Secret-to-Image Reversible Transformation for Generative Steganography

Zhili Zhou, Member, IEEE, Yuecheng Su, Jin Li, Keping Yu, Q. M. Jonathan Wu, Senior Member, IEEE, Zhangjie Fu, Member, IEEE, and Yunqing Shi, Life Fellow, IEEE

Abstract—Recently, generative steganography that transforms secret information to a generated image has been a promising technique to resist steganalysis detection. However, due to the inefficiency and irreversibility of the secret-to-image transformation, it is hard to find a good trade-off between the information hiding capacity and extraction accuracy. To address this issue, we propose a secret-to-image reversible transformation (S2IRT) scheme for generative steganography. The proposed S2IRT scheme is based on a generative model, i.e., Glow model, which enables a bijective-mapping between latent space with multivariate Gaussian distribution and image space with a complex distribution. In the process of S2I transformation, guided by a given secret message, we construct a latent vector and then map it to a generated image by the Glow model, so that the secret message is finally transformed to the generated image. Owing to good efficiency and reversibility of S2IRT scheme, the proposed steganographic approach achieves both high hiding capacity and accurate extraction of secret message from generated image. Furthermore, a separate encoding-based S2I transformation (SE-S2IRT) scheme is also proposed to improve the robustness to common image attacks. The experiments demonstrate the proposed steganographic approaches can achieve high hiding capacity (up to 4 bpp) and accurate information extraction (almost 100% accuracy rate) simultaneously, while maintaining desirable anti-detectability and imperceptibility.

Index Terms—coverless steganography, digital forensics, generative steganography, information hiding, steganography

1 INTRODUCTION

Image steganography is the technology of imperceptibly hiding secret information into a cover image so that the covert communication can be achieved by transmitting the cover image without arousing suspicion [1], [2]. As another way of covert communication, data encryption usually encodes the secret information in an incomprehensible and meaningless form, and thus it exposes the importance and confidentiality of the secret information. That makes the secret information easy to be intercepted and cracked. Therefore, compared to encryption, the advantage of steganography is that it can conceal the occurrence of covert communication to ensure the security of the transmitted secret information [1], [2].

The conventional steganographic approaches generally select an existing image as a cover and then embed secret information into the cover with a slight modification. However, due to the modification, image distortion is inevitably left more or less in the cover image, especially under a high hiding payload. That might cause the presence of hidden information to be successfully exposed by well-designed steganalyzers [3]. To resist the detection of steganalyzers, recently, some researchers have been devoted to the study of “generative steganography” [4], [5], [6]. Instead of modifying an existing cover image, the idea of generative steganography is to transform a given secret message to a generated image, which is then used as the stego-image for covert communication. Since the secret information is concealed without modification, generative steganography could achieve promising performance against the detection of steganalyzers.

Recently, generative steganography usually applied Generative Adversarial Networks (GANs) [7] to transform a given secret message to a new meaningful image that looks very realistic, i.e., “realistic-looking image”. Some generative steganographic approaches usually converted the secret message to a simple image label [8], [9], [10] or a low-dimensional noise signal [6], [11], [12], [13] as the input of a GAN model to generate an image, and thus the secret message is finally transformed to the generated image. However, these approaches can only convert a small-sized secret message
to a low-dimensional input information of GAN model, and only allow one-way mapping from the input information to the image content. Thus, the secret-to-image transformation process of those approaches is generally inefficient and irreversible. Consequently, their hiding capacity is quite limited, or they cannot accurately extract the secret information from the generated image. That makes these approaches inapplicable in practical steganographic tasks.

According to the above, to realize both high-capacity hiding and accurate extraction of secret information, the key issue of generative steganography is how to find an efficient and reversible transformation between secret messages and images. To this end, instead of using GANs, we explore a flow-based generative model, i.e., Glow model, which enables a bijective-mapping between high-dimensional latent vectors and images for generative steganography. By analyzing the properties of bijective-mapping process of Glow model, we propose the secret-to-image reversible transformation (S2IRT) scheme based on Glow model to realize an efficient and reversible transformation between secret messages and generated images for information hiding and extraction.

Specifically, we encode a given secret message as a high-dimensional latent vector in an efficient and robust manner, and then map the vector to a generated image by the Glow model for information hiding. Consequently, the secret message is finally transformed to a generated image. Also, a reverse transformation between the secret message and the generated image can be implemented for information extraction. Fig. 1 lists the framework of the S2IRT scheme for generative steganography based on Glow model. The main contributions are summarized as follows.

1) The properties of the bijective-mapping process of Glow model are analyzed. By the experiments and analysis, it is found that the bijective-mapping process of Glow model has the following properties: the elements of the latent vector mapped from a real image follow an approximate one-dimensional Gaussian distribution, and the mapping errors occur in the bijective-mapping process. The property analysis motivates us to propose the S2IRT scheme based on Glow model for generative steganography.

2) The S2IRT scheme is proposed for generative steganography. In the proposed S2IRT scheme, an efficient and robust encoding method is first designed to encode a given secret message as a high-dimensional latent vector. Specifically, guided by a given secret message, a large number of elements are arranged into the corresponding positions to construct a high-dimensional vector. Consequently, instead of simply encoding a given secret message as elements, the secret message is encoded as the position arrangement of these elements. The encoding method not only ensures the elements’ distributions of the constructed latent vector and the latent vector mapped from a real image are consistent, but also the encoded message is robust to the bijective-mapping errors. Then, the vector is mapped to a generated image by Glow model, and the generated image is used as the stego-image for covert communication. The S2IRT scheme has the following advantages.

Due to the efficient and robust message encoding and the bijective mapping processes, the proposed S2IRT scheme has good efficiency and reversibility, which lead to both high hiding capacity and accurate extraction of secret message.

Moreover, the quality of generated image can maintain at a high-level with the increase of hiding payload. That is because, although encoding secret messages of different sizes lead to different constructed vectors, each constructed vector in latent space can be mapped to a high-quality image in image space by Glow model.

In addition, the existing steganalyzers are still not likely to defeat the proposed steganographic approach, since the steganography is implemented by generating a new image as the stego-image rather than by modifying an existing image.

3) The SE-S2IRT scheme is proposed to improve the robustness of generative steganography. To improve the robustness to common attacks such as intensity changing, contrast enhancement, and noise addition, the SE-S2IRT scheme is proposed based on a designed separate encoding method for generative steganography.

The remainder of this paper is organized as follows. Section 2 introduces the related work. Section 3 briefly describes the Glow model. Sections 4 and 5 elaborate the proposed S2IRT and SE-S2IRT schemes for generative steganography, respectively. Section 6 presents and discusses the experimental results. Conclusions are drawn in Section 7.

2 RELATED WORK

2.1 Conventional Steganography

In the past two decades, a lot of conventional image steganographic approaches have been proposed. They generally select an existing image as the cover and then imperceptibly embed secret information into the cover image with a slight modification.

In the early work [14], [15], steganography is implemented by simply modifying the least significant bit (LSB) of each pixel of cover image with the same probability to embed secret information. To reduce image distortion caused by modification, Fridrich et al. [1] proposed the adaptive steganography based on the framework of minimizing additive distortion. Specifically, a distortion function is heuristically defined to assign an embedding cost to each pixel of the cover image, and then the secret information is embedded in the way of minimizing the sum of embedding costs, i.e., additive distortion. By designing different distortion functions, many adaptive steganographic approaches have been proposed, such as HUGO [16], S-UNIWARD.
[17], HILL [18], and MiPOD [19]. Also, some non-additive distortion-based adaptive steganographic approaches such as DeJoin [20], ACM [21], and GMRF [22] have been proposed. However, since all of these distortion functions are defined heuristically, it is still possible to design a superior distortion function to implement steganography with less image distortion.

As deep learning techniques have made remarkable achievements in the field of computer vision, some researchers used deep learning networks to automatically learn a distortion function for steganography, as done by ASDL-GAN [23], UT-6HPF-GAN [24], and reinforcement learning-based steganography [25]. Compared to the heuristically defined distortion functions, the distortion functions learned by deep learning networks can further reduce the image distortion for steganography. Recently, Baluja et al. [26] trained two fully convolutional networks [27] to directly embed a secret image within a cover image and to extract the secret image from the stego-image, respectively.

As an adversary of steganography, steganalysis aims to detect the existence of hidden information in stego-images. To this end, the handcrafted steganalytic features such as first-order statistical features [28] and second-order statistics [29], [30] are extracted from the stego-images, and then are fed into the advanced machine learning-based classifiers such as Support Vector Machine (SVM). Fridrich et al. [31] proposed the Spatial Rich Model (SRM)-based steganalyzer, in which a set of high-dimensional handcrafted steganalytic features are extracted to train the steganalyzer. In recent works [32], [33], [34], convolutional neural networks (CNNs) were applied to automatically learn steganalytic features and classify these features for steganalysis.

Although some conventional steganographic approaches [35], [36] could fool one or several certain steganalyzers, it is very hard for them to resist the detection of a set of well-designed steganalyzers. The main reason is that the modification operation of conventional steganography inevitably leads to some distortion in the stego-images, which would cause the presence of hidden information to be detected successfully.

### 2.2 Generative Steganography

Instead of modifying an existing cover image, the idea of generative steganography is to generate a new image on the basis of secret information, which is then used as the stego-image for covert communication. Since the secret information is concealed without modification, generative steganography could achieve promising performance against the detection of steganalyzers. The existing generative steganographic approaches can be roughly categorized into two classes: texture synthesis-based and GANs-based approaches.

Wu et al. [4] proposed a generative steganographic approach based on reversible texture synthesis. It hides a given secret message into a generated texture image through the process of texture synthesis. Qian et al. [37] proposed an improved version of this steganographic approach to enhance the robustness to JPEG compression. Xu et al. [38] directly transformed the secret message to an intricate texture image for steganography. However, the texture images are generally meaningless and they somewhat resemble the encrypted data. Thus, transmitting these images on networks will easily arouses attackers’ suspicion.

Fortunately, the popular generative deep learning models, i.e., Generative Adversarial Networks (GANs) [7], are able to produce new meaningful images that looks very realistic, i.e., “realistic-looking images”. Thus, some generative steganographic approaches transformed the secret messages to the realistic-looking images based on GANs.

Some researchers converted the secret message to a simple image label or semantic information as intermedia data, and then fed it into GANs to generate the stego-image. Liu et al. [8] transformed a given secret message to the class label information and then fed it into the Auxiliary Classifier GAN (ACGAN) [39] to generate the stego-image. Also, a discriminator was trained to determine the class label of the generated image to extract the hidden secret message. Zhang et al. [9] mapped the secret message to the image semantic information, and then used this information as a input condition of GANs to generate the corresponding stego-image. Qin et al. [10] detected image objects by using Faster RCNN [40] and established a mapping dictionary between binary message sequences and object labels, and then transformed the secret message to the corresponding image by GANs. However, since the simple label and semantic information can only contain very limited information, the hiding capacity of these generative steganographic approaches is quite low.

To enhance hiding capacity, some researchers converted the secret message to a noise signal, and then used the noise signal as the input of GANs to generate the stego-image. Hu et al. [6], [11] directly encoded the secret message as a noise signal, and then fed it into Deep Convolutional GANs (DCGANs) [41] to generate a realistic-looking image as the stego-image. Instead of using DCGANs, Li et al. [12] adopted Wasserstein GAN Gradient Penalty (WGAN-GP) [42] to transform the secret message to a generated image for steganography. Arifioanto et al. [13] used a word2vec model [43] to convert the secret message to a word vector, which is then used as the input vector of GANs to generate the stego-image. However, since GANs only allow one-way mapping from the low-dimensional input information to the high-dimensional images, those GANs-based steganographic approaches cannot accurately extract secret information from these generated images, especially under high hiding payloads.

In summary, the existing generative steganographic approaches cannot achieve a good trade-off between hiding capacity and extraction accuracy of secret message. That makes them inapplicable in practical steganographic tasks.

To achieve both high-capacity information hiding and accurate information extraction, in this paper, we attempt to propose the S2IRT scheme based on Glow model to realize an efficient and reversible transformation between secret messages and generated images for information hiding and extraction. Consequently, the proposed approach can achieve high-capacity information hiding and accurate information extraction simultaneously, while maintaining desirable anti-detectability and imperceptibility for generative steganography.
Suppose there is a training dataset $P_{data}$ containing a set of image samples $x$, and the values of their elements are in the range of [0,1]. According to the above equation, the learning of $f_{\theta}$ is realized via maximum log-likelihood estimation using the following equation:

$$
\max_{\theta \in \Theta} E_{x \sim P_{data}} \left[ \log p_{X}(x) \right] = \max_{\theta \in \Theta} E_{x \sim P_{data}} \left[ \log p_{Z}(f_{\theta}(x)) \left| det \frac{\partial f_{\theta}(x)}{\partial x} \right| \right]
$$  

(6)

Consequently, by using the learned Glow model, which consists of a pair of mapping functions $f_{\theta}$ and $g_{\theta}$ as shown in Eqs. (3) and (4), we can achieve the bijective-mapping between image space and latent space.

4 S2IRT SCHEME FOR GENERATIVE STEGANOGRAPHY

Fig. 1 shows the framework of the proposed S2IRT scheme for generative steganography based on the Glow model. It consists of two stages: S2I transformation for information hiding and S2I reverse transformation for information extraction. In this section, we first give the motivation of the proposed S2IRT scheme, and then elaborate on the two stages of S2IRT scheme.

4.1 Motivation

To realize the generative steganography, it is necessary to encode a given secret message as a latent vector at first, and then input the vector into the Glow model to generate the stego-image. To this end, we analyze the properties of bijective mapping of Glow model to propose the S2IRT scheme for generative steganography. More details are given as follows.

Property 1. The elements of the latent vector mapped from a real image follow an approximate one-dimensional Gaussian distribution.

As mentioned in Section 3, the Glow model builds a bijective-mapping between the high-dimensional latent vectors in which each dimension follows one-dimensional Gaussian distribution and the real images with a complex distribution. Moreover, the dimensions of the latent vectors are independent. Thus, for a given real image, the elements of its latent vector follow an approximate one-dimensional Gaussian distribution, as shown by a toy example in Fig. 2a.
To avoid the successful identification of generated images from real images on latent space, a set of elements are randomly selected from the standard Gaussian distribution to construct a latent vector for image generation, so that the elements of the latent vector of generated image also follow an approximate one-dimensional Gaussian distribution, as done in the literature [44], [46].

To resist the detection of the secret message hidden in a generated stego-image on latent space, in this paper, we also randomly sample a set of elements from the standard Gaussian distribution to construct the latent vector for the stego-image generation, so that the elements of the latent vector of each generated stego-image also follow an approximate one-dimensional Gaussian distribution. Then, these elements are divided into a number of groups according to their values. Afterward, guided by a given secret message, we arrange corresponding positions for these group elements sequentially to construct the latent vector, and thus the secret message is encoded as the position arrangement of these elements.

On the contrary, if we directly encode the secret message as the element values to construct the latent vector for stego-image generation, it is very likely that the distribution of these elements is quite different from the approximate one-dimensional Gaussian distribution, which will cause the generated stego-image to be easily detected by analyzing the distribution of its latent vector’s elements. Fig. 2a also illustrates the distributions of the elements of the latent vectors constructed by element position arrangement-based encoding and element value-based encoding methods.

Property 2. There are mapping errors in the bijective-mapping.

Since the bijective-mapping is implemented between continuous latent space and discrete image space, there are mapping errors in the bijective-mapping process.

As a result, although the constructed latent vector can be recovered from the generated stego-image by the Glow model for information extraction, there is still a slight difference between the original vector and the recovered one, as shown by a toy example in Fig. 2b. Thus, some elements of original vector near to the group boundaries will shift to the adjacent groups. That will affect the accuracy of information extraction. As shown by a toy example of an original vector and the recovered one in Fig. 2c, the second element shifts from Group 4 to Group 5, and thus it is hard to accurately obtain the position arrangement of the original vector’s elements to extract the hidden message. Thus, we choose and arrange the elements nearest to the center of each group, which are less likely to shift to the adjacent groups, for latent vector construction.

Therefore, motivated by the observed properties of bijective mapping of Glow model, we propose the S2IRI scheme for generative steganography. For information hiding, we group a set of elements randomly sampled from standard Gaussian distribution and choose the elements near to each group’s center. Then, guided by a given secret message, we arrange the positions of these elements to construct the latent vector. Finally, the constructed latent vector is input into the Glow model to generate the stego-image. Consequently, the secret message is transformed to the generated stego-image. For information extraction, the S2I reverse transformation is implemented to extract the secret message from the generated image. More details are given as follows.

### 4.2 S2I Transformation for Information Hiding

The S2I transformation is implemented for information hiding, as illustrated in Fig. 3. According to Section 4.1, the process of S2I transformation includes four main steps: element grouping and choosing, secret-guided position arrangement, latent vector construction, and image generation.

1. **Step (1): Element grouping and choosing.**
    - Set $S$.
    - Grouping and Choosing.
    - Element groups.

2. **Step (2): Secret-guided position arrangement.**
    - Given the secret message and $G_i$, $1 \leq i \leq K$.
    - Integerizing $m_i$.
    - 38-th choice $\{1, 3, 4, 8\}$.
    - 4-th choice $\{5, 6, 9, 12\}$.

3. **Step (3): Latent vector construction.**
    - Generating.
    - Constructing.
    - Latent vector.

4. **Step (4): Image generation.**
    - Glow model.

Fig. 3. The illustration of S2I transformation for information hiding. The process of information hiding includes four main steps: element grouping and choosing, secret-guided position arrangement, latent vector construction, and image generation.
Fig. 3. Suppose there are \( N = 12 \) elements chosen from \( K = 3 \) groups \( \{G_i \mid 1 \leq i \leq 3 \} \), and the number of elements chosen from each group is \( n = 4 \). As 12 elements need to be arranged, the initial position set used for arranging these elements is denoted as \( pos = \{1, 2, 3, \ldots, 12\} \). Guided by a given secret bitstream \( M = '00100101101001' \), we arrange the elements of each group into a set of positions selected from \( pos \). More details are given as follows.

To arrange \( n = 4 \) elements of the first group \( G_1 \), we should select 4 positions from the 12 positions in \( pos \), and thus the number of choices is \( C(12, 4) = 495 \). Since 495 choices can be used to represent \( \lfloor \log_2 495 \rfloor = 8 \) bits of secret message, we read the first 8 bits from the secret bitstream, i.e., \( '00100101' \), as the decimal integer \( m = 37 \). Then, the \((m + 1) = 38\)-th position choice in lexicographical order, i.e., the positions \( ‘1,3,4,8’ \), are selected for arranging the elements of \( G_1 \). Thus, we set \( Ind[1] = Ind[3] = Ind[4] = Ind[8] \) of position arrangement array \( Ind \) as the No. of first group, i.e., \( Ind[1] = Ind[3] = Ind[4] = Ind[8] = 1 \). That means the positions \( ‘1,3,4,8’ \) will be used to arrange the elements of \( G_1 \).

**Algorithm 1 Secret-guided position arrangement of elements**

**Input:**
- Secret bitstream: \( M = '00100101101001' \)
- Number of groups: \( K \)
- Number of chosen elements in each group: \( n \)
- Number of chosen elements in total: \( N \)

**Output:**
- Position arrangement array: \( Ind = \{Ind[1], Ind[2], \ldots, Ind[N]\} \)
  1. \( r \leftarrow N \) //Record number of remaining positions
  2. \( pos \leftarrow \{1, 2, 3, \ldots, N\} \) //Record remaining positions
  3. for \( i = 1 \) to \( K - 1 \) do
    4. \( \omega_i \leftarrow C(r, n) \) //Compute number of choices of selecting \( n \) positions from \( r \) positions
  5. \( m \leftarrow \text{Read}(M, \lfloor \log_2 \omega_0 \rfloor) \) //Read the next \( \lfloor \log_2 \omega_0 \rfloor \) bits from \( M \) as a positive decimal integer \( m \)
  6. \( sel \leftarrow \text{Select}(m, pos, \omega_i) \) //Select \( n \) positions from \( pos \) guided by \( m \)
  7. for \( p \) in \( sel \) do
    8. \( Ind[p] \leftarrow i \) //Set \( Ind[p] \) as the group No., i.e., \( i \)
  9. end for
  10. \( r \leftarrow r - n \)
  11. \( pos \leftarrow \text{PosDelete}(pos, sel) \) //Remove \( sel \) from \( pos \)
  12. end for
  13. return \( Ind \)
  14. function \( \text{Select}(m, pos, \omega_i) \) // Select \( n \) positions from \( pos \) guided by \( m \)
    15. \( sel \leftarrow \text{CreateArray}(n) \) //Create a \( N \)-dimensional array
    16. for \( j = n \) to 1 do
      17. \( p \leftarrow j \)
    18. while \( C(p, j) < m + 1 \) do
      19. \( p \leftarrow p + 1 \)
    20. end while
    21. \( sel[j] \leftarrow pos[p] \)
    22. \( m \leftarrow m - C(p - 1, j) \)
    23. end for
  24. return \( sel \)
  25. end function

Similarly, to arrange \( n = 4 \) elements of \( G_2 \), we should select 4 positions from the remaining \( 8 \) positions in \( pos \), and thus the number of choices is \( C(8, 4) = 70 \). The 70 choices enable the hiding of \( \lfloor \log_2 70 \rfloor = 6 \) bits of secret message. Thus, the next 6 bits are read from the secret bitstream, i.e., \( '101001' \), as the decimal integer \( m = 41 \). Then, the \((m + 1) = 42\)-th position choice in lexicographical order, i.e., the positions \( ‘5,6,9,12’ \), are selected for arranging the four elements of \( G_2 \). Thus, we set \( Ind[5] = Ind[6] = Ind[9] = Ind[12] = 2 \). That means the positions \( ‘5,6,9,12’ \) will be used to arrange the four elements of \( G_2 \).

Guided by the given secret message \( M \), we can finally obtain the position arrangement array \( Ind = \{1, 3, 1, 2, 2, 3, 1, 2, 3, 2\} \). Consequently, the secret message \( M \) is encoded as the position arrangement of those elements. Indicated by this array \( Ind \), the latent vector can be constructed by the following step.

**Step (3): Latent vector construction.** According to the obtained position arrangement array \( Ind = \{1, 3, 1, 2, 2, 3, 1, 2, 3, 2\} \), we put the four elements of \( G_1 \) into the \( ‘1-th, 3-th, 4-th, and 8-th’ \) positions, the four elements of \( G_2 \) into the \( ‘5-th, 6-th, 9-th, and 12-th’ \) positions, and the elements of groups \( G_3 \) into the remaining positions in a random order. These arranged elements are concatenated to form a \( N \)-dimensional vector. As there are \( N_T \) sampled elements in total, the other \((N_T - N)\) elements are directly concatenated at the end of the vector to obtain the final \( N_T \)-dimensional latent vector \( z \).

**Step (4): Image generation.** After constructing the \( N_T \)-dimensional latent vector \( z \) by using a private key shared between communication participants, we randomly scramble the elements of \( z \) and then map the scrambled vector to a generated high-quality image by the Glow model. Finally, the generated image is used as the stego-image for covert communication.

In the proposed S2IRT scheme, the room of position arrangement of \( N \) chosen elements is very large for constructing the latent vector and the group Nos. of these elements are stable, which lead to high efficiency and robustness for encoding the secret message as the constructed vector. Moreover, the bijective-mapping between the constructed vector and the generated image can be realized by the Glow model. Thus, the proposed S2IRT scheme has good efficiency and reversibility, which lead to both high hiding capacity and accurate extraction of secret message. In addition, the constructed vector can be mapped to a high-quality image by the Glow model and the information hiding is implemented by generating a new image rather than modifying an existing image. Therefore, the proposed steganographic approach also has desirable anti-detectability and imperceptibility. These are also proven by experiments in Section 6.

### 4.3 S2I Reverse Transformation for Information Extraction

As information extraction is the reverse process of information hiding, the S2I reverse transformation is implemented to extract the secret message from the generated image. It includes two main steps: latent vector recovery and secret message extraction.
Step (1): Latent vector recovery. At the receiving end, the received image is reversely mapped to a latent vector by the Glow model. Then, by the shared Key, the position order of the elements of latent vector $z'$ is restored to obtain the recovered latent vector $z$.

Step (2): Secret message extraction. By the shared information of $K$ and $n$, the first $N = Kn$ elements of the recovered latent vector $z'$ are first re-grouped into $K$ groups in the same manner as described in the stage of information hiding.

For $n$ elements of each group $G_i$, we obtain the set of their positions in $z'$ and compute the choice $N_i$ of this position set in all $C(N - (i - 1)n, n)$ possible choices. According to step (2) of information hiding stage, the choice $N_i$ is equal to $(m_i + 1)$, where $m_i$ is the decimal integer of corresponding secret bits. Thus, for each group $G_i$, we can obtain $m_i$ from the position choice $N_i$ of its $n$ elements, and then transform $m_i$ to the corresponding secret bits. After computing all the bits, we concatenate these bits to obtain the final secret message $M'$.

4.4 Hiding Capacity Analysis

As mentioned in the above subsection, for $i$-th group, $[\log_2 \omega_i]$-bit secret message can be encoded as the position arrangement of its elements, where $\omega_i = C(N - (i - 1)n, n)$. Thus, we can compute the total number of bits that can be encoded as the position arrangement for all groups by

$$BN_{S2I} = \sum_{i=1}^{K-1} [\log_2 \omega_i]$$

It satisfies the following equation.

$$\left(\sum_{i=1}^{K-1} \log_2 \omega_i\right) - K + 1 < BN_{S2I} \leq \sum_{i=1}^{K-1} \log_2 \omega_i$$

Where,

$$\sum_{i=1}^{K-1} \log_2 \omega_i = \log_2 \prod_{i=1}^{K-1} C(N - (i - 1)n, n)$$

According to the Stirling’s approximation, we can obtain the following result.

$$\sum_{i=1}^{K-1} \log_2 \omega_i \approx \log_2 \left(\frac{1}{\sqrt{2\pi}}\right) \frac{(Kn)^{K+n+\frac{1}{2}}}{n^{K+n+\frac{1}{2}}} \approx -(K-1)\log_2 \sqrt{2\pi} + \left(Kn + \frac{1}{2}\right) \log_2 K - \frac{K-1}{2} \log_2 n$$

Where, $K \geq 2$ and $n \geq 1$. Suppose all the $N_T$ sampled elements are chosen and divided into $N_T$ groups, i.e., $N = N_T$ and $K = N_T$, and each group contains only $n = 1$ element for information hiding. The sizes of the training images used in the experiments are $256 \times 256 \times 3 = 196,608$ and the dimension of constructed latent vector, i.e., $N_T$, is the equal to the sizes of those images, and thus $N_T = 196,608$. According to Eqs. (8) and (10), the maximum $BN_{S2I}$ is approximately estimated to be in the range of $(3,000,094, 3,196,701)$, and thus the maximum number of bits hidden in per image pixel per channel (bpp) can theoretically reach up to a very high value, i.e., $3,196,701/196,608 \approx 16.2$ bpp.

We analyze the impacts of $K$ and $n$ on the hiding capacity of S2IRT in Appendix A, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TDSC.2022.3217661. As proven by Appendix A, available in the online supplemental material, available online, larger $K$ and $n$ lead to higher hiding capacity for S2IRT. However, larger $K$ and $n$ will lead to lower accuracy of information extraction. The above conclusions are also proven by the experimental results in Section 6.2.

5 SE-S2IRT Scheme for Generative Steganography

In practice, images usually suffer from a variety of attacks such as intensity changing, contrast enhancement, and noise addition during transmitting on networks. In the S2IRT scheme, although the group Nos. of the chosen elements used for latent vector construction are relatively stable, those attacks will cause some elements of the constructed latent vector to shift from one group to its adjacent ones. Once the group No. of one element is changed, the position choice Nos. for arranging the elements of most groups would be changed, and thus most secret bits encoded as the position arrangement for these groups will be affected. Consequently, it is hard for the S2IRT scheme to accurately extract secret information from the recovered latent vector under those attacks. To improve the robustness to those attacks, we propose a separate encoding-based S2IRT (SE-S2I) scheme for steganography.

5.1 SE-S2I Transformation for Information Hiding

The proposed SE-S2IRT scheme is based on the idea of separate encoding of secret message. Fig. 4 illustrates the process of SE-S2I transformation for information hiding, which also includes following steps.

Step (1): Element grouping and choosing. This step is the same as that of S2IRT scheme. The $N_T$ sampled elements are randomly sampled from the standard Gaussian distribution and divided into $K$ groups denoted as $\{G_i|1 \leq i \leq K\}$. Then, $n$ elements nearest to the center of each group are chosen to obtain the $N$ elements, where $N = Kn$ and $N \leq N_T$.

Step (2): Secret-guided position arrangement of elements. To improve the robustness to image attacks, guided by the secret message $M$, for each group $G_i$, we select one position from every $\lfloor N - (i - 1)n \rfloor/n = (K - i + 1)$ adjacent positions of the position set pos to separately arrange the $n$ elements of $G_i$. Where, pos records the $\lfloor N - (i - 1)n \rfloor$ remaining positions, and $(i - 1)n$ means the number of positions occupied by the elements of 1-th to $(i-1)$-th groups. The number of choices for selecting one position from $(K - i + 1)$ positions can be computed by $\omega_i = C(K - i + 1, 1)$, and thus $[\log_2 \omega_i]$ bits can be encoded as the position arrangement for one element of $G_i$. As there are $n$ elements in $G_i$, totally $n[\log_2 \omega_i]$ can be encoded as the position arrangement for the $n$ elements of $G_i$. 

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
To illustrate the position arrangement, Fig. 4 shows an example of secret-guided position arrangement using the SE-S2IRT scheme. Suppose there are totally $N = 12$ chosen elements in $K = 3$ groups $\{G_i\}_{1 \leq i \leq 3}$ and the number of elements in each group is $n = 4$. As 12 elements need to be arranged, the initial position set used for arranging these elements is $pos = (1, 2, 3, \ldots, 12)$. Then, guided by a given secret bitstream $M = '10101100'$, we separately arrange the elements of each group into the corresponding selected positions. More details are given as follows.

For arranging one element of $G_1$, the number of choices of selecting one position from 3 adjacent positions is $C(3, 1) = 3$, and thus $\lceil \log_2 3 \rceil = 1$ can be encoded. Therefore, we read the first bit from the secret bitstream, i.e., '1', as the decimal integer $m = 1$. Then, we select the $(m + 1) = 2$-th position choice in lexicographical order, i.e., the position '2', to arrange one element of $G_1$. Next, we set the $Ind[2]$ of position arrangement array $Ind$ as the No. of first group, i.e., $Ind[2] = 1$. Similarly, we read the next bit, i.e., '0', as the decimal integer $m_2 = 0$, and then use the $(m + 1) = 1$-th position choice in the next three adjacent positions, i.e., the position '1', to arrange one element of $G_1$. Thus, $Ind[4] = 1$. After separately arranging all the 4 elements of $G_1$ in the same manner, totally 4 bits, i.e., '10101', are encoded, and $Ind[2], Ind[4], Ind[8]$ and $Ind[10]$ are set as 1. The element arrangement process is also shown in Fig. 4.

The position arrangement for the other groups is the same as that for $G_1$. Guided by the given secret message $M$, finally, we can select the corresponding positions for the elements of all groups and obtain the position arrangement array $Ind = (3, 1, 2, 1, 3, 2, 1, 3, 1, 2, 3)$. The secret message $M$ is encoded as the position arrangement of those elements, and $Ind$ will be used for latent vector construction.

Step (3): Latent vector construction and Step (4): Image generation are also the same as those of S2IRT scheme. Thus, they are skipped.

In the SE-S2IRT scheme, once the group No. of one element arranged into one of $K$ adjacent positions is changed, only the position choice Nos. for arranging the elements into the $K$ adjacent positions will be changed and thus only $\sum_{i=1}^{K-1} \lceil \log_2 (K - i + 1) \rceil$ bits will be affected at most. Therefore, the SE-S2IRT scheme provides higher robustness to common image attacks than the S2IRT scheme, which is also proven by experiments in Section 6.3.

5.2 SE-S2I Reverse Transformation for Information Extraction

Similarly, the reverse SE-S2I transformation is implemented for information extraction.

Step (1): Latent vector recovery. This step is the same as that of S2IRT scheme. Thus, it is also skipped.

Step (2): Secret message extraction. By the shared information of $K$ and $n$, we re-group the first $N = Kn$ elements of the recovered latent vector $z'$ in the same manner as described in the stage of information hiding of S2IRT scheme.

For the $n$ elements of each group $G_i$, we obtain their positions in $z'$. Then, we compute the No. of position choice of selecting one position from every $(K - i + 1)$ adjacent positions. According to the step (2) of information hiding stage in SE-S2IRT scheme, the No. of position choice is equal to $(m_i + 1)$, where $m_i$ is the decimal integer of corresponding secret bits. Then, the decimal $m_i$ is transformed to the $\lceil \log_2 C(K - i + 1, 1) \rceil$ bits. According to the position choices for the $n$ elements of each group $G_i$, we can compute $n \lceil \log_2 C(K - i + 1, 1) \rceil$ bits of secret message. After computing all bits from the element positions of all groups, we concatenate these bits to recover the final secret message $M'$.

5.3 Hiding Capacity Analysis

As mentioned in the above subsection, for $i$-th group, $n \lceil \log_2 \omega_i \rceil$-bit secret message can be encoded as the position arrangement of its elements, where $\omega_i = C(K - i + 1, 1)$. Thus, we can compute the total number of bits encoded as the position arrangement for all groups by

$$BN_{SE-S2I} = \sum_{i=1}^{K-1} n \lceil \log_2 \omega_i \rceil$$

(11)

It satisfies the following equation.

$$\sum_{i=1}^{K-1} n(\log_2 \omega_i) - n < BN_{SE-S2I} \leq \sum_{i=1}^{K-1} n \log_2 \omega_i$$

(12)

Where,

$$\sum_{i=1}^{K-1} n \log_2 \omega_i = n \log_2 \left( \prod_{i=1}^{K-1} (K - i + 1) \right) = n \log_2 (K!)$$

(13)
According to the Stirling’s approximation, we can obtain the following result.

\[
\sum_{i=1}^{K} n \log_2 \omega_i \approx n \log_2 \left( \sqrt{2\pi K} \left( \frac{K}{e} \right)^K \right) = Kn \log_2 \frac{2\sqrt{2\pi K}}{e}
\] (14)

Where, \( K \geq 2 \) and \( n \geq 1 \). Suppose all the \( N_T \) sampled elements are chosen and divided into \( N_T \) groups, i.e., \( N = N_T \) and \( K = N_T \), and each group contains only \( n = 1 \) element. The sizes of the training images used in the experiments are \( 256 \times 256 \times 3 = 196,608 \) and thus \( N_T = 196,608 \). According to the Eqs. (12) and (14), the maximum \( BN_{SE-S2IRT} \) is approximately estimated to be in the range of \((2,977,102, 3,173,709)\), and thus the maximum number of bits hidden in per image pixel can theoretically reach up to a very high value, i.e., about \( 3,173,709/196,608 \approx 16.1 \) bpp.

Also, we analyze the impacts of \( K \) and \( n \) on the hiding capacity of SE-S2IRT in Appendix B, available in the online supplemental material. As proven by Appendix B, available in the online supplemental material, the hiding capacity of SE-S2IRT increases with the increase of \( K \) and \( n \). However, larger \( K \) and \( n \) will lead to lower accuracy of information extraction. The above conclusions are also proven by the experimental results in Section 6.2.

6 Experiments

In this section, we first introduce the experimental settings. Then, the impacts of parameters of S2IRT and SE-S2IRT schemes—the number of element groups \( K \) and the size of each group \( n \)—are analyzed and discussed. Finally, the performances of proposed stenographic approaches are tested in the aspects of information extraction accuracy, anti-detectability, imperceptibility, and robustness, and are then compared with those of state-of-the-arts.

6.1 Experiment Settings

6.1.1 Training Datasets

In [46], the original Glow model trained on CelebA-HQ dataset [47] can effectively and efficiently produce high-quality “realistic-looking” images. Thus, in our experiments, we also adopt this dataset to train the Glow model. The CelebA-HQ dataset contains 30,000 high-resolution face images selected from the CelebA dataset. The sizes of the images in the dataset are \( 1024 \times 1024 \times 3 \). For the sake of training efficiency, we rescale the images to the sizes of \( 256 \times 256 \times 3 \) and train the Glow model on the rescaled images.

6.1.2 Model Training Details

The Glow model is trained by maximizing the objective function defined in Eq. (6) with the learning rate 0.001. The depth of each flow is set as 32 and the level of the multiscale architecture is set as 6. After 2,000,000 iterations on the dataset, the trained Glow model is obtained.
attacks, and the quality of each training image is scored manually. By inputting a test image into the trained model, the score of the image can be scored automatically. T-SNE is a statistical method of visualizing high-dimensional data by transforming each data to a datapoint in two-dimensional space.

6.1.4 Evaluation Environment

All the experiments are conducted on an NVIDIA RTX 3090 GPU platform using PyTorch with Python interface.

6.2 Parameter Impacts

In the proposed S2IR and SE-S2IR schemes, for the latent vector construction, there are two key parameters: $K$, which is the number of element groups, and $n$, which is the size of each group. We test the parameter impacts on the performances of proposed schemes in the aspects of hiding capacity and extraction accuracy. In this experiment, we adopt the trained Glow model, which is introduced in previous subsection.

The parameter impacts on the proposed S2IR and SE-S2IR schemes are shown in Fig. 5, respectively. From this figure, it is clear that larger $K$ and $n$ lead to higher hiding capacity. That is because larger $K$ and $n$ offer larger room of position arrangement of elements for latent vector construction, which enables more secret bits to be hidden in the image generated from the constructed latent vector. However, the accuracy of information extraction decreases with the increase of $K$ and $n$ for the following reason. Increasing $K$ and $n$ causes more elements to be near to the group boundaries, and thus the group Nos. of these elements are vulnerable. Therefore, the group Nos. of elements in the recovered latent vector would be quite different from those in the original latent vector, which affects the accuracy of information extraction significantly. Moreover, the hiding capacity of S2IR is generally higher than that of SE-S2IR with the same values of $K$ and $n$. The main reason is that the number of ways of selecting $n$ positions from $N$ positions is larger than that of selecting $n$ positions from every $K$ positions of $N$ ones, and thus more bits can be hidden.

According to Figs. 5a and 5c, under the condition of perfect accuracy rate of information extraction, i.e., $IE_A = 1.0$, the highest hiding capacity of S2IR and that of SE-S2IR can reach up to 4.3 bpp and 4.1 bpp, respectively. From Figs. 5a and 5c, to obtain a required level of hiding capacity $IH_C$, several combinations of $(K, n)$ can be chosen. For example, when the hiding capacity is required to be $IH_C = 4$ bpp for S2IR, the set of combinations of $(K, n)$ are $(23,8000)$, $(29,6000)$, $(33,5000)$, $(40,4000)$, $(50,3000)$, $(70,2000)$ and $(128,1000)$. Also, it is found that smaller $K$ generally leads to higher extraction accuracy at a certain level of hiding capacity. Thus, the parameter combination with smallest $K$, i.e., $(23,8000)$, is chosen to achieve the required hiding capacity $IH_C = 4.0$ bpp. In the above manner, to obtain a required level of hiding capacity, the parameter combinations $(K, n)$ can be determined for S2IR and SE-S2IR, and they are used in the following experiments.

6.3 Performance Evaluation and Comparison

In this subsection, we evaluate the performances of the proposed approaches in the aspects of information extraction accuracy, anti-detectability, imperceptibility and robustness under different hiding payloads, and compare with those of the state-of-the-arts. These approaches are listed as follows.

S-UNIWARD [17]: It is a famous conventional steganographic approach, which is based on a heuristically defined distortion function.
That is because the P1.0000 1.0000 1.0000 1.0000 0.9943 P to 4 used to generate the stego-images. S2IRT, the Glow model trained on the CelebA-HQ dataset is S2IRT scheme proposed in Section 5. In both S2IRT and SE-S2IRT scheme proposed in Section 4.

...images and the accuracy of information extraction. which the secret information is directly encoded as the flow-based model to hide a secret image within a cover image and to extract it.

SWE [6]: This is a generative steganographic approach, called as steganography without embedding (SWE), in which the secret information is directly encoded as the input noise of DCGAN to generate the stego-images.

ISWE [52]: This is an improved version of SWE. It used self-attention mechanism to improve the quality of generated images and the accuracy of information extraction.

S2IRT: This is the steganographic approach using S2IRT scheme proposed in Section 4.

SE-S2IRT: This is the steganographic approach using SE-S2IRT scheme proposed in Section 5. In both S2IRT and SE-S2IRT, the Glow model trained on the CelebA-HQ dataset is used to generate the stego-images.

6.3.1 Information Extraction Accuracy

Table 1 lists the performances of information extraction accuracy (IEA) of different steganographic approaches, i.e., Deep-Stego, HiNet, SWE, ISWE, S2IRT, and SE-S2IRT, with the increase of hiding payloads. From this table, it is clear that the proposed steganographic approaches, i.e., S2IRT and SE-S2IRT, achieve much higher information extraction accuracy than the conventional steganographic approaches, i.e., Deep-Stego and HiNet, and the generative steganographic approach, i.e., SWE and ISWE, under different hiding payloads. The extraction accuracy rates of S2IRT and SE-S2IRT keep at a very high-level (IEA ≈ 1.0) when the hiding payload ranges from 0.1 bpp to 4 bpp. That is because S2IRT and SE-S2IRT achieve the secret-to-image reversible transformation, mainly due to the efficient and robust message encoding and the bijective mapping between latent space and image space implemented by the Glow model. In summary, the proposed generative steganographic approaches can achieve high hiding capacity (up to 4 bpp) and accurate extraction of secret message (almost 100% accuracy rate), simultaneously.

In contrast, with the increase of hiding payload, the information extraction accuracies of the four other approaches decrease for the following reasons. For Deep-Stego and HiNet, the secret message is compressed and then is hidden into a cover image by training a fully convolutional network in a lossy manner, and thus it is hard to accurately extract the secret message from the stego-image. For SWE and ISWE, the secret message is encoded as a low-dimensional noise signal of GANs to generate the stego-image. However, GAN model only allows one-way mapping from the input information to the image, which makes the hidden secret message difficult to be extracted from the stego-image. Thus, the above four approaches cannot achieve accurate information extraction, especially under high hiding payloads.

6.3.2 Anti-Detectability

To evaluate and compare the anti-detectability performances of different steganographic approaches, we use the famous steganalyzers including SRM [31] and XuNet [33] to detect the presence of hidden information in the stego-images. Where, SRM is based on a set of high-dimensional hand-crafted steganalytic features, while XuNet is based on a modified CNN structure for steganalysis.

For the conventional steganographic approaches, the training dataset of the steganalyzers includes 5,000 natural images (cover images) and 5,000 stego-images obtained by embedding the secret messages into these natural images. It is notable that the generative steganographic approaches generate the stego-images by the secret-to-image transformation without the need of the natural images (cover images). For the generative steganographic approaches, as both the natural images and the generated images without information hiding can be regarded as cover images [53], the training dataset of the steganalyzers consists of 5,000 images without information hiding (including 2,500 natural images and 2,500 generated images without information hiding) and 5,000 generated images with information hiding. By using the two trained steganalyzers, the detection error rate $P_F$ is computed in the manner described in Section 6.1 to evaluate the anti-detectability performances of different steganographic approaches.

Table 2 lists the anti-detectability performances (values of $P_F$) of those approaches with different hiding payloads. From this table, there are several observations.

From this table, it is clear that the anti-detectability of the proposed generative steganographic approaches, i.e., S2IRT and SE-S2IRT, keeps at a high-level ($P_F$ is about 0.27) and generally outperform the other approaches significantly, when the hiding payload ranges from 0.5-4.0 bpp. That is because the proposed generative steganographic approaches directly generate new images as the stego-images without modification, and thus it is relatively hard for the steganalyzers to detect the presence of hidden information. In contrast, with the increase of hiding payload, the performances of the conventional steganographic approaches including S-UNIWARD, UT-6HPF-GAN, Deep-Stego, and HiNet are generally poor. In those conventional steganographic approaches, as the secret message is hidden by modifying the existing cover images, hiding more information into the cover images leads to more distortion. Thus, it is much easier for the steganalyzers to successfully
defeat these steganographic approaches, especially under high hiding payloads.

However, when the hiding payload is 0.1 bpp, the \( P_E \) values of the generative steganographic approaches, i.e., S2IRT, SE-S2IRT, SWE and ISWE, are also about 0.27, which are lower than those of the conventional steganographic approaches, i.e., S-UNIWARD, UT-6HPF-GAN, Deep-Stego, and HiNet. That is because the limited size of training dataset leads to the limited ability of generating realistic images of generative models, which makes some natural images to be distinguished from generated images correctly. However, this problem can be effectively alleviated by increasing the size of training dataset.

Also, it is notable that some \( P_E \) values of the other steganographic approaches including S-UNIWARD, UT-6HPF-GAN, Deep-Stego, HiNet, SWE, and ISWE are empty or equal to 0 in Table 1 for the following reasons. The \( P_E \) values of UT-6HPF-GAN and S-UNIWARD are empty when the payload is larger than 1.0 bpp, because they cannot achieve the hiding capacity larger than 1.0 bpp. Since Deep-Stego and HiNet aim to hide a secret image within an image, they cannot hide the messages of small sizes, and thus the \( P_E \) values are empty when the hiding payload is smaller than 1.0. Moreover, it does not consider the resistance to steganalyzers, the stego-images can be easily detected by the steganalyzers, and thus the \( P_E \) values of Deep-Stego and HiNet are equal to 0. Due to the irreversibility of GANs, SWE and ISWE cannot extract the hidden information when the hiding payload is larger than 0.1, and thus the corresponding \( P_E \) values of SWE and ISWE are empty. Therefore, the high hiding capacities of S-UNIWARD, UT-6HPF-GAN, Deep-Stego, HiNet, SWE, and ISWE come at the expense of security performance or information extraction accuracy.

### 6.3.3 Imperceptibility

Without the need of an existing cover image, the generative steganographic approaches, i.e., SWE, ISWE, S2IRT and SE-S2IRT, directly transform a given secret message to a generated image for steganography. As there is no need of cover images in these approaches, the traditional reference image quality assessment strategies such as PSNR and SSIM are not suitable for evaluating the imperceptibility of these approaches. Thus, the typical model of reference less image quality assessment strategy, i.e., \( BRISQUE \), is adopted in the experiments. However, \( BRISQUE \) is not sensitive to slight noise, and it is also experimentally found that the \( BRISQUE \) scores of the stego-images obtained by the conventional steganographic approaches are almost the same as those of original cover images, since those stego-images are obtained by slightly modifying the original cover images. Thus, \( BRISQUE \) is not suitable to evaluate the qualities of stego-images for the conventional steganographic approaches.

Therefore, we only compute and compare the \( BRISQUE \) scores for the stego-images generated by the generative steganographic approaches, i.e., SWE, ISWE, S2IRT, and SE-S2IRT, with different hiding payloads. Table 3 shows the average \( BRISQUE \) scores of the stego-images generated by different approaches with different hiding payloads. Smaller \( BRISQUE \) score means higher image quality. From this table, although the \( BRISQUE \) scores of SWE and ISWE are close to those of S2IRT and SE-S2IRT, SWE and ISWE cannot accurately extract the hidden information when the hiding payload is higher than 0.1. Also, in S2IRT and SE-S2IRT, directly transform a given secret message to a generated image for steganography. As there is no need of cover images in these approaches, the traditional reference image quality assessment strategies such as PSNR and SSIM are not suitable for evaluating the imperceptibility of these approaches. Thus, the typical model of reference less image quality assessment strategy, i.e., \( BRISQUE \), is adopted in the experiments. However, \( BRISQUE \) is not sensitive to slight noise, and it is also experimentally found that the \( BRISQUE \) scores of the stego-images obtained by the conventional steganographic approaches are almost the same as those of original cover images, since those stego-images are obtained by slightly modifying the original cover images. Thus, \( BRISQUE \) is not suitable to evaluate the qualities of stego-images for the conventional steganographic approaches.

| Approaches | Hiding payloads (bpp) |
|------------|-----------------------|
|            | 0.1       | 0.5     | 1.0   | 2.0   | 4.0   |
| SWE        | 9.07      | 8.69    | -     | -     | -     |
| ISWE       | 8.74      | 10.13   | -     | -     | -     |
| S2IRT      | 8.85      | 8.13    | 7.32  | 8.54  | 9.13  | 7.44  |
| SE-S2IRT   | 6.89      | 7.93    | 9.03  | 7.21  | 7.69  | 8.17  |

### Table 2

The Values of \( P_E \) of Steganographic Approaches with Different Hiding Payloads

| Approaches | Hiding payloads (bpp) |
|------------|-----------------------|
|            | 0.1       | 0.5     | 1.0   | 2.0   | 4.0   |
| S-UNIWARD  | 0.4229    | 0.1824  | 0.0493| -     | -     |
| UT-6HPF-GAN| 0.4414    | 0.2489  | 0.0615| -     | -     |
| Deep-Stego | -         | -       | 0.0000| 0.0000| 0.0000|
| HiNet      | -         | -       | 0.0000| 0.0000| 0.0000|
| SWE        | 0.2687    | -       | -     | -     | -     |
| ISWE       | 0.2699    | -       | -     | -     | -     |
| S2IRT      | 0.2707    | 0.2711  | 0.2698| 0.2702| 0.2701|
| SE-S2IRT   | 0.2696    | 0.2700  | 0.2710| 0.2703| 0.2694|
|            | 0.4461    | 0.1925  | 0.0712| -     | -     |
| UT-6HPF-GAN| 0.4690    | 0.2971  | 0.0787| -     | -     |
| Deep-Stego | -         | -       | 0.0000| 0.0000| 0.0000|
| HiNet      | -         | -       | 0.0000| 0.0000| 0.0000|
| SWE        | 0.2691    | -       | -     | -     | -     |
| ISWE       | 0.2698    | -       | -     | -     | -     |
| S2IRT      | 0.2701    | 0.2670  | 0.2686| 0.2708| 0.2702|
| SE-S2IRT   | 0.2703    | 0.2698  | 0.2695| 0.2697| 0.2695|

### Table 3

The \( BRISQUE \) Scores of Stego-Images of Those Approaches With Different Hiding Payloads

| Approaches | Hiding payloads (bpp) |
|------------|-----------------------|
|            | 0.1       | 0.5     | 1.0   | 2.0   | 4.0   |
| SWE        | 9.07      | 8.69    | -     | -     | -     |
| ISWE       | 8.74      | 10.13   | -     | -     | -     |
| S2IRT      | 8.85      | 8.13    | 7.32  | 8.54  | 9.13  | 7.44  |
| SE-S2IRT   | 6.89      | 7.93    | 9.03  | 7.21  | 7.69  | 8.17  |
S2IRT, it is clear that the imperceptibility of the generated images without hiding message (0 bpp) is similar to the generated images with the hiding payload ranging from 0.1-4.0 bpp. That indicates the quality of stego-images generated by S2IRT and SE-S2IRT will not be affected with the increase of hiding capacity, mainly due to the efficiency and robustness of message encoding and the powerful ability of generating high-quality images of Glow model.

To further illustrate imperceptibility of the proposed steganographic approaches, we also compute and compare the t-SNE data of the real images and the images generated with and without information hiding. Where, t-SNE is the dimensionality reduction strategy, which transforms high-dimensional image data to 2-dimensional data so as to observe the distribution of values of image data. From Fig. 6, it is clear that it is hard to distinguish the real images, the images generated without information hiding, and those with information hiding.

Fig. 7 shows toy examples of real images and images generated by S2IRT without and with information hiding. It is clear that S2IRT can achieve excellent hiding capacity, i.e., up to 4.0 bpp. Meanwhile, it is hard to distinguish the images generated with information hiding from the images generated without information hiding. Also, mainly due to the insufficient training of Glow model, it is notable that there are some slight visual differences between the real images and the generated images. The visual differences can be further reduced by training the Glow model on larger-sized datasets.

According to the above, the proposed steganographic approaches can consistently achieve desirable imperceptibility performances under different hiding payloads.

| Attack                                      | Approaches | Hiding payloads (bpp) |
|---------------------------------------------|------------|-----------------------|
| Intensity change with 1%                    | S2IRT      | 0.74                  |
|                                             | SE-S2IRT   | 1.00                  |
| Intensity change with 2%                    | S2IRT      | 0.68                  |
|                                             | SE-S2IRT   | 1.00                  |
| Intensity change with 4%                    | S2IRT      | 0.68                  |
|                                             | SE-S2IRT   | 0.87                  |
| Contrast enhancement with 3%                | S2IRT      | 0.70                  |
|                                             | SE-S2IRT   | 1.00                  |
| Contrast enhancement with 5%                | S2IRT      | 0.66                  |
|                                             | SE-S2IRT   | 1.00                  |
| Contrast enhancement with 7%                | S2IRT      | 0.66                  |
|                                             | SE-S2IRT   | 0.88                  |
| Salt & Pepper Noise with the factor of 1%   | S2IRT      | 0.73                  |
|                                             | SE-S2IRT   | 0.99                  |
| Salt & Pepper Noise with the factor of 3%   | S2IRT      | 0.69                  |
|                                             | SE-S2IRT   | 0.91                  |
| Salt & Pepper Noise with the factor of 5%   | S2IRT      | 0.66                  |
|                                             | SE-S2IRT   | 0.95                  |
| Gaussian noise with standard deviation σ = 10^{-4} | S2IRT | 0.68                  |
|                                             | SE-S2IRT   | 0.84                  |
| Gaussian noise with standard deviation σ = 10^{-3} | S2IRT | 0.66                  |
|                                             | SE-S2IRT   | 0.80                  |
| JPEG compression 90                         | S2IRT      | 0.66                  |
|                                             | SE-S2IRT   | 0.80                  |
| Rotation ±0.75°                             | S2IRT      | 0.66                  |
|                                             | SE-S2IRT   | 0.85                  |
| Image sterilization                         | S2IRT      | 1.00                  |
|                                             | SE-S2IRT   | 1.00                  |

| Attack                                      | Approaches | Hiding payloads (bpp) |
|---------------------------------------------|------------|-----------------------|
| Intensity change with 0.5%                  | S2IRT      | 0.63                  |
|                                             | SE-S2IRT   | 0.97                  |
| Intensity change with 1.0%                  | S2IRT      | 0.63                  |
|                                             | SE-S2IRT   | 0.91                  |
| Intensity change with 4.0%                  | S2IRT      | 0.62                  |
|                                             | SE-S2IRT   | 0.76                  |
| Contrast enhancement with 0.5%              | S2IRT      | 0.63                  |
|                                             | SE-S2IRT   | 0.99                  |
| Contrast enhancement with 1.0%              | S2IRT      | 0.62                  |
|                                             | SE-S2IRT   | 0.90                  |
| Contrast enhancement with 4.0%              | S2IRT      | 0.60                  |
|                                             | SE-S2IRT   | 0.69                  |
| Salt & Pepper Noise with the factor of 0.5% | S2IRT      | 0.63                  |
|                                             | SE-S2IRT   | 0.91                  |
| Salt & Pepper Noise with the factor of 1.0% | S2IRT      | 0.62                  |
|                                             | SE-S2IRT   | 0.78                  |
| Salt & Pepper Noise with the factor of 3.0% | S2IRT      | 0.61                  |
|                                             | SE-S2IRT   | 0.70                  |
| Salt & Pepper Noise with the factor of 5.0% | S2IRT      | 0.60                  |
|                                             | SE-S2IRT   | 0.68                  |
| Gaussian noise with standard deviation σ = 10^{-4} | S2IRT | 0.63                  |
|                                             | SE-S2IRT   | 0.78                  |
| Gaussian noise with standard deviation σ = 10^{-3} | S2IRT | 0.62                  |
|                                             | SE-S2IRT   | 0.69                  |
| JPEG compression 90                         | S2IRT      | 0.62                  |
|                                             | SE-S2IRT   | 0.61                  |
| Rotation ±0.75°                             | S2IRT      | 0.62                  |
|                                             | SE-S2IRT   | 0.61                  |
| Image sterilization                         | S2IRT      | 0.61                  |
|                                             | SE-S2IRT   | 0.59                  |

Fig. 6. The t-SNE data of real images and the images generated by (a) S2IRT and (b) SE-S2IRT with and without information hiding. The data of real images, images generated without information hiding, and images generated with hiding payload of 4 bpp is marked by blue, red, and yellow dots, respectively.

| Table 4                                                                 |
|------------------------------------------------------------------------|
| The IE4 Values of Stego-images of The Proposed Approaches Under Different Kinds of Attacks |
| Attack                                      | Approaches | Hiding payloads (bpp) |
|---------------------------------------------|------------|-----------------------|
| Intensity change with 1%                    | S2IRT      | 0.74                  |
|                                             | SE-S2IRT   | 1.00                  |
| Intensity change with 2%                    | S2IRT      | 0.68                  |
|                                             | SE-S2IRT   | 1.00                  |
| Intensity change with 4%                    | S2IRT      | 0.68                  |
|                                             | SE-S2IRT   | 0.87                  |
| Contrast enhancement with 3%                | S2IRT      | 0.70                  |
|                                             | SE-S2IRT   | 1.00                  |
| Contrast enhancement with 5%                | S2IRT      | 0.66                  |
|                                             | SE-S2IRT   | 1.00                  |
| Contrast enhancement with 7%                | S2IRT      | 0.66                  |
|                                             | SE-S2IRT   | 0.88                  |
| Salt & Pepper Noise with the factor of 1%   | S2IRT      | 0.73                  |
|                                             | SE-S2IRT   | 0.99                  |
| Salt & Pepper Noise with the factor of 3%   | S2IRT      | 0.69                  |
|                                             | SE-S2IRT   | 0.91                  |
| Salt & Pepper Noise with the factor of 5%   | S2IRT      | 0.66                  |
|                                             | SE-S2IRT   | 0.91                  |
| Gaussian noise with standard deviation σ = 10^{-4} | S2IRT | 0.68                  |
|                                             | SE-S2IRT   | 0.95                  |
| Gaussian noise with standard deviation σ = 10^{-3} | S2IRT | 0.66                  |
|                                             | SE-S2IRT   | 0.84                  |
| JPEG compression 90                         | S2IRT      | 0.66                  |
|                                             | SE-S2IRT   | 0.80                  |
| Rotation ±0.75°                             | S2IRT      | 0.66                  |
|                                             | SE-S2IRT   | 0.85                  |
| Image sterilization                         | S2IRT      | 1.00                  |
|                                             | SE-S2IRT   | 1.00                  |

| Table 5                                                                 |
|------------------------------------------------------------------------|
| The Computational Complexity of Each Step of S2IRT and SE-S2IRT        |
| Approaches | Hiding Stage | Extraction Stage |
|------------|--------------|------------------|
| S2IRT      | Step(1)      | O(N log N)      |
|            | Step(2)      | O(N^2)          |
|            | Step(3)      | O(N)            |
|            | Step(4)      | O(N^2)          |
|            | Step(1)      | O(N^2)          |
|            | Step(2)      | O(N)            |
| SE-S2IRT   | Step(1)      | O(N log N)      |
|            | Step(2)      | O(N)            |
|            | Step(3)      | O(N)            |
|            | Step(4)      | O(N^2)          |
|            | Step(1)      | O(N^2)          |
|            | Step(2)      | O(N)            |
6.3.4 Robustness

To test and compare the robustness performances of S2IRT and SE-S2IRT, we adopt a set of common attacks including intensity change, contrast enhancement, salt & pepper noise, Gaussian noise, JPEG compression, rotation as well as image sterilization [50], which aims to remove secret data hidden within an image. Then, these attacks are conducted on the generated stego-images. The accuracy rates of information extraction of the two approaches are listed in Table 4.

From Table 4, it can be clearly observed that SE-S2IRT performs better than S2IRT in the aspect of robustness for the following reasons. In SE-S2IRT, once the group No. of one element arranged into one of K adjacent positions is changed, only the position choice Nos. for arranging the elements into the K adjacent positions will be changed, and thus only \( \sum_{i=1}^{K} \lfloor \log_2 (K - i + 1) \rfloor \) bits will be affected at most. On the contrary, in S2IRT, once the group No. of one element is changed, the position choice Nos. for arranging the elements of most groups would be changed. Thus, most of the secret bits encoded as the position arrangement for the elements of most groups would be changed. Thus, most of the secret bits encoded as the position arrangement for the elements of most groups would be changed. Therefore, SE-S2IRT can improve the robustness performance of S2IRT significantly. Also, it is clear that SE-S2IRT achieves good robustness to intensity change, contrast enhancement, image sterilization, and slight noise addition, but relatively low robustness to JPEG compression, rotation, and strong noise addition.

6.3.5 Computational Complexity

The computational complexity of each step of S2IRT and SE-S2IRT schemes is shown in Table 5. From Table 5, the main steps of SE-S2IRT have the same computation complexities with those of S2IRT, except for step (2), since SE-S2IRT uses a more efficient information encoding method, i.e., separate encoding. To further observe the computational complexity of S2IRT and SE-S2IRT schemes, Table 6 lists the time consumption of information hiding and extraction of the two schemes, when generating different sized images for steganography. According to Table 6, in most cases, the time consumption of information hiding and extraction of the two schemes are ranging from several microseconds to several seconds. Thus, if real-time communication is required in some practical scenarios, we can set proper hiding payload and generate moderate-sized stego-images for covert communication to meet the real-time requirement.

6.4 Security Analysis

We analyze the security of the proposed steganographic approaches in this subsection. Suppose an attacker can intercept the shared information, i.e., K and \( n_i \), but cannot obtain the \( Key \), which is protected well by communication participants. According to Sections 4.4 and 5.3, the probabilities of successfully cracking the secret messages hidden by S2IRT and SE-S2IRT can be computed by

\[
P_e = \frac{1}{\prod_{i=1}^{K} s_i}
\]

where, \( s_i = C(N - (i - 1)n, n) \) in the S2IRT scheme and \( s_i = [C(K - i + 1, 1)]^n \) in the SE-S2IRT scheme. The computed \( P_e \) values with different K and n are listed in Table 7.
It is clear that the cracking probabilities of proposed steganographic approaches are very small, and thus they are secure enough against a malicious attack of random guess.

7 Conclusion

This paper has provided a novel idea for generative steganography. It presents two secret-to-image reversible transformation schemes, i.e., S2IRT and SE-S2IRT, based on the Glow model to implement generative steganography. The proposed steganographic approaches can achieve excellent hiding capacity (over 4 bpp) and accurate information extraction (almost 100% accuracy rate), while maintaining desirable anti-detectability and imperceptibility. It outperforms the state-of-the-arts significantly. Moreover, SE-S2IRT can improve the robustness to a variety of common attacks significantly.

In many practical steganographic tasks, it is required to hide a large amount of information in an image while maintaining high extraction accuracy and desirable anti-detectability, imperceptibility, and robustness. The proposed steganographic approaches can handle these tasks well. Hence, we can conclude that the proposed approaches have important practical significance in the area of information hiding. However, our approaches still have some shortcomings. First, due to the small sizes of training datasets and limited image generation ability of Glow model, there is still room for improving the quality of generated images. Second, our approaches are not robust enough to strong image attacks, since the elements of latent vectors mapped from generated images will be affected significantly by those attacks. In future, we attempt to design more sophisticated generative models and build larger-sized training databases to improve the quality of generated images. Moreover, we also consider how to achieve high robustness to strong image attacks for generative steganography.

References

[1] T. Filler, J. Judas, and J. Fridrich, “Minimizing additive distortion in steganography using syndrome-trellis codes,” IEEE Trans. Inf. Forensics Secur., vol. 6, no. 3, pp. 920–935, Sep. 2011.
[2] Z. Zhou, Y. Mu, and Q. J. Wu, “Coverless image steganography using partial-duplicate image retrieval,” Soft. Comput., vol. 23, no. 13, pp. 4927–4938, 2019.
[3] Z. Zhou, H. Sun, R. Harit, X. Chen, and X. Sun, “Coverless image steganography without embedding,” in Proc. Int. Conf. Cloud Comput. Secur., 2015, pp. 123–132.
[4] K.-C. Wu and C.-M. Wang, “Steganography using reversible texture synthesis,” IEEE Trans. Image Process., vol. 24, no. 1, pp. 130–139, Jan. 2015.
[5] S. Li and X. Zhang, “Toward construction-based data hiding: From secrets to fingerprint images,” IEEE Trans. Image Process., vol. 28, no. 3, pp. 1482–1497, Mar. 2019.
[6] D. Hu, L. Wang, W. Jiang, S. Zheng, and B. Li, “A novel image steganography method via deep convolutional generative adversarial networks,” IEEE Access, vol. 6, pp. 38303–38314, 2018.
[7] J. Goodfellow et al., “Generative adversarial nets,” in Proc. Adv. Neural Inf. Process. Syst., 2014, pp. 2672–2680.
[8] M.-M. Liu, M.-Q. Zhang, J. Liu, Y.-N. Zhang, and Y. Ke, “Coverless information hiding based on generative adversarial networks,” 2017, arXiv:1708.04651.
[9] Z. Zhang, G. Fu, R. Ni, J. Liu, and X. Yang, “A generative method for steganography by cover synthesis with auxiliary semantics,” Tsinghua Sci. Technol., vol. 25, no. 4, pp. 516–527, 2020.
[10] Y. Luo, J. Qin, X. Xiang, and Y. Tan, “Coverless image steganography based on multi-object recognition,” IEEE Trans. Circuits Syst. Video Technol., vol. 31, no. 7, pp. 2779–2791, Jul. 2021.
[11] W. Jiang, D. Hu, C. Yu, M. Li, and N. Zhao, “A new steganography without embedding based on adversarial training,” in Proc. Proc. ACM Tering Celebation Conf., 2020, pp. 219–223.
[12] J. Li et al., “A generative steganography method based on WGAN-VP,” in Proc. Int. Conf. Artif. Intell. Secur., 2020, pp. 386–397.
[13] A. Arifinanto et al., “EDGAN: Disguising text as image using generative adversarial network,” in Proc. Int. Conf. Inf. Commun. Technol., 2020, pp. 1–4.
[14] C.-K. Chan and L.-M. Cheng, “Hiding data in images by simple LSB substitution,” Pattern Recognit., vol. 37, no. 3, pp. 469–474, 2004.
[15] J. Mieliikainen, “LSB matching revisited,” IEEE Signal Process Lett., vol. 13, no. 5, pp. 285–287, May 2006.
[16] T. Pevny, T. Filler, and P. Bas, “Using high-dimensional image models to perform highly undetectable steganography,” in Proc. Int. Workshop Inf. Hiding, 2010, pp. 161–177.
[17] V. Holub, J. Fridrich, and T. Denemark, “Universal distortion function for steganography in an arbitrary domain,” EURASIP J. Inf. Secur., vol. 2014, no. 1, 2014, Art. no. 1.
[18] B. Li, M. Wang, J. Huang, and X. Li, “A new cost function for spatial image steganography,” in Proc. IEEE Int. Conf. Image Process., 2014, pp. 4206–4210.
[19] V. Sedighi, R. Cogranne, and J. Fridrich, “Content-adaptive steganography by minimizing statistical detectability,” IEEE Trans. Inf. Forensics Secur., vol. 11, no. 2, pp. 221–234, Feb. 2016.
[20] W. Zhang, Z. Zhang, L. Zhang, H. Li, and N. Yu, “Decomposing joint distortion for adaptive steganography,” IEEE Trans. Circuits Syst. Video Technol., vol. 27, no. 10, pp. 2274–2280, Oct. 2017.
[21] X. Liao, Y. Yu, B. Li, Z. Li, and Z. Qin, “Payload partition strategy in color image,” IEEE Trans. Circuits Syst. Video Technol., vol. 30, no. 3, pp. 685–696, Mar. 2020.
[22] W. Su, J. Ni, X. Hu, and J. Fridrich, “Image steganography with symmetric embedding using gaussian markov random field model,” IEEE Trans. Circuits Syst. Video Technol., vol. 31, no. 2, pp. 1001–1015, Mar. 2021.
[23] W. Tang, S. Tan, B. Li, and J. Huang, “Automatic steganographic distortion learning using a generative adversarial network,” IEEE Signal Process Lett., vol. 24, no. 10, pp. 1547–1551, Oct. 2017.
[24] J. Yang, D. Ruan, J. Huang, X. Kang, and Y.-Q. Shi, “An embedding cost learning framework using GAN,” IEEE Trans. Inf. Forensics Secur., vol. 15, pp. 839–851, 2019.
[25] W. Tang, B. Li, M. Barni, J. Li, and J. Huang, “An automatic cost learning framework for image steganography using deep reinforcement learning,” IEEE Trans. Inf. Forensics Secur., vol. 16, pp. 952–967, 2020.
[26] S. Baluja, “Hiding images within images,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 42, no. 7, pp. 1685–1697, Jul. 2020.
[27] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2015, pp. 3431–3440.
[28] J. Fridrich and M. Goljan, “An estimation of secret message length in LSB steganography in spatial domain,” in Proc. Secur. Steganogr. Watermarking Multimedia Contents VI, 2004, pp. 23–34.
[29] C. Chen and Y. Q. Shi, “JPEG image steganalysis utilizing both intrablock and interblock correlations,” in Proc. IEEE Int. Symp. Circuits Syst., 2008, pp. 3029–3032.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Zhili Zhou (Member, IEEE) received the MS and PhD degrees in computer application from the School of Information Science and Engineering, Hunan University, in 2010 and 2014, respectively. He is currently a professor with the Institute of Artificial Intelligence and Blockchain, Guangzhou University, China. Also, he was a postdoctoral fellow with the Department of Electrical and Computer Engineering, University of Windsor, Canada. His current research interests include multimedia security, information hiding, digital forensics, blockchain, and secret sharing. He got ACM SEGWEB, China Rising Star Award. He is serving as an associate editor of Journal of Real-Time Image Processing, Security and Communication Networks, and International Journal on Semantic Web and Information Systems.

Yuecheng Su currently working toward the MS degree with the Department of Computer and Software, Nanjing University of Information Science and Technology, China. His current research interests include information hiding and digital forensics.

Jin Li received the BS degree in mathematics from Southwest University, in 2002, the MS degree in mathematics from Sun Yat-sen University, in 2004, and the PhD degree in information security from Sun Yat-sen University, in 2007. He is currently a professor and vice dean with the School of Computer Science, Guangzhou University. His research interests include design of secure protocols in Cloud Computing and cryptographic protocols. He has published more than 100 papers in international conferences and journals, including IEEE INFOCOM, IEEE Transactions on Information Forensics and Security, IEEE Transactions on Computers and ESORICS etc. His work has been cited more than 10000 times at Google Scholar and the H-index is 34. He also served as program chair and committee for many international conferences. He received NSFC Outstanding Youth Foundation in 2017. He is serving as editor-in-chief of International Journal of Intelligent Systems, and an associate editor of IEEE Transactions on Dependable and Secure Computing.

Keping YU received the ME and PhD degrees from the Graduate School of Global Information and Telecommunication Studies, Waseda University, Tokyo, Japan, in 2012 and 2016, respectively. He was a research associate, junior researcher, researcher with the Global Information and Telecommunication Institute, Waseda University, from 2015 to 2019, 2019 to 2020, 2020 to 2022, respectively. He is currently an associate professor with the Graduate School of Science and Engineering, Hosei University, Tokyo, Japan. He has authored more than 100 publications including papers in prestigious journals/conferences. He is an associate editor of IEEE Open Journal of Vehicular Technology, Journal of Intelligent Manufacturing, Journal of Circuits, Systems and Computers. His research interests include smart grids, information-centric networking, the Internet of Things, artificial intelligence, blockchain, and information security.
Q. M. Jonathan Wu (Senior Member, IEEE) received the PhD degree in electrical engineering from the University of Wales, Swansea, United Kingdom, in 1990. He was affiliated with the National Research Council of Canada for ten years beginning, in 1995, where he became a senior research officer and a group leader. He is currently a professor with the Department of Electrical and Computer Engineering, University of Windsor, Windsor, Ontario, Canada. He has published more than 300 peer-reviewed papers in computer vision, image processing, intelligent systems, robotics, and integrated microsystems. His current research interests include 3-D computer vision, active video object tracking and extraction, interactive multimedia, sensor analysis and fusion, and visual sensor networks. He is the fellow of the Canadian Academy of Engineering. He is an associate editor for the IEEE Transaction on Cybernetics, the IEEE Transactions on Circuits and Systems for Video Technology. He has served on technical program committees and international advisory committees for many prestigious conferences.

Zhangjie Fu (Member, IEEE) received the PhD degree in computer science from the College of Computer, Hunan University, China, in 2012. He is currently a professor with the School of Computer, Nanjing University of Information Science and Technology, China. His research interests include cloud and outsourcing security, digital forensics, network and information security. His research has been supported by NSFC, PAPD, and GYHY. He is a member of ACM.

Yunqing Shi (Life Fellow, IEEE) received the MS degree from Shanghai Jiao Tong University, China, and the PhD degree from the University of Pittsburgh. In 1987, he joined the New Jersey Institute of Technology. He is the author or co-author of more than 400 articles, one book, five book chapters, and an editor of ten books. He holds 30 U.S. patents. His research interests include data hiding, forensics, information assurance, and visual signal processing and communications. He has been a member of a few IEEE technical committees and a fellow of the National Academy of Inventors (NAI) since 2017. He has served as the Technical Program Chair for the IEEE ICME07, the co-Technical chair for the IEEE MMSP05, as well as IWWD since 2006, the Co-General Chair for the IEEE MMSP02, and a Distinguished Lecturer for the IEEE CASS. He has served as an Associate Editor for the IEEE Transactions on Signal Processing, the IEEE Transactions on Information Forensics and Security, and the IEEE Transactions on Circuits and Systems and an editorial board member for few journals.

For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/csdl.