JPEG Compression-Resistant Generative Image Hiding Utilizing Cascaded Invertible Networks

Tiewei Qin, Bingwen Feng, Bingbing Zhou, Jilian Zhang, *Member, IEEE*, Zhihua Xia, *Member, IEEE*, Jian Weng, *Senior Member, IEEE*, and Wei Lu, *Member, IEEE*

***Abstract*—Generative steganography is renowned for its exceptional undetectability. However, prevalent generative methods often have insufficient capacity for concealing secret images. Furthermore, the sensitivity of commonly utilized generative models exacerbates the challenge of ensuring robustness against channel distortions such as JPEG compression. In this paper, we introduce a generative image hiding network that employs two invertible generators to transform secret images into stego images within a disparate image domain. Additionally, we seamlessly integrate an up-and-down sampling module (UDM) within these generators to facilitate efficient decoupling of the intermediate representations obtained by each generator. The UDM serves multiple purposes: preserving coherence between the intermediate representations, enhancing resilience against JPEG compression, and safeguarding the confidentiality of the concealed images. To address the complexity of mapping both uncompressed and compressed stego images to a unified intermediary representation, we implement two distinct flows for the forward and backward processes of the generator associated with the stego images. The experimental results show that our scheme offers concurrent advantages in terms of full-size image hiding ability, undetectability, confidentiality, and robustness.**

***Index Terms*—Generative steganography, invertible network (INN), robustness, undetectability, confidentiality.**

I. Introduction

IMAGE steganography is a covert communication technique that involves hiding secret information in images. Secret information is hidden into stego images, which should look innocent so as not to arouse the suspicion of third parties. The essential properties of steganographic schemes include capacity, security, and robustness.

Secret information can be binary sequences, images, or other types of data. Compared with embedding binary sequences, embedding images is more challenging because of their high capacity requirement. Baluja [1] proposed the first deep learning-based steganography scheme that hides an image within another image. Since then, numerous techniques have emerged for image hiding [2], [3], [4]. These methods frequently utilize an encoder-decoder architecture to embed and subsequently recover the secret image. StegFormer [5] suggested an autoencoder-based steganographic model and presented a normalizing training strategy and a restricted loss to enhance reliability under realistic conditions. An invertible neural network (INN) [6] can construct an invertible mapping between two distributions, which benefits its application in image steganography [7], [8], [9], [10]. Lu et al. [7] regarded the embedding and extraction of secret images as a pair of inverse problems. In the embedding stage, the secret images and the cover images are input into an invertible network to obtain stego images by forward propagation. In the extraction stage, the stego images and the constant matrix are input to recover the secret images by backward propagation. Another approach, HiNet [8], inputs stego images and random vectors that follow a Gaussian distribution for backward propagation. Furthermore, it hides the secret image in the wavelet domain to improve its invisibility. In RIIS [9], the redundant high-frequency component is modeled with the aid of conditional normalizing flow to reduce information loss in backward propagation. DeepMIH [10] proposes a multiple image hiding framework that is based on an invertible neural network. These INN-based schemes have presented good performance in terms of the quality of stego and recovered secret images.

However, the aforementioned schemes require modification of the cover image. This inevitably results in deviation of the stego distribution, which reduces security [11], [12], [13]. To solve this problem, generative steganography transforms the secret information directly into a generated image, thus avoiding the operation of modification. Therefore, it can resist steganography analysis tools well. Wei et al. [14] proposed a generative steganography network (GSN) that is based on StyleGan2 [15]. PARIS [16] maps a binary message to a latent vector according to a standard Gaussian distribution. Liu et al. [17] proposed a deniable carrier-free generative steganography method that is based on diffusion models [18]. Wei et al. [19] proposed a generative steganographic flow (GSF) that is based on Glow [20]. Zhou et al. [21] encoded a secret

Received 22 July 2024; revised 1 February 2025, 11 April 2025, and 23 May 2025; accepted 4 June 2025. Date of publication 10 June 2025; date of current version 20 June 2025. This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFB3103100; in part by the National Natural Science Foundation of China under Grant 62472199, Grant 62441237, Grant 62472197, Grant 62261160653, Grant 62020106013, Grant U23B2023, and Grant 62122032; in part by the Natural Science Foundation of Guangdong Province, China, under Grant 2025A1515011601; in part by Guangdong Key Laboratory of Data Security and Privacy Preservation under Grant 2023B1212060036; and in part by the Opening Project of the Ministry of Education (MoE) Key Laboratory of Information Technology (Sun Yat-sen University) under Grant 2024ZD001. The associate editor coordinating the review of this article and approving it for publication was Dr. Yansong Gao. *(Corresponding author: Bingwen Feng.)*

Tiewei Qin, Bingwen Feng, Bingbing Zhou, Jilian Zhang, and Jian Weng are with the College of Cyber Security, Jinan University, Guangzhou 510632, China (e-mail: qtiewei@stu2021.jnu.edu.cn; bingwfeng@gmail.com; zbbing0908@163.com; zhangjilian@jnu.edu.cn; cryptjweng@gmail.com).

Zhihua Xia is with the College of Cyber Security, Engineering Research Center for Trustworthy AI, Ministry of Education, Jinan University, Guangzhou 510632, China (e-mail: xia_zhihua@163.com).

Wei Lu is with the School of Computer Science and Engineering, Ministry of Education Key Laboratory of Information Technology, Guangdong Province Key Laboratory of Information Security Technology, Sun Yat-sen University, Guangzhou 510006, China (e-mail: luwei3@mail.sysu.edu.cn).

Digital Object Identifier 10.1109/TIFS.2025.3578231

message as the position arrangement of the elements in the latent representation. These schemes are able to hide binary sequences in generated images. However, they do not provide enough capacity to hide secret images. Li et al. [22] hid full-size secret images during cross-domain image transformation. Zhang et al. [23] hid secret images within stylized images. However, both of these schemes alter the distribution of latent variables, thereby potentially leading to detection by specific steganalytic tools.

Moreover, compression loss is commonly observed in public channels. Therefore, robustness to this type of attack is critical for practical steganographic schemes. Many steganographic schemes use a simulated perturbation layer for this purpose. RIIS [9] uses a container enhancement module (CEM) to enhance robust reconstruction. ABDH [24] hides secret images in inconspicuous regions through an attentional mechanism while using adversarial training to improve robustness. Zhang et al. [3] proposed a deep adaptive hiding network that extracts frequency information from secret and cover images and fuses this information incrementally to improve robustness. However, hiding secret images requires steganography schemes to have a large capacity, which makes improving robustness difficult. Furthermore, flow-based steganographic schemes tend to be more vulnerable to intermediate distortion because of the dependence on inherent invertible bijective transformation properties [9], [21].

For generative steganography, several schemes can achieve robustness. PARIS [16] uses gradient descent optimization to increase robustness. Zhou et al. [21] used the elements nearest to the center of each group and suggested the idea of separate encoding to increase robustness. Sun et al. [25] embedded messages in guidance features that were used in image synthesis. Zhou et al. [26] encoded secret messages as the object contours of the stego image. Owing to the resistance of these semantic features, these two schemes present high robustness against various image attacks. However, redundancy of the embedded information is necessary for these schemes to achieve robustness. This inevitably reduces their capacity, thus making them unsuitable for hiding images. Both CRoSS [27] and DiffStega [28] use conditional diffusion model-based image translation to hide an image within a synthesized image in a robust manner. Nevertheless, these methods encounter difficulties in concealing the comprehensive contents of the secret images and maintain a noticeable correlation between the secret and stego images. The hierarchical framework in [29] amalgamates a diffusion model with a flow model. However, it still relies on cover modification in its second phase. Improving the capacity of robust generative steganography remains a challenge.

The proposed generative steganographic scheme, which is inspired by the impressive achievements of INNs in image hiding applications, uses a novel approach to securely and resiliently conceal images. By incorporating two Glow-based generators, $G1$ and $G2$, the scheme establishes mappings between a secret image and generated images that may belong to diverse image domains. Each generator crafts a connection between an intermediate distribution and its respective image domain. In the context of $G2$, distinct flows are utilized for the forward and backward processes. In the forward process, a flow that mirrors Glow is employed, whereas a separate, dedicated flow is established for the compressed stego images in the backward process. This strategy ensures that the stego images are not only of high quality but also robust. Furthermore, an up-and-down sampling module (UDM) is introduced to facilitate random mapping between two intermediate distributions while bolstering both robustness and confidentiality. The main contributions of this paper are summarized as follows:

1) We propose a robust generative steganography method that is based on cascaded INNs. By incorporating a random mapping between two INNs, the correlation between the secret and stego images is effectively decoupled, thereby bolstering the security of the system.
2) In $G2$, we use distinct flows to overcome the challenge of asymmetric mapping of both uncompressed and compressed images onto the same intermediate variable. It creates a dedicated flow specifically for compressed stego images, thereby ensuring precise mapping to the intermediate variables.
3) We suggest the UDM for maintaining consistency across different generators. Additionally, this module enhances the robustness of the proposed scheme and guarantees that unauthorized users are unable to recover the secret image.

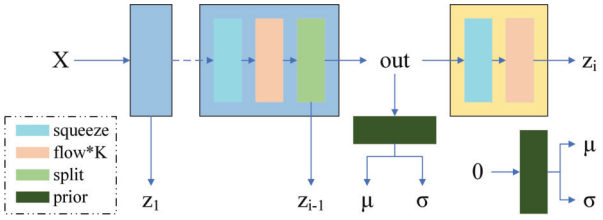


Fig. 1. Structure of Glow. Z is a normally distributed variable. μ and σ are the mean and standard deviation of the prediction, respectively. $\mathbf{0}$ denotes a matrix in which all the elements are zeros.

II. Related Work: Normalizing Flow-Based Model

Normalizing flow [6], [20], [30], which is a type of invertible generative model, is distinguished by its ability to recover the original variable x directly through its inverse function $f^{-1}(z)$ given a transformation function $z = f(x)$. Notably, both $f(x)$ and $f^{-1}(z)$ share the same set of model parameters. This flow-based approach accomplishes a bijective mapping from a complex distribution $p_X(x)$ to a simple distribution $p_Z(z)$ through maximum likelihood estimation. Dinh et al. [6] first proposed the NICE model. Subsequent models, such as RealNVP [30] and Glow [20], have realized improved performance.

The Glow model is renowned for its generative capabilities, thus making it a natural choice for our proposed framework. Its structure is shown in Fig. 1. Within the model, each block incorporates a prior layer that is tasked with predicting the mean μ and standard deviation σ of a normal distribution. These parameters subsequently contribute to the computation of the loss function. The loss function, designated L_{Glow}, is formulated as follows:

$$\begin{aligned} L_{\text{Glow}} &= \log p_X(x) \\ &= \log p_Z(z) \left| det \frac{\partial z}{\partial x} \right| \\ &= \log p_Z(z) + \log \left| det \frac{\partial f(x)}{\partial x} \right| \end{aligned} \tag{1}$$

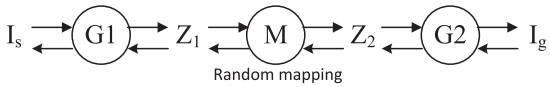


Fig. 2. Overall framework of the proposed scheme.

where f is a bijective mapping function learned by the Glow model and where $p_Z(z)$ is a distribution with a simple and tractable density, such as a normal distribution:$p_Z(z) = \mathcal{N}(z;\mu,\sigma)$.

Notably, there are generative steganographic schemes that leverage the Glow framework [19], [21]. However, these schemes are tailored for binary message sequences, thus differing from our proposed approach.

III. Motivation

Generative steganography uses secret information to directly craft a stego medium. In our context, an irrelevant image is crafted on the basis of a secret image, which remains concealed within it. Essentially, this process involves finding an image translation between two distinct domains. The source domain, which houses the secret images, remains arbitrary, whereas the target domain, which represents the stego images, can be specified and often differs significantly from the source. The content, style, structure, and other pertinent aspects of each secret image must be completely decoupled from those of the stego image.

Two generative models can be utilized to align two distinct image domains [31], [32]. They can attain exact cycle consistency $G1(G2^{-1}(G2(G1^{-1}(I_1)))) \approx I_1$, where the two generative models are respectively defined in $\mathcal{I}_1$ and $\mathcal{I}_2$ as

$$G1 : Z \to I_1, I_1 \in \mathcal{I}_1 \tag{2}$$

$$G2 : Z \to I_2, I_2 \in \mathcal{I}_2 \tag{3}$$

Herein, the forward process of $G1$ signifies the translation from the intermediate variable Z to the image I_1, whereas the backward process, $G1^{-1}$, indicates the reverse translation. These methods preserve large amounts of mutual information $I[\mathcal{I}_1,\mathcal{I}_2]$ within the shared latent representation Z. However, when utilized for steganography, this preserved mutual information may lead to a high degree of pertinence between the secret and stego images, thereby potentially leaving traces that steganalytic tools can detect.

To fully decouple the pertinence between the secret and stego images, we introduce a random mapping between $G1$ and $G2$. Specifically, $G1$ transforms the secret image $I_s \in \mathcal{I}_s$ into an intermediate variable $Z_1 \in \mathcal{Z}_1$, whereas $G2$ maps the generated image $I_g \in \mathcal{I}_g$ to another intermediate variable $Z_2 \in \mathcal{Z}_2$. An intermediate mapping function, M, is then utilized to randomly map samples from Z_1 to samples in Z_2. The overall framework is schematically depicted in Fig. 2. With this framework, the generation of stego images can be achieved by

$$I_g = G2(M(G1^{-1}(I_s))) \tag{4}$$

and the secret image can be recovered by

$$I_s = G1(M^{-1}(G2^{-1}(I_g))) \tag{5}$$

Moreover, our steganographic scheme must possess the following crucial characteristics to ensure its practical applicability in image hiding:

(1) **Security.** Security consists of undetectability and confidentiality. Undetectability ensures that the generated stego images remain undetectable even by the most advanced steganalytic tools, and confidentiality demands that unauthorized users be unable to retrieve the secret image, regardless of their access to the network. Consequently, a secret key is imperative for the recovery of the secret image.

(2) **Robustness.** Stego images are typically transmitted over public channels and are often subject to JPEG compression. Therefore, the primary content of the concealed image remaining recoverable even after the stego images have undergone compression is crucial.

(3) **High image quality.** Both the generated stego images and the recovered secret images must maintain a high level of image quality. This requires that all the mapping rules employed within the scheme minimally disrupt the distributions of $\mathcal{Z}_1$ and $\mathcal{Z}_2$; otherwise, the stego images will deviate from normally generated images or the recovered images will suffer from degraded quality.

Notably, the flow model is sensitive to distortion [9]. Consequently, the intermediate mapping M serves as an important means for ensuring robustness. Furthermore, to ensure this robustness, crucially, both Z_2 and its perturbed counterpart must consistently map to the identical Z_1 through a certain error correction methodology. This constitutes a many-to-one mapping. Consequently, the entropies of these two intermediary distributions adhere to the inequality $H[\mathcal{Z}_1] < H[\mathcal{Z}_2]$ [33]. This, in turn, suggests that $H[\mathcal{I}_s] < H[\mathcal{I}_g]$. Intriguingly, our ultimate goal is to embed a secret image within a generated image of equivalent dimensions, which presents an apparent paradox. To resolve this seeming contradiction, we initially downsample I_s prior to its input into G_1 and subsequently upsample the recovered $\tilde{I}_s$ once it has been output by G_2.

IV. Proposed Method

A. Overview

As illustrated in Fig. 3, the proposed scheme comprises four components: two Glow-based generators, $G1$ and $G2$; an up-and-down sampling module (UDM); and an image superresolution module (ISM). $G1$ is responsible for the translation between $\mathcal{I}_s$ and $\mathcal{Z}_1$, whereas $G2$ handles the translation between $\mathcal{Z}_2$ and $\mathcal{I}_g$. The UDM implements a robust and random mapping between $\mathcal{Z}_1$ and $\mathcal{Z}_2$ while maintaining their distribution consistency. The ISM is responsible for scaling the recovered image back to its original size.

During the hiding phase, the proposed scheme transforms the secret image $I_s \in [0,1]^{3\times h\times w}$ into a stego image $I_g \in [0,1]^{3\times h\times w}$. I_s first undergoes downsampling to yield an image of reduced size $3\times h/2\times w/2$. This downsampled image is then fed into the backward path of $G1$ to derive an intermediate variable $Z_1 \in \mathbb{R}^{3\times h/2\times w/2}$. Subsequently, Z_1 is mapped to $Z_2 \in \mathbb{R}^{3\times h\times w}$ using the UDM. Finally, Z_2 is processed by the forward path of $G2$ to generate the stego image I_g.

In the recovery procedure, the stego image has been transmitted over the public channel and potentially compressed as $\tilde{I}_g$. This received image $\tilde{I}_g$ is fed into the backward path of $G2$, which produces the intermediate variable $\tilde{Z}_2 \in \mathbb{R}^{3\times h\times w}$. Subsequently, $\tilde{Z}_2$ is mapped to $\tilde{Z}_1 \in \mathbb{R}^{3\times h/2\times w/2}$ using UDM. Finally, $\tilde{Z}_1$ sequentially traverses through $G1$ and the ISM, thus yielding the recovered image $\tilde{I}_s$.

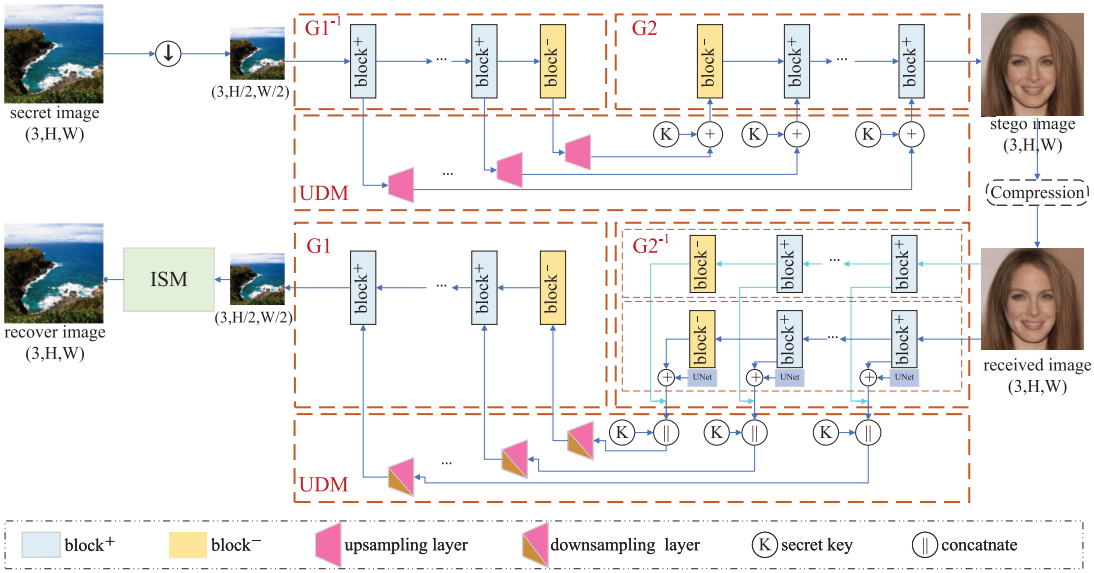


Fig. 3. Architecture of the proposed scheme.

B. Generator $G1$

$G1$ shares the same structure and loss function as Glow [20]. Given that the secret images can be arbitrary, $G1$ can be trained on any dataset. In our proposed scheme, we demonstrate the use of the COCO dataset [34]. Notably, $G1$ is trained solely beforehand, with the objective of achieving high-quality reconstruction of the original input image by independent and identically distributed (i.i.d.) Gaussian sampling.

C. Generator $G2$

We recall that the forward process of $G2$ transforms the generated Z_2 into the stego image I_g, whereas its inverse, $G2^{-1}$, reverses this mapping. The input image for the inverse process may have undergone JPEG compression. However, the inherent bijectivity of INNs poses a challenge for $G2^{-1}$ in accurately mapping both uncompressed and compressed images to identical intermediate variables. To address this issue, we employ two distinct flows for executing the forward and inverse processes.

The forward process, $G2$, adopts a flow identical to that of the Glow model, which ensures that the appearance of the resulting stego image aligns consistently with the appearance of the innocent generated images. For the inverse process, we devise two flows: $G2^{-1}_{(1)}$ and $G2^{-1}_{(2)}$. The selection of which flow to use depends on the nature of the input image. Specifically, for uncompressed input images, $G2^{-1}_{(1)}$ utilizes the same flow as in the forward process. In contrast, for compressed input images $\tilde{I}_g$, we establish a separate flow $G2^{-1}_{(2)}$ to map them back to Z_2. It uses a distortion-guided INN similar to that in RIIS [9], which accepts the QF as an input condition feature. Additionally, a U-Net branch is incorporated to facilitate mapping from images that have undergone compression at diverse levels to the same intermediate representation. Figure 4 illustrates the structure of $G2^{-1}$.

$G2^{-1}_{(2)}$ undergoes independent training, with its INN branch and U-Net branch being trained separately. Both branches are optimized using the loss

$$L_{\text{Grbst}} = \left\| Z_2 - G2^{-1}_{(2)}(\tilde{I}_g) \right\|_1 \tag{6}$$

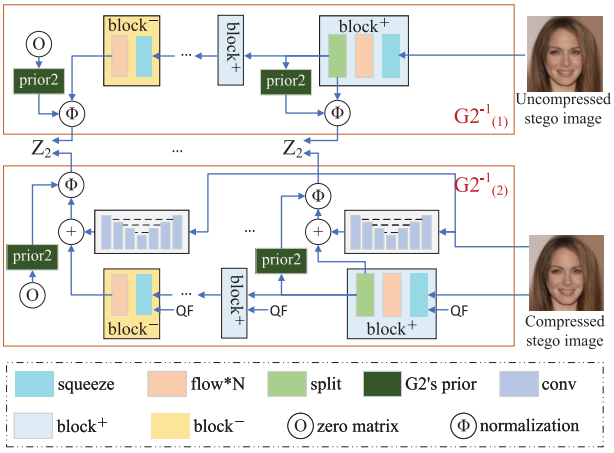


Fig. 4. Structure of $G2^{-1}$.

The INN branch of $G2^{-1}_{(2)}$ is initialized with the parameters $G2^{-1}_{(1)}$. This strategy leverages the pretrained model to expedite the training of $G2^{-1}_{(2)}$. The training process for $G2$ and its inverse processes is outlined in Algorithm 1.

D. Up-and-Down Sampling Module (UDM)

Since $G1$ and $G2$ are trained separately, the mapped intermediate distributions $\mathcal{Z}_1$ and $\mathcal{Z}_2$ may adhere to different normal distributions. Consequently, when the intermediate variable traverses different generators, it requires renormalization to align with the target distribution. Furthermore, the size of $Z_1 \in \mathcal{Z}_1$ is only half the size of $Z_2 \in \mathcal{Z}_2$. Therefore, resizing the intermediate variable becomes necessary during its transition from one generator to another. Given these considerations, an up-and-down sampling module (UDM) is proposed for maintaining the consistency of intermediate variables across $G1$ and $G2$.

Furthermore, the UDM functions as a random mapping to decouple the pertinence between $\mathcal{Z}_1$ and $\mathcal{Z}_2$. Recognizing the

Algorithm 1 Training of $G2$

Input: Normally distributed variable Z_2, Images I.
Output: The trained $G2$, $G2^{-1}_{(1)}$, and $G2^{-1}_{(2)}$.

1: Optimize $G2$ by minimizing L_{Glow} using Z_2 and I.
2: Share the parameters of $G2$ and $G2^{-1}_{(1)}$.
3: Initialize the INN branch of $G2^{-1}_{(2)}$ using the parameters of $G2^{-1}_{(1)}$.
4: Bypass the U-Net branch of $G2^{-1}_{(2)}$, allowing the model to focus solely on refining the INN branch.
5: JPEG Simulation: $\tilde{I} \leftarrow \text{JPEG}(I)$ with QF $\in \{40, 50, 60, 70, 80, 90\}$.
6: Optimize the INN branch by minimizing L_{Grbst} using $\tilde{I}$ and QF.
7: Introduce the U-Net branch, and freeze the INN branch.
8: Optimize the U-Net branch by minimizing L_{Grbst} using $\tilde{I}$.

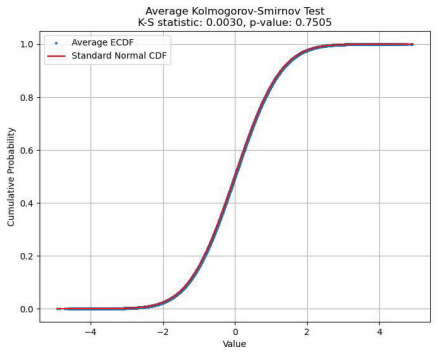


Fig. 5. Results of the averaged Kolmogorov-Smirnov test on Z_2.

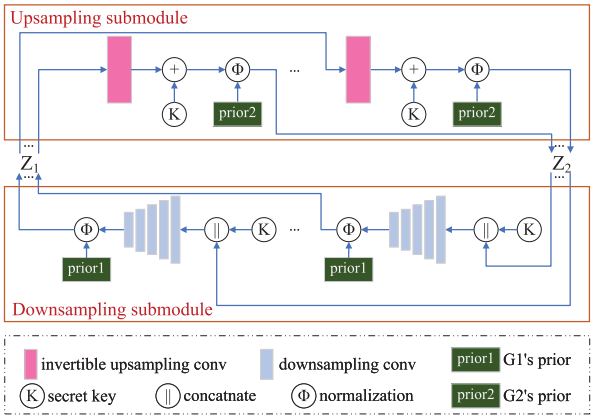


Fig. 6. Structure of the UDM.

importance of safeguarding Z_1 from unauthorized access, the UDM randomizes the mapping by adding a secret key $K \in \mathbb{R}^{3\times h/2\times w/2}$ to Z_1 before its input into $G2$. K is a pseudorandom sequence that follows the standard normal distribution, which ensures that the encrypted variable also follows a normal distribution. Conversely, when the encrypted variable needs to be input into $G1$, the UDM decrypts it using the same secret key K.

The UDM comprises two submodules, namely, an upsampling submodule and a downsampling submodule, as illustrated in Fig. 6. The former is employed during the hiding phase, whereas the latter is utilized in the recovery phase. The upsampling submodule doubles the size of Z_1 to increase its redundancy to tolerate potential channel distortion. An encryption is subsequently performed using K. This process can be expressed mathematically as

$$Z_2 = \text{UP}(Z_1) + K \tag{7}$$

where UP() denotes the upsampling operation.

Notably, the original Glow architecture requires Z_2 to follow an i.i.d. Gaussian distribution. However, this i.i.d. property fundamentally conflicts with robustness requirements, as the zero mutual information between entries in Z_2 eliminates essential redundancy for error correction. To address this limitation, our solution strategically trades independence for spatial redundancy while maintaining the Gaussian property of the distribution. We implement this through an invertible upsampling layer adapted from [35]. First, the input channel dimension is quadrupled by replicating Z_1 four times. Then, a Gaussian-distributed random bias N_b is introduced to mitigate deviations from the ideal i.i.d. property. This upsampling operation can be expressed as follows:

$$\text{UP}(Z_1) = \hat{A} \cdot (\text{Vec}(Z_1)||\text{Vec}(Z_1)||\text{Vec}(Z_1)||\text{Vec}(Z_1) + N_b) \tag{8}$$

where $\hat{A}$ is an orthogonal block diagonal matrix and where Vec() represents an appropriate vectorization operation that reshapes the concatenated inputs into a column vector. Given that the width of the subblock in $\hat{A}$ is significantly smaller than the length of $\text{Vec}(Z_1)$, Equation (8) can be considered a weighted sum of multiple Z_1 samples, thereby preserving the Gaussian property of the output. After normalization using μ and σ from the original Glow model's input distribution, the resulting Z_2 becomes statistically aligned with the Glow model's requirements while crucial spatial redundancy is incorporated. We further employ the Kolmogorov-Smirnov test to evaluate the distribution of Z_2. The experimental results averaged over 100 secret images are shown in Fig. 5. These results indicate that Z_2 passes the test with a p value of nearly 0.7505.

The downsampling submodule reconstructs $\tilde{Z}_1$ from $\tilde{Z}_2$ obtained from $G2^{-1}(\tilde{I}_g)$. This submodule incorporates both decryption and downsampling. Although decryption might appear reducible to a basic subtraction operation in linear systems, the nonlinear nature of $G2$ introduces complications from channel distortion N_c. This is formally evident from

$$G2^{-1}\left(\tilde{I}_g\right) \tag{9}$$
$$= G2^{-1}\left(G2\left(\text{UP}(Z_1) + K\right) + N_c\right) \tag{10}$$
$$\neq G2^{-1}\left(G2\left(\text{UP}(Z_1) + K\right)\right) + G2^{-1}(N_c) \tag{11}$$
$$= \text{UP}(Z_1) + K + G2^{-1}(N_c) \tag{12}$$

To address this, the downsampling submodule serves another purpose: concurrent separation of K from $\tilde{Z}_2$.

The implemented architecture comprises a 5-layer convolutional block stack that processes concatenated inputs $\tilde{Z}_2 \parallel K$, which is formed as follows:

$$\tilde{Z}_1 = \text{DOWN}(\tilde{Z}_2 \parallel K) \tag{13}$$

Owing to the invertibility of the upsampling submodule, this design guarantees strict bijective mapping in noise-free scenarios. This ensures distribution preservation in $Z_2 \rightarrow Z_1$.

TABLE I
FUNCTIONALITY COMPARISON OF VARIOUS SCHEMES

	Type of Message	Capacity	Robustness	Hidden Method	Confidentiality
PARIS[16]	binary stream	2.98×10^{-1} bpp	✓	generative-based	✓
GFIS[25]	binary stream	3.9×10^{-3} bpp	✓	generative-based	✓
SE-S2I [21]	binary stream	16.1 bpp	✓	generative-based	✓
ISN[7]	image	$\geq$ 24bpp	×	modification-based	×
HiNet[8]	image	24bpp	×	modification-based	×
StegFormer[5]	image	$\geq$ 24bpp	×	modification-based	×
RIIS[9]	image	24bpp	✓	modification-based	×
HCCS[22]	image	24bpp	×	modification-based	×
IHST[23]	image	24bpp	×	modification-based	×
CRoSS[27]	image	24bpp	✓	generative-based	✓
DiffStega[28]	image	24bpp	✓	generative-based	✓
HIS[29]	image	$\geq$ 24bpp	✓	generative+modification	✓
Ours	image	$\geq$ 24bpp	✓	generative-based	✓

Algorithm 2 Training of the UDM

Input: Normally distributed variable Z_1, secret key K, trained $G2$.

Output: The trained UDM.

1: **for** Each iteration **do**
2: Calculate $Z_2 = \mathrm{UP}(Z_1) + K$.
3: Input Z_2 into $G2$ to output the generated image I_g.
4: Calculate $\tilde{I}_g = \mathrm{JPEG}(I_g)$.
5: Calculate $\tilde{Z}_1 = \mathrm{DOWN}(G2^{-1}(\tilde{I}_g) \parallel K)$.
6: Optimize UDM by minimizing L_{UDM}.
7: **end for**

The UDM is trained independently. Its loss function is defined as

$$L_{\mathrm{UDM}} = \|Z_1 - \mathrm{DOWN}\left(G2^{-1}\left(\mathrm{JPEG}(G2(\mathrm{UP}(Z_1) + K))\right)\right)\|_1 \quad (14)$$

The training procedure is detailed in Algorithm 2.

E. Image Superresolution Module (ISM)

Given that the image output by $G1$ is only half the size of the secret image I_s, it is necessary to enlarge it to match the original size. This enlargement is achieved through the image scaling module (ISM). We utilize the superresolution network defined in [36] as our ISM. This module is also trained independently, with its loss function defined as

$$L_{ISM} = \|I_s - \mathrm{ISM}(\downarrow I_s)\|_1 \quad (15)$$

where $\downarrow$ denotes the downsampling operation, which implemented by simple interval sampling.

F. Overall Training Procedure

In the proposed framework, $G1$, $G2$, the UDM, and the ISM undergo sequential training. With the exception of the ISM, each of these modules is trained independently of the other modules. Initially, $G1$ is trained, followed by $G2$, utilizing Algorithm 1. Subsequently, the UDM is trained using Algorithm 2. Finally, with the already trained $G1$, $G2$, and UDM, the ISM is trained on the secret image set I_s.

Fig. 7. Visual comparison between the images produced by our scheme and those generated by the Glow model.

V. EXPERIMENTAL RESULTS

The proposed scheme is a robust generative steganographic scheme that is capable of seamlessly concealing a full-sized color image. In this section, we experimentally compare our scheme with modification-based methodologies, including ISN [7], HiNet [8], StegFormer [5], and RIIS [9], alongside generative approaches such as HCCS [22], IHST [23], CRoSS [27], HIS [29], and DiffStega [28]. Table I delineates the distinctions in the functionalities of these schemes. Notably, certain schemes lack the ability to conceal images; hence, we refrain from comparing our scheme with them in subsequent experiments.

A. Experimental Settings

The proposed scheme is trained in PyTorch 1.9 with an NVIDIA RTX3090 GPU. The COCO dataset [34] serves as the training dataset for $G1$. This dataset comprises images that depict daily scenes that feature common objects in their natural settings and encompasses photographs of 91 distinct object types. Conversely, the CelebA-HD dataset [37] is employed to train $G2$ and encompasses a comprehensive archive of 30,000 high-resolution facial images. All the images are resized to 128×128 in the experiments.

The secret key K is composed of pseudorandom sequences that adhere to a normal distribution $\mathcal{N}(0, 1)$. The Adam optimizer [38] is consistently utilized to optimize all the modules. During the training of $G1$, the optimizer's learning rate is set to 0.0001, and the input images are resized to 64×64. For the remaining modules, the optimizer maintains a learning rate of 0.00001. The parameters specified in Algorithm 1 are $\lambda_1 = 0.1, r_{\lambda_1} = 40,000$, and $\Delta_{\lambda_1} = 0.1$. This implies that the

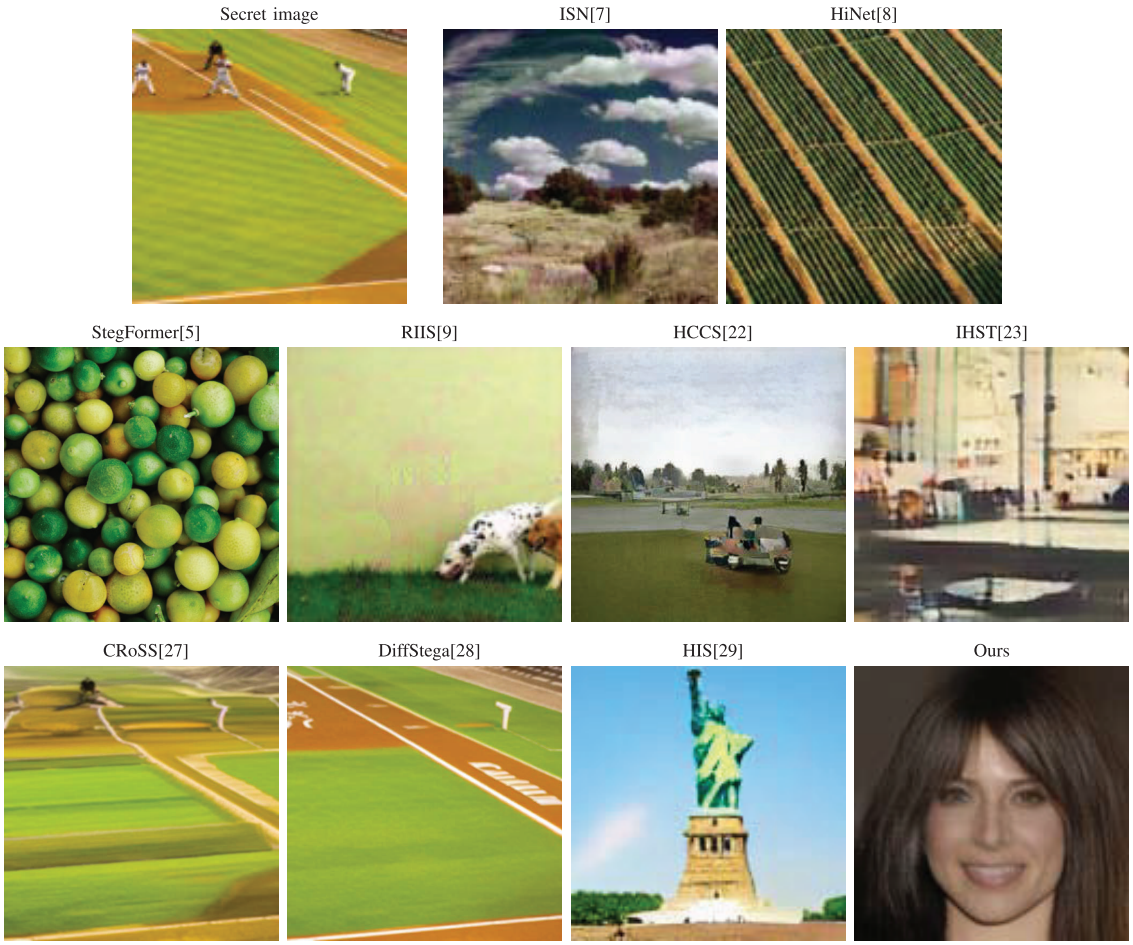


Fig. 8. Stego images generated by different schemes. The first image is the secret image, while the others are the stego images obtained by different schemes.

initial weight of λ_1 is set to 0.1 and incremented by 0.1 every $40,000$ iterations.

B. Evaluation of Image Synthesis Capabilities

The proposed scheme integrates the Glow-based $G2$ to impart image synthesis capabilities. Notably, conventionally, the Glow model accommodates an intermediate variable that adheres to an i.i.d. Gaussian distribution. However, to maintain consistency between $G1$ and $G2$ while increasing robustness, the input distribution of $G2$ is modified to $\mathcal{Z}_2$ which deviates from the original Glow's accepted inputs. In this section, we delve into the ramifications of this alteration on the image synthesis ability of $G2$.

To gauge the quality of the synthesized images, we employ the FID metric [39]. It is defined as the Wasserstein-2 distance between two image distributions. We use the generated stego images and natural images to calculate FID scores. A lower FID score indicates that the stego image is more realistic. In addition to the images generated by our scheme, we randomly sample variables from an i.i.d. Gaussian distribution with the same mean and standard deviation as those in $\mathcal{Z}_2$ and feed them into $G2$ to generate images. In this way, we evaluate the deviation of $\mathcal{Z}_2$ from the ideal distribution.

Table II shows the average FID scores obtained by various methods. Notably, utilizing the Z_2 output by the UDM marginally increases the FID scores compared with sampling random variables from the ideal distribution. This is attributed to Z_2 exhibiting slight deviations from i.i.d. random variables, as it is derived through linear operations on Z_1. Nevertheless, the FID scores remain within acceptable ranges. Figure 7 shows several stego images generated by our scheme, which resemble the images produced by the original Glow model. This underscores the efficacy of the UDM in managing the distribution of $\mathcal{Z}_2$.

TABLE II
FID SCORES OBTAINED BY DIFFERENT METHODS WITH NATURAL IMAGES

scheme	FID↓
$G2$ with normal random input	53.09
$G2$ with $\mathcal{Z}_2$ as the input	64.19
original Glow	52.35

C. Evaluation of Stego Image Quality

We further compare the quality of the stego images generated by different methods. Figure 8 presents a comprehensive visual comparison of stego images generated by various schemes when identical secret content is embedded. All the investigated methods successfully produce visually plausible stego images. Notably, except for CRoSS and DiffStega, all the compared schemes effectively decouple visual-semantic correlations between the secret and stego images. These find

TABLE III
COMPARISON OF THE QUALITY OF STEGO IMAGES GENERATED BY VARIOUS SCHEMES

Method	ISN[7]	HiNet[8]	StegFormer[5]	RIIS[9]	HCCS[22]	IHST[23]	CRoSS[27]	DiffStega[28]	HIS[29]	Ours
FID	19.25	15.06	7.56	82.62	93.19	68.81	49.05	48.70	83.22	25.09

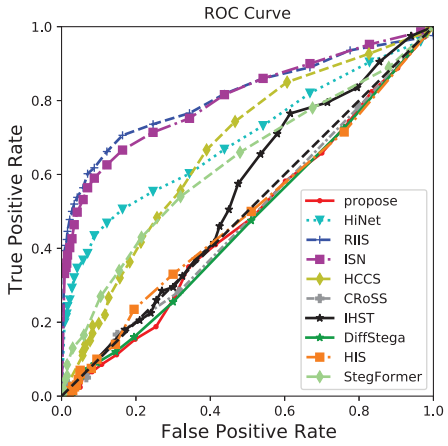
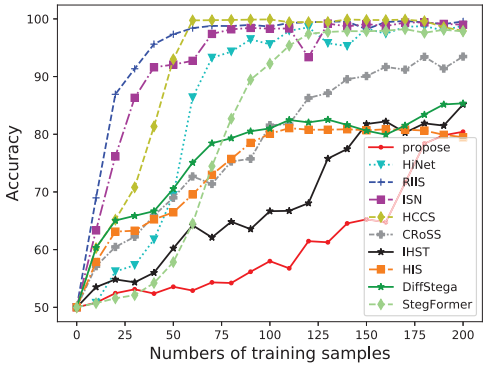
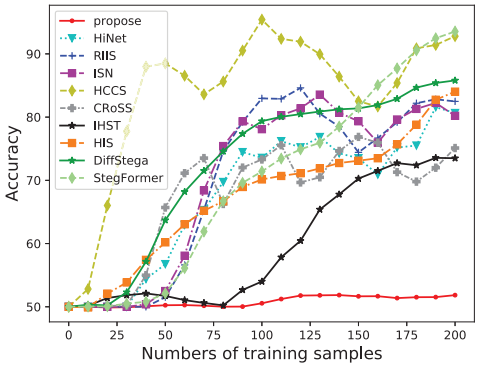


Fig. 9. ROC curves of StegExpose for detecting various schemes.



(a) SRNet



(b) XuNet

Fig. 10. Detection accuracies of a) SRNet and b) XuNet when different numbers of leaked samples are used.

ings confirm that our scheme is capable of eliminating content pertinence between secret and stego images.

Furthermore, we employ the FID metric to provide a quantitative assessment of these stego images. For modification-based steganographic schemes, we leverage the cover and stego images to compute the FID scores. Conversely, for generative approaches, we utilize the stego images and the images generated by the corresponding backbone generative model. Specifically, we adopt the cross-domain image translation technique in [40] to calculate the FID scores for HCCS; the StyTR^2 style transferring model [41] for IHST; the pretrained stable diffusion v1.5[1] for CRoSS, DiffStega, and HIS; and the Glow model for our scheme. The lower the FID score is, the better the stego image quality. Table III compares the FID scores obtained by these schemes. Modification-based steganographic schemes typically attain lower FID scores because of cover-guided distortion constraints. However, this may compromise security by leaving detectable traces. In contrast, our scheme is generative-based. This inherently sacrifices some stego image quality but ensures superior security. Furthermore, our scheme achieves the best FID scores among the generative-based methods, thus demonstrating a good security-quality tradeoff.

D. Evaluation of Undetectability

This section compares the undetectability of various schemes by using various steganalytic tools to discern between cover and stego images in alignment with each respective approach. Notably, there are no cover images for the generative steganographic schemes. To address this problem, following [14], [19], [25], [26], the cover images for these schemes are synthesized utilizing random variables, such as Gaussian noise, rather than variables derived from secret data.

We first utilize the open-source steganalytic toolkit StegExpose [42], which is renowned for its powerful methods such as RS analysis and chi square attack, to benchmark these schemes. The receiver operating characteristic (ROC) curve is employed as the metric. It represents the true-positive rate versus the false-positive rate at various threshold settings. The closer the curve is to the counter diagonal, the better the undetectability of the scheme. For each technique, we collect 1000 cover images and 1000 stego images from the COCO dataset. Figure 9 depicts the resulting ROC curves. Our scheme's ROC curve is near the diagonal, which suggests its resilience against detection by this toolkit.

We subsequently employ two advanced learning-based steganalytic tools, SRNet [12] and XuNet [11], to further analyze these schemes. The detection accuracy serves as the metric. It quantifies the ratio of the number of correctly identified instances to the total number of instances examined. A lower detection accuracy indicates that the steganographic scheme has a higher level of undetectability. Both models are retrained using leaked samples from the testing images of various methods. As we increase the number of leaked samples, we monitor the change in detection accuracy. Figure 10 illustrates the detection accuracies of various schemes as a function of the leaked sample count. As the number of leaked

[1]https://huggingface.co/runwayml/stable-diffusion-v1-5

TABLE IV
ROBUSTNESS COMPARISON (PSNR) UNDER VARIOUS LEVELS OF JPEG COMPRESSION

Metric	ISN[7]	HiNet[8]	StegFormer[5]	RIIS[9]	HCCS [22]	IHST[23]	CRoSS[27]	DiffStega[28]	HIS[29]	Ours
Clean	43.33	46.98	49.21	44.19	26.32	32.03	23.79	23.95	33.44	29.99
JPEG (QF= 90)	13.76	13.86	10.66	27.40	13.68	9.88	23.62	23.70	25.74	27.75
JPEG (QF= 80)	13.44	13.23	10.60	27.02	12.88	9.85	23.51	23.59	24.11	24.14
JPEG (QF= 40)	9.69	9.78	9.71	25.41	11.28	10.43	20.46	21.16	23.90	21.21

samples increases, all schemes become more susceptible to detection.

Owing to the minor discrepancies between Z_2 and the original input of Glow, SRNet may still discern the proposed scheme given a sufficient number of training samples, as illustrated in Fig. 10. Nevertheless, our proposed scheme demonstrates the slowest increase in detection accuracy, thereby exhibiting superior undetectability. This is governed by the generic structure of $G2$ and the distribution preservation of UDM, which minimize statistical deviations in stego images.

E. Evaluation of Robustness

We validate the robustness of the proposed scheme against varying degrees of JPEG compression. The peak signal-to-noise ratio (PSNR) is employed as the metric. A higher PSNR signifies superior recovered image quality. We first establish baseline recovery capability using uncompressed stego images. Figure 11 shows a visual comparison between original secret images and their corresponding recovered versions. All the schemes successfully recover secret content with perceptual fidelity.

Table IV provides a quantitative comparison of the recovered image quality. Modification-based methods achieve superior PSNRs because of cover-image reference advantages. Our scheme has to preserve redundancy in the intermediate variable Z_2, which leads to minor detail loss on the recovered images. Nonetheless, our proposed scheme outperforms most generative steganographic approaches.

We then compare these schemes under varying JPEG compression ratios. Figure 11 shows several images recovered under JPEG compression, which indicate that our scheme incurs minimal degradation in recovered image quality. Table IV details the PSNR scores across QF $\in \{90, 80, 40\}$. Our approach achieves better PSNR scores than the other schemes, except for RIIS. This is attributable to RIIS's ability to learn a more stable mapping without the need for image generation. Nevertheless, our scheme can achieve a good balance between undetectability and robustness.

Finally, we assess the generalization ability by evaluating our scheme on untrained QF values within $[40, 90]$. Figure 12 shows the average PSNRs of the recovered images, where CRoSS and DiffStega are employed as the baselines. Our scheme achieves a stable PSNR across all the tested QF values. The robustness stems from the distortion-guided flow in $G2^1_{(2)}$ and the redundancy via the UDM. By conditioning the inverse flow on wide QF values during training, our model learns a generalized mapping that adapts to compression intensity variations. Furthermore, the upsampling operation in the UDM introduces redundancy into Z_2, which enables error correction even for untrained QF values. These results demonstrate that our scheme generalizes effectively to unseen QF values within the common range, thus ensuring practicality in real-world scenarios.

F. Evaluation of Confidentiality

Table I shows that many steganographic schemes that are capable of concealing secret images fall short in providing confidentiality. Any user with access to the decoder can effortlessly recover the concealed secret image. Conversely, the proposed scheme incorporates a secret key K into the UDM to safeguard against unauthorized access. Recognizing the potential for an attacker to uncover this key, we assess the sensitivity of the encryption algorithms employed to variations in the key.

Given the genuine K, we randomly draw another variable N from the same distribution and gradually introduce noise into the key through

$$\tilde{K} = K + d \times (N - K) \tag{16}$$

where d ranges from 0 to 1. This perturbed key is then used to recover the secret image. Figure 13 illustrates the decline in recovery accuracy as d increases. As d increases, the quality of the recovered images decreases significantly. When $\tilde{K}$ is independent of K, the PSNR score of the recovered image falls below 12. Conversely, as depicted in Fig. 13, the employed robust mapping unexpectedly heightens the tolerance to perturbed keys, thereby reducing the security of our scheme.

However, owing to the vast key space $\mathbb{R}^{3\times h/2\times w/2}$, guessing a value with a small distance to K is challenging. The nonlinear nature of the Glow model further increases the difficulty of accurate estimation K. Figure 14 shows several images recovered via perturbed keys, which demonstrate that when the PSNR scores dip below 15, the image content becomes indecipherable. Consequently, the proposed scheme offers satisfactory confidentiality.

G. Multiple Image Hiding

The proposed scheme is suitable for a multireceiver environment, where each receiver possesses its own secret images. The sender is capable of generating a stego image that hides multiple secret images, each of which is tied to a distinct key. This stego image, along with its respective key, is dispatched to each intended receiver.

In the scenario where three secret images, ($I_s^{(1)}, I_s^{(2)}$ and $I_s^{(3)}$), are to be transmitted, we leverage generator $G1$ to derive the corresponding intermediate variables: ($Z_1^{(1)}, Z_1^{(2)}$ and $Z_1^{(3)}$). The aggregate $(Z_1^{(1)} + Z_1^{(2)} + Z_1^{(3)})/\sqrt{3}$ subsequently serves as the input for the UDM, which ultimately yields the generated image I_g. In this case, the secret key associated with a specific secret image $I_s^{(i)}$ is determined by

$$K^{(i)} = \sum_{j,j\neq i} Z_1^{(j)} \tag{17}$$

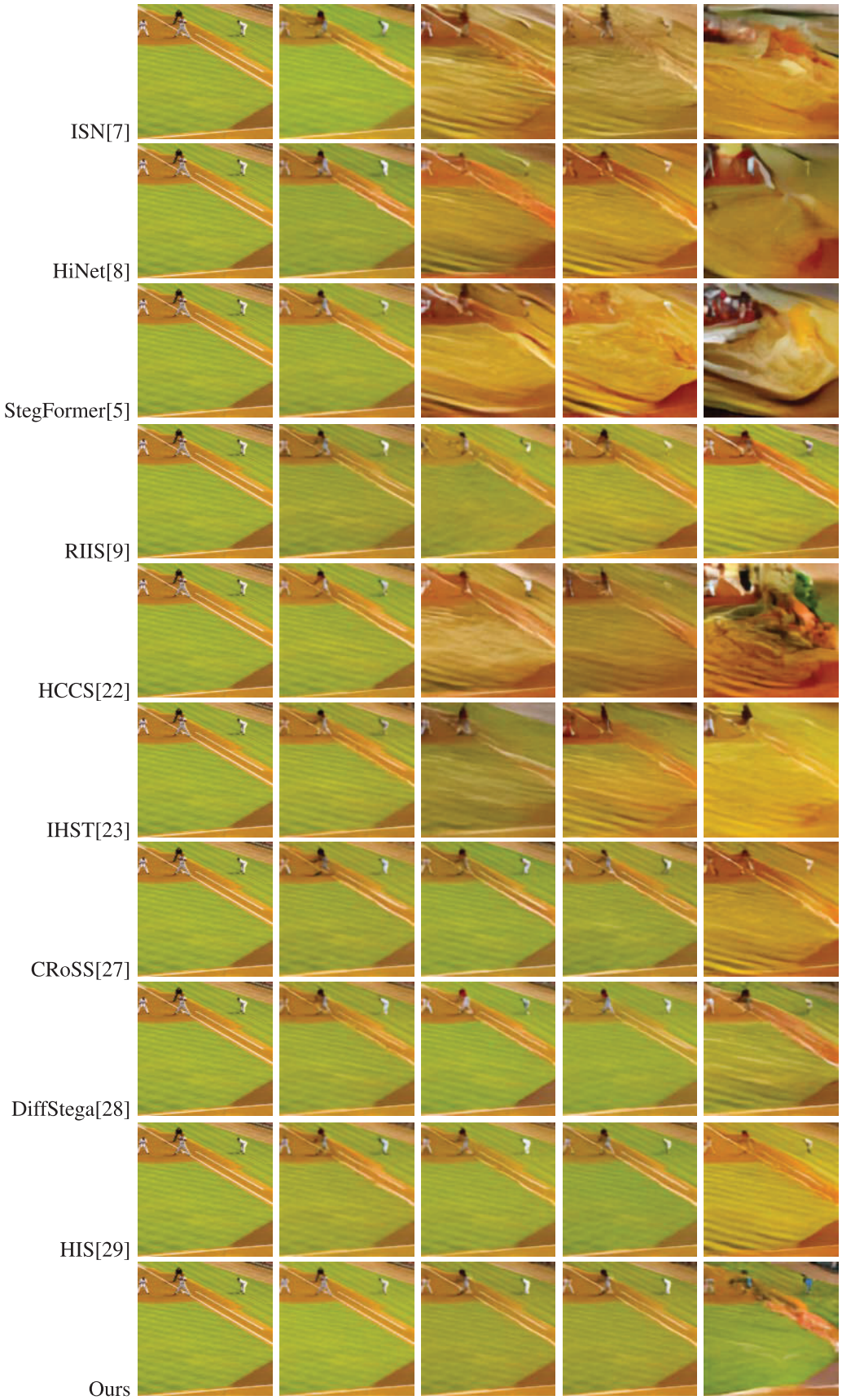


Fig. 11. Examples of recovered secret images. The first column shows the original secret images, the second column shows the recovered images without compression, and the remaining columns show the recovered images under different compression quality factors (QF = 90, 80, and 40).

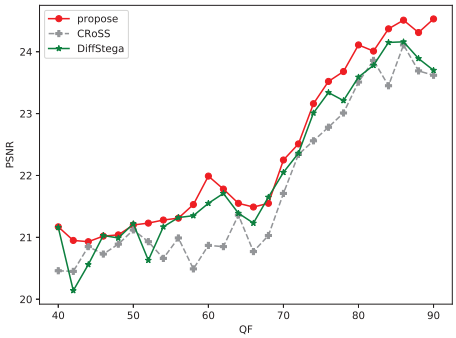


Fig. 12. Robustness comparison under untrained JPEG compression.

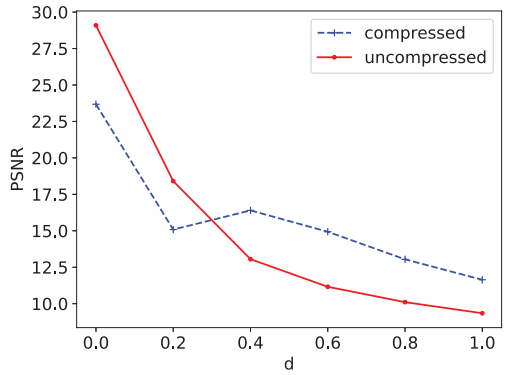


Fig. 13. Curves of the recovered image quality when disturbed keys are used.

TABLE V
RECOVERED IMAGE QUALITY WHEN HIDING MULTIPLE IMAGES

	2 images	3 images	4 images
ISN[7]	37.48	35.49	31.12
HIS[29]	23.04	24.09	22.56
Stegformer[5]	40.67	36.89	33.66
Ours	29.91	29.24	30.09

With this key in hand, the receiver can successfully recover the intended secret image $I_s^{(i)}$ while remaining unable to access other secret images, as the estimation of $Z_1^{(j)}, j \neq i$ from $K^{(i)}$ is infeasible.

Notably, the receiver must possess the key $K^{(i)}$ to recover the secret image. Owing to the substantial data volume, transmitting this key incurs a significant cost. To avoid the need for separate transmissions of $K^{(i)}$, we can embed it into the stego image I_g utilizing steganographic methods such as HiNet [8]. The resulting stego image, designated I_{gg} can then be dispatched to the receiver. Upon acquiring I_{gg} the recipient can effortlessly extract both the secret key $K^{(i)}$ and the original stego image I_g thereby enabling the subsequent recovery of the secret image $I_s^{(i)}$.

For validation, we randomly select 100 test image triplets $\{(I_s^{(1)}, I_s^{(2)}, I_s^{(3)})\}$ from the COCO dataset and employ them to generate stego images. Figure 15 provides visual examples of the images recovered using their respective keys, whereas Table V summarizes the average PSNR scores achieved using these image triplets. Although our scheme does not outperform ISN or StegFormer, the PSNR scores of the recovered images remain impressively high, at approximately 29. Furthermore, the quality of the recovered images remains unaffected as the number of hidden images increases.

Notably, this embedding strategy comes at the cost of compromised robustness and undetectability of the scheme. Therefore, it should be considered only in scenarios where robustness and undetectability are not stringent requirements.

TABLE VI
COMPUTATIONAL COSTS OF VARIOUS SCHEMES

	Time		Memory	
	Image hiding	Image recovery	Image hiding	Image recovery
Glow	65.05s	50.60s	2269MiB	2269MiB
ISN[7]	61.04s	54.15s	1923MiB	1923MiB
HiNet[8]	67.77s	62.53s	1911MiB	1911MiB
Stegformer[5]	93.12s	88.73s	2986MiB	2986MiB
RIIS[9]	58.34s	56.19s	1877MiB	1877MiB
HCCS[22]	86.34s	79.89s	2457MiB	2457MiB
IHST[23]	77.54s	76.22s	2178MiB	2178MiB
CRoSS[27]	83.15s	80.81s	2357MiB	2357MiB
DiffStega[28]	96.54s	95.89s	2869MiB	2869MiB
HIS[29]	132.56s	129.68s	2798MiB	2798MiB
Ours	113.54s	143.09s	2983MiB	3037MiB

TABLE VII
COMPARISON OF DIFFERENT STRUCTURES OF THE PROPOSED SCHEME

modification	$\|Z_1 - \tilde{Z}_1\|_1$	FID of generated images
original Glow	4.3102	52.35
$+G2_{(2)}^{-1}$	2.1682	53.31
+ UDM (without N_b)	1.9549	77.80
+ UDM	1.9549	64.19

H. Evaluation of Computational Cost

This section discusses the access time and memory consumption of our scheme. To ensure fairness, we benchmark our method against the compared schemes with identical hardware (NVIDIA RTX3090 GPU) and standardized metrics, including time (seconds) and peak memory consumption (MiB). The cumulative consumption for processing 500 images is reported in Table VI. Considering that our proposed scheme can be loosely viewed as a cascade of two Glow models, our scheme necessitates approximately double the time cost in comparison to the benchmark. Our method requires 113.54s, which is comparable to the time costs of diffusion-based methods such as DiffStega but higher than those of lightweight modification-based approaches such as ISN. The recovery phase takes 143.09s, which is due primarily to the invertible mapping and UDM decryption processes. The peak memory usage reaches 3037MiB during recovery, which is slightly higher than those of the other methods. This increase stems from the cascaded INN architecture and the UDM. While our method incurs higher computational costs than lightweight schemes do, it can balance full-size image hiding, JPEG robustness, and provable security within a generative framework. Moreover, concealing a single image requires merely 0.23s, which makes the trade-off a viable option for satisfying practical requirements for real-world covert communication.

I. Ablation Experiment

The proposed scheme enhances the original Glow model by incorporating a new flow $G2_{(2)}^{-1}$ into the backward pass of $G2$. This modification aims to strike a balance between the

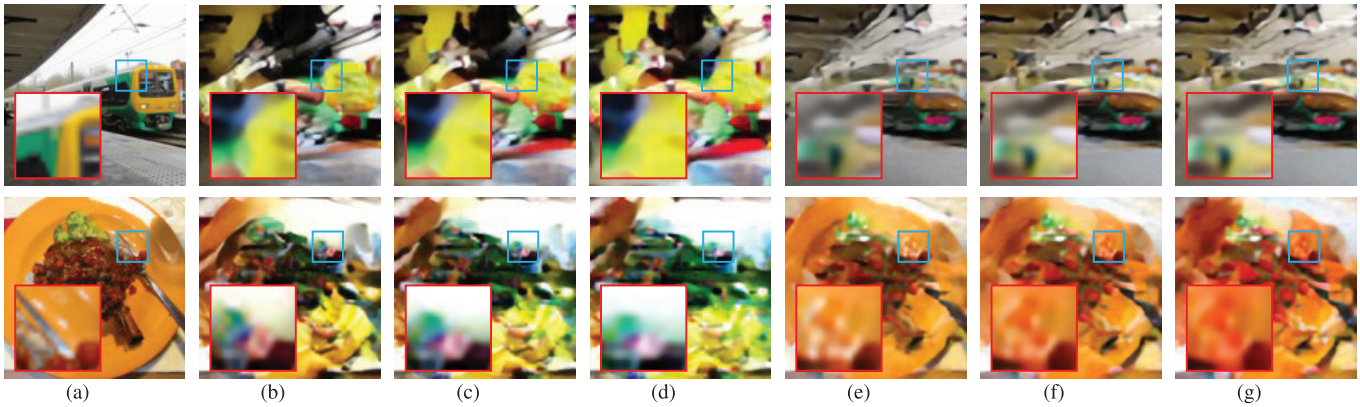


Fig. 14. Demonstration of images recovered by using disturbed keys. The images in a) are the original secret images; those in b)–d) are the images recovered with the disturbed keys obtained with d equal to 0.6, 0.8, and 1.0, respectively; and those in e)–g) are the images recovered with the same disturbed keys with d values of 0.6, 0.8, and 1.0, respectively, but in the presence of JPEG compression.

Fig. 15. Examples of the recovered secret images when hiding multiple images. The images in the first row are the original secret images, and those in the second row are the corresponding recovered images.

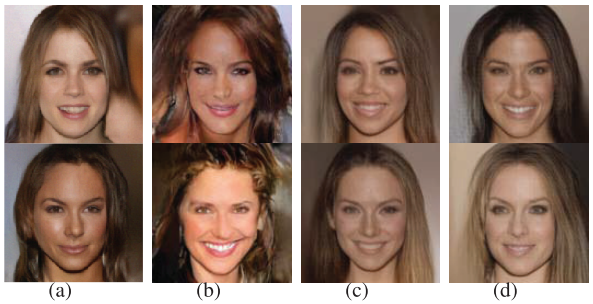


Fig. 16. Images generated by using different structures of the proposed scheme. The images in a) were generated by using the original Glow model, whereas those in b), c) and d) were generated by the scheme that successively integrates $G2^{-1}_{(2)}$, the UDM (without N_b), and the UDM.

quality of the stego image and its robustness. Additionally, we introduce the UDM to further increase the robustness of our scheme. We successively integrate each of these modifications and evaluate their effects on robustness. For illustrative purposes, the JPEG compression ratio is set to 40. We quantify the discrepancy between the input intermediate variable Z_1 and its recovered counterpart $\tilde{Z}_1$, as well as the FID scores of the generated images. The comprehensive results are listed in Table VII. Several example images are shown in Fig. 16.

The unmodified Glow model exhibits limited robustness, as evidenced by the significant $L1$ distance between Z_1 and $\tilde{Z}_1$, which surpasses a value of 4. Upon integrating $G2^{-1}_{(2)}$, we observe a nearly halved distance between Z_1 and $\tilde{Z}_1$. Furthermore, the stego images still retain a high level of quality, as demonstrated in Fig. 16. As per Table VII, the integration of the UDM further decreases the L1 distance. This confirms the necessity of these modules.

In our UDM design, we compromise on independence to achieve spatial redundancy while retaining only the Gaussian property. To counteract deviations from the ideal i.i.d. property that result from this compromise, we introduce a Gaussian bias N_b as illustrated in Eq. (7). Herein, we evaluate this dependence regulatory mechanism. The experimental results presented in Table VII demonstrate that it can maintain generation quality (which decreases from 77.80 to 64.19) while preserving robustness (which remains at 1.9549). This confirms that N_b successfully strikes a balance between the conflicting objectives of correlation suppression and robustness preservation.

The UDM uses a convolution layer to discern K from $\tilde{Z}_2$ In this instance, we assess the efficacy of this decryption technique. A random set of 100 test images is constructed from the dataset to create stego images, which are subsequently compressed with a QF of 80. During the decryption process at the receiver's end, when a straightforward subtraction method is utilized, the average PSNR of the retrieved image is only 17.65 dB. However, when the proposed decryption method is adopted, the average PSNR score increases to 24.14 dB, thereby validating its effectiveness.

VI. Conclusion

This paper proposes a robust generative steganographic scheme that is based on cascaded invertible neural networks

(INNs) for concealing images securely. To guarantee the independence of the stego image from the secret image, we utilize two invertible generators that facilitate a randomized transformation between two unrelated image domains. Our approach encompasses 4 components: Glow-based generators $G1$ and $G2$, an up-and-down sampling module (UDM), and an image superresolution module (ISM). Specifically, $G1$ transforms the secret image into an intermediate variable Z_1, whereas $G2$ maps a separate intermediate variable Z_2 to the stego image. To bolster resilience against JPEG compression, $G2$ incorporates a separate flow that maps compressed stego images to the same Z_2 used for the original uncompressed stego images. Furthermore, Z_1 is intentionally designed to be half the size of Z_2, which creates redundancy for the correction of channel distortions. To maintain consistency between Z_1 and Z_2 while also encrypting Z_1, we introduce the UDM. Additionally, we incorporate the ISM to restore the recovered images to their original dimensions. The experimental results validate that our scheme generates realistic stego images, thus enabling the effective retrieval of secret images, even when subjected to JPEG compression. Moreover, our approach is adaptable to multireceiver environments to enable the sender to conceal multiple secret images within a single stego image. Only a recipient who possesses the correct key can unveil the corresponding secret image.

The proposed scheme holds potential for expansion to ensure robustness against a variety of attacks, including median filtering and brightness adjustments. However, achieving robustness against noise remains a significant challenge because of the complexity of establishing a flow for random distortion values. This highlights the necessity of our future efforts to increase resilience against this specific type of attack. Furthermore, our framework also reveals an inherent trade-off between robustness and generation: increased spatial redundancy improves error correction but inevitably sacrifices the statistical independence of the latent space. The empirical results presented in Section V-I demonstrate that, while our current implementation maintains a viable balance, the systematic optimization of this dual objective remains a challenge for future research.

REFERENCES

[1] S. Baluja, "Hiding images within images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 7, pp. 1685–1697, Jul. 2020.

[2] C. Zhang, P. Benz, A. Karjauv, G. Sun, and I. S. Kweon, "UDH: Universal deep hiding for steganography, watermarking, and light field messaging," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, Dec. 2020, pp. 10223–10234.

[3] L. Zhang, Y. Lu, J. Li, F. Chen, G. Lu, and D. Zhang, "Deep adaptive hiding network for image hiding using attentive frequency extraction and gradual depth extraction," *Neural Comput. Appl.*, vol. 35, no. 15, pp. 10909–10927, May 2023.

[4] X. Liu et al., "Joint compressive autoencoders for full-image-to-image hiding," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 7743–7750.

[5] X. Ke, H. Wu, and W. Guo, "StegFormer: Rebuilding the glory of autoencoder-based steganography," in *Proc. AAAI Conf. Artif. Intell.*, Mar. 2024, vol. 38, no. 3, pp. 2723–2731.

[6] L. Dinh, D. Krueger, and Y. Bengio, "Nice: Non-linear independent components estimation," *arXiv preprint arXiv:1410.8516*, 2014.

[7] S.-P. Lu, R. Wang, T. Zhong, and P. L. Rosin, "Large-capacity image steganography based on invertible neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 10816–10825.

[8] J. Jing, X. Deng, M. Xu, J. Wang, and Z. Guan, "HiNet: Deep image hiding by invertible network," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 4733–4742.

[9] Y. Xu, C. Mou, Y. Hu, J. Xie, and J. Zhang, "Robust invertible image steganography," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 7875–7884.

[10] Z. Guan et al., "DeepMIH: Deep invertible network for multiple image hiding," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 372–390, Jan. 2023.

[11] G. Xu, H.-Z. Wu, and Y.-Q. Shi, "Structural design of convolutional neural networks for steganalysis," *IEEE Signal Process. Lett.*, vol. 23, no. 5, pp. 708–712, May 2016.

[12] M. Boroumand, M. Chen, and J. Fridrich, "Deep residual network for steganalysis of digital images," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 5, pp. 1181–1193, May 2018.

[13] J. Wang, R. Wang, L. Dong, D. Yan, X. Zhang, and Y. Lin, "Towards breaking DNN-based audio steganalysis with GAN," *Int. J. Auto. Adapt. Commun. Syst.*, vol. 14, no. 4, pp. 371–383, 2021.

[14] P. Wei, S. Li, X. Zhang, G. Luo, Z. Qian, and Q. Zhou, "Generative steganography network," in *Proc. 30th ACM Int. Conf. Multimedia*, Oct. 2022, pp. 1621–1629.

[15] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of StyleGAN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8110–8119.

[16] Z. Yang, K. Chen, K. Zeng, W. Zhang, and N. Yu, "Provably secure robust image steganography," *IEEE Trans. Multimedia*, vol. 26, pp. 5040–5053, 2024.

[17] T. Liu, Y. Chen, and W. Gu, "Deniable diffusion generative steganography," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2023, pp. 67–71.

[18] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proc. NIPS*, vol. 33. Vancouver, BC, Canada: Curran Associates, 2020, pp. 6840–6851.

[19] P. Wei, G. Luo, Q. Song, X. Zhang, Z. Qian, and S. Li, "Generative steganographic flow," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2022, pp. 1–6.

[20] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, Jan. 2018, pp. 10215–10224.

[21] Z. Zhou et al., "Secret-to-image reversible transformation for generative steganography," *IEEE Trans. Depend. Secure Comput.*, vol. 20, no. 5, pp. 4118–4134, Sep. 2023.

[22] G. Li, B. Feng, M. He, J. Weng, and W. Lu, "High-capacity coverless image steganographic scheme based on image synthesis," *Signal Process., Image Commun.*, vol. 111, Feb. 2023, Art. no. 116894.

[23] F. Zhang, B. Feng, Z. Xia, J. Weng, W. Lu, and B. Chen, "Conditional image hiding network based on style transfer," *Inf. Sci.*, vol. 662, Mar. 2024, Art. no. 120225.

[24] C. Yu, "Attention based data hiding with generative adversarial networks," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 1, 2020, pp. 1120–1128.

[25] Y. Sun, J. Liu, and R. Zhang, "A robust generative image steganography method based on guidance features in image synthesis," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2023, pp. 55–60.

[26] Z. Zhou et al., "Generative steganography via auto-generation of semantic object contours," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 2751–2765, 2023.

[27] J. Yu, X. Zhang, Y. Xu, and J. Zhang, "CRoSS: Diffusion model makes controllable, robust and secure image steganography," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2023, pp. 1–13.

[28] Y. Yang et al., "DiffStega: Towards universal training-free coverless image steganography with diffusion models," in *Proc. 33rd Int. Joint Conf. Artif. Intell.*, Aug. 2024, pp. 1579–1587.

[29] Y. Xu, X. Zhang, J. Yu, C. Mou, X. Meng, and J. Zhang, "Diffusion-based hierarchical image steganography," 2024, *arXiv:2405.11523*.

[30] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real NVP," 2016, *arXiv:1605.08803*.

[31] A. Grover, C. Chute, R. Shu, Z. Cao, and S. Ermon, "AlignFlow: Cycle consistent learning from multiple domains via normalizing flows," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2020, vol. 34, no. 4, pp. 4028–4035.

[32] X. Su, J. Song, C. Meng, and S. Ermon, "Dual diffusion implicit bridges for image-to-image translation," 2022, *arXiv:2203.08382*.

[33] T. M. Cover, *Elements of Information Theory*. Hoboken, NJ, USA: Wiley, 1999.

[34] T. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

[35] C. Etmann, R. Ke, and C.-B. Schönlieb, "IUNets: Learnable invertible up- and downsampling for large-scale inverse problems," in *Proc. IEEE 30th Int. Workshop Mach. Learn. Signal Process. (MLSP)*, Sep. 2020, pp. 1–6.
[36] M. V. Conde, U.-J. Choi, M. Burchi, and R. Timofte, "Swin2SR: SwinV2 transformer for compressed image super-resolution and restoration," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2022, pp. 669–687.
[37] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," 2017, *arXiv:1710.10196*.
[38] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[39] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local Nash equilibrium," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2017, pp. 6626–6637.
[40] P. Zhang, B. Zhang, D. Chen, L. Yuan, and F. Wen, "Cross-domain correspondence learning for exemplar-based image translation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 5143–5153.
[41] Y. Deng et al., "StyTr2: Image style transfer with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 11326–11336.
[42] B. Boehm, "StegExpose—A tool for detecting LSB steganography," 2014, *arXiv:1410.6656*.

Tiewei Qin received the B.S. degree from Zhejiang University of Technology in 2021 and the M.S. degree from Jinan University in 2024.

Bingwen Feng received the B.E. and Ph.D. degrees from Sun Yat-sen University, Guangzhou, China, in 2008 and 2014, respectively.

He is currently an Associate Professor with the College of Cyber Security, Jinan University, Guangzhou. His research interests include multimedia security, AI security, and privacy protection.

Bingbing Zhou received the B.S. degree from Guangdong University of Foreign Studies, Guangzhou, China, in 2024. He is currently pursuing the M.S. degree with Jinan University, Guangzhou.

His research interests include multimedia security and image steganography.

Jilian Zhang (Member, IEEE) received the M.S. degree from Guangxi Normal University, Guilin China, in 2006, and the Ph.D. degree from Singapore Management University, Singapore, in 2014.

He is currently a Professor with the College of Cyber Security, Jinan University, Guangzhou, China. He has published on international journals and conferences, including IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, ACM SIGMOD, VLDB, IJCAI, and WWW. His research interests include data management, query processing, and information security.

Zhihua Xia (Member, IEEE) received the Ph.D. degree in computer science and technology from Hunan University, China, in 2011.

He worked successively as a Lecturer, an Associate Professor, and a Professor with the College of Computer and Software, Nanjing University of Information Science and Technology. He is currently a Professor with the College of Cyber Security, Jinan University, Guangzhou, China. He was a Visiting Scholar at New Jersey Institute of Technology, USA, in 2015, and a Visiting Professor at Sungkyunkwan University, South Korea, in 2016. His research interests include AI security, cloud computing security, and digital forensics. He serves as a Managing Editor for IJAACS.

Jian Weng (Senior Member, IEEE) received the Ph.D. degree in computer science and engineering from Shanghai Jiao Tong University in 2008.

From 2008 to 2010, he held a post-doctoral position with the School of Information Systems, Singapore Management University. He is currently a Professor and the Vice Chancellor of Jinan University, Guangzhou, China. He has published more than 100 papers in cryptography and security conferences and journals, such as CRYPTO, EUROCRYPT, ASIACRYPT, TCC, PKC, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, and IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING. His research interests include public key cryptography, cloud security, and blockchain. He served as the PC co-chair or a PC member for more than 30 international conferences. He also serves as an Associate Editor for IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY.

Wei Lu (Member, IEEE) received the B.S. degree in automation from Northeast University, China in 2002, and the M.S. and Ph.D. degrees in computer science from Shanghai Jiao Tong University, China, in 2005 and 2007, respectively.

He was a Research Assistant at The Hong Kong Polytechnic University from 2006 to 2007. He is currently a Professor with the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China. His research interests include multimedia forensics and security, data hiding and watermarking, and privacy protection. He is an Associate Editor of *Signal Processing* and *Journal of Visual Communication and Image Representation*.