Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.064901

ARTICLE

Fixed Neural Network Image Steganography Based on Secure Diffusion Models

Yixin Tang[1,2], Minqing Zhang[1,2,3,*], Peizheng Lai[1,2], Ya Yue[1,2] and Fuqiang Di[1,2,*]

[1]College of Cryptography Engineering, Engineering University of People's Armed Police, Xi'an, 710086, China
[2]Key Laboratory of People's Armed Police for Cryptology and Information Security, Engineering University of People's Armed Police, Xi'an, 710086, China
[3]Key Laboratory of CTC & Information Engineering, Ministry of Education, Engineering University of People's Armed Police, Xi'an, 710086, China

*Corresponding Authors: Minqing Zhang. Email: api_zmq@126.com; Fuqiang Di. Email: 18710752607@163.com

Received: 27 February 2025; Accepted: 26 June 2025; Published: 30 July 2025

ABSTRACT: Traditional steganography conceals information by modifying cover data, but steganalysis tools easily detect such alterations. While deep learning-based steganography often involves high training costs and complex deployment. Diffusion model-based methods face security vulnerabilities, particularly due to potential information leakage during generation. We propose a fixed neural network image steganography framework based on secure diffusion models to address these challenges. Unlike conventional approaches, our method minimizes cover modifications through neural network optimization, achieving superior steganographic performance in human visual perception and computer vision analyses. The cover images are generated in an anime style using state-of-the-art diffusion models, ensuring the transmitted images appear more natural. This study introduces fixed neural network technology that allows senders to transmit only minimal critical information alongside stego-images. Recipients can accurately reconstruct secret images using this compact data, significantly reducing transmission overhead compared to conventional deep steganography. Furthermore, our framework innovatively integrates ElGamal, a cryptographic algorithm, to protect critical information during transmission, enhancing overall system security and ensuring end-to-end information protection. This dual optimization of payload reduction and cryptographic reinforcement establishes a new paradigm for secure and efficient image steganography.

KEYWORDS: Image steganography; fixed neural network; secure diffusion models; ElGamal

1 Introduction

Image steganography is a crucial tool for covert communication, ensuring that only authorized users possess the necessary information to extract and decrypt secret messages, thereby recovering the original data. To counter human visual inspection and machine-driven steganalysis, stego-images must be indistinguishable from cover images in visual appearance and statistical properties. The foundational covert communication framework, the "prisoner's model," was introduced by Simmons [1] in 1983. This model involves two prisoners (Alice and Bob) and a warden (Cain). Alice embeds a secret escape plan into a natural-looking image, transmitting it to Bob through a monitored channel without Cain's awareness.

Traditional steganographic methods primarily modify cover media to embed secret data. Classic spatial-domain techniques include LSB (Least Significant Bit) [2], WOW (Weighted Optimization Watermarking) [3], and HUGO (Highly Undetectable Steganography) [4], while transform-domain approaches leverage

DWT (Discrete Wavelet Transform) [5] and DCT (Discrete Cosine Transform) [6]. However, these methods often suffer from high distortion and are easily detectable by advanced steganalysis tools.

In the field of traditional steganography, there are also approaches that utilize encryption of images or texts, known as Reversible Data Hiding in the Encrypted Domain (RDH-ED). RDH-ED not only enables embedding additional information within the ciphertext but also allows for the exact, lossless recovery of the original plaintext. This technique is particularly valuable in applications where even minor distortions are unacceptable. Based on how they leverage redundancy in the cover data for embedding purposes, symmetric encryption-based RDH-ED methods can be divided into two main types: "vacating room before encryption (VRBE)" [7,8], and "vacating room after encryption (VRAE)" [9,10].

With advancements in generative networks, steganography has transitioned from traditional methods to data-driven paradigms. Hu et al. [11] further improved extraction accuracy and payload capacity using deep convolutional generative adversarial network (DCGAN), mapping secret information to noise vectors for generating stego-images via GANs.

The rise of DNNs (Deep Neural Networks) has significantly advanced steganography. Notable frameworks like Learning-Based Neural Network Steganography (LNNS) [12–15] replace manual cover modification with data-driven embedding. LNNS employs an encoder-decoder architecture: the encoder embeds secrets into cover images, while the decoder extracts them. Training objectives aim to balance minimal visual distortion with high extraction accuracy. However, DNN-based methods demand extensive datasets and computational resources, resulting in large model files. Securely transmitting these models between parties is impractical, as the overhead often exceeds that of directly transmitting encrypted images. Training a high-performance neural network to complete generalized steganography tasks takes several hours or even days.

Recent breakthroughs focus on fixed neural networks [16–18] and other emerging frameworks [19,20]. In fixed neural networks, pre-trained models remain static, with only input data being optimized. This approach drastically reduces training time and model complexity. Fixed Neural Network Steganography (FNNS) eliminates the need for trainable encoders. Instead, decoders generate weights from a seed without training. This method minimizes system complexity and enhances security by perturbing cover images to adapt to the fixed decoder. However, fixed neural networks [16–18] focus only on the protection of steganography itself and do not emphasize the need for ensuring the security of content transmitted over public channels to guarantee the safety of steganography.

In summary, traditional steganography involves minor changes to an image to hide data. It's easy to do and quick, but it's also easier for someone to detect. Image encryption scrambles the entire image so the hidden data is protected and hard to see, but it can be more complicated to decrypt. Neural network (NN) based steganography uses AI (Artificial Intelligence) to hide data very secretly and make detection difficult. However, they need a lot of training data and powerful computers. Overall, traditional methods are simple but less secure, encryption offers better protection but is more complex, and NN-based methods are very good at hiding data but harder to develop.

1.1 Related Work

Hu et al. [11] introduced a novel DCGAN-based image steganography method, which demonstrated outstanding performance in steganographic capacity, undetectability, and image reconstruction quality, achieving better results in metrics like PSNR (Peak Signal to Noise Ratio) and SSIM (Structural Similarity). Regarding Learning Neural Network-based steganography, Shumeet [12] employed Laplacian Pyramid Networks to enhance the steganographic process, enabling secret information embedding without significantly

altering the appearance of generated carrier images. Through multi-scale decomposition, this approach facilitates the effective reconstruction of embedded content during information extraction, ensuring high quality and precision. To further optimize algorithm performance, Rahim et al. [13] incorporated loss penalties during steganographic network training to enhance training stability. Zhang et al. [14] improved secret information processing by converting secret images into invisible high-frequency information prior to embedding. In the latest research, Jing et al. [15] enhanced both steganographic and extraction networks by proposing HiNet—a deep learning framework utilizing invertible neural network technology. This innovation not only embeds secret information into images but also enables accurate extraction of hidden data from stego images.

During the development of FNNS, to enhance the robustness of steganography and confirmed that their embedding is elusive, encode data as the labels of the image that the evasion attacks produce. But the limited steganographic capacity restricted its practical application value. Kishore et al. [18] proposed a method that trains only on input images (without modifying the network architecture) to effectively hide secret information. By expanding the decoder's output dimensions to enhance steganographic capacity, this approach balances the fidelity of original images with the ability to extract and decode hidden messages. They improved the robustness of FNNS to JPEG (Joint Photographic Experts Group) compression by adding a JPEG layer (with quality factor 80) in their optimization pipeline, in which the back-propagated gradients are approximated with identity transformation. However, existing FNNS methods still suffer from high distortion in generated stego images, where visual artifacts become perceptible under high embedding capacities.

Recent research has introduced several innovative approaches to neural network-based steganography. Luo et al. [19] proposed a framework based on the State Space Model (SSM) that dynamically adjusts the information distribution among multiple carrier images to optimize the embedding process. They introduced the concept of "Immune-Cover" to resist steganalytic tools based on statistical analysis. Since images with distortion resistance reduce embedding damage, the original cover image is reconstructed through the Immune-cover construction module (ICCM) and associated with the steganography task. The method demonstrates excellent robustness and efficiency. Li et al. [17] proposed Cover-separable Fixed Neural Network Steganography via Deep Generative Models. This method leverages mature deep generative models to create carrier images and embeds secret information into AI-generated carriers using a minimal perturbation strategy. The sender and receiver share a seed as a key to generate identical carrier images and decoders. This ensures no visual distortion even under high embedding capacities while accurately extracting hidden information, in terms of optimizing neural networks. To address the loss of reconstruction accuracy caused by lossy transmission during the steganographic process, they incorporated existing denoising networks into the steganography procedure to enhance its robustness. Zhou et al. [20] further improved concealment and image quality through a channel attention mechanism and multi-module collaborative optimization, reducing detection risks. This provides an efficient solution for secure communication and privacy protection. However, Zhou's method overlooks the security risks in key transmission and employs insufficiently complex prompt words for deep generative models, resulting in overly simplistic carrier images. To overcome these challenges. We propose a Secure Diffusion Model-based Fixed Neural Network Steganography for image steganography. This method efficiently accomplishes steganographic tasks under high embedding capacities while maintaining robust resistance to steganalysis but also ensures the secure transmission of prompt words, random seeds, and other minimal critical information through public-key encryption, guaranteeing the security of generated images during transmission. The key advantage of this approach lies in its dual assurance: it secures the steganographic process with minimal critical information while also safeguarding the secure transmission of such information.

1.2 Objective of This Study

Our research contributions can be summarized as follows:

- **Enhanced Anti-Steganalysis and Reduced Distortion**
  To improve the anti-steganalysis capability of stego-images while minimizing distortion, we map secret information onto a mask image and design a composite loss function to train the mask image. This dual optimization ensures undetectability by steganalysis tools and high-fidelity preservation of the original image.
- **Increased Complexity of Stego Images**
  To enhance the complexity of generated stego images, we develop a method leveraging a finely-tuned Stable Diffusion model. Our approach ensures rich diversity in stego image outputs while maintaining semantic consistency by employing simple-to-complex prompt words and generating images guided by random seeds.
- **Secure Transmission of Critical Parameters**
  We implement an ElGamal algorithm-based encryption scheme for these critical parameters to safeguard the security of prompt words and seeds during transmission. This cryptographic layer guarantees end-to-end security throughout the steganographic process.

Our work aims to advance steganography in reliability, security, anti-steganalysis resilience, and application versatility, addressing real-world challenges to ensure information security across diverse scenarios.

2 The Proposed Scheme

The core architecture of our method comprises three distinct phases: stego image generation, secure transmission of critical parameters, and secret information extraction.

During the stego image generation stage, the process is divided into two parts: generating a cover image using a diffusion model and embedding secret information into the carrier image via a mask image. Here, the cover image is not a natural image selected from traditional fixed datasets but rather a non-natural image synthesized through the denoising process of the diffusion model. Under a fixed neural network generated from a shared seed, we construct a mask image with minimal impact on the carrier image by balancing anti-steganalysis capability and the fixed neural network's accuracy of secret information extraction. The stego-image is then obtained by adding the mask image to the cover image.

During the secret information extraction, the receiver utilizes the securely transmitted critical information (e.g., decrypted prompt words and seeds) to regenerate the identical cover image and fixed neural network. The mask image is derived by subtracting the carrier image from the received stego image, and the original secret information is accurately reconstructed from the mask image through the preconfigured fixed neural network. As shown in Fig. 1, this section presents the overall framework of the proposed algorithm.

In this context, the acronym I_s stands for the secret image. I_m denotes the trained and optimized mask image. SD represents the Stable Diffusion. I_c is the cover image, which is generated by the SD model under the seed (S_2) and prompt conditions. The steganographic image I_{st} is obtained by adding the I_m to the I_c. *KI* stands for the key information. *KI* includes the seed (S_2) and prompts, which are encrypted and transmitted alongside the I_{st} via a public channel.

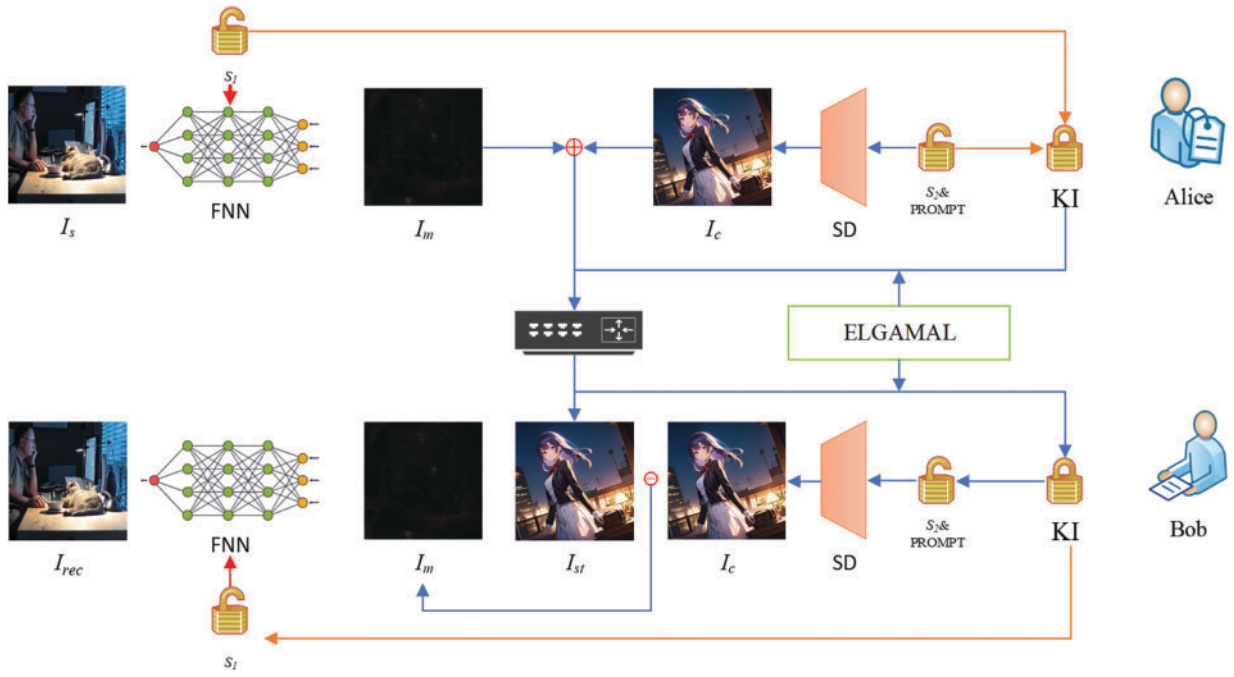

Figure 1: The overall framework of steganography using fixed neural networks based on secure diffusion models

2.1 Generate the Cover Image

In the steganography algorithm proposed in this article, the generation of the cover image differs from traditional approaches to cover images. Conventional cover images typically rely on naturally captured or artist-created images in various styles, as procedurally generated images were not widely transmitted across the internet then. In recent years, however, advanced deep generative models have rapidly created vast quantities of exquisite, elaborate, and high-quality images. Many companies and individuals now use these models for tasks like cartoon character design, comic creation, and illustration. Building on state-of-the-art deep generative models [21], we employ Stable Diffusion as the foundational generator for cover images. To address the growing prevalence of images generated online by deep anime-style, we adopt AwPainting as the generator (denoted as G). AwPainting is a generative model fine-tuned from Stable Diffusion v1.5 through further training on a large dataset of anime-style images, achieving superior performance in generating professional-grade anime artwork. As shown in Fig. 2, when comparing realistic images generated by other series of Stable Diffusion with real photographs, distinct artificial features are evident, drawing considerable attention from adversaries. However, when comparing generated anime images to authentic hand-drawn anime images, the details are much closer, which can reduce the likelihood of suspicion. We tested the entropy of the first image in each group. Entropy measures the amount of uncertainty or information contained in an image. Lower entropy indicates more straightforward content or more repetitive patterns, with pixel distribution more concentrated and less overall information. As shown in Table 1, the results demonstrate statistically that the generated anime images are more similar to natural paintings, which is more conducive to completing the steganography task. By introducing a random or specified seed S_2 and a designated prompt, these elements are synthesized into S_C, which enables the generator to precisely produce the image I_c, as shown in Eq. (1).

$$I_c = G(S_C) \tag{1}$$

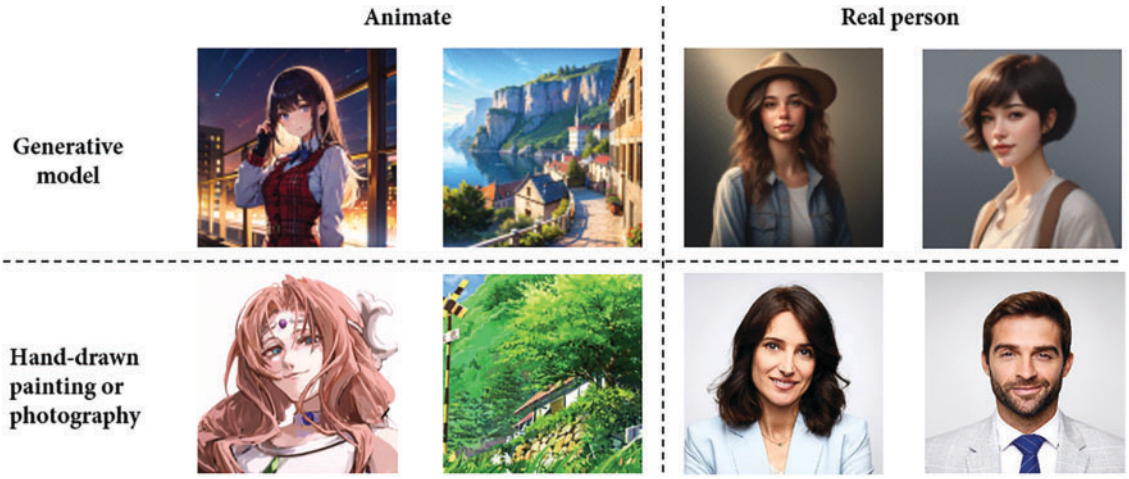

Figure 2: Comparison between generating anime images and generating real images

Table 1: Entropy of the first image in each group

Types	Animate	Real person
Generative model	[7.1162, 6.9981, 6.9451]	[4.4389, 4.2001, 3.9596]
Hand-drawn painting or photography	[6.3328, 6.9053, 6.7464]	[6.7927, 6.9064, 6.7698]

2.2 Randomly Generated Decoding Network

Convolutional neural networks (CNNs) have consistently demonstrated strong performance for image processing tasks. A CNN typically comprises two critical components: its network architecture and weight parameters. Traditional implementations involve constructing a loss function based on input-output relationships and iteratively updating the network's weights via methods like stochastic gradient descent (SGD) to minimize the loss. However, due to the massive scale of training data, the complexity of CNN architectures, and the substantial computational resources required for training—some algorithms employ up to 19 convolutional layers [22]—the secure transmission of such models may even exceed the cost associated with securely transmitting secret information. The proposed algorithm adopts a FNN as the decoder to address these challenges. Experimental validation was conducted to identify the optimal FNN architecture, as shown in Fig. 3.

The FNN network in the algorithm design resembles classical DNN [23]. The first block comprises a Pixel Unshuffle (PNS) layer, Conv layer, LeakyReLU layer [24], and InstanceNorm2d (IN) layer. The second block consists of a Conv, LeakyReLU, and IN layers. The final block includes a Conv layer, Sigmoid activation function, and PixelShuffle (PS) layer for output. The LeakyReLU and Sigmoid activation functions effectively map the secret image from feature space back to conventional image space, enhancing model performance and gradient propagation efficiency.

The IN layer calculates the mean and standard deviation for each channel of every sample and performs normalization. Compared to Batch Normalization, IN focuses more on the independence of individual samples, making it particularly suitable for processing images with diverse styles and helping the network learn finer-grained features. The Pixel Unshuffle/Shuffle layers perform tensor upsampling and downsampling, adjusting tensor dimensions for input/output compatibility. Modifying the upsampling and downsampling values allows adjustment of the embedding capacity.

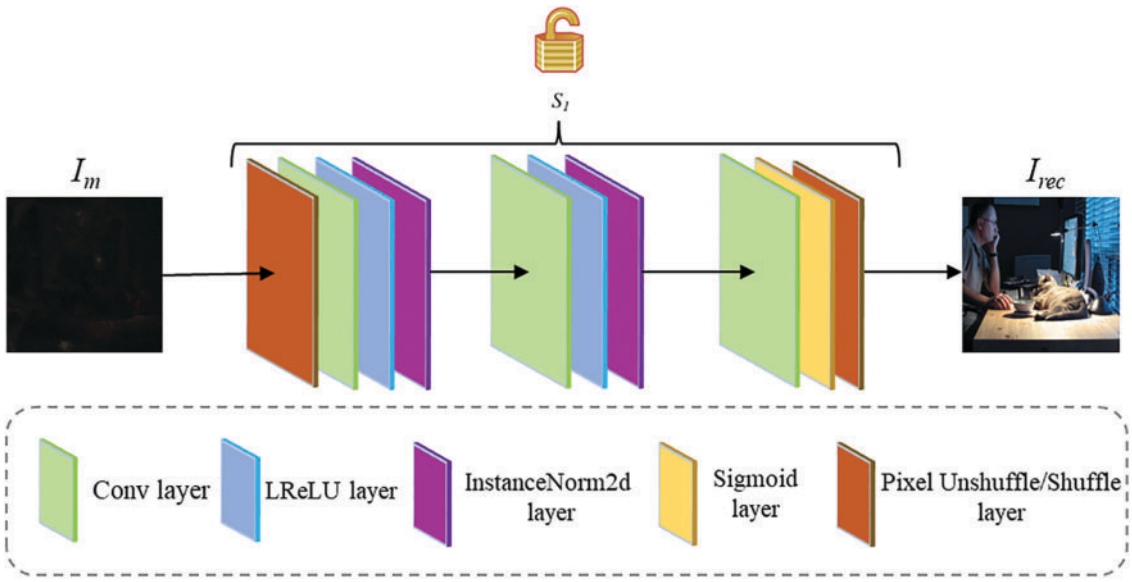

Figure 3: The optimal FNN architecture

The FNN weights are generated by S_1 and remain fixed after generation. Instead, the algorithm trains the mask image I_m to minimize the loss function between the extracted secret image I_{rec} (after FNN processing) and the original secret image I_s.

Let w denote the weights of the decoding network. The generation process of these weights can be represented by Eq. (2).

$$w = S(s_1, D) \tag{2}$$

where S is the weight setting function and D represents the architecture of the decoder network. We present the specific structure of the decoding network, as shown in Table 2.

Table 2: Provides specific parameter details of the network structure

Network layer	Channels	Convolutional kernel size	Size of the feature map	Activation function	Normalization layer	Tensor reshaping
Input	12	–	12*512*512	–	–	PUS
Conv	12	3*3	96*256*256	LReLU	IN	–
Conv	96	3*3	96*128*128	LReLU	IN	–
Conv	96	3*3	3*128*128	Sigmoid	–	PS
Output	3	–	3*256*256	–	–	–

Then, we provide the full name of some crucial acronyms in Table 3 for easier reference and reading.

Table 3: Important acronyms list

Acronyms	Full name
FNNS	Fixed Neural Network Steganography
GAN	Generative Adversarial Network
CNNs	Convolutional Neural Networks
SGD	Stochastic Gradient Descent
KI	Key Imfomation
SD	Stable Diffusion
PNS	Pixel Unshuffle
IN	InstanceNorm2d
PS	PixelShuffle

2.3 Training the Mark Image

In our algorithm, I_c denotes the cover image generated by the deep generative network G and I_s represents the secret image to be hidden. Both images are RGB three-channel. Our objective is to ensure that the stego image remains undetectable by steganalysis while maintaining high accuracy in the extracted secret image. Eq. (3) specifies the algorithm's formalized goals, including minimal interference, boundedness, accurate extractability, and resistance to steganalysis. Eq. (4) provides the expression of the loss function, where ea parameter weights each term.

$$\begin{cases} \min\limits_{I_m} d_{L1}\left(I_c + I_m, I_c\right) \\ I_c + I_m \in [0,1] \\ D_w\left(I_m\right) = I_s \\ F\left(I_c + I_m\right) = 0 \end{cases} \tag{3}$$

$$LOSS = \alpha * d_{L1}\left(I_c + I_m, I_c\right) + \beta * F\left(I_c + I_m\right) + \gamma * d_{L1}(I_s, D_w\left(I_m\right)) \tag{4}$$

In Eqs. (3) and (4), $d_{L1}\left(I_c + I_m, I_c\right)$ denotes the MSE loss between the stego and cover images. $d_{L1}(I_s, D_w\left(I_m\right))$ denotes the MSE loss between the stego and recovered images. $F\left(I_c + I_m\right)$ represents the cross-entropy between the output and label generated by the steganalysis tool. The values of α, β and γ must be adjusted based on the prompt (i.e., the content generated for the cover image). We performed ablation studies to optimize the three parameters. The value of $\beta = 10^{-5}$ was set based on the previous work [17], ensuring that the scale of the loss function for anti-steganography analysis is comparable to the other two items. We employed a grid search approach. In the first stage, a coarse search was conducted: the parameters α and γ were set within the range [0, 2] with a step size of 0.5, quickly identifying the optimal region where α is in [0.5, 1.5] and γ is in [0.5, 1]. In the second stage, the search was refined around the coarse results, using a smaller step size of 0.1, which yielded the optimal values of $\alpha = 1$ and $\gamma = 0.6$. Further, more precise searches resulted in minimal performance improvements, so we adopted these values as the parameters for subsequent experiments.

2.4 ElGamal for Ensuring Algorithm Security

ElGamal encryption [25] is a public-key cryptosystem designed based on the intractability of solving the discrete logarithm problem over finite fields. Proposed by Taher ElGamal in 1985, it is widely used in

secure communication, digital signatures, and key exchange, serving as the foundation for many modern cryptographic protocols (e.g., the Diffie-Hellman key exchange). By leveraging ElGamal encryption, the critical information KI can be securely transmitted. In Algorithm 1, we provide pseudocode demonstrating the encryption of KI (composed of S_1, S_2, and prompt) using ElGamal and the subsequent key exchange between Alice and Bob.

Algorithm 1: ElGamal Embedding ()

Input: Key Information KI, large prime p, generator g;
Output: Encrypted key information $\widetilde{KI}$; Decrypted KI
1. G = $\{1, \ldots, q-1\}$.//G is a cyclic group of order q generated by g
2.Choose x, where $x \in G$.
3. y ≡ $g^x mod p$.//Compute y
4.Bob sets (y, g, p) as the public key and x as the private key.
5.Choose k, where $(k, p-1) = 1 \&\& k \in Z$
6.$C_1 \equiv g^k mod p$, $C_2 \equiv y^k KI mod p$, $\widetilde{KI} = (C_1, C_2)$// Alice Compute C_1, C_2 as $\widetilde{KI}$
7. $KI = \frac{C_2}{C_1^x} \bmod p$.// Decryption by Bob

Notably, the implementation of ElGamal over elliptic curves combines enhanced security, computational efficiency, and optimized resource utilization. We further present the encryption method on elliptic curves:

Select an elliptic curve $E_P(a, b)$ and embed KI into a point P_{KI} on the curve. Then Choose a generator G of $E_P(a, b)$. Bob publishes $E_P(a, b)$ and G as public parameters, selects n_b as the private key, and computes the public key $P_B = n_B G$. To send P_{KI} to Bob, Alice selects a random positive integer k and generates the ciphertext using Eq. (5):

$$\widetilde{KI} = \{kG, KI + kP_B\} \tag{5}$$

Bob decrypts $\widetilde{KI}$ via Eq. (6):

$$KI + kP_B - n_B kG = KI + k(n_B G) - n_B kG = KI \tag{6}$$

An adversary attempting to recover KI from $\widetilde{KI}$ must solve for k, which requires computing the discrete logarithm of kG on the elliptic curve (i.e., determining k from G and kG). This problem is computationally infeasible, ensuring the security of KI under the ElGamal framework.

3 Experimental Results and Analysis

The experimental platform adopts PyTorch1.11.0, Cuda 11.3, programming languagePython3.8(ubuntu20.04), RTX 4090D graphics card with 24 GB of computing capacity and 18 vCPU AMD EPYC 9754 128-Core Processor. We utilized secret images from five benchmark datasets: ImageNet [26], COCO dataset [27], CelebA dataset [28], DIV2K dataset and APTOS 2019 Blindness Detection dataset. From each dataset, 1000 images were randomly selected and resized to 256 × 256 pixels. AwPainting was employed as the cover image generator, with the following prompts: Prompt-1: “1 girl”, Prompt-2: “European and American scenery” and Prompt-3: “1 girl, golden hair, sunset, tuxedo”. The generation parameters were configured as follows: sampling method DPM++2M Karras, sampling steps chose 30, resolution chose 512 × 512 pixels, CFG scale is 7, and Clip Skip is 2. Using these distinct prompts, we generated 1000 cover images.

The secret images were embedded into the cover images and extracted using our proposed method. We then conducted comprehensive evaluations, including security and feasibility analysis, quantitative metrics (PSNR, SSIM, LPIPS), and anti-steganalysis testing. The algorithm's performance was evaluated from human visual perception and machine-based detection perspectives to ensure robustness and imperceptibility.

3.1 Security and Feasibility Analysis

The primary objective and function of a steganography algorithm are to conceal secret information within a cover medium, ensuring secure and covert transmission over public channels. Thus, security is a critical evaluation metric for steganographic algorithms [29]. Unlike traditional steganography methods that modify cover images at the pixel level, this study employs a mask image to minimize alterations to the cover image. The modifications are applied to the image tensor, which offers higher security. To validate the security and feasibility of our method, we designed experiments using linear pixel value differencing operations to generate differential images between the cover and stego images, as well as between the original secret image and the extracted image. The results confirm the effectiveness of our approach, as shown in Fig. 4.

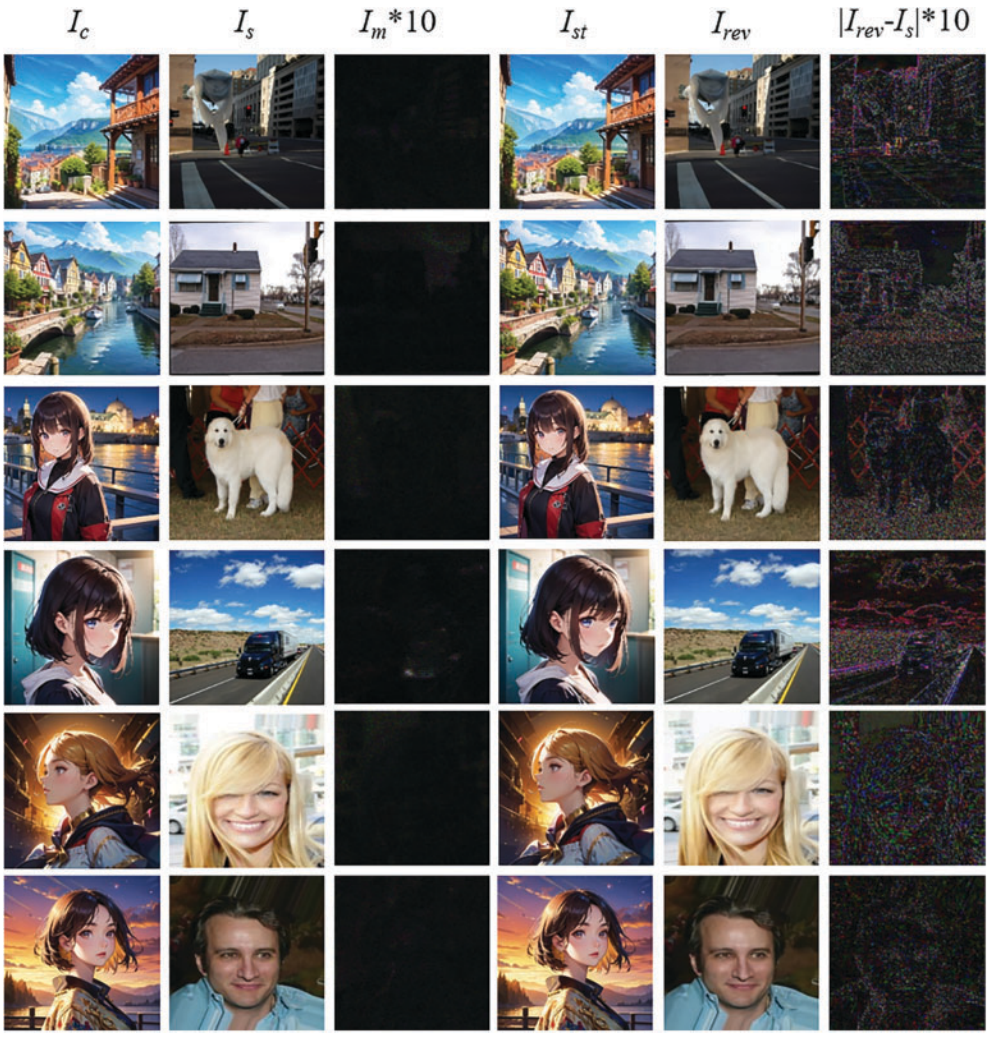

Figure 4: Residual images between secret images and recovered images

The columns in the figure, from left to right, display the cover image, secret image, the residual between the stego image and cover image (equivalent to the mask image magnified 10 times), the stego image, the recovered image, and the residual between the recovered image and secret image magnified 10 times. Experimental results demonstrate that the stego image does not leak any secret information during transmission while enabling complete recovery of the secret data, thereby validating the security and feasibility of the proposed scheme. We further compared our method with the approach in [18]. As shown in Fig. 5, our method significantly outperforms [18] in stego image quality, providing stronger evidence for our solution's superior security and feasibility.

Figure 5: Comparison of steganographic image residuals

3.2 Quantitative Experimental Analysis

Quantitative experiments numerically evaluate the performance of the steganography algorithm, with primary metrics including PSNR, SSIM, and LPIPS.

PSNR is a widely used metric for assessing image quality. It measures distortion by comparing the quality difference between a processed image and its original version, detecting subtle distortions caused by noise or algorithmic operations that may be imperceptible to the human eye. For the color images used in this

algorithm, the PSNR calculation is defined by Eq. (7) where MSE computed via Eq. (8), represents the average error between the original and processed images.

$$PSNR = 10 \cdot log10\left(\frac{255^2}{MSE}\right) \tag{7}$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(I(i) - J(i))^2 \tag{8}$$

Here, N is the total number of pixels, $I(i)$ is the pixel value of the original image, and $J(i)$ is the pixel value of the processed image.

SSIM evaluates the similarity between two images based on luminance, contrast, and structural information. Luminance compares the average brightness of the two images. Contrast quantifies the color range through standard deviation and Structural similarity is derived from normalized covariance, reflecting texture, edges, and other structural features. The calculation of SSIM is given by Eq. (9).

$$SSIM(x, y) = \frac{\left(2\mu_x\mu_y + C_1\right)\left(2\sigma_{xy} + C_2\right)}{\left(\mu_x^2 + \mu_y^2 + C_1\right)\left(\sigma_x^2 + \sigma_y^2 + C_2\right)} \tag{9}$$

where x and y represent the two images to be compared, μ_x and μ_y are the means of the images, indicating brightness, σ_x^2 and σ_y^2 are the variances of the images, indicating contrast, and σ_{xy} is the covariance, indicating structural similarity.

LPIPS is a deep learning-based metric for assessing image similarity, designed to better emulate the human visual system (HVS) in perceiving image differences. Unlike traditional metrics such as SSIM or PSNR, LPIPS extracts image features using pre-trained deep neural networks and evaluates similarity by computing distances in the feature space. We compared our method with related steganography approaches [16,18]. Tables 4 and 5 evaluate the cover and stego images, while Tables 6 and 7 assess the secret image and recovered image. In each group, the first table fixes the cover image used, and the second table fixes the prompts. In Luo et al. [16], a FNN is employed for the embedding and extraction of secret information, utilizing key-controlled perturbations generated by the FNN to embed the data. In contrast, Kishore et al. [18] capitalized on the sensitivity of neural networks to minor perturbations, achieving satisfactory results in the steganography task. However, experimental data indicate that the performance of both methods is inferior to ours when high-accuracy steganographic tasks are required. Experimental results demonstrate that our method outperforms others across all metrics. Our algorithm ensures security and undetectability during the steganographic process by utilizing generated anime-style images as cover images. Additionally, the results confirm that more complex prompts for generating cover images do not compromise the steganographic task, further enhancing the algorithm's security. Because based on the experimental data from the more complex prompt-3, the results across various metrics are close to those of the simpler prompt.

Table 4: Comparison of experimental values between the cover image and the stego image under different prompts

Methods	Prompt-1			Prompt-2			Prompt-3		
	PSNR (dB)↑	SSIM↑	LPIPS↓	PSNR (dB)↑	SSIM↑	LPIPS↓	PSNR (dB)↑	SSIM↑	LPIPS↓
Kishore et al. [18]	23.65	0.5164	0.2098	22.98	0.5361	0.2023	21.76	0.5112	0.2234
Luo et al. [16]	24.62	0.5735	0.1643	23.59	0.5915	0.1632	22.91	0.5632	0.1709
Ours	42.23	0.9768	0.0030	42.39	0.9728	0.0045	42.18	0.9716	0.0038

Table 5: Comparison of experimental values between the cover image and the stego image under different datasets

Datasets	Kishore et al. [18]			Luo et al. [16]			Ours		
	PSNR (dB)↑	SSIM↑	LPIPS↓	PSNR (dB)↑	SSIM↑	LPIPS↓	PSNR (dB)↑	SSIM↑	LPIPS↓
ImageNet	21.45	0.5244	0.2094	23.58	0.5685	0.1623	42.16	0.9718	0.0038
COCO	21.76	0.5112	0.2234	22.91	0.5632	0.1709	42.18	0.9716	0.0038
CelebA	22.36	0.5082	0.2164	23.41	0.5572	0.1859	41.93	0.9690	0.0040
DIV2K	22.53	0.5143	0.2214	23.75	0.5849	0.1736	42.12	0.9742	0.0039
APTOS 2019	20.45	0.5024	0.2316	22.86	0.5923	0.1689	40.32	0.9523	0.0042

Table 6: Comparison of experimental values between the secret image and the recovered image under different prompts

Methods	Prompt-1			Prompt-2			Prompt-3		
	PSNR (dB)↑	SSIM↑	LPIPS↓	PSNR (dB)↑	SSIM↑	LPIPS↓	PSNR (dB)↑	SSIM↑	LPIPS↓
Kishore et al. [18]	22.91	0.7084	0.1698	22.48	0.7162	0.1705	23.51	0.7895	0.1732
Luo et al. [16]	23.67	0.7165	0.2413	23.61	0.7468	0.2387	22.83	0.7435	0.2313
Ours	33.48	0.9271	0.0376	33.26	0.9248	0.0396	33.51	0.9274	0.0377

Table 7: Comparison of experimental values between the secret image and the recovered image under different datasets

Datasets	Kishore et al. [18]			Luo et al. [16]			Ours		
	PSNR (dB)↑	SSIM↑	LPIPS↓	PSNR (dB)↑	SSIM↑	LPIPS↓	PSNR (dB)↑	SSIM↑	LPIPS↓
ImageNet	22.95	0.7816	0.1765	21.62	0.7451	0.2353	33.57	0.9184	0.0442
COCO	23.51	0.7895	0.1732	22.83	0.7435	0.2313	33.51	0.9274	0.0377
CelebA	22.45	0.7996	0.2035	20.78	0.7568	0.2476	37.14	0.9497	0.0302
DIV2K	22.15	0.7689	0.1847	21.89	0.7638	0.2315	34.29	0.9288	0.0315
APTOS 2019	23.15	0.7783	0.1786	22.74	0.7789	0.1634	36.43	0.9354	0.0311

3.3 *Resistance to Steganalysis*

To evaluate the algorithm's robustness against steganalysis tools, we employed the traditional steganalysis tool StegExpose [30] and the deep learning-based steganalysis tool YeNet [31]. Each experiment utilized 3000 generated images.

As shown in Fig. 6a, the experimental results from the StegExpose are plotted as curves. The ideal curve represents random guessing, where the correct and error rates are 50%, indicating perfect deception of the steganalysis tool. Our results are notably closer to this ideal curve than the other two steganographic schemes, demonstrating superior resistance to traditional statistical steganalysis.

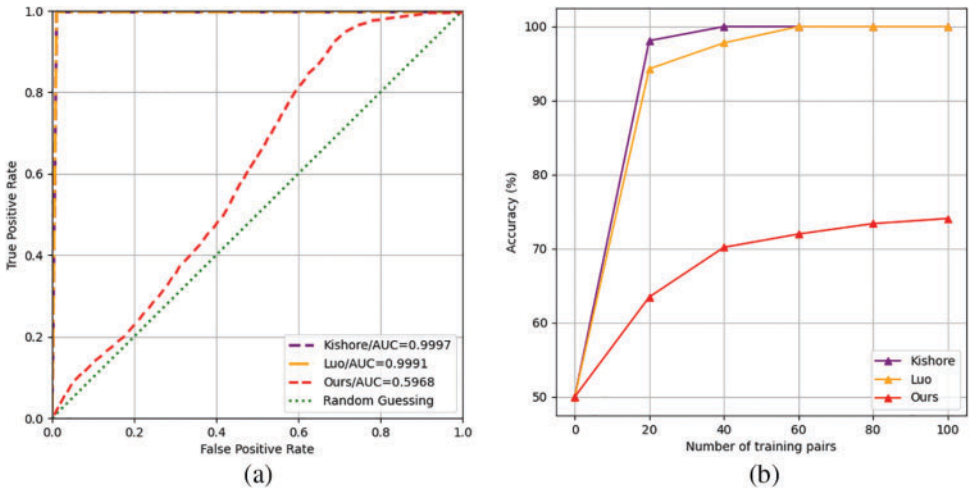

Figure 6: Comparison of experimental results for steganalysis

In Fig. 6b, YeNet's detection accuracy of all three methods increases as the training samples are progressively augmented. However, our method exhibits a slower rise in detection accuracy: the other two schemes approach 100% accuracy when trained on 20 sample pairs, while our method remains around 73% even at 100 pairs.

These experiments confirm that our scheme achieves enhanced security and undetectability against traditional statistical and deep learning-based steganalysis. In addition to the two steganalysis methods mentioned above, the most recent advanced steganalysis technique is described in reference [32]. We plan to incorporate this method in future work to evaluate the steganographic performance of our approach. This steganalysis method, which combines active learning with off-policy deep reinforcement learning (DRL), may pose significant challenges to our steganographic scheme. HSDetect-Net [33] uses specialized small-sized convolutional kernels to extract complex details and incorporates a fuzzy layer to improve classification accuracy. It performs better when handling complex textures generated by steganographic techniques. In future work, we will consider further countermeasures against these steganographic methods.

3.4 *Embedding Capacity*

Embedding capacity is an important performance criterion in image steganography. A larger embedding capacity in images of the same size indicates that more information can be transmitted. Under the premise of ensuring transmission performance, we compared our algorithm with traditional neural network image steganography [11] and reversible data hiding over encrypted images [34]. The results are shown in Table 8, indicate that our algorithm can achieve a higher embedding capacity, allowing for the transmission of the

same amount of secret information while requiring less carrier information, thereby reducing the likelihood of detection.

Table 8: Comparison of embedding capacity

Methods	Embedding capacity (bpp)
Hu et al. [11]	7.328×10^{-2}
Hua et al. [34]	4.0×10^{-1}~3.5
Our	4.0

3.5 PSNR Degradation

Our algorithm's performance declines under real-world channel conditions; therefore, we evaluated how image quality varies with the intensity of Gaussian noise. A random sample of 100 image pairs was used for testing. As shown in Table 9, when the standard deviation of the Gaussian noise is 1, it significantly impacts the masked images in our algorithm, resulting in poor quality of the recovered secret images. After the standard deviation reaches 5, recovering the original secret image becomes nearly impossible. We integrated a denoising network into the entire steganography process, which helps to recover the image to some extent after the Gaussian noise attack and improves the quality of the extracted secret image.

Table 9: PSNR variations with Gaussian noise intensity

Standard deviation	PSNR (db)	Denoising PSNR (db)
1	28.15	28.56
5	20.36	25.42
10	15.29	22.37
15	8.65	20.43

3.6 Computational Complexity

We conducted tests on the segmented runtime of the scheme. The program's runtime refers to the actual time it takes for a program to execute from start to finish (measured in seconds, milliseconds, etc.). It reflects the time required for code to run on particular hardware, operating system, programming language, and input data. As shown in Table 10, we tested the time for cover-image generation, the time for fixed neural network to generate stego-images, the encryption and decryption time for key information, and the secret image extraction time. The results indicate that compared to traditional steganographic networks that require a large amount of time for training, our method can complete the embedding and extraction processes in a very short time.

Table 10: Program's runtimes

Steganography stage	Runtime (seconds)
Cover-image generation	2.0637
Generate stego-images	7.6588
Encryption	0.0136

(Continued)

Table 10 (continued)

Steganography stage	Runtime (seconds)
Decryption	0.0004
Secret image extraction	7.6349

4 Conclusion

In this study, we propose a constructive image steganography technique that leverages the fixed neural network steganography technology introduced by Li et al. [17]. By fully leveraging the generative advantages of diffusion models and their innovative combination with security, we use the latest high-fidelity anime generation model to create cover images. We find and complete the embedding of the secret information with minimal disturbance relative to the cover image. We also design a method to securely transmit the key information for the generated cover images using the ElGamal algorithm, ensuring the security of the entire image steganography process. Compared to Li et al. [17], our method innovatively combines with cryptography, providing stronger confidentiality during transmission and usage. In comparison to Kishore et al. [18], our algorithm performs better and exhibits superior steganographic effects. A series of extensive experiments demonstrate that the proposed algorithm can effectively resist steganalysis while achieving a high extraction accuracy and better steganographic security. However, our algorithm currently suffers from limited robustness against certain noise attacks. To address this issue, future research will focus on enhancing robustness to improve fidelity under noisy channel conditions. Potential strategies include applying traditional scrambling and chaos-based techniques to promote better noise dispersion across the targeted pixel regions, and incorporating denoising neural networks to process attacked images, thereby strengthening overall robustness.

Acknowledgement: We thank all the members who have contributed to this work with us.

Funding Statement: This work was supported in part by the National Natural Science Foundation of China under Grants 62102450, 62272478 and the Independent Research Project of a Certain Unit under Grant ZZKY20243127.

Author Contributions: Conceptualization: Yixin Tang, Minqing Zhang; Experimental operation and data proofreading: Peizheng Lai, Ya Yue, Fuqiang Di; Analysis and interpretation of results: Yixin Tang, Minqing Zhang, Fuqiang Di; Draft Manuscript preparation: Yixin Tang, Peizheng Lai, Ya Yue; Figure design and drawing: Yixin Tang. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Due to the fact that the source code and data include research findings that our experimental team has not yet made public, we are currently unable to share them openly. Additionally, our academic institution is bound by confidentiality protocols that require us to disclose the source code and data only after the designated decryption period has elapsed.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Simmons GJ. The prisoners' problem and the subliminal channel. In: Advances in cryptology. Boston, MA, USA: Springer; 1984. p. 51–67. doi:10.1007/978-1-4684-4730-9_5.
2. Yang CH, Weng CY, Wang SJ, Sun HM. Adaptive data hiding in edge areas of images with spatial LSB domain systems. IEEE Trans Inf Forensics Secur. 2008;3(3):488–97. doi:10.1109/TIFS.2008.926097.

3. Holub V, Fridrich J. Designing steganographic distortion using directional filters. In: 2012 IEEE International Workshop on Information Forensics and Security (WIFS); 2012 Dec 2–5; Costa Adeje, Spain. p. 234–9. doi:10.1109/WIFS.2012.6412655.
4. Pevný T, Filler T, Bas P. Using high-dimensional image models to perform highly undetectable steganography. In: Information hiding. Berlin/Heidelberg, Germany: Springer; 2010. p. 161–77. doi:10.1007/978-3-642-16435-4_13.
5. Chen WY. Color image steganography scheme using set partitioning in hierarchical trees coding, digital Fourier transform and adaptive phase modulation. Appl Math Comput. 2007;185(1):432–48. doi:10.1016/j.amc.2006.07.041.
6. Cox IJ, Kilian J, Leighton FT, Shamoon T. Secure spread spectrum watermarking for multimedia. IEEE Trans Image Process. 1997;6(12):1673–87. doi:10.1109/83.650120.
7. Puech W, Chaumont M, Strauss O. A reversible data hiding method for encrypted images. In: Security, forensics, steganography, and watermarking of multimedia contents X. San Jose, CA, USA: SPIE; 2008. p. 68191E. doi:10.1117/12.766754.
8. Ma K, Zhang W, Zhao X, Yu N, Li F. Reversible data hiding in encrypted images by reserving room before encryption. IEEE Trans Inf Forensics Secur. 2013;8(3):553–62. doi:10.1109/TIFS.2013.2248725.
9. Zhang X. Reversible data hiding in encrypted image. IEEE Signal Process Lett. 2011;18(4):255–8. doi:10.1109/LSP.2011.2114651.
10. Zhou J, Sun W, Dong L, Liu X, Au OC, Tang YY. Secure reversible image data hiding over encrypted domain via key modulation. IEEE Trans Circuits Syst Video Technol. 2016;26(3):441–52. doi:10.1109/TCSVT.2015.2416591.
11. Hu D, Wang L, Jiang W, Zheng S, Li B. A novel image steganography method via deep convolutional generative adversarial networks. IEEE Access. 2018;6:38303–14. doi:10.1109/access.2018.2852771.
12. Shumeet B. Hiding images in plain sight: deep steganography. Adv Neural Inform Process Syst. 2017;30:2066–76.
13. Rehman A, Rahim R, Nadeem S, Hussain S. End-to-end trained cnn encoder-decoder networks for image steganography. In: Proceedings of the European Conference on Computer Vision (ECCV) Workshops; 2018 Sep 8–14; Munich, Germany. p. 723–9. doi:10.1007/978-3-030-11018-5_64.
14. Zhang CN, Benz P, Karjauv A, Sun G, Kweon IS. Udh: universal deep hiding for steganography, watermarking, and light field messaging. Adv Neural Inf Process Syst. 2020;33:10223–34.
15. Jing J, Deng X, Xu M, Wang J, Guan Z. HiNet: deep image hiding by invertible network. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV); 2021 Oct 10–17; Montreal, QC, Canada. p. 4713–22. doi:10.1109/ICCV48922.2021.00469.
16. Luo Z, Li S, Li G, Qian Z, Zhang X. Securing fixed neural network steganography. In: Proceedings of the 31st ACM International Conference on Multimedia. Ottawa, ON, Canada: ACM; 2023. p. 7943–51. doi:10.1145/3581783.3611920.
17. Li G, Li S, Qian Z, Zhang X. Cover-separable fixed neural network steganography via deep generative models. In: Proceedings of the 32nd ACM International Conference on Multimedia. Melbourne, VIC, Australia: ACM; 2024. p. 10238–47. doi:10.1145/3664647.3680824.
18. Kishore V, Chen XY, Wang Y, Li BY, Weinberger KQ. Fixed neural network steganography: train the images, not the network. In: International Conference on Learning Representations; 2022 Apr 25–29; Washington, DC, USA.
19. Luo T, Zhou Y, He Z, Jiang G, Xu H, Qi S, et al. StegMamba: distortion-free immune-cover for multi-image steganography with state space model. IEEE Trans Circuits Syst Video Technol. 2025;35(5):4576–91. doi:10.1109/TCSVT.2024.3515652.
20. Zhou Y, Luo T, He Z, Jiang G, Xu H, Chang CC. CAISFormer: channel-wise attention transformer for image steganography. Neurocomputing. 2024;603(2):128295. doi:10.1016/j.neucom.2024.128295.
21. Ho J, Saharia C, Chan W, Fleet DJ, Norouzi M, Salimans T. Cascaded diffusion models for high fidelity image generation. J Mach Learn Res. 2022;23(1):2249–81. doi:10.2139/ssrn.4960907.
22. Li G, Li S, Luo Z, Qian Z, Zhang X. Purified and unified steganographic network. In: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2024 Jun 16–22; Seattle, WA, USA. p. 27559–68. doi:10.1109/CVPR52733.2024.02603.
23. Zhang K, Zuo W, Zhang L. FFDNet: toward a fast and flexible solution for CNN-based image denoising. IEEE Trans Image Process. 2018;27(9):4608–22. doi:10.1109/TIP.2018.2839891.

24. He K, Zhang X, Ren S, Sun J. Delving deep into rectifiers: surpassing human-level performance on ImageNet classification. In: 2015 IEEE International Conference on Computer Vision (ICCV); 2015 Dec 7–13; Santiago, Chile. p. 1026–34. doi:10.1109/ICCV.2015.123.
25. Elgamal T. A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Trans Inf Theory. 1985;31(4):469–72. doi:10.1109/TIT.1985.1057074.
26. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, et al. ImageNet large scale visual recognition challenge. Int J Comput Vis. 2015;115(3):211–52. doi:10.1007/s11263-015-0816-y.
27. Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, et al. Microsoft COCO: commonobjects in context. In: Computer Vision—ECCV 2014: 13th European Conference; 2014 Sep 6–12; Zurich, Switzerland. p. 740–55. doi:10.1007/978-3-319-10602-1_48.
28. Liu Z, Luo P, Wang X, Tang X. Deep learning face attributes in the wild. In: 2015 IEEE International Conference on Computer Vision (ICCV); 2015 Dec 7–13; Santiago, Chile. p. 3730–8. doi:10.1109/ICCV.2015.425.
29. Zhang X, Zhang M, Wang XA, Huang S, Di F. Constructive robust steganography algorithm based on style transfer. Comput Mater Continua. 2024;81(1):1433–48. doi:10.32604/cmc.2024.056742.
30. Boehm B. Stegexpose—A tool for detecting LSB steganography. arXiv:1410.6656. 2014.
31. Ye J, Ni J, Yi Y. Deep learning hierarchical representations for image steganalysis. IEEE Trans Inf Forensics Secur. 2017;12(11):2545–57. doi:10.1109/TIFS.2017.2710946.
32. Li B, Li N, Yang J, Alfarraj O, Albelhai F, Tolba A, et al. Image steganalysis using active learning and hyperparameter optimization. Sci Rep. 2025;15(1):7340. doi:10.1038/s41598-025-92082-w.
33. de la Croix NJ, Ahmad T, Han F, Ijtihadie RM. HSDetect-net: a fuzzy-based deep learning steganalysis framework to detect possible hidden data in digital images. IEEE Access. 2025;13:43013–27. doi:10.1109/access.2025.3546510.
34. Hua Z, Liu X, Zheng Y, Yi S, Zhang Y. Reversible data hiding over encrypted images via preprocessing-free matrix secret sharing. IEEE Trans Circuits Syst Video Technol. 2024;34(3):1799–814. doi:10.1109/TCSVT.2023.3298803.