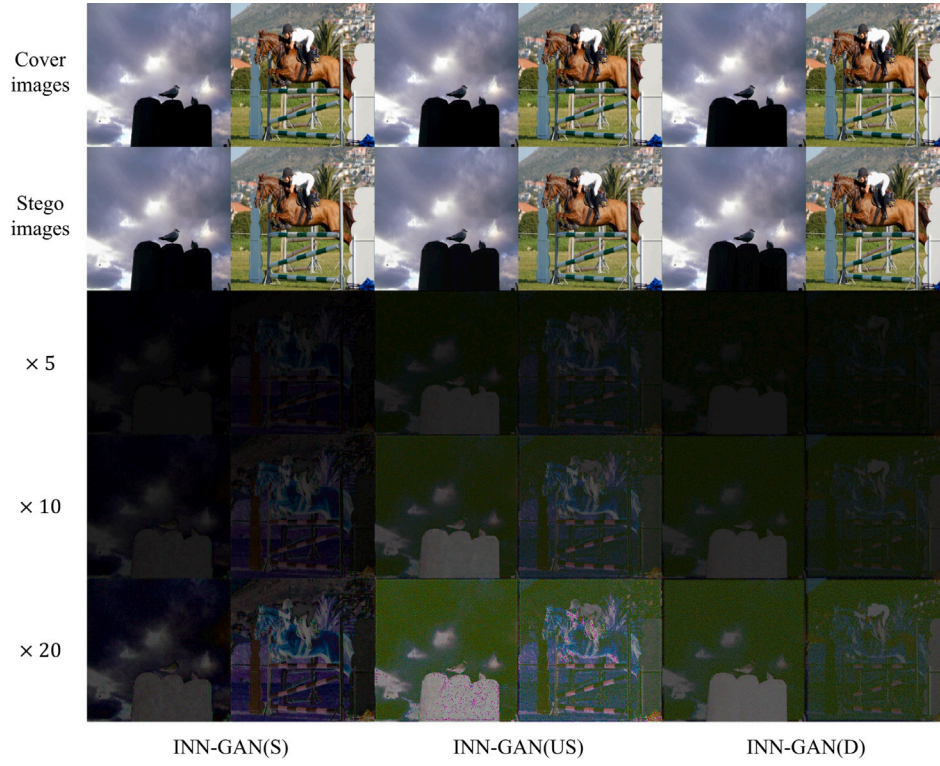


Fig. 6. Subjective visual quality at 1 bpp embedding rate. From top to bottom are the cover images X , the stego images S , the residual signal of $5|X - S|$, $10|X - S|$, and $20|X - S|$. Each pair of columns corresponds to the proposed network structures INN-GAN(S), INN-GAN(US), and INN-GAN(D).

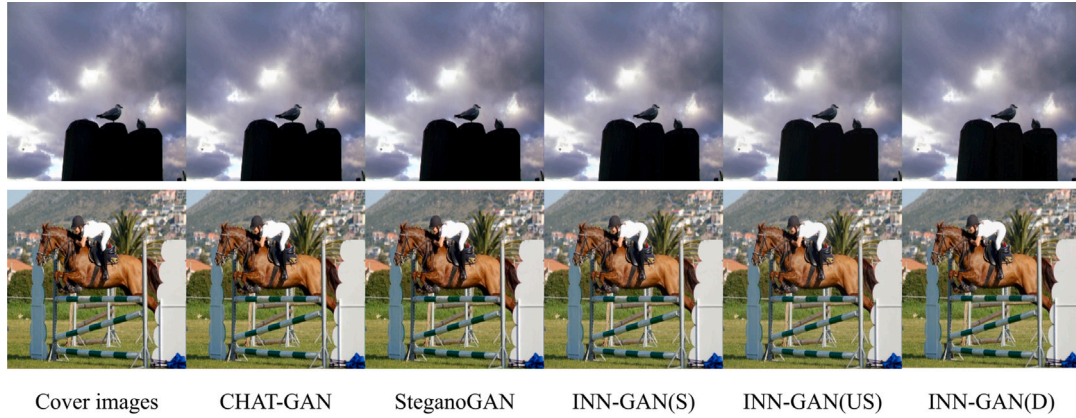


Fig. 7. Subjective visual quality at 1 bpp Embedding rate, each column corresponds to the cover images or the stego images generated by different network structures.

suggests that the proposed algorithm makes more significant modifications in the high-frequency region, which aligns with the human eye's lower sensitivity to higher frequencies. The proposed method is also compared with other methods in terms of visual quality. The experimental results in Fig. 7 depict two stego images generated by different methods. It is evident that stego images produced by all methods are visually identical to the cover images, making it extremely difficult to differentiate between them with the naked eye.

To demonstrate the superiority of the proposed method over other similar schemes, we use the metrics PSNR, SSIM, LP, and ACC to evaluate the invisibility of the stego images and the effectiveness of data extraction. Besides, to demonstrate the unique advantages of the proposed extraction network structures, we utilized CNNs as the extraction network in the original INN-GAN architecture and named it INN-GAN(C). Table 5 presents the performance of different methods on three test image datasets, where ACC values close to 1 indicate nearly perfect accuracy, exceeding 0.999. The performance metrics for

INN-GAN(C) have also been added to the table for comparison. From the table, it can be seen that INN-GAN(S) surpasses other comparative methods in terms of PSNR and SSIM. Specifically, INN-GAN(S) achieves PSNR values 1.77 dB, 5.58 dB, and 7.58 dB higher than the second-best method on the three datasets, respectively. Additionally, INN-GAN(D) exhibits the smallest LP, less than half that of other comparative methods. All three proposed network structures achieve ACC values close to 1 on each image dataset, higher than any of the comparative methods. INN-GAN(C) underperforms the three proposed structures across all four metrics. These results demonstrate that our method not only achieves better visual quality but also higher effectiveness in data extraction compared to state-of-the-art methods. Among the three different network structures, INN-GAN(S) performs the best overall. This is because, unlike INN-GAN(US) and INN-GAN(D) where the hidden and extraction networks are completely independent, INN-GAN(S) utilizes the same network weights for data embedding and extraction, and

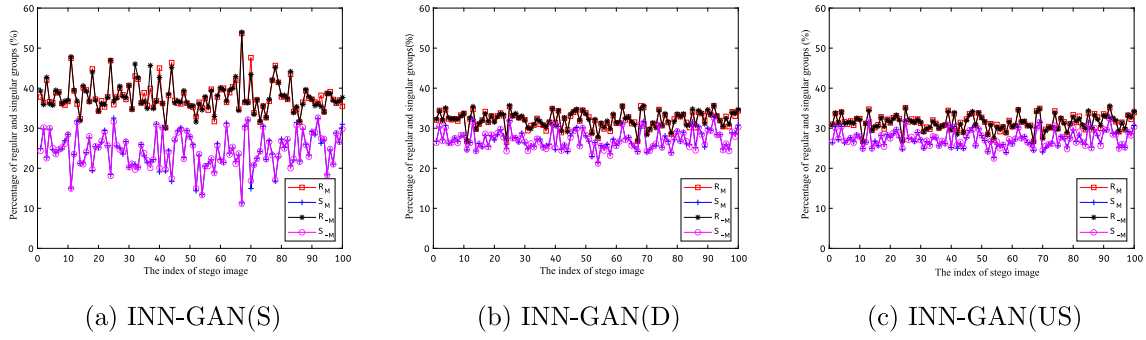


Fig. 8. RS steganalysis results for INN-GAN(S), INN-GAN(D), and INN-GAN(US). R_M and R_{-M} represents the result of the regular group under different flip functions; S_M and S_{-M} represents the result of the singular group under different flip functions.

Table 5

Comparative evaluation metrics across different datasets, the best results are highlighted in bold. $\uparrow$ denotes higher value is better, and vice versa.

Method	COCO				VOC				ImageNet			
	PSNR $\uparrow$	SSIM $\uparrow$	LP $\downarrow$	ACC $\uparrow$	PSNR $\uparrow$	SSIM $\uparrow$	LP $\downarrow$	ACC $\uparrow$	PSNR $\uparrow$	SSIM $\uparrow$	LP $\downarrow$	ACC $\uparrow$
ADV-EMB [9]	35.83	0.9716	4.3016	0.962	37.95	0.9858	3.6155	0.966	37.33	0.9883	3.6271	0.965
SteganoGAN [18]	42.42	0.9812	1.0946	0.993	45.01	0.9863	0.5324	0.996	46.82	0.9786	0.6706	0.997
CHAT-GAN [19]	42.54	0.9880	0.4961	0.993	45.25	0.9922	0.3234	0.994	47.58	0.9877	0.5626	0.999
INN-GAN(C)	29.58	0.9381	5.5178	0.914	33.11	0.9451	4.1790	0.935	34.33	0.9575	4.1724	0.948
INN-GAN(S)	44.31	0.9953	0.4241	≈ 1	50.83	0.9987	0.1009	≈ 1	55.16	0.9991	0.1976	≈ 1
INN-GAN(US)	36.93	0.9556	0.5580	≈ 1	38.11	0.9545	0.1965	≈ 1	38.62	0.9184	0.5255	≈ 1
INN-GAN(D)	40.34	0.9733	0.1326	≈ 1	41.71	0.9790	0.0291	≈ 1	41.33	0.9582	0.0801	≈ 1

Table 6

The detection accuracy of comparative methods and the three proposed network structures under different embedding payloads using SRM steganalysis.

Network structure	0.4 bpp	0.6 bpp	0.8 bpp	1.0 bpp	1.2 bpp
ADV-EMB [9]	72.7%	79.6%	85.5%	89.7%	95.1%
SteganoGAN [18]	76.0%	82.8%	85.5%	90.3%	93.7%
CHAT-GAN [19]	55.8%	58.8%	65.5%	72.1%	79.4%
INN-GAN(S)	55.1%	60.4%	66.0%	74.7%	84.9%
INN-GAN(US)	57.9%	61.6%	65.6%	72.9%	81.0%
INN-GAN(D)	53.6%	57.2%	62.7%	68.3%	75.5%

shares information between the hidden and extraction network, leading to better performance.

4.5. Security analysis

To comprehensively evaluate the security of the proposed method, this section will conduct analysis on three different network structures using both traditional statistical feature-based steganalysis methods [35] and deep learning-based steganalysis methods [25,36].

Steganalysis based on traditional statistical features involves analyzing the statistical properties of image pixels to distinguish between cover and stego images. The RS steganalysis [35] is used in the experiments and the RS steganalysis results for the three network structures are shown in Fig. 8, where R_M and R_{-M} represent regular group results under different flipping functions, while S_M and S_{-M} represent singular group results. When $R_M \approx R_{-M}$ and $S_M \approx S_{-M}$, it indicates that the steganography algorithm can effectively resist RS steganalysis. As shown as in graph, the curves representing R_M and R_{-M} , as well as S_M and S_{-M} , almost overlap, indicating that all three proposed network structures can effectively resist RS steganalysis.

In image steganography, the size of the embedding payload directly influences the security of the steganographic method: lower embedding payloads typically make stego images harder to detect, while higher payloads increase the risk of detection by steganalysis tools. Therefore, we decided to use SRM [37] as steganalyzers to evaluate the detection performance on stego images generated by our proposed method under

different embedding payloads. Specifically, we utilize the SOTA hand-crafted feature set SRM-34,671D with the ensemble classifier [38], where the fisher linear discriminant is employed as the base learner. A total of 4000 cover images and their corresponding stego images were selected from the COCO dataset as the training dataset, while an additional 1000 cover images and their corresponding stego images were used as the testing dataset. The detection accuracy of comparative methods [9,18,19] and the three proposed network structures at different embedding payloads is recorded in Table 6, where each entry represents the corresponding results for SRM. The data indicates that at lower embedding payloads, the proposed network structures and CHAT-GAN [19] exhibit characteristics of random guessing, as it is nearly unable to effectively distinguish between cover images and stego images. As the embedding payload increases, the detection accuracy of SRM gradually improves; however, the growth rate of the corresponding values for methods such as INN-GAN(D) remains relatively low. In contrast, ADV-EMB [9] and SteganoGAN [18] exhibit significantly higher steganalysis detection accuracy at lower embedding rates and almost completely lose their ability to resist SRM-based steganalysis when the embedding rate becomes sufficiently high. This demonstrates that the proposed three network structures can effectively resist SRM-based steganalysis across different embedding rates.

Deep learning-based steganalysis methods train neural network models to discern differences between cover and stego images, providing superior capabilities compared to traditional statistical feature-based methods. The steganalysis methods ZhuNet [36], YeNet [25] and SRNet [39] are employed in the experiments, using 16,000 cover images and 16,000 stego images for training, and 4000 cover images and 4000 stego images for testing. All cover images are 256×256 -sized color images sourced from the COCO database. The experimental results of our proposed network structures are shown in Table 7, we have also incorporated the detection accuracy of comparative methods, including ADV-EMB [9], SteganoGAN [18], and CHAT-GAN [19], under steganalysis into the table. It can be observed that when the data embedding rate is 0.6 bpp, the three steganalyzers can easily distinguish the stego images generated by ADV-EMB and SteganoGAN. Although the stego images produced by CHAT-GAN may be harder to

Table 7

Detection accuracy for deep learning-based steganalysis (ZhuNet, YeNet, and SRNet) of comparative methods and our proposed three network structures (embedding rate = 0.6 bpp).

Network structure	ZhuNet [36]	YeNet [25]	SRNet [39]
ADV-EMB [9]	83.8%	83.5%	87.8%
SteganoGAN [18]	80.3%	77.0%	91.0%
CHAT-GAN [19]	72.5%	63.2%	87.4%
INN-GAN(S)	77.1%	72.4%	84.8%
INN-GAN(US)	76.6%	71.9%	84.1%
INN-GAN(D)	69.0%	63.7%	80.9%

detect using certain steganalyzers compared to INN-GAN(S) or INN-GAN(US), the proposed network structure INN-GAN(D) demonstrates superior overall performance against most steganalyzers. Among all the evaluated architectures, the INN-GAN(D) network structure performs the best, indicating that at a certain data embedding rate, our proposed network structures can resist the detection of these steganalysis methods to a certain extent. As shown in Table 5, the LP of INN-GAN(D) is significantly lower than that of the other two structures. For instance, on the COCO dataset, INN-GAN(S) and INN-GAN(US) have LP around 0.42 and 0.56, respectively, whereas INN-GAN(D) has the LP of less than 0.13. The stego images generated by INN-GAN(D) are closer to the cover images in terms of features and this contributes to the higher security. INN-GAN(S) and INN-GAN(US) demonstrate weaker resistance to deep learning-based steganalysis compared to INN-GAN(D), possibly due to the similarity in the structure of the hidden and extraction networks. Balancing the selection of suitable network structures in various real-world applications can maximize performance benefits.

5. Conclusion

In this paper, we propose a novel deep learning-based image steganography network architecture capable of generating three structures: INN-GAN(D), INN-GAN(S), and INN-GAN(US). Specifically, leveraging the relationship with the hidden network, the different extraction networks are designed based on DenseNet, weight sharing, and non-weight sharing to identify more suitable embedding positions. The discriminator compares the cover image and the stego image in a patch-to-patch manner, effectively enhancing the invisibility and security of the method. Extensive experiments demonstrate that our proposed network structures outperform other GAN-based image steganography methods in various performance aspects, particularly achieving near-100% ACC without the need for error correction codes.

In the future, we aim to enhance the robustness and security of steganography methods. In real-world network environments, images often undergo compression, noise addition, and other processing. Simulating various image processing operations by adding noise layers can improve the method's robustness and serve as a future research direction. Additionally, the effectiveness of the proposed network structures against deep learning-based steganalysis requires further improvement. We can explore optimizing adaptive payload distribution and adjusting intensities in multiple images steganography [40] to enhance resistance against steganalysis. Furthermore, different network models need to be trained for hiding secret information of varying capacities in our method. Designing a capacity-adaptive steganography method would be a meaningful research direction with practical applications.

CRedit authorship contribution statement

Ke Wang: Writing – original draft, Validation, Software, Methodology, Conceptualization. **Yani Zhu:** Writing – review & editing, Supervision, Conceptualization. **Qi Chang:** Writing – review & editing, Supervision. **Junyu Wang:** Validation, Software, Methodology, Investigation. **Ye Yao:** Writing – review & editing, Supervision, Conceptualization.

Funding

This work was supported by the Zhejiang Provincial Natural Science Foundation of China under Grant Number LY24F020017 and LQ24F020036, the National Natural Science Foundation of China under Grant Number 62302135 and 62471167, and the Humanities and Social Sciences Foundation of Ministry of Education of China under Grant Number 23YJA870013.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

- [1] W. Bender, D. Gruhl, N. Morimoto, A. Lu, Techniques for data hiding, *IBM Syst. J.* 35 (3.4) (1996) 313–336.
- [2] Y. Yao, K. Wang, Q. Chang, S. Weng, Reversible data hiding in encrypted images using global compression of zero-valued high bit-planes and block rearrangement, *IEEE Trans. Multimed.* (2023).
- [3] S. Rustad, P.N. Andono, G.F. Shidik, et al., Digital image steganography survey and investigation (goal, assessment, method, development, and dataset), *Signal Process.* 206 (2023) 108908.
- [4] B. Li, M. Wang, J. Huang, X. Li, A new cost function for spatial image steganography, in: 2014 IEEE International Conference on Image Processing, ICIP, IEEE, 2014, pp. 4206–4210.
- [5] V. Sedighi, R. Cogranne, J. Fridrich, Content-adaptive steganography by minimizing statistical detectability, *IEEE Trans. Inf. Forensics Secur.* 11 (2) (2015) 221–234.
- [6] I.J. Kadhim, P. Premaratne, P.J. Vial, Improved image steganography based on super-pixel and coefficient-plane-selection, *Signal Process.* 171 (2020) 107481.
- [7] Y. Huang, Z. Liu, Q. Wu, X. Liu, Robust image steganography against JPEG compression based on DCT residual modulation, *Signal Process.* (2024) 109431.
- [8] D.R.I.M. Setiadi, S. Rustad, P.N. Andono, G.F. Shidik, Maximizing complex features to minimize the detectability of content-adaptive steganography, *Multimedia Tools Appl.* (2024) 1–19.
- [9] W. Tang, B. Li, S. Tan, M. Barni, J. Huang, CNN-based adversarial embedding for image steganography, *IEEE Trans. Inf. Forensics Secur.* 14 (8) (2019) 2074–2087.
- [10] W. Tang, S. Tan, B. Li, J. Huang, Automatic steganographic distortion learning using a generative adversarial network, *IEEE Signal Process. Lett.* 24 (10) (2017) 1547–1551.
- [11] J. Yang, D. Ruan, J. Huang, X. Kang, Y.-Q. Shi, An embedding cost learning framework using GAN, *IEEE Trans. Inf. Forensics Secur.* 15 (2019) 839–851.
- [12] W. Tang, B. Li, M. Barni, J. Li, J. Huang, An automatic cost learning framework for image steganography using deep reinforcement learning, *IEEE Trans. Inf. Forensics Secur.* 16 (2020) 952–967.
- [13] D. Huang, W. Luo, M. Liu, W. Tang, J. Huang, Steganography embedding cost learning with generative multi-adversarial network, *IEEE Trans. Inf. Forensics Secur.* (2023).
- [14] L. Zhou, G. Feng, L. Shen, X. Zhang, On security enhancement of steganography via generative adversarial image, *IEEE Signal Process. Lett.* 27 (2019) 166–170.
- [15] J. Yang, X. Liao, ACGIS: Adversarial cover generator for image steganography with noise residuals features-preserving, *Signal Process., Image Commun.* 113 (2023) 116927.
- [16] A. Martín, A. Hernández, M. Alazab, J. Jung, D. Camacho, Evolving generative adversarial networks to improve image steganography, *Expert Syst. Appl.* 222 (2023) 119841.
- [17] J. Zhu, R. Kaplan, J. Johnson, L. Fei-Fei, Hidden: Hiding data with deep networks, in: Proceedings of the European Conference on Computer Vision, ECCV, 2018, pp. 657–672.
- [18] K.A. Zhang, A. Cuesta-Infante, L. Xu, K. Veeramachaneni, SteganoGAN: High capacity image steganography with GANs, 2019, arXiv preprint arXiv:1901.03892.
- [19] J. Tan, X. Liao, J. Liu, Y. Cao, H. Jiang, Channel attention image steganography with generative adversarial networks, *IEEE Trans. Netw. Sci. Eng.* 9 (2) (2022) 888–903.
- [20] C. Yuan, H. Wang, P. He, J. Luo, B. Li, GAN-based image steganography for enhancing security via adversarial attack and pixel-wise deep fusion, *Multimedia Tools Appl.* 81 (5) (2022) 6681–6701.

- [21] S. Bernard, P. Bas, J. Klein, T. Pevný, Backpack: A backpropagable adversarial embedding scheme, *IEEE Trans. Inf. Forensics Secur.* 17 (2022) 3539–3554.
- [22] T. Filler, J. Judas, J. Fridrich, Minimizing additive distortion in steganography using syndrome-trellis codes, *IEEE Trans. Inf. Forensics Secur.* 6 (3) (2011) 920–935.
- [23] O. Ronneberger, P. Fischer, T. Brox, 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*, Springer, 2015, pp. 234–241.
- [24] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.
- [25] J. Ye, J. Ni, Y. Yi, Deep learning hierarchical representations for image steganalysis, *IEEE Trans. Inf. Forensics Secur.* 12 (11) (2017) 2545–2557.
- [26] K. Fu, D.-P. Fan, G.-P. Ji, Q. Zhao, J. Shen, C. Zhu, Siamese network for RGB-D salient object detection and beyond, *IEEE Trans. Pattern Anal. Mach. Intell.* 44 (9) (2021) 5541–5559.
- [27] Q. Giboulot, R. Cogranne, D. Borghys, P. Bas, Effects and solutions of cover-source mismatch in image steganalysis, *Signal Process., Image Commun.* 86 (2020) 115888.
- [28] G. Xu, H.-Z. Wu, Y.-Q. Shi, Structural design of convolutional neural networks for steganalysis, *IEEE Signal Process. Lett.* 23 (5) (2016) 708–712.
- [29] J. Hayes, G. Danezis, Learning universal adversarial perturbations with generative models, in: *2018 IEEE Security and Privacy Workshops, SPW*, IEEE, 2018, pp. 43–49.
- [30] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein generative adversarial networks, in: *International Conference on Machine Learning*, PMLR, 2017, pp. 214–223.
- [31] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft coco: Common objects in context, in: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*, Springer, 2014, pp. 740–755.
- [32] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, A. Zisserman, The pascal visual object classes (VOC) challenge, *Int. J. Comput. Vis.* 88 (2010) 303–338.
- [33] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: A large-scale hierarchical image database, in: *2009 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2009, pp. 248–255.
- [34] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2014, arXiv preprint arXiv:1409.1556.
- [35] J. Fridrich, M. Goljan, R. Du, Reliable detection of LSB steganography in color and grayscale images, in: *Proceedings of the 2001 Workshop on Multimedia and Security: New Challenges*, 2001, pp. 27–30.
- [36] R. Zhang, F. Zhu, J. Liu, G. Liu, Depth-wise separable convolutions and multi-level pooling for an efficient spatial CNN-based steganalysis, *IEEE Trans. Inf. Forensics Secur.* 15 (2019) 1138–1150.
- [37] J. Fridrich, J. Kodovsky, Rich models for steganalysis of digital images, *IEEE Trans. Inf. Forensics Secur.* 7 (3) (2012) 868–882.
- [38] J. Kodovsky, J. Fridrich, V. Holub, Ensemble classifiers for steganalysis of digital media, *IEEE Trans. Inf. Forensics Secur.* 7 (2) (2011) 432–444.
- [39] M. Boroumand, M. Chen, J. Fridrich, Deep residual network for steganalysis of digital images, *IEEE Trans. Inf. Forensics Secur.* 14 (5) (2018) 1181–1193.
- [40] X. Liao, J. Yin, M. Chen, Z. Qin, Adaptive payload distribution in multiple images steganography based on image texture features, *IEEE Trans. Dependable Secur. Comput.* 19 (2) (2022) 897–911.