Signal, Image and Video Processing (2025) 19:262
https://doi.org/10.1007/s11760-025-03818-0

ORIGINAL PAPER

Research on image steganography based on a conditional invertible neural network

Menghua Liang[1] · **Hongtu Zhao**[1]

Received: 5 October 2024 / Revised: 24 December 2024 / Accepted: 4 January 2025 / Published online: 28 January 2025

Abstract

To improve the imperceptibility of image steganography, an image steganography method based on a conditional invertible neural network is proposed in this paper. First, we design a conditional invertible neural network to obtain high-quality stego images with rich high-level semantic information and clear spatial details. On the basis of the conditional directivity of the conditional invertible neural network, we can adjust the semantic information of the stego image accurately and ensure the controllability of the stego image content. We introduce a dual cross-attention module into the network structure. The integration of dual cross-attention modules enhances feature extraction and captures complex image details to improve steganographic accuracy. In addition, the introduction of the convolutional block attention module in the convolutional layer directs the model's focus to key image regions, refining stego image quality. We increase the number of convolutional blocks, which improves the ability of feature extraction and reuse. Many experiments are carried out on datasets. For the cover and stego image pairs, the PSNR value reached 43.62 dB, and for the secret and recovery image pairs, the PSNR value reached 46.48 dB. The experimental results show that the image quality and imperceptibility of this method are better than those of other state-of-the-art image steganography methods.

Keywords Image steganography · Conditional invertible neural network · DCA · CBAM

1 Introduction

Image steganography, as an effective means of hiding information, originated in the 1940s. It has a wide range of applications in military, commercial, medical, and other fields. With the popularization of the internet and the advent of the big data era, digital images, as one of the main carriers of information, have become increasingly prominent in terms of security issues.

Traditional image steganography methods usually embed the secret information into the spatial domain or transform domain of the image through cover modification. The existing image steganography methods in the spatial domain include the least significant bit (LSB) method [1], histogram shifting method [2], and difference expansion method [3]. The robustness and security of spatial domain steganography are weak; it is difficult to recover images after they have been attacked and to resist detection by powerful steganalytic models. Unlike spatial domain image steganography, in the transform domain, the cover image needs to be converted from the spatial domain to the transform domain first, and then the secret data are hidden. Image steganography methods in transform domain include discrete cosine transform (DCT) [4], discrete wavelet transform (DWT) [5], discrete Fourier transform (DFT) [6], singular value decomposition (SVD) [7] and so on. However, the capacity for image steganographic embedding in the transform domain is small, and the visual quality of the image may not be ideal. In addition, there are Fractional order weighted spherical Bessel-Fourier moments[8], a powerful image analysis tool that can effectively extract image features and provide support in stegographic applications.

With the development of massive data availability, deep learning has become a trend and has been widely used in image steganography. Lin et al. [9] proposed a deep neural image steganography framework based on Y channel information and novel structure loss. This method exhibits

✉ Hongtu Zhao
hongtuzhao@hpu.edu.cn

[1] School of Physics & Electronic Information Engineering, Henan Polytechnic University, Jiaozuo 454003, Henan Province, China

strong robustness and expands the application scope of image steganography, but its embedding capacity is low. Ma et al. [10] proposed an adversarial embedding scheme with low operational complexity, which improved the performance of the adversarial steganography analyzer. Yu [11] proposed an end-to-end framework based on attention mechanisms and a generative adversarial network (GAN), which enhances the robustness of the model, but the invisibility of stego images needs further improvement. Jing et al. [12] considered image concealment and image revealing as forward and reverse processes of the same invertible neural network (INN) for the first time and designed an image steganography method called HiNet. To enhance INNs' proficiency in handling image-related tasks, Dinh et al. [13] introduced convolutional layers and multiscale layers into the coupling model to reduce computing costs and improve regularization capabilities. Kingma et al. [14] introduced reversible 1 $\times$ 1 convolution in an INN and proposed the Glow algorithm, which has significant effects on image synthesis and processing. As an important variant of the INN, the conditional invertible neural network (cINN) has been successfully applied to audio-hiding in video [15] and steg-TTS, which is based on a text-to-speech system [16]. In addition, the cINN has also been applied in the field of image steganography, such as steg-glow [17], a generative image steganography algorithm based on the invertible network Glow. It performs well in terms of statistical security but cannot control the synthesized semantic content, thus failing to guarantee the security of covert communication behavior. This is also a common problem faced by existing generative image steganography methods.

To solve this problem, a generative image steganography algorithm based on a cINN is proposed in this paper. On the one hand, our method leverages the reversibility of the INN structure to realize the reversible extraction of secret images and improve the extraction accuracy of the steganography algorithm. On the other hand, the proposed method utilizes the conditional guidance of cINN to ensure the security of image steganography. Our main contributions are as follows:

- We propose an image steganography method based on cINN. Through the conditional directivity of cINN, the security of the steganography algorithm can be ensured.
- We propose dual cross-attention (DCA), a simple yet effective attention module. DCA has a stronger feature extraction capability, which can better capture the detailed information of the image, thereby enhancing the accuracy of steganography.
- We introduce the convolutional block attention module into the convolutional layer to direct the model's focus to key image regions, refining the stego image quality.
- The number of convolutional blocks is increased to improve the ability of feature extraction and reuse.

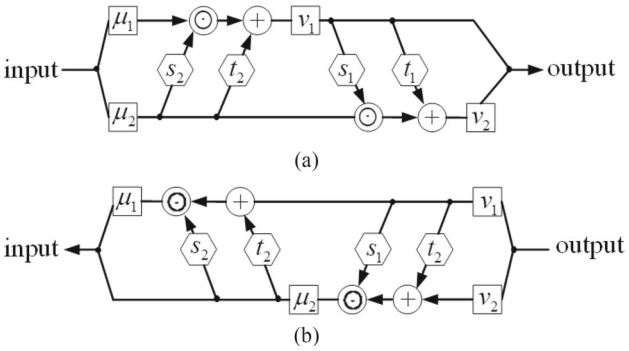

Fig. 1 Forward process and inverse process of the INN

2 Conditional invertible neural network

2.1 Invertible neural network

In recent years, the invertible neural network has attracted the attention of many researchers because of its excellent performance. The INN maps high-dimensional complex distributions p_x to simple latent distributions p_z through a series of invertible transformations and uses the neural network to learn the mapping relationship between p_x and p_z. The forward process takes high-dimensional complex data x as input and outputs data z that conform to a simple distribution. The inverse process is a generative modeling process that takes sampled data z as input to generate high-dimensional complex data x. The basic network architecture of the INN is the affine coupling layer generalized from the real NVP model, which works by splitting the input data into two parts, μ_1 and μ_2, which are then converted by the learning functions s_i and t_i, which are coupled in an alternating manner. The INN forward process is shown in Fig. 1a, and the computational formulas are shown in Eqs. (1) and (2). The learning functions s_i and t_i can be any complex function, and the function itself need not be invertible.

$$v_1 = \mu_1 \odot \exp(s_1\mu_2) + t_2(\mu_2) \tag{1}$$

$$v_2 = \mu_2 \odot \exp(s_1 v_2) + t_1(\mu_1) \tag{2}$$

where exp is the exponential function in mathematics and where $\odot$ represents the Hadamard product. The Hadamard product is the result of multiplying the corresponding elements of two matrices, and the resulting matrix is the same size as the original matrix.

Owing to the invertibility of the INN, the inverse process is shown in Fig. 1b, and the computational formulas are shown in Eqs. (3) and (4).

$$\mu_2 = (v_2 - t_1(v_1)) \odot \exp(-s_1(v_1)) \tag{3}$$

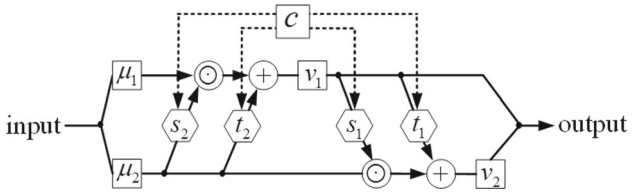

Fig. 2 Framework structure of cINN

$$\mu_1 = (v_1 - t_2(\mu_2)) \odot \exp(-s_2(\mu_2)) \tag{4}$$

Because of its powerful network representation capabilities, the INN is suitable for a variety of inference tasks, such as image hiding, image coloring, image scaling, and image compression.

2.2 Conditional invertible neural network

The conditional invertible neural network (cINN) is an INN model with conditional guidance. In reference [18], the development process of the flow-based generation model of reversible networks is explained in detail. The inverse proxy model developed can be applied to the inversion of multiphase flow problems. We apply conditional invertible neural network to image steganography, where c stands for conditional information, which in this paper refers to specific image features and context information. By introducing this conditional information, cINN is able to better control the embedded information in the generated image. In the process of image generation, conditional information can affect the network's learning and representation of specific features, so that the generated image remains visually similar to the original image, and at the same time, the secret information can be effectively embedded. When extracting steganographic information, cINN can use the same conditional information to decode the embedded information and ensure the consistency of the information recovery process and the embedding process.

The framework structure of cINN is shown in Fig. 2. Its basic principle is to fit a conditional distribution $q(x; c)$ of real-world data x with the help of a simple distribution of hidden variables z, where c is the condition. cINN's inference process is divided into two steps. First, the standard normal distribution is sampled to obtain the hidden variables z. Then, calculate $x = f^{-1}(z; c)$, where x represents the data we want to synthesize, f represents the mapping of the cINN feedforward direction, and f^{-1} represents the mapping of the cINN generation direction. z is the latent variable that drives the cINN model to perform the generative task. Notably, f^{-1} or f is composed of a series of invertible transformations f_i, which means that $z = f(x; c)$ can be written as $z = f_k$, $f_{k-1} \ldots f_2 \cdot f_1(x; c)$. In terms of structure, the affine coupling layer is the basic component of cINN, and its role is to realize reversible transformations f_i.

During training, the training goal of a cINN is to ensure that the hidden variables z are close to the standard normal distribution. Since cINN is naturally structurally reversible, as long as z is close to the standard normal distribution, $x = f^{-1}(z; c)$ will be close to the distribution $q(x; c)$ of data x in the real world.

To solve the security problem in the steganography process, this paper proposes a generative image steganography algorithm based on cINN. This method can make use of the conditional directivity of cINN so that the semantic information of the generated image can be adjusted and controlled to ensure that the content of the classified image data is controllable, which ensures the behavioral security of the steganography algorithm.

3 Method

3.1 Network

The framework of the network is shown in Fig. 3. The secret image x_{secret} and the cover image x_{cover} are jointly used as the inputs to the forward network, and images are first preprocessed via the discrete wavelet transform (DWT) to decompose them into low-frequency and high-frequency wavelet subbands. The processed feature map is then input into the concealing block of the network, and a new feature map is obtained via the concealing block transform. Then, the data are processed by the DCA module to better capture detailed image information. Then, the image information is processed via the inverse wavelet transform (IWT) to obtain the stego image x_{stego} and the loss of information r. The lost secret information and the corrupted cover information together constitute the loss information r. Owing to the reversibility of the network, the backward revealing process requires the introduction of a random variable z. The variable z is randomly drawn from an arbitrary Gaussian distribution, which is the same as the distribution of r. In complete contrast to the forward concealing process, the stego image x_{stego} together with the random vector z serves as the input to the inverse network. The image is preprocessed by the DWT, and the feature map is fed into the revealing block of the network to obtain a new feature map, which then undergoes the DCA module. The recovered image is subsequently processed via the IWT to obtain the recovered secret image and the cover image.

3.2 Dual cross-attention

We introduce a dual cross-attention module into the network structure to improve steganographic accuracy. DCA is able to

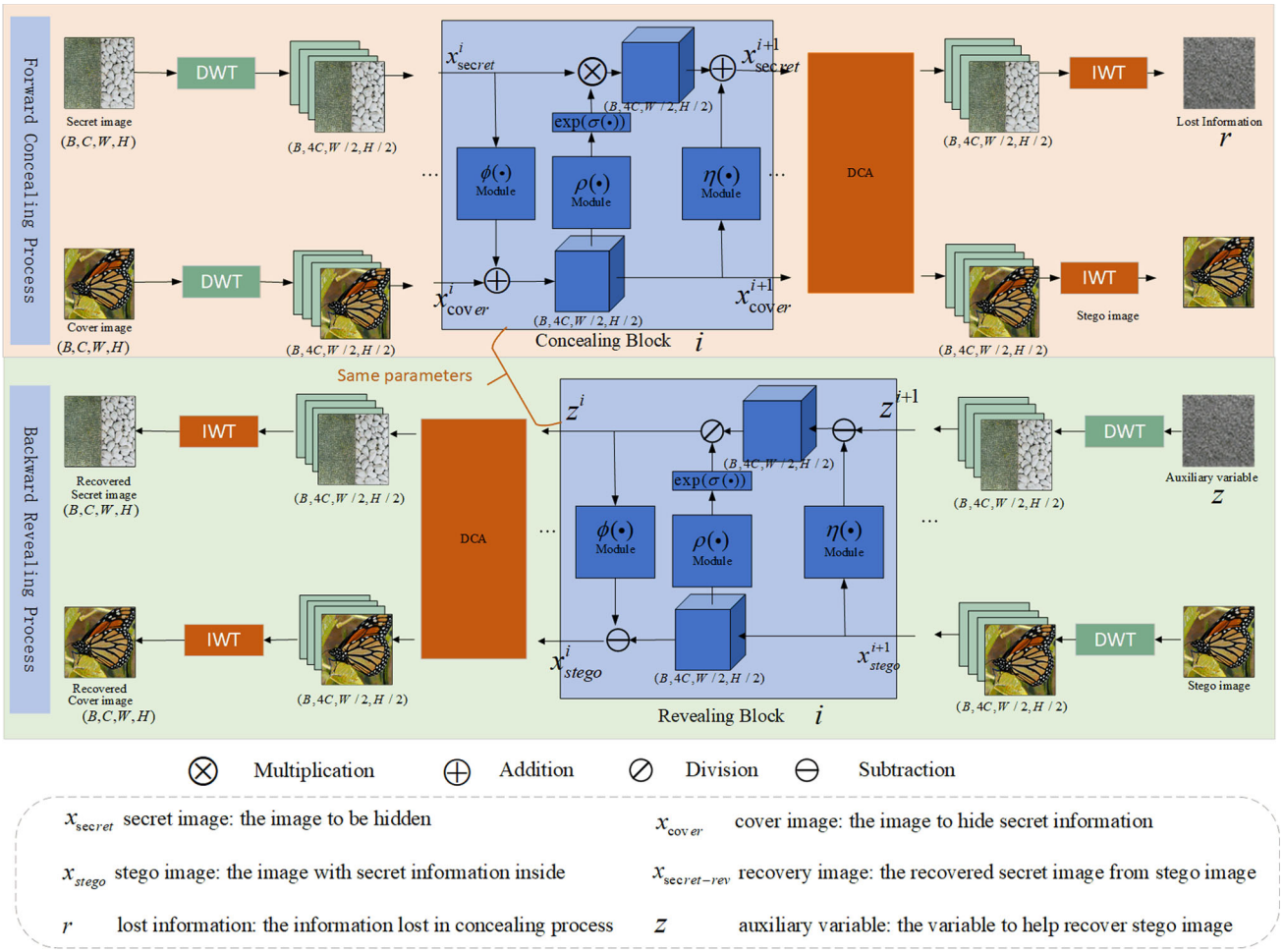

Fig. 3 Framework of the network

improve the performance of the model with a slight increase in parameters and complexity. In addition, DCA can effectively extract the channel and spatial dependencies between multiscale encoder features, thus improving the accuracy of steganography.

As shown in Fig. 4a, the DCA block is divided into two main phases. The first phase consists of multiscale patch embedding modules to acquire encoder tokens. In the second phase, we perform DCA mechanisms on encoder tokens via channel cross-attention (CCA) and spatial cross-attention (SCA) modules to capture remote dependencies. Finally, we apply layer normalization and GeLU sequences and upsample the tokens, connecting them to their decoder counterparts.

3.2.1 Multi-scale patch embedding

We firstly extracts patches from n multi-scale encoder stages. Given n different scales of encoder stages $E_i \in \mathbb{R}^{C_i \times \frac{H}{2^{i-1}} \times \frac{W}{2^{i-1}}}$ and patch sizes $P_1^S = \frac{P^S}{2^{i-1}}$, where $i = 1, 2, \ldots, n$. We extract the patches using a 2D average pool with a pool size and step size of P_1^S, and apply projection using a 1×1 deep convolution on a flat 2D patch:

$$T_i = DConv1D_{E_i}(\mathrm{Reshape}(AvgPool2D_{E_i}(E_i))) \quad (5)$$

where $T_i \in \mathbb{R}^{P \times C_i}$, $(i = 1, 2, \ldots, n)$ denotes the planarized patch of the ith encoder stage. Note that P is the number of patches, which is the same for each T_i, so we can exploit the cross-attention of these tokens.

3.2.2 Channel cross-attention module

The CCA module is shown in Fig. 4b. Each token Ti (i = 1, 2,..., n) is fed into the CCA module. First, the execution level of each Ti is normalized. Then, by following the cross-attention strategy, we replace all linear projections with 1 $\times$ 1 depthwise convolution projections by creating keys and values for token Ti along the channel dimension:

$$Q_i = DConv1(D_Q(T_i)) \quad (6)$$

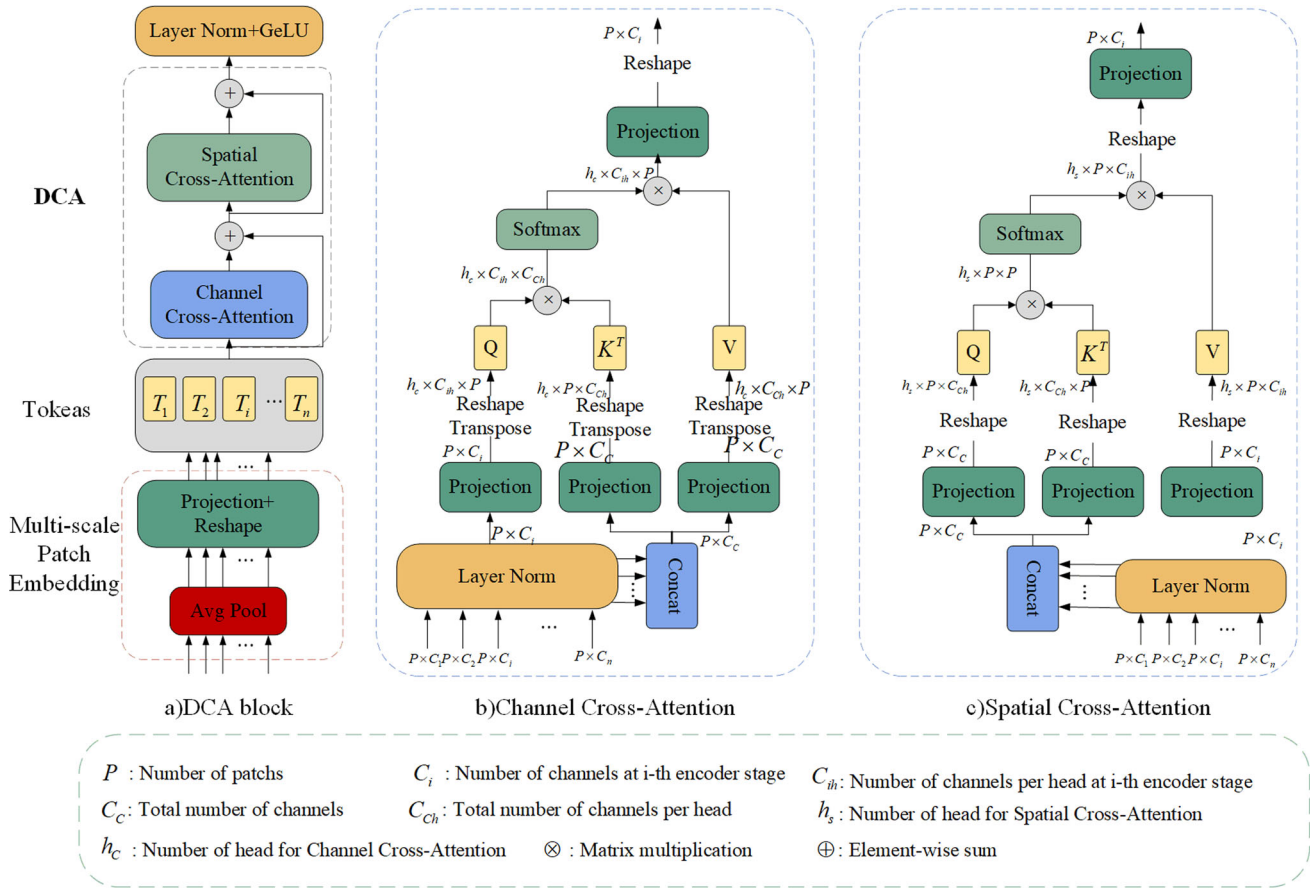

Fig. 4 Architecture of the DCA block

$$K = DConv1(D_Q(T_c)) \tag{7}$$

$$V = DConv1(D_Q(T_c)) \tag{8}$$

where $Q_i \in \mathbb{R}^{P\times C_i}$, $K \in \mathbb{R}^{P\times C_c}$, and $V \in \mathbb{R}^{P\times C_c}$ are the projected queries, keys, and values, respectively. To utilize cross-attention on the channel dimension, Q_i, K and V are transposed. Thus, the CCA takes the form:

$$CCA(Q_i,\ K,\ V) = Soft\max\left(\frac{Q_i^T K}{\sqrt{C_c}}\right)V^T \tag{9}$$

where Q_i, K and V are matrices and where $\frac{1}{\sqrt{C_c}}$ is the scaling factor. The output is a weighted sum of V, weighted by the similarity between Q_i and K. The Softmax output weights are V. Finally, deepwise convolutional projections are applied to the cross-attention output, which is fed into the SCA module.

3.2.3 Spatial cross-attention module

The SCA module is shown in Fig. 4c. Given reshaped outputs $\bar{T}_i \in \mathbb{R}^{P\times C_c}(i = 1, 2, ..., n)$, layers are normalized and cascaded along the channel dimension in the CCA module. The connected tokens $\bar{T}_c$ are used as queries and keys, while each token $\bar{T}_i$ is used as a value. We use 1×1 depthwise projections over queries, keys, and values:

$$Q = DConv1D_Q(\bar{T}_c) \tag{10}$$

$$K = DConv1D_K(\bar{T}_c) \tag{11}$$

$$V = DConv1D_{V_i}(\bar{T}_i) \tag{12}$$

where $Q \in \mathbb{R}^{P\times C_c}$, $K \in \mathbb{R}^{P\times C_c}$, and $V_i \in \mathbb{R}^{P\times C_i}$ are the projected queries, keys and values, respectively. Then, the

SCA can be represented as:

$$SCA(Q,\ K,\ V_i) = Soft\max(\frac{QK^T}{\sqrt{d_k}})V_i \tag{13}$$

where Q, K, V are matrices representing the query, key and value embeddings, respectively, and where $\frac{1}{\sqrt{d_k}}$ is the scale factor. For the multihead case $\frac{1}{\sqrt{d_k}} = \frac{C_c}{h_c}$, h_c is the number of heads. The outputs of the SCA module are then projected via depthwise convolutions to form the DCA outputs.

3.3 CBAM

The convolutional block attention module (CBAM) is an attentional mechanism for computer vision tasks designed to improve the performance of convolutional neural networks.

In the channel attention module (CAM), the global information of each channel is first obtained through the global average pooling operation, and each channel has a corresponding value that represents the global characteristics of the channel. Next, the channel weights are learned by a small fully connected network. The network can have one or more fully connected layers and eventually output a weight tensor of shape $(C,\ 1,\ 1)$. The learned channel weights are multiplied by the original input feature map to scale the features of each channel.

In the spatial attention module (SAM), the weights for each spatial position are learned in a similar manner, using the output of the CAM as input. The pooling operation in the CAM is repeated in the spatial dimension, and then the features obtained from two different pools are concatenated. Spatial correlation is learned by the convolutional layer, and the SAM is obtained via a sigmoid activation function. The outputs of the CAM and SAM are multiplied to obtain the modulated feature map. This feature map is used as the output of the CBAM module of the subsequent network layer.

In this work, as shown in Fig. 5, the number of convolutional layers in the reversible block is increased from 5 to 7 layers, and the convolutional output is fed into the CBAM module. By adaptively adjusting the weights of different channels and allocating attention to different locations, CBAM enables the network to capture key information more accurately, thereby improving the performance of the model.

3.4 Loss function

The loss function proposed in this paper consists of three main components:

(1) Concealing loss L_{con}.
In the forward process, we hide the secret images $x_{\mathrm{sec}ret}$ to the cover images x_{cover} to generate the stego images x_{stego}. To ensure that the generated images x_{stego} and cover images x_{cover} are as visually similar as possible so that observers cannot easily distinguish between them, we propose the concealing loss function L_{con}. We define the loss function as:

$$L_{con}(\theta) = \sum\nolimits_{n=1}^{N} \ell_c(x^n_{\mathrm{cover}},\ x^n_{stego}) \tag{14}$$

where N represents the number of training samples and ℓ_c calculates the difference between the cover images and stego images.

(2) Low-frequency wavelet loss L_{freq}.
To keep the information hidden as much as possible in the high-frequency region of images and to enhance the network's ability to resist steganalysis, we propose a low-frequency wavelet loss function. According to the literature [19], information hidden in high-frequency components is less likely to be detected than it is in low-frequency components. To ensure that most of the information is hidden in the high-frequency subbands, the low-frequency subbands of stego images after wavelet decomposition must be similar to those of cover images. In this paper, the loss constraint of the low-frequency subbands of stego images and cover images is used, as shown in the following equation:

$$L_{freq}(\theta) = \sum\nolimits_{n=1}^{N} \ell_f(H(x^n_{\mathrm{cover}})_{ll},\ H(x^n_{stego})_{ll}) \tag{15}$$

where N represents the number of training samples, ℓ_f represents the low-frequency difference between stego images and cover images, and $H(\cdot)_{ll}$ represents the low-frequency subband operation of extracted images.

(3) Revealing loss L_{rev}.
To ensure consistency between the recovered secret image and the embedded secret image, minimizing the difference between the recovered secret image and the secret image improves the model accuracy, as shown in the following equation:

$$L_{rev}(\theta) = \sum\nolimits_{n=1}^{N} E_{z-p(z)}\left[\ell_r\left(x^n_{\mathrm{sec}ret},\ x^n_{\mathrm{sec}ret-rev}\right)\right] \tag{16}$$

where N represents the number of training samples and where ℓ_r calculates the difference between the recovered secret images and the secret images. The process of random vector z sampling is random. The total loss function is a weighted sum of concealing loss, low frequency wavelet loss and revealing loss:

$$L_{total}(\theta) = \lambda_1 L_{con} + \lambda_2 L_{freq} + \lambda_3 L_{rev} \tag{17}$$

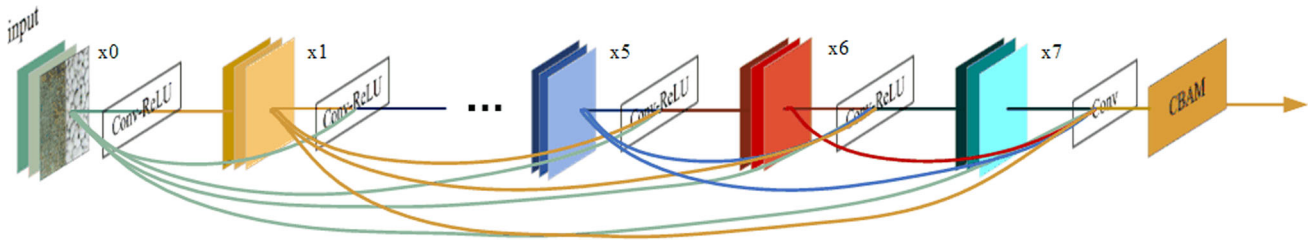

Fig. 5 The structure of dense blocks $\phi(\cdot)$, $\rho(\cdot)$, $\eta(\cdot)$

In the training process, λ2 is first set to 0, which means that the network model is directly pretrained without considering the effect of low frequency on the network. Then, the low-frequency constraint term is gradually added to further optimize the network model to hide the secret image in the high-frequency region of the cover image to enhance the model's ability to resist steganalysis. In this paper, we experimentally find that different λ2 values have a certain effect on the quality of the steganographic image of the network model. We found through experiments that the effect is the best when λ2 = 0.5. Therefore, in this experiment, we set $\lambda_1 = 5$, $\lambda_2 = 0.5$, $\lambda_3 = 1$.

4 Experiments

4.1 Experimental environment

The experiments are conducted on a deep learning server with a GeForce RTX 3090 GPU and 24 GB of RAM. It is built on the PyTorch 1.7.1 deep learning framework, with Python interpreter version 3.7.16 and CUDA version 11.0.

4.2 Experimental datasets and evaluation index

Datesets. The DIV2K dataset is a diverse, high-resolution, realistic dataset covering natural landscapes, people, animals, and buildings and is well suited for information hiding. We use the DIV2K dataset for our experiments. To verify the generalization ability of the algorithm proposed in this paper, we also use the COCO dataset and ImageNet dataset.

Experimental setup. The network model is trained via the Adam optimizer with a learning rate $= 1 \times 10 - 4.5$, batch size $= 32$, the number of invertible blocks for the whole network model is 16, the number of convolutional blocks contained in each invertible block is 7, λ1 = 5, λ2 = 0.5, and λ3 = 1, and the hiding loss uses the L2 regularization loss function.

Evaluation metrics. There are four indicators used to measure the quality of cover & stego and secret & secret-rev pairs, including Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), Root Mean Square Error(RMSE) and Mean Absolute Error (MAE). The larger the PSNR and SSIM values are, the smaller the RMSE and MAE values are, indicating better image quality. The relative payload (RP) is used as an evaluation metric to compare the steganographic ability of each stego image. The mathematical expression for RP is given below:

$$RP = \frac{bits(\text{sec}\,ret\ \text{inf}\,ormation)}{bits(\text{cover}\,capacity)} \times 100\% \tag{18}$$

In addition, we perform security analysis experimentally in Sect. 4.3.6.

4.3 Experimental results and analysis

4.3.1 Ablation experiment

To verify the effectiveness of the cINN, DCA and CBAM designed in this paper, we conducted many ablation experiments on the DIV2K dataset, and the experimental results are shown in Table 1. For the cover and stego image pairs, the PSNR of our method reaches 42.6257 dB, and the SSIM reaches 0.9851. For the secret and recovery image pairs, 46.4863 dB is achieved for the PSNR, and 0.9985 is achieved for the SSIM. The simultaneous use of cINN, DCA, and CBAM in our method significantly improves the quality of both the hidden and recovered images.

We obtain the residual map by subtracting the pixels of the cover image from the stego image. Figure 6 shows the visual effect of the stego image generated via our method and the residual plot between the cover image and stego image. Stego and cover images are difficult to distinguish with the eyes. The residual plot between the two images is almost pure black, indicating that the difference between the stego image and the cover image is minimal.

4.3.2 Effect of the number of convolutional blocks

Owing to the reversibility of the network, $\phi(\cdot)$, $\rho(\cdot)$, $\eta(\cdot)$ in the reversible block can use any function; in this paper, for

Table 1 The results of ablation experiments (the first line is the baseline of this experiment, without any modification)

cINN	DCA	CBAM	Cover & stego		Secret & recovery	
			PSNR	SSIM	PSNR	SSIM
			34.3198	0.8724	40.3013	0.9772
√			40.1469	0.9537	45.2955	0.9947
	√		39.9550	0.9637	44.2596	0.9796
		√	38.2720	0.9374	44.8776	0.9879
√	√		42.1124	0.9789	45.5369	0.9950
	√	√	40.4812	0.9452	45.2581	0.9941
√		√	41.8562	0.9548	45.6583	0.9936
√	√	√	*42.6257*	*0.9851*	*46.4863*	*0.9985*

The best results are shown in italics

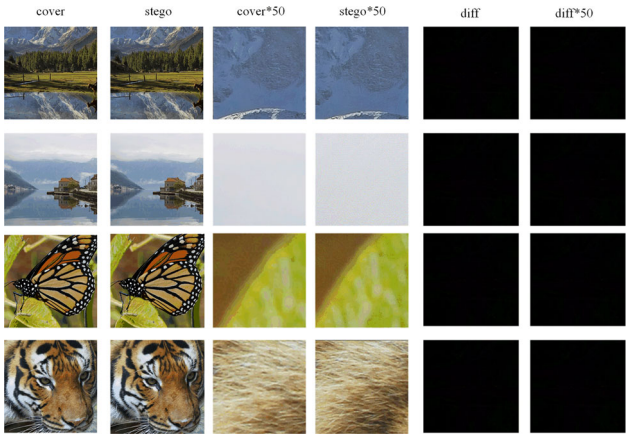

Fig. 6 Residual map between the cover image and stego image. From left to right are the cover image, stego image, cover image with 50 enlarged local feature maps, stego image with 50 enlarged local feature maps, residual image, and residual image with 50 enlarged local feature maps

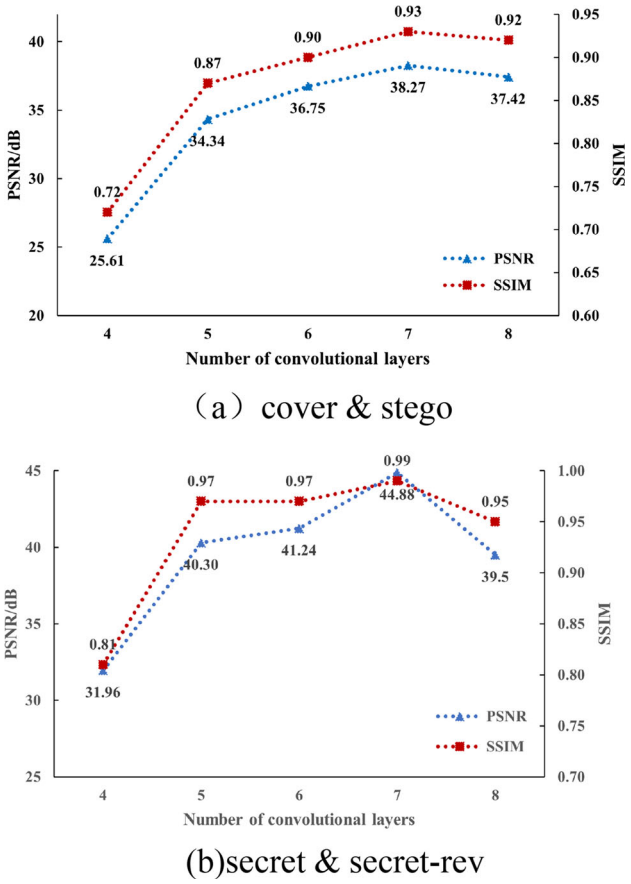

Fig. 7 Effect of the number of convolutional blocks on image quality

$\phi(\cdot)$, $\rho(\cdot)$, $\eta(\cdot)$, DenseNet is used. The number of convolution blocks used by DenseNet in HiNet is 5, but we found that the number of convolutional layers in DenseNet has a certain effect on the network's cryptographic ability through experiments. Figure 7 shows the quality of the stego image and the quality of the recovered image corresponding to different structures of $\phi(\cdot)$, $\rho(\cdot)$, $\eta(\cdot)$ and analyzes the effects of different structures on the performance of the network model. Figure 7 shows the hidden image quality and recovered image quality corresponding to different structures. As shown in Fig. 7, when the number of convolutional layers is 7, the PSNR value for the cover and stego image pairs is 38.27 dB, and the SSIM is 0.93, which are 11.4% and 6.9% higher than those when the number of convolutional layers is 5. For the secret image and recovery image pairs, the PSNR value is 44.88 dB, and the SSIM is 0.99, which are 11.3% and 2.1% higher than those when the number of convolutional layers is 5. This is because as the number of convolutions increases, deep feature extraction of the image becomes more effective to achieve feature reuse. When the number of convolutional layers exceeds 7, the model complexity and number of computations of the network greatly increase, and overfitting occurs. Therefore, the number of convolution layers is set to 7 in this paper.

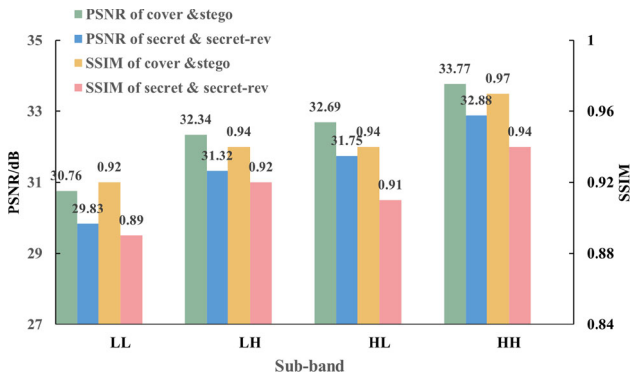

Fig. 8 Effect of the hiding image in the LL, HL, LH and HH subbands on image quality

4.3.3 Effect of hiding frequency on steganographic performance

In order to verify that the information hidden in the high-frequency component is less likely to be detected than that in the low-frequency component is, we perform DWT on the cover image and obtain four wavelet subbands, LL, LH, HL, and HH. Among them, LL is the low-frequency subband, and HL, LH, and HH are high-frequency subbands. The size of each subband is one-fourth that of the original cover image. Then, secret images of the same size as the subbands are hidden in the LL, LH, HL, and HH subbands via the Baluja method. The PSNR and SSIM are used to measure the results of the hidden images and recovery images. As shown in Fig. 8, hiding the secret image in the high-frequency subbands (LH, HL, HH) results in higher PSNR and SSIM values, suggesting that the high-frequency subbands may be more suitable for image hiding in terms of image quality preservation (Fig. 9).

4.3.4 Effect of different λ_2 values

To further optimize the network model to hide the secret image in the high-frequency region of the carrier image to enhance the model's ability to resist steganalysis, we find that different values of λ_2 have a certain effect on the quality of the steganographic image of the network model in the process of gradually adding low-frequency constraint terms. As shown in Fig. 10, fixing the values of λ_1 and λ_3 unchanged, we take the value of λ_2 from 0 to 1.2 with a step size of 0.1. Through many experiments, we show that the network model has the best steganalysis ability when $\lambda_2 = 0.5$.

4.3.5 Comparison with other methods

To prove the superiority of the steganography algorithm proposed in this paper, we compare five steganography methods. One of them is traditional LSB-1bit steganography;

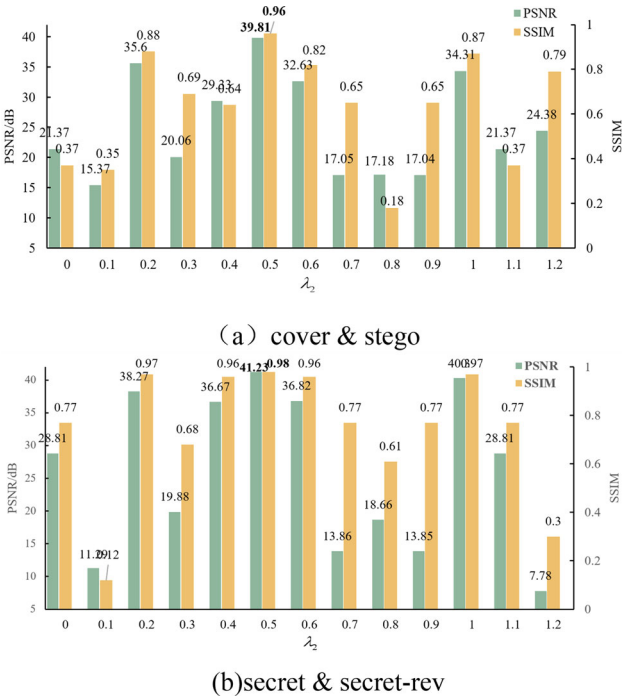

Fig. 9 Effect of low-frequency loss weights

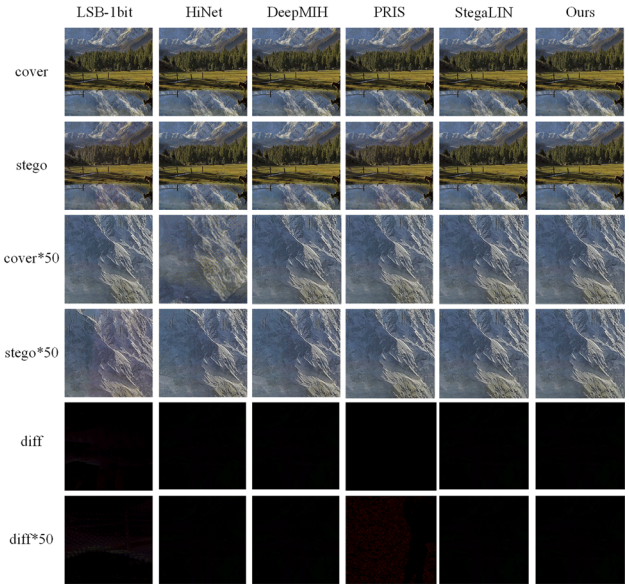

Fig. 10 Comparison of cover and stego images between different methods. From top to bottom are the cover image, stego image, cover image enlarged 50 times, stego image enlarged 50 times, residuals between the cover image and stego image, and residuals enlarged 50 times

HiNet [12] is a classical steganography algorithm based on reversible neural networks. StegaLIN [20] is a lightweight image steganography method based on invertible networks. PRIS [21] is a practical robust invertible network based on HiNet. DeepMIH [22] is a multi-image hiding method based on an invertible neural network. To ensure the visual quality

Table 2 Benchmark comparisons on different datasets, with the best results shown in italic and the second best in bold

Method	RP (%)	DIV2K				COCO				ImageNet			
		PSNR	SSIM	MAE	RMSE	PSNR	SSIM	MAE	RMSE	PSNR	SSIM	MAE	RMSE
Cover & Stego image													
LSB-1bit	50	33.1948	0.9453	6.93	9.52	33.2522	0.9420	7.24	9.77	33.2149	0.9428	7.04	9.63
HiNet	300	32.4314	0.9482	3.81	4.25	33.0015	0.9537	3.37	4.10	33.5914	0.9516	3.72	4.59
DeepMIH	300	33.7719	**0.9645**	3.70	4.14	32.1458	**0.9584**	3.79	4.26	**34.3368**	**0.9572**	**3.10**	3.94
PRIS	300	32.4630	0.8874	3.13	3.78	32.1654	0.9016	3.66	4.48	33.6120	0.9152	3.33	4.17
StegaLIN	300	**34.3198**	0.8724	**2.77**	**3.16**	**33.9544**	0.8756	**3.15**	**4.00**	34.2693	0.9016	3.14	**3.82**
Ours	300	*42.6257*	*0.9851*	*2.15*	*2.88*	*41.3457*	*0.9824*	*2.69*	*3.13*	*40.8731*	*0.9766*	*2.71*	*3.25*
Secret & recovery image													
LSB-1bit	50	30.8172	0.9020	8.64	12.37	30.7650	0.9125	8.91	11.94	31.5249	0.9168	9.72	12.98
HiNet	300	31.7726	0.9476	4.28	4.71	32.0469	0.9695	3.90	4.67	32.7513	0.9571	3.76	4.84
DeepMIH	300	32.8823	0.9377	3.99	4.18	33.2148	0.9687	3.62	4.06	33.4295	0.9595	3.71	4.25
PRIS	300	32.7024	0.9053	3.28	3.47	32.8951	0.9347	3.33	4.14	33.7162	0.9365	3.44	3.97
StegaLIN	300	**40.3013**	**0.9772**	**2.51**	**3.00**	**39.1537**	**0.9831**	**2.41**	**3.17**	**40.9245**	**0.9850**	**2.66**	**3.28**
Ours	300	*46.4863*	*0.9985*	*2.06*	*2.35*	*43.2596*	*0.9886*	*2.28*	*2.38*	*46.2671*	*0.9914*	*2.12*	*2.48*

of the steganographic images, we set the embedding capacity of the LSB method to 1 bpp. The comparison results are shown in Table 2. As shown in Table 2, compared with other state-of-the-art methods, the method proposed in this paper has the best image quality.

Table 2 compares our method with the experimental results from LSB-1bit, HiNet, DeepMIH, PRIS and StegLIN. As shown in Table 2, the steganographic effect of our method on the three datasets is significantly superior to that of the other methods. Specifically, for the cover and stego image pairs, our method yields 8.3059 dB, 7.3913 dB and 6.5363 dB higher than the second-best results on the DIV2K, COCO and ImageNet datasets, respectively. For the secret and recovered image pairs, our method achieves improvements of 6.1850 dB, 4.1059 dB and 5.3426 dB over the second-best results on the DIV2K, COCO and ImageNet datasets, respectively.

The residual map is obtained by subtracting the cover image from the stego image. As shown in Fig. 10, color distortion occurs in the stego images of DeepMIH, HiNet and PRIS. The color distortion of the LSB-1bit stego image is more obvious, and the residual map is the most obvious. The stego image generated by our method is visually indistinguishable from the cover image. This finding also shows that our method of steganography works best.

4.3.6 Imperceptibility

Histogram analysis is a method of specifying the distribution of pixel values in an image. Different images inherently exhibit different distribution trends, and any change in the pixel value of the image subsequently alters its histogram distribution. Therefore, evaluating whether the histogram distribution of an image is similar or the same as that of its original counterpart has become a classic way to discern whether an image is hiding any secret information. After training, we randomly select 8 images and compare their histograms. Figure 11 shows the histogram of the cover image and the steganographic image in our experiment, and it can be observed that there is little visual difference between the steganographic image and the cover image, and that the secret image is visually identical to the extracted image. According to the histogram of the cover image and steganographic image shown in Fig. 11, the distribution of pixels before and after steganography changes very slightly, which also proves the good invisibility of the algorithm.

4.3.7 Steganpgraphic analysis

Steganalysis results measure the likelihood that a steganalysis tool [23] can distinguish between stego images and cover images. The current mainstream steganalysis methods can be divided into two categories: traditional methods based on statistics and new methods based on deep learning.

Traditional steganalysis tools, such as the one described in the literature [24], use the open source steganalysis tool StegExpose [25] to evaluate the steganalysis resistance of the network model in this paper. One hundred randomly selected cover images and secret images from the test set are passed through the network model of this paper to generate the stego image. To plot the ROC curves, the detection threshold in StegExpose varies over a wide range. Figure 12 shows the

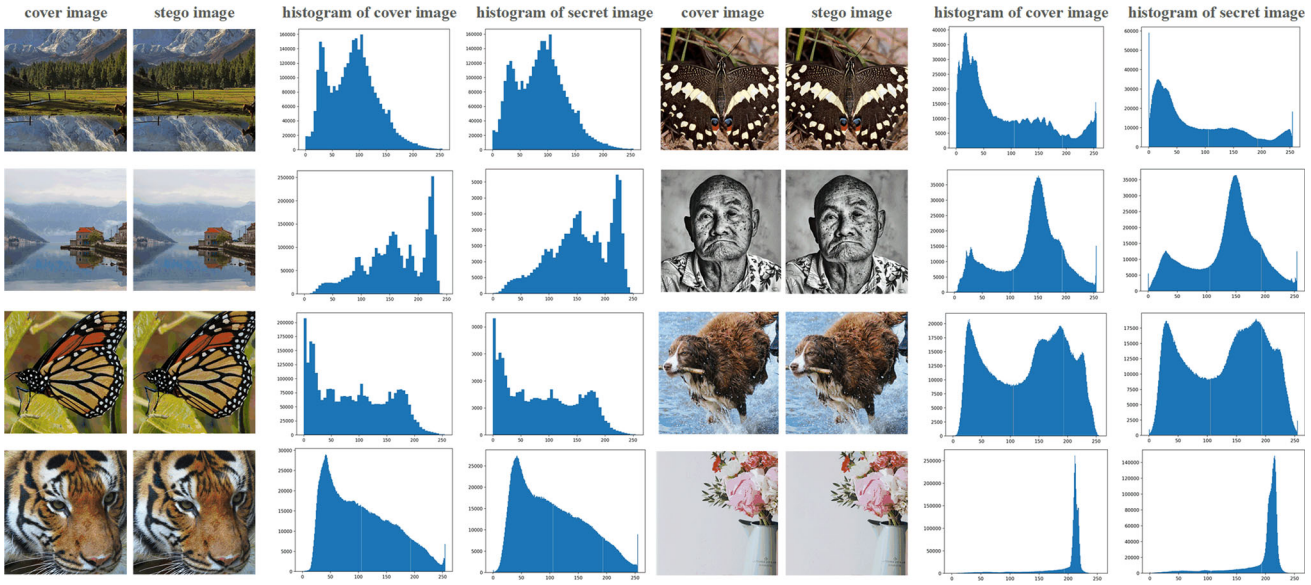

Fig. 11 Histogram analysis results

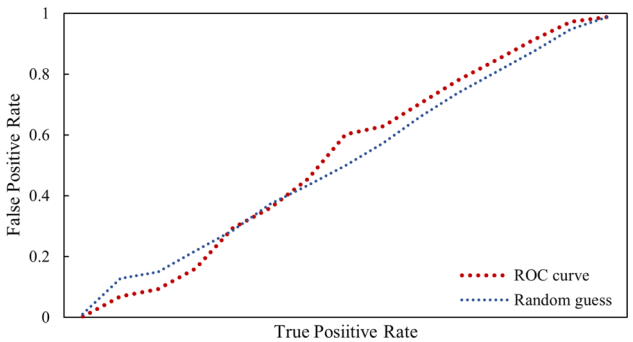

Fig. 12 The ROC curve generated from the StegExpose analysis results

Table 3 Detection errors of different methods in three steganographic analysis networks, with the best results shown in itlic and the second best in bold

Method	Detection error(%)		
	XuNet	SRNet	WISERNet
LSB-1bit	0.26	0	2.41
HiNet	2.34	0.11	1.27
DeepMIH	4.31	1.74	2.58
PRIS	**10.15**	2.67	**4.68**
StegaLIN	9.36	**3.71**	4.28
Ours	*10.31*	*3.96*	*6.72*

ROC curve of the network. Figure 12 shows that StegExpose does not perform well on our method, with detection accuracies close to random guessing. This finding indicates that the stego images generated by our method are highly secure against detection by traditional steganalysis tools.

Deep learning-based steganalysis. To verify the security, we used three state-of-the-art steganographic networks, XuNet [26], SRNet [27], and WISERNet [28], to evaluate the steganographic security of different steganographic methods with embedded payloads of 1 bpp. We generated 3000 cover/stego image pairs for each method to retrain the steganalysis network. Table 3 lists the detection errors of the steganalysis networks using the three different methods. The detection errors of our methods in the three steganalysis networks are considerably greater than those of the other methods, which indicates that our methods have better steganographic security.

5 Conclusion

This paper presents an image steganography method based on a conditional invertible neural network. This method makes full use of the reversible property of the INN and guarantees the reversible extraction of secret information, thus significantly improving the extraction precision of the algorithm. Moreover, with the conditional directivity of cINN, the semantic information of steganographic images can be accurately adjusted to ensure the controllability of steganographic image content to further ensure the security of steganographic algorithms at the behavioral level. DCA has a strong feature extraction ability. Introducing DCA into the network can better capture the details of the image. In addition, CBAM is introduced into the convolutional layer to make the model pay more attention to the key areas of the image, thus improving

the quality of the hidden image. To extract the deep features of images better and realize feature reuse, the number of convolutional blocks is increased, and the effect of the number of convolutional blocks on image steganography is verified by experiments. We conducted many experiments and steganography on different datasets. The results show that the steganography method proposed in this paper is significantly superior to other state-of-the-art hiding methods in terms of visual quality and imperceptibility.

However, this algorithm also has several shortcomings. On the one hand, the network model is very large, requiring high GPU performance and long training times. On the other hand, real transmission channels often face problems such as malicious eavesdropping, channel noise and image compression, which present challenges for the transmission of confidential information. Addressing these challenges while improving capacity, safety, and robustness in practical applications is a relevant and necessary direction for future research. In addition, we study lightweight image steganography models.

Author contributions Hongtu Zhao conceived the idea of the study; Hongtu Zhao and Menghua Liang analysed the data; Hongtu Zhao and Menghua Liang interpreted the results; Menghua Liang wrote the paper; all authors discussed the results and revised the manuscript.

Data availability No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Ehsan Ali, U.A., Ali, E., Sohrawordi, M., et al.: A LSB based image steganography using random pixel and bit selection for high payload. Int. J. Math. Sci. Comput. **7**, 24–31 (2021)
2. Anushiadevi, R., Amirtharajan, R.: Separable reversible data hiding in an encrypted image using the adjacency pixel difference histogram. J. Inf. Secur. Appl. **72**, 103407 (2023)
3. Ge, Y., Zhang, M., Yang, P.: Reversible data hiding in encrypted domain based on color image channel correlation. In: Third International Seminar on Artificial Intelligence, Networking, and Information Technology (AINIT 2022). SPIE, vol. 12587, pp. 317-324 (2023)
4. Mao, B.H., Wang, Z.C., Zhang, X.P.: Asymmetric JPEG steganography based on correlation in DCT domain. Comput. Sci. **46**(01), 203–207 (2019)
5. Rashmi, P., Supriya, M.C., Hua, Q.: Enhanced lorenz-chaotic encryption method for partial medical image encryption and data hiding in big data healthcare. Secur Commun Netw **2022**(1), 9363377 (2022)
6. Soni, A., Jain, J., Roshan, R.: Image steganography using discrete fractional Fourier transform. In: 2013 international conference on intelligent systems and signal processing (ISSP). IEEE, pp. 97–100 (2013)
7. Singh, J., Singla, M.: Image steganography technique based on singular value decomposition and discrete wavelet transform. Int. J. Electr. Electron. Res. **10**(2), 122–125 (2022)
8. Yang, T., Liu, Z., Guo, J., et al.: Image analysis by fractional-order weighted spherical Bessel-Fourier moments. Pattern Recogn. **157**, 110872 (2025)
9. Lin, W., Zhu, X., Ye, W., et al.: An improved image steganography framework based on Y channel information for neural style transfer. Secur. Commun. Netw. **2022**(1), 2641615 (2022)
10. Ma, S., Zhao, X.: Generating JPEG steganographic adversarial example via segmented adversarial embedding. In: International Workshop on Digital Watermarking pp. 68-79. Springer International Publishing, Cham (2020)
11. Yu, C.: Attention based data hiding with generative adversarial networks. In: Proceedings of the AAAI conference on artificial intelligence, vol. 34(01), pp. 1120–1128 (2020)
12. Jing, J., Deng, X., Xu, M., et al.: Hinet: deep image hiding by invertible network. In: Proceedings of the IEEE/CVF international conference on computer vision. Pp. 4733–4742 (2021)
13. Dinh, L., Sohl-Dickstein, J., Bengio, S.: Density estimation using real nvp. arXiv preprint arXiv:1605.08803 (2016)
14. Kingma, D. P., Dhariwal, P.: Glow: generative flow with invertible 1x1 convolutions. Adv. Neural Inf. Process. Syst. 31 (2018)
15. Yang, H., Ouyang, H., Koltun, V., et al.: Hiding video in audio via reversible generative models. In: Proceedings of the IEEE/CVF international conference on computer vision, pp. 1100–1109 (2019). https://doi.org/10.1109/ICCV.2019.00119
16. Chen, K., Zhou, H., Zhao, H., et al.: Distribution-preserving steganography based on text-to-speech generative models. IEEE Trans. Dependable Secure Comput. **19**(5), 3343–3356 (2022). https://doi.org/10.1109/TDSC.2021.3095072
17. Chen, K., Zhou, H., Hou, D., et al.: Provably secure steganography on generative media. arXiv preprint, (2018)
18. Padmanabha, G.A., Zabaras, N.: Solving inverse problems using conditional invertible neural networks. J. Comput. Phys. **433**, 110194 (2021). https://doi.org/10.1016/j.jcp.2021.110194
19. Baluja, S.: Hiding images in plain sight: deep steganography. Adv. Neural Inf. Process. Syst. 30 (2017)
20. Sun, W., Liu, J., Niu, K., et al.: Lightweight image steganography scheme based on invertible neural network. Appl. Res. Comput. **41**(01), 266–271 (2024). https://doi.org/10.19734/j.issn.1001-3695.2023.05.0215
21. Yang, H., Xu, Y., Liu, X., et al.: PRIS: practical robust invertible network for image steganography. Eng. Appl. Artif. Intell. **133**, 108419 (2024)
22. Guan, Z., Jing, J., Deng, X., et al.: DeepMIH: deep invertible network for multiple image hiding. IEEE Trans. Pattern Anal. Mach. Intell. **45**(1), 372–390 (2022). https://doi.org/10.1109/TPAMI.2022.3141725t
23. Xiao, M., Zheng, S., Liu, C., et al.: Invertible image rescaling. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16, pp. 126-144. Springer International Publishing (2020)
24. Rahim, R., Nadeem, S.: End-to-end trained CNN encoder-decoder networks for image steganography. In: Proceedings of the European conference on computer vision (ECCV) workshops, pp. 1–6 (2018)
25. Boehm, B.: Stegexpose-A tool for detecting LSB steganography. arXiv preprint arXiv:1410.6656 (2014)
26. Xu, G., Wu, H. Z., Shi, Y. Q.: Ensemble of CNNs for steganalysis: an empirical study. In: Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security, pp. 103–107 (2016)
27. Boroumand, M., Chen, M., Fridrich, J.: Deep residual network for steganalysis of digital images. IEEE Trans. Inf. Forensics Secur. **14**(5), 1181–1193 (2018)

28. Zeng, J., Tan, S., Liu, G., et al.: WISERNet: wider separate-then-reunion network for steganalysis of color images. IEEE Trans. Inf. Forensics Secur. **14**(10), 2735–2748 (2019)