Robust Generative Steganography for Image Hiding Using Concatenated Mappings

Liyan Chen, Bingwen Feng, Zhihua Xia, *Member, IEEE*, Wei Lu, *Member, IEEE*, and Jian Weng, *Senior Member, IEEE*

***Abstract*—Generative steganography stands as a promising technique for information hiding, primarily due to its remarkable resistance to steganalysis detection. Despite its potential, hiding a secret image using existing generative steganographic models remains a challenge, especially in lossy or noisy communication channels. This paper proposes a robust generative steganography model for hiding full-size image. It lies on three reversible concatenated mappings proposed. The first mapping uses VQGAN with an order-preserving codebook to compress an image into a more concise representation. The second mapping incorporates error correction to further convert the representation into a robust binary representation. The third mapping devises a distribution-preserving sampling mapping that transforms the binary representation into the latent representation. This latent representation is then used as input for a text-to-image Diffusion model, which generates the final stego image. Experimental results show that our proposed scheme can freely customize the stego image content. Moreover, it simultaneously attains high stego and recovery image quality, high robustness, and provable security.**

***Index Terms*—Generative steganography, reversible mappings, customizability, robustness, provable security.**

I. Introduction

IMAGE steganography aims to conceal secret messages within unassuming stego images, ensuring that no suspicion is aroused among unauthorized individuals. Only those with the proper authorization can retrieve the hidden information. This technique holds paramount importance in the realm of multimedia security and privacy preservation. Given the pervasive use of digital images for conveying vital information across various sectors, including military, medical, and commercial domains, image steganography that can hide secret images has emerged as a pivotal subset within this field. It calls for that a steganographic method can integrate the secret image into the stego image without degrading its visual quality or introducing detectable anomalies.

Traditional steganography modifies a natural cover image to embed a secret message, yielding a stego image that closely mirrors the cover for covert communication. Early techniques, for instance, embedding message bits in the least significant bits (LSBs) of pixels [1], [2], often led to noticeable changes in the statistics of natural images. To mitigate this issue, content-adaptive steganographic methods such as Syndrome-Trellis Codes (STC) [3], [4] and Steganographic Polar Codes (SPC) [5] were developed. These methods encode messages in a way that minimizes heuristically defined distortion functions. With the advent of deep learning, numerous approaches have leveraged various network models to conceal secret data [6], [7], [8]. By acknowledging that perfect encoding of secret images is not necessary, Baluja [9] pioneered the successful hiding of an image within another using neural networks. Recently, Invertible Neural Network (INN)-based image steganography models [10], [11], [12], [13], [14] have emerged, effectively harnessing the capabilities of INNs to establish invertible mappings between cover+secret and stego images. Despite these advancements, all aforementioned methods still require modifying cover images, inevitably introducing statistical anomalies. Consequently, the attackers can develop steganalytic tools to discern the distributions of cover and stego images [15], [16], [17].

In contrast, generative steganography eliminates the drawbacks of modification-based methods by directly synthesizing stego images driven by secret messages. This characteristic theoretically enhances the security of steganography. Wei et al. [18] hide secret data in feature maps using StyleGAN [19]. Su et al. [20] develop a distribution-preserving secret data modulator to achieve provable security. Liu et al. [21] exploit the stability of structural features to robustly hide secret messages. However, these methods may require extensive and costly training due to their intricate adversarial objectives [22]. Some approaches have employed Flow models [23], [24]. Wei et al. [23] establish a reversible bijective mapping between

Received 9 December 2024; revised 30 April 2025; accepted 16 May 2025. Date of publication 29 May 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 62261160653, and Grant U23B2023; 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 Preserving under Grant 02023B1212060036; and in part by the Opening Project of 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. Jan Butora. *(Corresponding author: Bingwen Feng.)*

Liyan Chen, Bingwen Feng, and Jian Weng are with the College of Cyber Security, Jinan University, Guangzhou 510632, China (e-mail: chenliyan@stu.jnu.edu.cn; bingwfeng@gmail.com; cryptjweng@gmail.com).

Zhihua Xia is with the College of Cyber Security, Engineering Research Center of 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, Guangdong Province Key Laboratory of Information Security Technology, Ministry of Education Key Laboratory of Machine Intelligence and Advanced Computing, Sun Yat-sen University, Guangzhou 510006, China (e-mail: luwei3@mail.sysu.edu.cn).

Digital Object Identifier 10.1109/TIFS.2025.3573669

secret data and generated stego images using Glow [25], while Zhou et al. [24] design a reversible mapping between secret bits and latent vectors. Recently, thanks to their powerful generative capabilities, Diffusion models have been adopted as the foundation. Yang et al. [26] map the watermark to latent representation following a standard Gaussian distribution, while Hu et al. [27] design an orthogonal transformation kernel to transform binary messages into Gaussian-distributed latent vectors. These methods create diverse mappings capable of maintaining the input distributions within the utilized generative models, thereby ensuring provable steganographic security. However, they may not provide sufficient capacity to conceal entire secret images. Yu et al. [28] build a secret-stego invertible mapping known as CRoSS, and DiffStega [29] further employs pre-determined passwords instead of text prompts alone to ensure security. Nevertheless, these methods can only conceal a portion of a secret image, leaving a significant part unchanged.

The mapping between message and latent representation space is pivotal in generative methods. However, existing mappings often fall short in accommodating a message space with sufficient dimension to represent secret images. Furthermore, the inherent numerical instability and sensitivity to perturbations in generative models pose significant challenges for achieving robustness [12], [27]. Consequently, it is often necessary to sacrifice limited capacity to enhance robustness. Given that images inherently contain significant redundancy, compressing the image before mapping it to the latent representation presents a prudent approach. Techniques such as vector quantization [30], [31], image degradation model [32], and dictionary learning [33] can learn compact representations of high-dimensional data. Among them, VQGAN [31] stands out as a formidable learned image compression architecture, integrating a learnable vector-quantization codebook with compressive VAEs. By leveraging this technique, we can significantly reduce the amount of information that needs to be concealed. However, the compressed representations generated by VQGAN are sensitive to distortions. To address this issue, we propose an order-preserving coding to bolster the robustness of VQGAN.

In this paper, we propose a robust generative steganographic scheme that can hide an image within a stego image of the same size. We utilize the concepts of neural image compression to craft a series of robust mappings that not only enhance capacity but also guarantee robustness. Specifically, three invertible mappings are designed. The first mapping compresses an image into a more concise representation by leveraging a robust version of VQGAN. The second mapping introduces an error correction mechanism by appending check codes. In the third mapping, we devise a distribution-preserving sampling mapping that transforms the secret information into a latent representation that follows a standard Gaussian distribution. Finally, by employing a text-to-image Diffusion model, users can freely customize the content of stego images. The proposed scheme can generate stego images that are indistinguishable from typical AI-generated ones, which also present remarkable robustness. Our contributions can be summarized as follows:

- A robust generative image steganography scheme that is founded on three reversible concatenated mappings is presented. It enables the hiding of an image within a user-customizable AI-generated image of the same size, fulfilling the desirable properties of security, diversity, robustness, and capacity, simultaneously.
- An order-preserving codebook is introduced in the first mapping. It is constructed using the proposed hierarchical clustering-based index assignment algorithm. By leveraging the capabilities of VQGAN in conjunction with this codebook, the first mapping achieves efficient compression efficiency and robustness.
- A distribution-preserving noise sampling method is suggested in the third mapping. This method ensures that only the coefficients with the largest absolute amplitude are used to carry message bits. Additionally, a dual-sample strategy is employed to append the secret information. The third mapping can provide provable security while further enhancing the robustness.

II. Related Work

A. Diffusion Models

Diffusion models [34], [35] have emerged as a powerful framework for generative tasks, enabling the synthesis of high-quality images from Gaussian noise through progressive denoising. The training procedure consists of two processes: the forward diffusion process and the backward denoising process. In the forward process, Gaussian noise of varying scales is gradually added to the training image at each step. In the backward process, the original image is recovered by sequentially removing the noise, which is estimated by a neural network based on U-Net architecture [36]. To reduce computational costs of directly operating in pixel space while retaining the quality and controllability, the Latent Diffusion Model (LDM) [37] applies diffusion processes in the latent space of pretrained autoencoders. By using deterministic sampling such as DDIM [34] and EDICT [35], Diffusion can identify the initial noise that generates the image during the diffusion process. In this paper, we use Stable Diffusion to construct a novel covert channel for transmitting secret images.

B. Diffusion-Based Generative Image Steganography

Compared to other generative models, Diffusion-based image steganography models have become a focal point in generative steganography due to their exceptional image generation quality. Most of them are based on DDIM Inversion [34] to achieve reversible mapping between the initial noise distribution and the generated image distribution. Yang et al. [26] and Hu et al. [27] embed secret bits into the initial noise latent, achieving great robustness and diversity in the stego image. However, constrained by the limited embedding size of the initial noise latent, [26] can only hide 256 bits (capacity is 0.000975 bpp (bits per pixel)), while [27] can hide 16384 bits (capacity is 0.0625 bpp). For high-capacity models, CRoSS [28] creates a mapping between secret and stego image of the same size by using two DDIM Inversion loops with different prompts to achieve dual-direction image translation

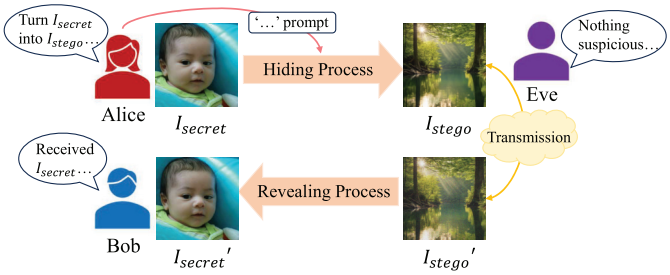

Fig. 1. Application scenario.

between them. Based on CRoSS, DiffStega [29] overcomes the risk of text prompts leakage in CRoSS by incorporating pre-determined passwords. However, even though they can formally achieve image-in-image concealment (capacity is 24 bpp), they struggle to conceal the overall content of the secret image. This is because image translation in CRoSS and DiffStega can only replace specific components of the secret image, leaving noticeable overlaps between the secret and stego images. In this paper, we propose a controllable and robust generative steganography method that is capable of hiding full-size images and overcoming the shortcomings related to the similarities between secret and stego images.

III. Proposed Method

A. Application Scenario

This paper considers a covert communication application based on AI-generated images. As shown in Fig. 1, there are three roles in the scenario: the sender Alice, the receiver Bob, and the potential attacker Eve. Alice synthesizes a stego image I_{stego} with the secret image I_{secret} hidden in it. I_{stego} is then transmitted through a public channel without arousing Eve's suspicions. Note that the channel may introduce distortions, such as noise and compression, leading Bob to receive a degraded version of the stego image, denoted as I'_{stego}. Consequently, there is a need for Bob to be able to recover an approximate secret image I'_{secret} from I'_{stego}.

To achieve this, the generative image steganography requires the following four essential properties:

- **Security**: We consider security from three dimensions: 1) provable steganographic security, ensuring that Eve, regardless of her steganalysis prowess, cannot detect the presence of secret information. 2) perfect secrecy, which mandates that the stego image I_{stego} cannot be recovered without the decryption keys, even if Eve has access to the secret extractor. 3) undetectability, where the generated I_{stego} maintains high visual fidelity, rendering it indistinguishable from the output produced by comparable image synthesis tools.
- **Diversity**: Alice must have the liberty to customize the stego image I_{stego} to suit various application contexts, unfettered by any particular stylistic or content constraints. It is in fact a one-to-many mapping, where a single secret image corresponds to multiple potential candidates. Nevertheless, the large volume of information in I_{secret} poses challenges in maintaining independence between the styles and content of I_{secret} and I_{stego}, particularly in generative methods, often resulting in limited diversity.
- **Robustness**: Bob must be able to retrieve a high quality approximation of the secret image $I'_{secret} \approx I_{secret}$ from the degraded stego image I'_{stego}. This necessitates steganographic strategies that can withstand common signal manipulations like JPEG compression and Gaussian noise. However, achieving robustness often comes at the sacrifice of hiding capacity, exacerbating the challenge of concealing an entire image.
- **Capacity**: The scheme must possess sufficient capacity to hide a secret image. However, preserving the diversity of I_{stego} necessitates a one-to-many mapping within the generative model. This introduces a constraint where the space of secret images is inherently smaller than that of stego images, conflicting with the aspiration of maintaining equal-sized spaces for hiding image in image.

B. Network Framework

We introduce a generative steganographic network designed to fulfill the four properties mentioned above. The architecture of the proposed network is illustrated in Figure 2. It comprises two primary components: an embedder and an extractor. The embedder takes a secret image and a prompt as inputs. It then generates a stego image that is in harmony with the provided prompt. The extractor, on the other hand, is tasked with recovering the secret image from the potentially distorted stego image. Both the embedder and the extractor are constructed as compositions of three concatenated invertible mappings, followed by a generative Diffusion model.

Mapping Γ_1 for Perceptual Image Compression compresses an RGB image into a more concise representation by leveraging a robust vector-quantized codebook. This transformation creates a favorable representation for subsequent robust coding endeavors.

Mapping Γ_2 for Error-Resilient Presentation further enhances the robustness by appending check codes to the output of Γ_1. This step imparts error correction capabilities, mitigating potential transmission errors.

Mapping Γ_3 for Distribution-Preserving Sampling transforms the obtained codes into a latent space whose distribution aligns with the input requirements of the Diffusion model, thus enabling its utilization as input for generating stego images. This mapping not only reinforces the robustness but also ensures the provable steganography security.

Consequently, the embedder can be articulated as:

$$I_{stego} = \mathrm{Diff}(\Gamma_3(\Gamma_2(\Gamma_1(I_{secret})))) \tag{1}$$

Here, Diff represents a Diffusion model that translates a latent representation into an image. For the realization of this model, we employ a pre-trained, shared text-to-image Diffusion model [37], leveraging its capabilities to synthesize high-fidelity images. Correspondingly, the extractor can be formed as:

$$I'_{secret} = \Gamma_1^{-1}(\Gamma_2^{-1}(\Gamma_3^{-1}(\mathrm{Diff}^{-1}(I'_{stego})))) \tag{2}$$

The three mappings are detailed in the following sections.

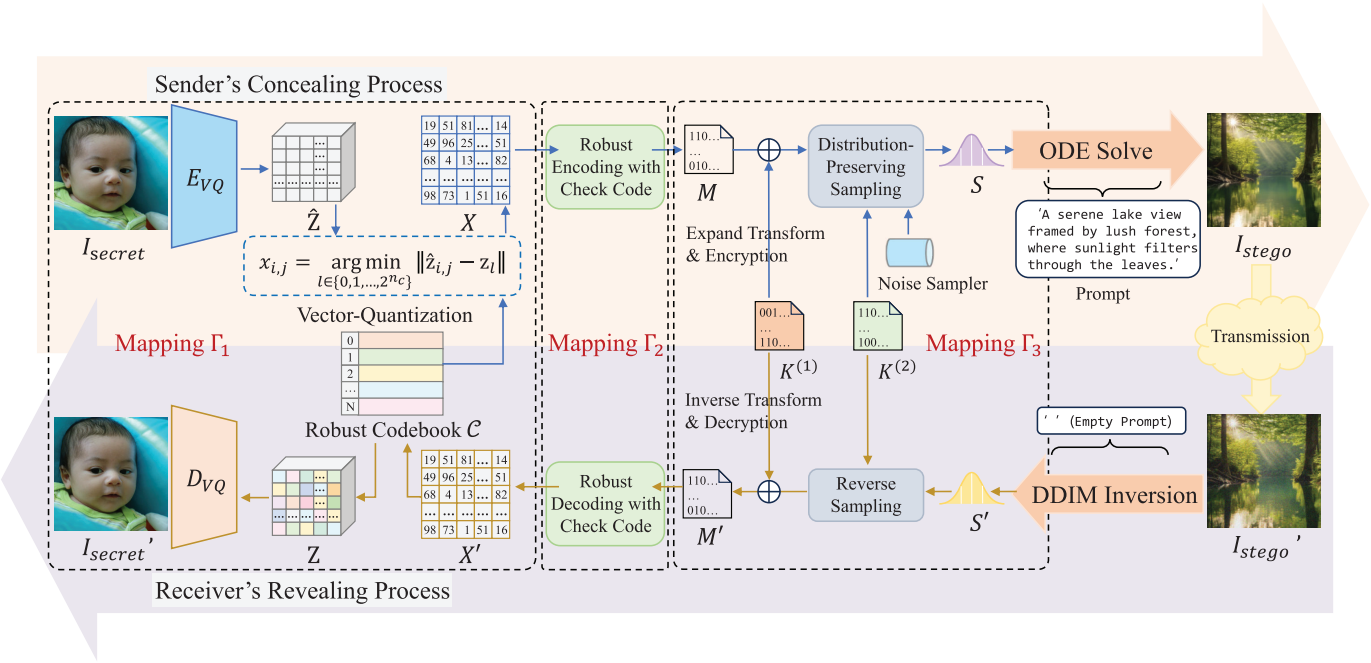

Fig. 2. The proposed generative image steganography framework.

C. Mapping Γ_1 for Perceptual Image Compression

1) Compression Based on VQGAN: The first mapping is implemented by Perceptual Image Compression based on VQGAN [31], which consists of an encoder E_{VQ}, a decoder D_{VQ}, and a discrete codebook $\mathcal{C} = \{z_l\}_{l=1}^{2^{n_c}} \subset \mathbb{R}^{n_z}$ with 2^{n_c} codewords, where n_z is the dimension of each codeword. An image $I \in \mathbb{R}^{n_H \times n_W \times 3}$ is represented by a collection of codewords $Z \in \mathbb{R}^{n_{hb} \times n_{wb} \times n_z}$. The (i, j)-th entry of Z, denoted as $z_{i,j}$, is obtained by quantizing the (i, j)-th entry of $\hat{Z} = E_{VQ}(I) \in \mathbb{R}^{n_{hb} \times n_{wb} \times n_z}$ to the closest codeword in $\mathcal{C}$:

$$z_{i,j} = \left(\arg\min_{z_l \in \mathcal{C}} \left\| \hat{z}_{i,j} - z_l \right\| \right) \in \mathbb{R}^{n_z} \tag{3}$$

The decoder D_{VQ} maps Z back to the image space to realize high quality reconstruction $\hat{I} \approx I$:

$$\hat{I} = D_{VQ}(Z) \tag{4}$$

In the original VQGAN, each $\frac{n_H}{n_{hb}} \times \frac{n_W}{n_{wb}}$ size image block is represented by a floating-point codeword $z_{i,j}$ of length n_z. However, the volume of $z_{i,j}$ is still too large to be hidden. As a result, we use the index x of the codeword to represent the image block, which is given by:

$$x_{i,j} = \left(\arg\min_{l \in \{0,1,\cdots,2^{n_c}\}} \left\| \hat{z}_{i,j} - z_l \right\| \right) \in \{0, 1, \cdots, 2^{n_c} - 1\} \tag{5}$$

Consequently, each decimal index $x_{i,j}$ can represent an image block of size $\frac{n_H}{n_{hb}} \times \frac{n_W}{n_{wb}}$, which serves as the output of Γ_1. Then, the image-to-indices bijective mapping Γ_1 can be formed as:

$$\Gamma_1 : I \in \mathbb{R}^{n_H \times n_W \times 3} \mapsto X \in \{0, \cdots, 2^{n_c} - 1\}^{n_{hb} \times n_{wb}} \tag{6}$$

The codebook $\mathcal{C}$ is shared by both sender and receiver. Therefore, with the index collection X, the original image I can be reconstructed with high quality.

2) Robust Codebook Generation: The inversion of Γ_1, denoted as Γ_1^{-1}, maps an index $x_{i,j}$ in a compact space to an image block in a much larger space. Consequently, any errors present in $x_{i,j}$ undergo significant amplification in the recovered image I'_{secret}. To enhance the robustness of Γ_1, we devise an order-preserving codebook $\mathcal{C}^o$, ensuring that indices with close proximity are mapped to codewords exhibiting small differences. Specifically, for any $z_l \in \mathcal{C}^o$ indexed by l, the following property holds:

$$L_z(z_{l_1}, z_{l_2}) < L_z(z_{l_1}, z_{l_3}) \tag{7}$$

$$\text{if } L_l(l_1, l_2) < L_l(l_1, l_3) \tag{8}$$

where $L_l(l_1, l_2)$ denotes a distance metric between l_1 and l_2, and $L_z(z_{l_1}, z_{l_2})$ denotes a distance metric between z_{l_1} and z_{l_2}. Since l will be encoded in binary form via Γ_2, we adopt the Hamming distance as L_l, measuring the difference between the binary representations of l_1 and l_2. This design of the codebook ensures that minimal errors in indices translate into minimal disturbances in codewords, thereby mitigating degradation in the recovered image.

To construct $\mathcal{C}^o$, we introduce an index assignment algorithm based on hierarchical clustering. It groups all the codewords from the original codebook in a top-down, divisive fashion, iteratively assigning unused indices to the central codeword of each formed cluster. The index assignment is detailed in Algorithm 1, which operates recursively. To tackle the challenge posed by the curse of dimensionality in high-dimensional data, which frequently undermines clustering performance, we utilize dimensionality reduction algorithms, such as UMAP [38]. It maps codewords into two- or three-dimensional space, while meticulously preserving their intricate patterns. Our algorithm then operates within this dimensionality-reduced codebook. At each iteration, the

Algorithm 1 *Index_Assignment* $\left(\mathfrak{C}, \mathfrak{Z}, n_f\right)$

Input: Cluster set $\mathfrak{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_n\}$, central codewords for each cluster $\mathfrak{Z} = \left\{z_{ct_1}, z_{ct_2}, \ldots, z_{ct_n}\right\}$, Number of flipping bits n_f

Output: Robust codebook $\mathcal{C}^o$

for $i \leftarrow 1$ **to** n **do**
  if *the number of clusters* $n = 1$ **then**
    Set the number of clusters for the next clustering: $n' \leftarrow 2$.
  else
    Set $n' \leftarrow \min\left(\text{number of codewords in } \mathcal{C}_i,\ \text{permutation number}(n_c, n_f)\right)$.
  end
  $\left\{\mathcal{C}_1^*, \mathcal{C}_2^*, \cdots, \mathcal{C}_{n'}^*\right\} \leftarrow Clustering(\mathcal{C}_i, n')$
  $\mathfrak{L}^* \leftarrow Index\_Generation\left(ct_i, n_f, n', \mathcal{C}^o\right)$, where ct_i is the index of z_{ct_i}.
  Initialize distance set $\mathfrak{D} \leftarrow \emptyset$.
  for $j \leftarrow 1$ **to** n' **do**
    Find $z_k \in \mathcal{C}_j^*$ closest to the center of $\mathcal{C}_j^*$, and set it as the central codeword of $\mathcal{C}_j^*$, namely $z_{ct_j}^*$.
    Find the nearest uncle codeword of $z_{ct_j}^*$ by $z_{un_j} \leftarrow \underset{z_{ct_k} \text{for each } \mathcal{C}_k \in \mathfrak{C} \backslash \{\mathcal{C}_j\}}{\arg\min} L_z\left(z_{ct_k}, z_{ct_j}^*\right)$.
    Set $d_j \leftarrow L_z\left(z_{un_j}, z_{ct_j}^*\right)$, then $\mathfrak{D} \leftarrow \mathfrak{D} \cup \{(d_j, z_{un_j})\}$.
  end
  Rearrange $\mathfrak{D}$ in ascending order based on the values of d_j.
  foreach (d_j, z_{un_j}) *in* $\mathfrak{D}$ **do**
    Set $ct_j^* \leftarrow \underset{l_k^* \in \mathfrak{L}^*}{\arg\min} L_l\left(l_k^*, un_j\right)$ where un_j is the index of z_{un_j}.
    Assign the index of $z_{ct_j}^*$ with ct_j^*, yielding $z_{ct_j^*}^*$.
    Update $\mathcal{C}^o \leftarrow \mathcal{C}^o \cup \{z_{ct_j^*}^*\}$.
  end
  if *number of elements in* $\mathcal{C}_i > n_c$ **then**
    $Index\_Assignment\left(\left\{\mathcal{C}_1^*, \mathcal{C}_2^*, \ldots, \mathcal{C}_k^*\right\}, \left\{z_{ct_1}^*, z_{ct_2}^*, \ldots, z_{ct_k}^*\right\}, \max(\lfloor n_f/4 \rfloor, 1)\right)$
  end
end

number of subclusters n' is first estimated. Then a clustering method, such as K-Means [39], is employed to obtain these subclusters. Subsequently, the *Index_Generation* function (detailed in Algorithm 2) generates n' unused indices $\mathfrak{L}^*$, ensuring that for any $l_i^*, l_j^* \in \mathfrak{L}^*$, it satisfies that $L_l(l_i^*, l_j^*) \leq 2 \times n_f$. Finally, these indices are assigned to the central codeword of each subcluster.

In Algorithm 1, n_f serves to regulate the Hamming distance among the assigned indices. Empirically, we find that semantically related codewords can be effectively clustered and mapped to contiguous indices during the early stages of the algorithm. Two constraints should be observed in order to maintain a good order-preserving property.

(1) **Intra-cluster distance constraint**. The codeword in a subcluster $\mathcal{C}_{i^*}^*$ should be closer to the central codeword of its parent cluster $\mathcal{C}_j$ than to any other uncle cluster's central codeword. Mathematically, this can be expressed as

$$L_l\left(l_i^*, ct_j\right) < L_l\left(l_i^*, ct_{j'}\right),\ \forall z_{l_i^*}^* \in \mathcal{C}_{i^*}^*, \forall \mathcal{C}_{j'} \in \mathfrak{C}/\{\mathcal{C}_j\} \quad (9)$$

where l_i^* denotes the index of $z_{l_i^*}^*$. z_{ct_j} and $z_{ct_{j'}}$ denote the central codewords of $\mathcal{C}_j$ and $\mathcal{C}_{j'}$, respectively, with ct_j and $ct_{j'}$ being their indices. To achieve this, the following distance constraint should be maintained:

$$L_l\left(ct_i^*, ct_j\right) < 4 \times L_l\left(ct_j, ct_{j'}\right),\ \forall \mathcal{C}_{j'} \in \mathfrak{C}/\{\mathcal{C}_j\} \quad (10)$$

Algorithm 2 *Index_Generation*$\left(l, n_f, n, \mathcal{C}^o\right)$

Input: Seed index l, Number of flipping bits n_f, Number of generated indices n, Set of correct robust codebook $\mathcal{C}^o$

Output: The set of unused indices $\mathfrak{L}^* = \left\{l_1^*, l_2^*, \cdots, l_n^*\right\}$

Initialize $i \leftarrow 1$.
while *true* **do**
  Generate all the indices $\mathfrak{L}$ that can be obtained by flipping n_f bits of l.
  foreach l_j *in* $\mathfrak{L}$ **do**
    if $l_j \notin \mathcal{C}^o$ **then**
      Set $l_i^* \leftarrow l_j$ and update $\mathfrak{L}^* \leftarrow \mathfrak{L}^* \cup \{l_i^*\}$.
      $i = i + 1$.
      if $i = n$ **then**
        return $\mathfrak{L}^*$.
      end
    end
  end
  $n_f = n_f + 1$.
end

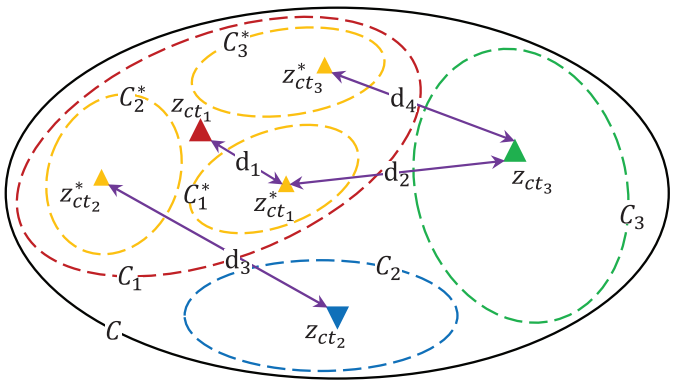

Fig. 3. Example of index assignment process: After the first iteration, all the codewords $\mathcal{C}$ have been clustered into 3 subclusters $\mathcal{C}_1$, $\mathcal{C}_2$, $\mathcal{C}_3$ and their central codewords $z_{ct_1}, z_{ct_2}, z_{ct_3}$ have been assigned with indices. In the second iteration, $\mathcal{C}_1$ have been clustered into $\mathcal{C}_1^*$, $\mathcal{C}_2^*$, $\mathcal{C}_3^*$ and their central codewords $z_{ct_1}^*, z_{ct_2}^*, z_{ct_3}^*$ need to be assigned. Denote $d_1 = L_l\left(ct_1^*, ct_1\right)$, $d_2 = L_l\left(ct_1^*, ct_3\right)$, $d_3 = L_l\left(ct_2^*, ct_2\right)$, $d_4 = L_l\left(ct_3^*, ct_3\right)$. The Intra-cluster distance constraint is to assure $d_1 < d_2$, the Inter-cluster distance constraint is to assure $d_4 < d_3$.

where ct_i^* is the index of $z_{ct_i^*}^*$, which is the central codeword of $\mathcal{C}_{i^*}^*$. This requires that the n_f used for the next iteration of *Index_Assignment* should be smaller than $\lceil n_f/4 \rfloor$ (See Line 22 in Algorithm 1).

(2) **Inter-cluster distance constraint**. If a cluster is closer to one of its uncle clusters than to others, the index of its central codeword should reflect this proximity. Formally, for $\mathcal{C}_{i^*}^* \subset \mathcal{C}_j$ and its uncle clusters $\mathcal{C}_k, \mathcal{C}_{k'} \in \mathfrak{C}/\{\mathcal{C}_j\}$,

$$L_l\left(ct_i^*, ct_k\right) < L_l\left(ct_i^*, ct_{k'}\right) \tag{11}$$

$$\text{if } L_z\left(z_{ct_i^*}^*, z_{ct_k}\right) < L_z\left(z_{ct_i^*}^*, z_{ct_{k'}}\right) \tag{12}$$

Line 9 to 20 in Algorithm 1 is used to assign indices so that they can satisfy Eq. (11).

Figure 3 illustrates an example of these two constraints.

D. Mapping Γ_2 for Error-Resilient Presentation

The second mapping Γ_2 converts the decimal index collection X into its robust binary presentation M. Note that in its inverse mapping Γ_2^{-1}, even a slight dispersion of errors across the extracted binary indices can significantly perturb a substantial portion of indices upon their conversion back to decimal form. To mitigate this issue, we append a check code that serves to detect and correct potential errors.

Given the index collection $X \in \{0, \cdots, 2^{n_c} - 1\}^{n_{hb} \times n_{wb}}$, it is first converted into its binary presentation by:

$$m_{i,j,k} = \lfloor x_{i,j}/2^{(k-1)} \rfloor \mod 2, \ k \in \{1, \cdots, n_c\} \tag{13}$$

After that, a binary check code of length 2^{n_c} is generated, which is denoted as m_{chk}. Specifically, the l-th bit of m_{chk} is set to 1 if and only if there is an $x_{i,j}$ whose value is exactly l. Otherwise, it is set to 0. This check code is then appended to the end of M, constituting the output of Γ_2. Formally, the decimal-to-binary mapping Γ_2 can be described as:

$$\Gamma_2 : X \in \{0, \cdots, 2^{n_c} - 1\}^{n_{hb} \times n_{wb}} \mapsto M \in \{0, 1\}^{n_{hb} \times n_{wb} \times n_c + 2^{n_c}} \tag{14}$$

At the receiver, the binary code M' is extracted from the received image I'_{stego}. The extracted indicator code m'_{chk} is then used to validate each $x'_{i,j}$ derived through the inverse mapping Γ_2^{-1}. If $x'_{i,j}$ is recovered as l' yet the l'-th bit of m'_{chk} is 0, an extraction error is flagged. Given the assumption that the majority of M' has been accurately extracted, we posit that there is at most one single bit error in either $x'_{i,j}$ or m'_{chk}. As a result, we iterate through all possible n_c indices by flipping a single bit in $x'_{i,j}$ and ascertain whether there exists an index value l'' for which the l''-th bit of m'_{chk} is 1. If such an index is identified, we conclude that the error lies in $x'_{i,j}$ and accordingly set $x'_{i,j} = l''$. Conversely, if no such index is found, we deduce that the l'-th bit of m'_{chk} is erroneous and proceed to flip it, while retaining $x'_{i,j} = l'$. Through this correction logic, we address potential errors in both the message and the check code itself.

E. Mapping Γ_3 for Distribution-Preserving Sampling

The third mapping Γ_3 transforms the binary code M to the final sampled noise $S \sim \mathcal{N}(0, 1)$ that can be used as the input to the designated Diffusion model. Moreover, Γ_3 plays a pivotal role in enhancing robustness and ensuring perfect secrecy.

To address the Diffusion model's sensitivity to diverse distortions, we employ a dual-sample strategy within the set Z as a cover to embed each bit of M. Initially, M is augmented twofold via the following transformation:

$$\tilde{m}_{2i-1} = m_i \tag{15}$$

$$\tilde{m}_{2i} = m_i \oplus 1 \tag{16}$$

where $\tilde{m}_i$ represents the i-th bit in the expanded $\tilde{M} \in \{0, 1\}^{2 \times (n_{hb} \times n_{wb} \times n_c + 2^{n_c})}$. Subsequently, $\tilde{M}$ will be encrypted to $\vec{M}$ with a secret key $K^{(1)} \in \{0, 1\}^{2 \times (n_{hb} \times n_{wb} \times n_c + 2^{n_c})}$ using a stream cipher:

$$\vec{m}_i = \tilde{m}_i \oplus k_i^{(1)} \tag{17}$$

It can be noted that the encrypted bit $\vec{m}_i$ follows a uniform distribution:

$$p(\vec{m}_i = 0) = p(\vec{m}_i = 1) = \frac{1}{2} \tag{18}$$

Next, a latent representation $S \in \mathbb{R}^{n_{hi} \times n_{wi} \times n_{ci}}$ will be sampled as the input of the Diffusion model. Coefficients possessing larger absolute amplitudes exhibit a heightened capacity to accommodate numerical shifts, thereby diminishing the likelihood of message extraction errors. Leveraging this principle, the dual-sample strategy employs two substantial coefficients with opposite signs for message mapping, significantly enhancing its robustness. As a result, we use $2 \times (n_{hb} \times n_{wb} \times n_c + 2^{n_c})$ coefficients with the largest absolute amplitudes to carry message bits. First, the positions of these coefficients are selected in S using a pseudorandom permutation with secret key $K^{(2)}$. It is executed by permuting S through a chaotic pixel shuffle [40], followed by selecting the first specified length of positions from the permuted S. Let p_s denote the probability that the position s in S is selected. This probability, p_s, is equivalent to that of a simple random sampling in S:

$$p_s = \frac{2 \times (n_{hb} \times n_{wb} \times n_c + 2^{n_c})}{n_{hi} \times n_{wi} \times n_{ci}} \tag{19}$$

By combining Eq. (18), it can be derived that the probability of a position being selected to carry $\vec{m}_i = 0$ or $\vec{m}_i = 1$ is equal to:

$$p(s_i, \vec{m}_i = 0) = p(s_i, \vec{m}_i = 1) = p_s \cdot p(\vec{m}_i = 0) = \frac{p_s}{2} \tag{20}$$

We then sample a coefficient in $\mathcal{N}(0,1)$ according to the following rule. Let $pdf(x)$ denote the probability density function of the Gaussian distribution $\mathcal{N}(0,1)$, and ppf denote its corresponding quantile function. We divide $pdf(x)$ into three cumulative probability portions by using thresholds $\tau = ppf(p_s/2)$ and $-\tau = ppf(1-p_s/2)$. When a coefficient s_i is selected to represent $\vec{m}_i = 0$, s_i should fall into the interval $(-\infty, ppf(p_s/2)]$. When s_i is selected to represent $\vec{m}_i = 1$, s_i should fall into $(ppf(1-p_s/2), +\infty)$. Otherwise, s_i falls into $(ppf(p_s/2), ppf(1-p_s/2)]$. This implies that s_i should follow the conditional distribution:

$$p\left(s_i \mid \vec{m}_i = 0\right) = \begin{cases} \frac{2pdf(s)}{p_s} & \text{if } s_i \in \left(-\infty, ppf\left(\frac{p_s}{2}\right)\right] \\ 0 & otherwise \end{cases} \tag{21}$$

$$p\left(s_i \mid \perp\right) = \begin{cases} \frac{pdf(s)}{1-p_s} & \text{if } s_i \in \left(ppf\left(\frac{p_s}{2}\right), ppf\left(1-\frac{p_s}{2}\right)\right] \\ 0 & otherwise \end{cases} \tag{22}$$

$$p\left(s_i \mid \vec{m}_i = 1\right) = \begin{cases} \frac{2pdf(s)}{p_s} & \text{if } s_i \in \left(ppf\left(1-\frac{p_s}{2}\right), +\infty\right) \\ 0 & otherwise \end{cases} \tag{23}$$

where $\perp$ denotes that s is not selected to carry message bits.

This sampling can be carried out by a rejection sampling similar to that in [26] and [41]. The suggested sampling algorithm is detailed in Algorithm 3. The distribution of s can be computed as:

$$\begin{aligned} p(s_i) = {} & p(s_i|\vec{m}_i = 0)\cdot p(s_i, \vec{m}_i = 0) + p(s_i|\vec{m}_i = \perp)\cdot p(s_i, \perp) \\ & + p(s_i|\vec{m}_i = 1)\cdot p(s_i, \vec{m}_i = 1) = pdf(s_i) \end{aligned} \tag{24}$$

That is, s_i follows the same distribution as that directly sampled from $\mathcal{N}(0,1)$. Therefore, the suggested sampling method can effectively preserve the input distribution.

This noise S serves as the input to the Diffusion model. Mathematically, the code-to-sampling mapping Γ_3 can be expressed as:

$$\Gamma_3 : M \in \{0,1\}^{n_{hb}\times n_{wb}\times n_c + 2^{n_c}} \mapsto S \in \mathbb{R}^{n_{hi}\times n_{wi}\times n_{ci}} \tag{25}$$

It can be observed that without the secret keys $K^{(1)}$ and $K^{(2)}$, the receiver is incapable of recovering M. Furthermore, Eq. (24) has proved that the generated S precisely mirrors the input distribution employed within the specified Diffusion model, thereby establishing Γ_3 as a provably secure steganographic mechanism with exceptional undetectability.

During the extraction process, upon obtaining S' through the inverse mapping of the Diffusion model, we select the $2\times(n_{hb}\times n_{wb}\times n_c + 2^{n_c})$ elements from S' using the permutation key $K^{(2)}$, denoted as S''. Then M' can be estimated by applying the inverse transformation and decryption processes, expressed as:

$$m'_i = \begin{cases} 1, & \text{if } s''_{2i-1} \times (1 - 2 \times k^{(1)}_{2i-1}) > s''_{2i} \times (1 - 2 \times k^{(1)}_{2i}) \\ 0, & \text{otherwise} \end{cases} \tag{26}$$

Algorithm 3 *Distribution_Preserving_Sampling* $\left(\vec{M}, K^{(2)}\right)$

Input: Encrypted bits $\vec{M}$, permutation key $K^{(2)}$
Output: Latent representation S

Divide $(-\infty, +\infty)$ into three intervals by using thresholds $\tau = ppf(p_s/2)$ and $-\tau = ppf(1-p_s/2)$.
Pseudorandomly permute $[1, 2, \cdots, n_H \times n_W \times n_C]$ by using $K^{(2)}$, and select the first $2 \times (n_{hb} \times n_{wb} \times n_c + 2^{n_c})$ values as the position set $\mathcal{P}$ of the coefficients used to carry message bits.
for $i = 1$ *to* $n_H \times n_W \times n_C$ **do**
  if $i \in \mathcal{P}$ **then**
    if s_i *is to represent* $\vec{m}_i = 0$ **then**
      Set interval $I = \left(-\infty, ppf(\frac{p_s}{2})\right]$.
    else
      Set $I = \left(ppf(1-\frac{p_s}{2}), +\infty\right)$.
    end
  else
    Set $I = \left(ppf(\frac{p_s}{2}), ppf(1-\frac{p_s}{2})\right]$.
  end
  repeat
    Sample s_i from the Gaussian distribution $\mathcal{N}(0,1)$.
  until s_i *drops into* I;
end

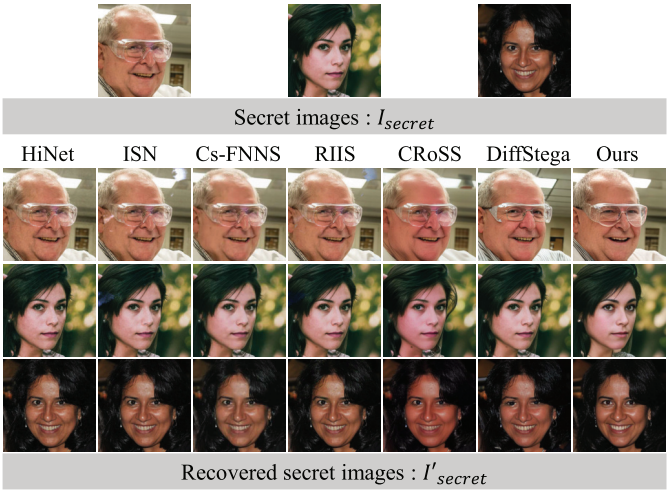

Fig. 4. Demonstrations of the recovered images obtained by different schemes.

IV. Experiments

This section presents an experimental analysis, including details of the experimental setup, comparisons to baseline methods, and ablation studies.

A. Experimental Setup

1) VQGAN: We choose FFHQ dataset [42], which includes 70,000 high-quality faces, and resize them into 512×512 as the secret images set. We follow the setting of VQGAN [31] to train the encoder E_{VQ}, decoder D_{VQ}, and codebook $\mathcal{C}$ with 2^{10} codewords, where the dimension of each codeword is 256. That is, $n_c = 10$ and $n_z = 256$. The parameter n_f in Algorithm 1 for robust codebook generation is set to 8, a value that has been found to yield good performance.

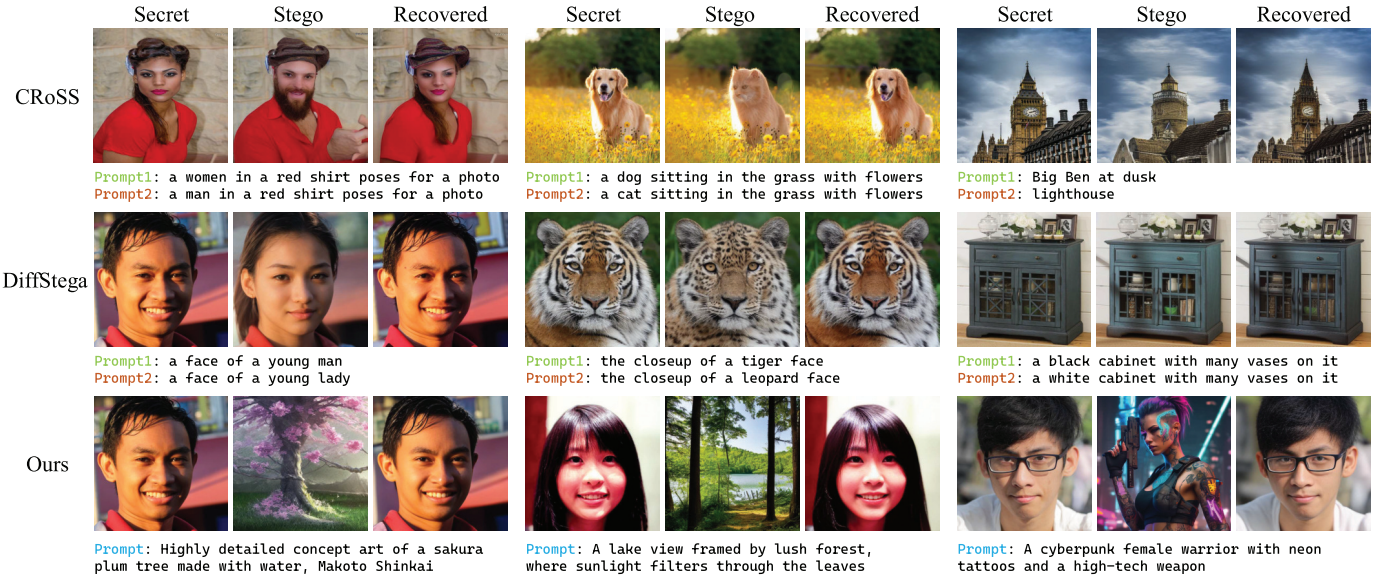

Fig. 5. Demonstration of stego images generated by three generative-based schemes.

2) Diffusion Model: As a demonstration, we utilize text-to-image Diffusion models, specifically selecting Stable Diffusion V2 [37] provided by HuggingFace as the baseline. The size of the generated images is 512×512, and the dimension of the latent space is $64 \times 64 \times 4$. That is, $n_H = n_W = 512$, $n_{hi} = n_{wi} = 64$, and $n_{ci} = 4$.

During inference, we randomly select prompts from Stable-Diffusion-Prompts [43], with a guidance scale of 7.5. We sample 50 steps using DPMSolver [44]. Considering the scenario that users only need to transmit the generated images without using the corresponding prompts, we simply use an empty prompt for inversion, with a scale of 1. We perform 50 steps of inversion using EDICT Inversion [35].

3) Baseline Methods: In our experiments, we focus on high-capacity image steganography, selecting six state-of-the-art baseline methods. These encompass modification-based image hiding schemes, including HiNet [10], ISN [11], Cs-FNNS [8], and RIIS [12], as well as generative-based image hiding schemes including CRoSS [28] and DiffStega [29]. For the modification-based models, the cover image set is generated using the aforementioned Diffusion model. The secret image set consists of the same FFHQ dataset referenced previously. The embedding capacity for all baselines is set at 24 bpp (bits per pixel), enabling one image to be hidden within another of the same resolution, specifically 512×512 pixels.

B. Visual Quality Comparison

We first compare the visual quality of stego images. Given that modification-based schemes can attain a high level of stego image quality, comparable to that of the covers, our focus in the comparison is solely on stego images produced through generative-based approaches.

Figure 5 exhibits several stego images created by different generative-based schemes. It is evident that all these schemes yield stego images of commendable visual quality. To delve deeper, we employ various metrics for a quantitative assessment of these techniques. Specifically, we calculate NIQE score [45] for the generated stego images, Fréchet Inception Distance (FID) [46] between the set of stego images and a set of images generated independently of secret images, as well as CLIP score [47] between the generated image and the given prompt. NIQE serves as a no-reference image quality assessment method, evaluating the naturalness of an image without a comparator. FID, on the other hand, compares the distribution of features from two sets of images to gauge the quality of the generated images. Herein we computed FID score for each baseline model using respective sets of 10,000 images. Lower NIQE and FID scores indicate a higher resemblance of the stego images to benign images. The three generative-based models are rooted in the text-to-image Diffusion model, prompting us to utilize CLIP score to assess the alignment of the stego image with the intended text prompt. A higher CLIP score underscores the superior consistency of the proposed method with the customized target prompt.

TABLE I

VISUAL COMPARISON OF STEGO IMAGES GENERATED BY DIFFERENT GENERATIVE-BASED SCHEMES. THE BEST RESULTS ARE BOLD AND THE SECOND-BEST RESULTS ARE UNDERLINED

Method \ Metric	NIQE ↓	FID ↓	CLIP Score ↑
Stable Diffusion V2	3.218	15.237	0.363
CRoSS	<u>4.452</u>	49.052	0.269
DiffStega	4.966	<u>48.698</u>	<u>0.294</u>
Ours	**3.250**	**17.709**	**0.363**

The comparison results are summarized in Table I. It can be observed that our scheme boasts the lowest NIQE and FID scores, confirming that the quality of the generated images remains unaffected by the secret-driven process. Moreover, our scheme attains the highest CLIP scores, highlighting its impressive customization capabilities. To further validate our

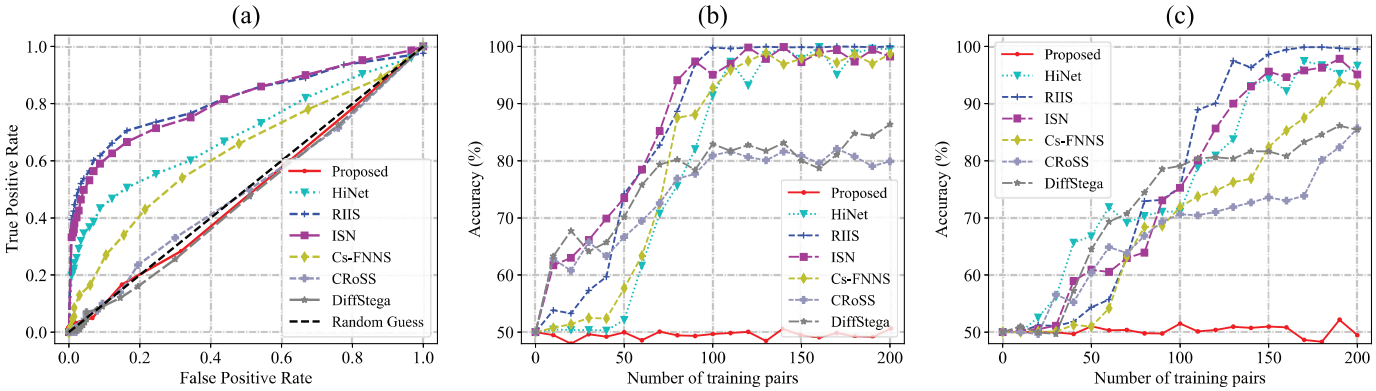

Fig. 6. Comparison of undetectability among different baseline schemes using (a) StegExpose, (b) SRNet, and (c) XuNet.

TABLE II

PSNR/SSIM RESULTS OF ALL THE BASELINE SCHEMES UNDER DIFFERENT DEGRADATIONS, INCLUDING GAUSSIAN NOISE AND JPEG COMPRESSION. THE BEST RESULTS ARE BOLD AND THE SECOND-BEST RESULTS ARE UNDERLINED

Method \ Attack	Clean	Gaussian noise				JPEG compression			
		$\sigma = 0.035$	$\sigma = 0.025$	$\sigma = 0.015$	$\sigma = 0.005$	$QF = 30$	$QF = 50$	$QF = 70$	$QF = 90$
HiNet	37.05 / 0.93	10.64 / 0.05	12.72 / 0.09	16.46 / 0.18	19.65 / 0.28	11.16 / 0.32	11.23 / 0.32	11.25 / 0.33	11.30 / 0.33
ISN	36.77 / 0.96	13.42 / 0.12	16.04 / 0.19	17.85 / 0.24	20.19 / 0.33	8.30 / 0.38	8.61 / 0.40	8.75 / 0.40	8.82 / 0.42
Cs-FNNS	35.80 / 0.93	12.22 / 0.42	12.95 / 0.44	14.51 / 0.51	22.07 / 0.66	11.14 / 0.33	11.35 / 0.34	11.45 / 0.35	11.83 / 0.37
RIIS	**39.69 / 0.97**	16.81 / 0.17	19.47 / 0.26	**23.63** / 0.44	**26.73** / 0.59	9.88 / 0.33	10.04 / 0.35	10.25 / 0.36	10.99 / 0.37
CRoSS	22.75 / 0.74	20.98 / 0.51	21.55 / 0.58	22.28 / 0.65	22.77 / 0.68	20.36 / 0.58	21.01 / 0.60	22.01 / 0.67	22.54 / 0.70
DiffStega	23.13 / 0.74	21.08 / 0.52	21.66 / 0.59	22.34 / 0.65	22.67 / 0.68	20.57 / 0.60	21.23 / 0.63	22.13 / **0.69**	22.64 / **0.71**
Ours	23.65 / 0.72	**21.20 / 0.63**	**21.74 / 0.64**	22.47 / **0.66**	23.09 / **0.69**	**20.86 / 0.62**	**21.39 / 0.64**	**22.24** / 0.67	**22.88** / 0.69

findings, we calculated these scores using the native Stable Diffusion V2 model. Two sets of images, both uninfluenced by secret images, were generated by this model and used to compute the aforementioned metrics. The results, also presented in Table I, reveal that our proposed method performs on a par with the native Stable Diffusion V2. In conclusion, the proposed method exhibits performance nearly identical to that of public image synthesis techniques.

We further evaluate the visual quality of the recovered images. Figure 7 presents a visual comparison of all baseline schemes. All schemes manage to restore the overall content of the secret images. However, a closer inspection reveals discernible differences in image details. Specifically, the recovery quality of the generative-based schemes in the last three columns falls short compared to that of the modification-based schemes in the first four columns. To quantify this disparity, we utilize Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity (SSIM) metrics [48], with the results tabulated in the "Clean" column of Table II. These metrics further confirm that the generative-based schemes lag behind the modification-based ones. This disparity could be attributed to the inherent capacity advantage enjoyed by the modification-based schemes, which do not have to grapple with the undetectability demands, thereby affording more space for concealing secret images. Nevertheless, it is worth noting that our scheme still leads among the 3 generative-based approaches.

C. Undetectability Comparison

Undetectability is critical for steganography applications. It is paramount that the observer remains oblivious to the presence of hidden information within the transmitted images.

TABLE III

FID SCORES BETWEEN SECRET AND STEGO IMAGE SETS OBTAINED BY DIFFERENT GENERATIVE-BASED MODELS. THE BEST RESULTS ARE BOLD

Method	CRoSS	DiffStega	Ours
FID between *secret* and *stego* ↑	71.608	74.055	**292.933**

The two generative-based schemes, CRoSS and DiffStega, achieve steganography by substituting a key object in the secret image with another, excelling in local modifications but struggling with global alterations. Consequently, as illustrated in Figure 5, these schemes may inadvertently retain significant overlap between the content of the secret and stego images, compromising confidentiality. Additionally, an observer who is privy to some content of the secret images could potentially distinguish between stego and innocent images.

In contrast, our proposed scheme transcends the limitations imposed by the content of the secret images. It grants users the liberty to specify the content of the generated stego images through customizable prompts. As evident in Figure 5, our scheme ensures independence between the secret and stego images. Furthermore, when we calculated the FID scores between the secret and stego image sets for three generative-based schemes, our scheme exhibited significantly higher scores, as listed in Table III, indicative of greater diversity between the secret and stego images. This implies the superior undetectability and confidentiality offered by our scheme.

We then adopt three popular image steganalytic tools, including StegExpose [15], SRNet [17], and XuNet [16], to quantitatively evaluate the undetectability of these baseline

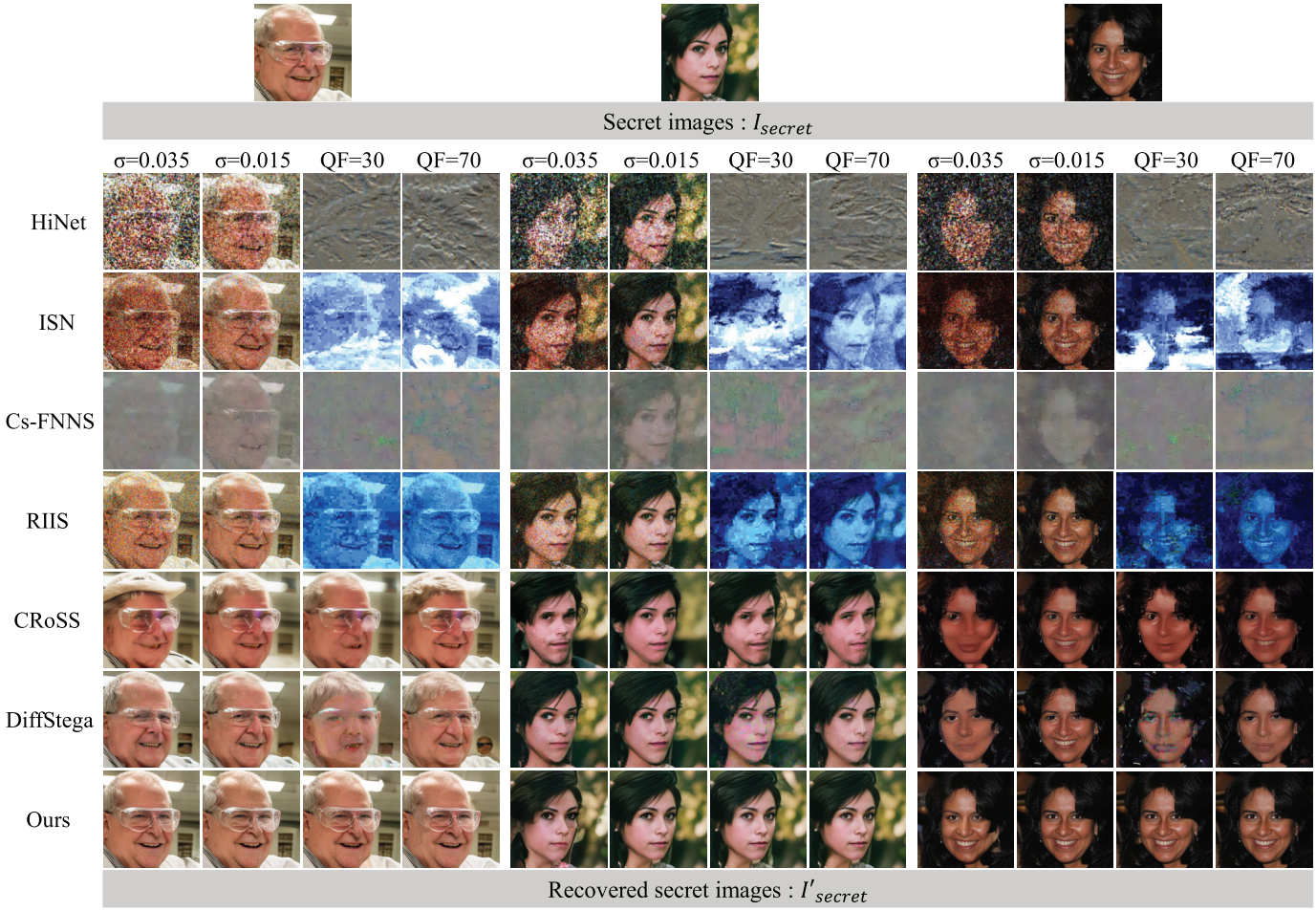

Fig. 7. Demonstration of robustness of different schemes under Gaussian noise with $\sigma \in \{0.035, 0.015\}$ and JPEG compression with $QF \in \{30, 70\}$.

schemes. StegExpose is a well-established steganalytic tool that incorporates various statistical methods such as RS analysis, Chi Square Attack, and so on. SRNet and XuNet, on the other hand, represent the pinnacle of deep learning-based steganalytic techniques. Cover/stego-image pairs are required to train these steganalytic tools. Due to the lack of cover images, we utilize images generated by the same Diffusion model and prompts from the same library, but which are not driven by secret images, as the cover images for the generative-based schemes. For a thorough evaluation, we provide 2,000 cover/stego-image pairs for each steganalytic tool and each baseline scheme.

We use all the cover/stego image pairs for testing via StegExpose. Figure 6 illustrates the receiver operating characteristic (ROC) curves for different detection thresholds in StegExpose. Ideally, a perfect steganographic method would yield an ROC curve that mirrors the diagonal reference line, suggesting that steganalytic tools would be reduced to random guessing. The proximity of the ROC curves for the generative-based schemes to this ideal diagonal line is remarkable. This performance surpasses that of modification-based approaches, indicating that generative-based schemes offer greater resilience against detection by StegExpose.

When it comes to SRNet and XuNet, we randomly split the 2,000 cover/stego-image pairs into two equal sets for training and testing. By adhering to the settings outlined in [16] and [17], we gradually increase the number of leaked sample pairs for training and record the corresponding detection accuracy on the test set. The comparison results are illustrated in Figs. 6 (b) and 6 (c). A consistent trend can be observed that, as the number of training pairs increases, the detection accuracies of all schemes improve. However, our scheme maintained a detection accuracy hovering around 0.5, suggesting that it remains largely undetectable even as the steganalytic tools become more informed. This underscores the superior undetectability of our proposed method.

D. Robustness Comparison

To evaluate the robustness of our scheme, we conduct experiments under various distortions, including Gaussian noise and JPEG compression. For all baseline schemes, we intentionally degraded the stego images and attempted to retrieve the secret images from these degraded versions. By computing the PSNR and SSIM between the recovered images and the original secret images, we are able to gauge the resilience of each scheme against these specified distortions.

Figure 7 provides a visual representation, showing the original secret images alongside their corresponding recovered versions after undergoing different levels of distortion. Our scheme demonstrates remarkable fidelity in recovering the content of the secret images, whereas the images recovered from other baseline schemes exhibited notable distortions in both color and content, or even failed to recover the secret images.

Table II presents a quantitative analysis of the quality of recovered images under various distortion scenarios. It is evident that when stego images are subjected to attacks, the modification-based schemes experience a significant decline in fidelity metrics. In stark contrast, generative-based schemes maintain relatively stable scores, showcasing their resilience.

The robustness demonstrated by CRoSS and DiffStega, despite the absence of specialized robustness designs, is intriguing. This phenomenon can be ascribed to the inherent stability of the Diffusion model. Furthermore, the approach of modifying only a fraction of the secret images while leaving the remainder unchanged enhances their robustness. Among the generative-based schemes, our proposed method emerges as the superior performer, thereby validating the efficacy of our robustness mechanism.

E. Evaluation of the Sensitivity of Encryption Keys

The proposed scheme employs two encryption keys, $K^{(1)}$ and $K^{(2)}$, to protect confidentiality against potential unauthorized access. To assess the sensitivity of the encryption algorithms to these keys, we simulate a scenario where an attacker can obtain parts of these keys.

Given an authentic key $K^{(i)}$, we disclose $p\%$ of its bits through the following process. We create a binary mask $M^{(i)}$ of the same length as the key. Within this mask, $p\%$ of the entries are set to 1, indicating that the corresponding bits in the key have been leaked, while the remaining entries are set to 0. The attacker can then construct a similar key $K^{(i)'}$ using the equation:

$$K^{(i)'} = (K^{(i)} M^{(i)}) | (\Delta \neg M^{(i)}) \tag{27}$$

where Δ represents a pseudorandom binary sequence of the same length as $K^{(i)}$. The attacker subsequently attempts to retrieve the secret image using $K^{(i)'}$.

In all instances where both $K^{(1)}$ and $K^{(2)}$ are leaked, we progressively decrease the percentage of leaked key bits, $p\%$, from 100% to 96%. Figure 8 visually compares the recovered images using keys with varying $p\%$ values. It is evident that, with only 1% of the bits of both keys remaining unknown, the recovered secret images exhibit considerable visual distortion, making the content unrecognizable. Table IV outlines the decline in PSNR scores of the recovered images as $p\%$ decreases. It can be inferred that the proposed method offers adequate confidentiality, as the PSNR scores of the recovered images drop significantly below 10 when the percentage of leaked bits for any key is below 100%.

F. Ablation Studies

To assess the effectiveness of the key components within Γ_1, Γ_2, and Γ_3, we conduct ablation experiments using six configurations of our scheme. Our focus is on three crucial components: the robust codebook in Γ_1, the check code in Γ_2, and the dual-sample in Γ_3. The six configurations encompass: the baseline devoid of the three mappings, the inclusion of ① only the robust codebook, ② only the check code, either ③ the dual-sample or ④ the triple-sample, and finally, the integration of three components ① ② ③.

Figure 9 offers a visual comparison between the original secret images obtained through our scheme with various configurations and their corresponding recovered versions in the present of distortions. It becomes evident that the incorporation of both components markedly diminishes the distortion in the finer details of the recovered images. Table V presents the PSNR scores of the recovered images obtained using different configurations under various distortion levels, along with the improvement values compared to the baseline. Notably, the robust codebook in Γ_1 contributes marginally more than the check code in Γ_2 and dual-sample in Γ_3. Additionally, we implemented a majority voting strategy in the triple-sample approach. However, this approach led to an increase in message errors due to the selection of more coefficients with small amplitudes. The fusion of the three components ① ② and ③ results in a substantial enhancement in robustness, with this enhancement becoming more evident as the distortion intensity increases. Consequently, the integration of the robust codebook, the check code, and the dual-sample is indispensable in our proposed scheme.

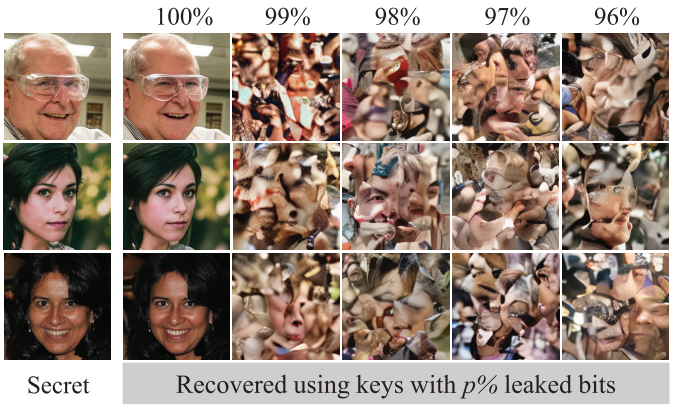

Fig. 8. Visual comparison of the recovered images using keys with varying degrees of leakage.

TABLE IV

PSNR SCORES OF THE RECOVERED IMAGES USING CONSTRUCTED KEYS, WHERE THE PERCENTAGE OF LEAKED BITS RANGES FROM 100% TO 96%

$K^{(1)'}$ \ $K^{(2)'}$	100%	99%	98%	97%	96%
100%	23.65	9.32	9.42	9.48	9.32
99%	9.45	9.38	9.29	9.47	9.36
98%	9.26	9.51	9.24	9.49	9.45
97%	9.32	9.26	9.14	9.37	9.38
96%	9.10	9.45	9.48	9.46	9.28

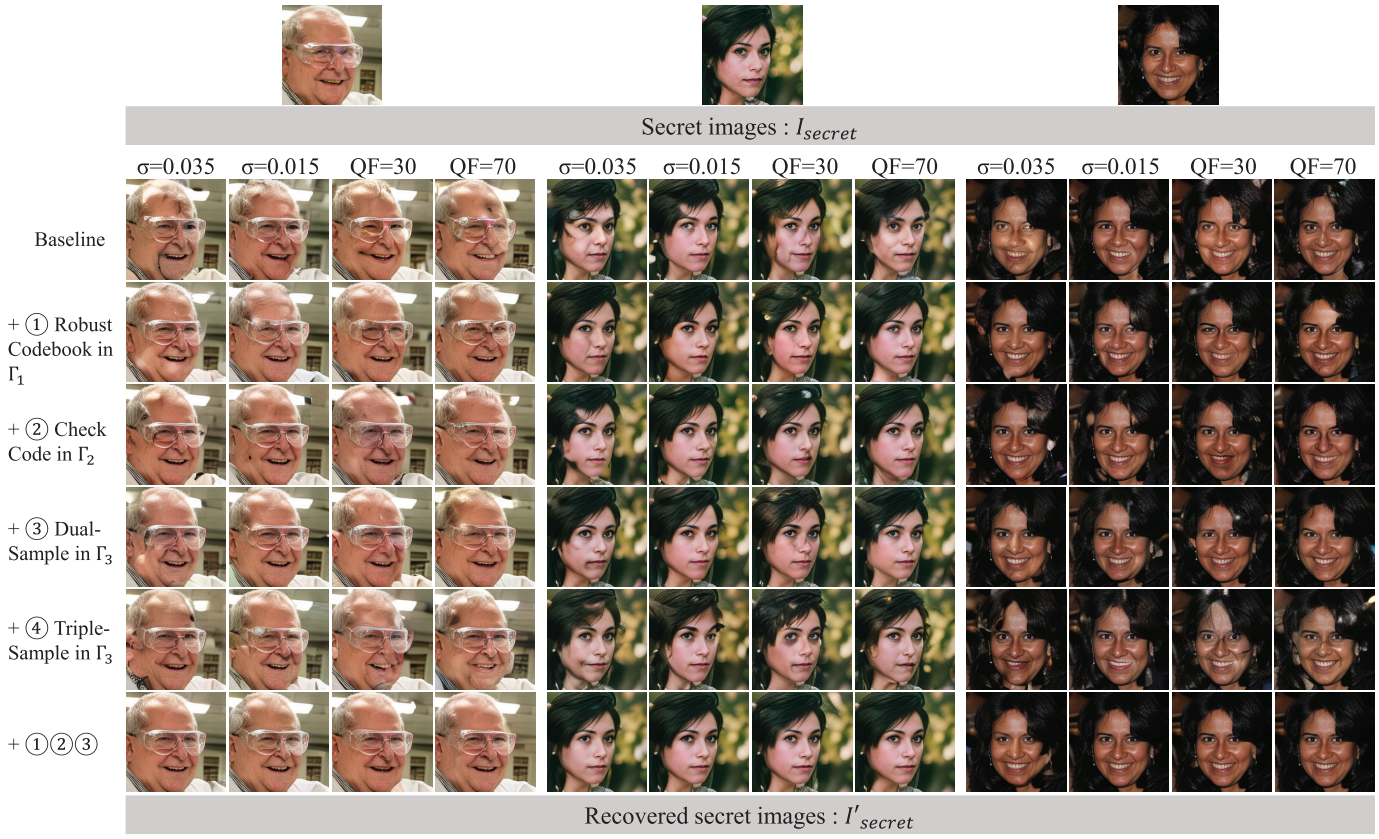

Fig. 9. Visual comparisons of the robustness obtained by the proposed scheme with different configurations.

TABLE V

PSNR SCORES OF THE RECOVERED IMAGES USING DIFFERENT CONFIGURATIONS. THE VALUE AFTER '+' OR '-' IN THE BOTTOM RIGHT CORNER INDICATES THE IMPROVEMENT OR DECLINE COMPARED TO THE BASELINE

Settings \ Attack	Gaussian noise				JPEG compression			
	$\sigma = 0.035$	$\sigma = 0.025$	$\sigma = 0.015$	$\sigma = 0.005$	$QF = 30$	$QF = 50$	$QF = 70$	$QF = 90$
Baseline	18.69	19.72	20.64	22.32	17.62	19.33	20.14	22.26
+ ① Robust Codebook in Γ_1	$19.73_{+1.04}$	$20.53_{+0.81}$	$21.38_{+0.74}$	$22.77_{+0.45}$	$19.49_{+1.87}$	$20.54_{+1.21}$	$21.06_{+0.92}$	$22.47_{+0.21}$
+ ② Check Code in Γ_2	$19.56_{+0.87}$	$20.51_{+0.79}$	$21.47_{+0.83}$	$22.61_{+0.29}$	$18.74_{+1.12}$	$20.09_{+0.76}$	$20.83_{+0.69}$	$22.56_{+0.30}$
+ ③ Dual-Sample in Γ_3	$19.74_{+1.05}$	$20.48_{+0.76}$	$21.31_{+0.67}$	$22.66_{+0.34}$	$18.25_{+0.63}$	$20.03_{+0.70}$	$20.70_{+0.56}$	$22.32_{+0.06}$
+ ④ Triple-Sample in Γ_3	$17.96_{-0.73}$	$19.27_{-0.45}$	$20.37_{-0.27}$	$22.39_{+0.07}$	$16.38_{-1.24}$	$18.35_{-0.98}$	$19.69_{-0.45}$	$22.13_{-0.13}$
+ ① ② ③	$\mathbf{21.20_{+2.51}}$	$\mathbf{21.74_{+2.02}}$	$\mathbf{22.47_{+1.83}}$	$\mathbf{23.09_{+0.77}}$	$\mathbf{20.86_{+3.24}}$	$\mathbf{21.39_{+2.06}}$	$\mathbf{22.24_{+2.10}}$	$\mathbf{22.88_{+0.62}}$

V. CONCLUSION

This paper introduces a generative steganography model that satisfies four crucial properties: security, diversity, robustness, and capacity. It is worth noting that the inherent numerical instability and sensitivity to perturbations in most generative models present significant obstacles to achieving robustness. To address this issue, the proposed scheme integrates three reversible mappings. In the first mapping, we utilize the concept of perceptual image compression and design a compression mapping based on VQGAN. Additionally, an order-preserving codebook generation algorithm is introduced, which lays a solid foundation for achieving robust image encoding. In the second mapping, to bolster robustness against potential network errors, we append check codes, thereby providing error correction capabilities. The third mapping features a novel distribution-preserving noise sampling module, which not only offers provable security but also enhances robustness. Following these three mappings, the secret image is transformed into an initial noise latent that adheres to a standard Gaussian distribution. This latent representation is then fed into the Stable Diffusion model to generate a high-quality stego image. By utilizing EDICT Inversion and the three reverse mappings, the secret image can be recovered with high fidelity, even in the presence of channel distortion. Furthermore, the proposed scheme provides both provable steganographic security and perfect secrecy. However, we acknowledge that achieving high capacity comes at the expense of reduced recovered quality. This trade-off motivates our future efforts to design improved compressive mappings that can maintain both high capacity and superior recovered quality.

While the vector-quantized codebook employed ensures robustness against various attacks, it unfortunately imposes constraints on the perceptual quality of the recovered

images. To overcome this limitation, our future research will delve into exploring robust latent representation that possesses heightened generative capabilities. Moreover, the present implementation is contingent upon Diffusion models, which entails considerable computational overhead. Exploring lightweight architectures represents another avenue for future work.

REFERENCES

[1] J. Mielikainen, "LSB matching revisited," *IEEE Signal Process. Lett.*, vol. 13, no. 5, pp. 285–287, May 2006.

[2] B. Feng, W. Lu, and W. Sun, "Novel steganographic method based on generalized K-distance N-dimensional pixel matching," *Multimedia Tools Appl.*, vol. 74, no. 21, pp. 9623–9646, Jun. 2014.

[3] T. Filler, J. Judas, and J. Fridrich, "Minimizing additive distortion in steganography using syndrome-trellis codes," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 3, pp. 920–935, Sep. 2011.

[4] B. Feng, W. Lu, and W. Sun, "Secure binary image steganography based on minimizing the distortion on the texture," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 2, pp. 243–255, Feb. 2015.

[5] W. Li, W. Zhang, L. Li, H. Zhou, and N. Yu, "Designing near-optimal steganographic codes in practice based on polar codes," *IEEE Trans. Commun.*, vol. 68, no. 7, pp. 3948–3962, Jul. 2020.

[6] C. Yuan, P. Yu, and Y. Wu, "A survey on neural network-based image data hiding for secure communication," *Int. J. Auto. Adapt. Commun. Syst.*, vol. 17, no. 2, pp. 476–493, 2024.

[7] Z. Luo, S. Li, G. Li, Z. Qian, and X. Zhang, "Securing fixed neural network steganography," in *Proc. 31st ACM Int. Conf. Multimedia*, Oct. 2023, pp. 7943–7951.

[8] G. Li, S. Li, Z. Qian, and X. Zhang, "Cover-separable fixed neural network steganography via deep generative models," in *Proc. 32nd ACM Int. Conf. Multimedia*, Oct. 2024, pp. 10238–10247.

[9] S. Baluja, "Hiding images in plain sight: Deep steganography," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, Dec. 2017, pp. 2066–2076.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] X. Deng, C. Zhang, L. Jiang, J. Xia, and M. Xu, "DeepSN-net: Deep semi-smooth Newton driven network for blind image restoration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 47, no. 4, pp. 2632–2646, Apr. 2025.

[15] B. Boehm, "StegExpose—A tool for detecting LSB steganography," 2014, *arXiv:1410.6656*.

[16] 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.

[17] 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, Sep. 2018.

[18] 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.

[19] 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.

[20] W. Su, J. Ni, and Y. Sun, "StegaStyleGAN: Towards generic and practical generative image steganography," in *Proc. AAAI Conf. Artif. Intell.*, Mar. 2024, vol. 38, no. 1, pp. 240–248.

[21] X. Liu, Z. Ma, J. Ma, J. Zhang, G. Schaefer, and H. Fang, "Image disentanglement autoencoder for steganography without embedding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 2303–2312.

[22] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2016, pp. 2234–2242.

[23] 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.

[24] Z. Zhou et al., "Secret-to-image reversible transformation for generative steganography," *IEEE Trans. Depend. Secure Comput.*, vol. 20, no. 5, pp. 4118–4134, Oct. 2022.

[25] 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.

[26] Z. Yang, K. Zeng, K. Chen, H. Fang, W. Zhang, and N. Yu, "Gaussian shading: Provable performance-lossless image watermarking for diffusion models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 12162–12171.

[27] X. Hu, S. Li, Q. Ying, W. Peng, X. Zhang, and Z. Qian, "Establishing robust generative image steganography via popular stable diffusion," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 8094–8108, 2024.

[28] 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. 80730–80743.

[29] 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.

[30] A. Babenko and V. Lempitsky, "Additive quantization for extreme vector compression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 931–938.

[31] P. Esser, R. Rombach, and B. Ommer, "Taming transformers for high-resolution image synthesis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 12873–12883.

[32] X. Deng, C. Gao, and M. Xu, "PIRNet: Privacy-preserving image restoration network via wavelet lifting," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 22311–22320.

[33] X. Deng, J. Xu, F. Gao, X. Sun, and M. Xu, "DeepM^{2}M2CDL: Deep multi-scale multi-modal convolutional dictionary learning network," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 5, pp. 2770–2787, May 2024.

[34] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *Proc. Int. Conf. Learn. Represent.*, Jan. 2020, pp. 1–20.

[35] B. Wallace, A. Gokul, and N. Naik, "EDICT: Exact diffusion inversion via coupled transformations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 22532–22541.

[36] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. 18th Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, vol. 9351. Cham, Switzerland: Springer, 2015, pp. 234–241.

[37] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 10684–10695.

[38] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform manifold approximation and projection for dimension reduction," 2018, *arXiv:1802.03426*.

[39] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. IT-28, no. 2, pp. 129–137, Mar. 1982.

[40] C. K. Huang and H. H. Nien, "Multi chaotic systems based pixel shuffle for image encryption," *Opt. Commun.*, vol. 282, no. 11, pp. 2123–2127, Jun. 2009.

[41] K. Chen, H. Zhou, H. Zhao, D. Chen, W. Zhang, and N. Yu, "Distribution-preserving steganography based on text-to-speech generative models," *IEEE Trans. Depend. Secure Comput.*, vol. 19, no. 5, pp. 3343–3356, Sep. 2022.

[42] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4401–4410.

[43] G. Santana. (2022). *Stable-diffusion-Prompts*. [Online]. Available: https://huggingface.co/datasets/Gustavosta/Stable-Diffusion-Prompts

[44] C. Lu, Y. Zhou, F. Bao, J. F. Chen, C. Li, and J. Zhu, "DPM-solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2022, pp. 5775–5787.

[45] L. Zhang, L. Zhang, and A. C. Bovik, "A feature-enriched completely blind image quality evaluator," *IEEE Trans. Image Process.*, vol. 24, pp. 2579–2591, 2015.

[46] 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. 6629–6640.
[47] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn.*, vol. 139, 2021, pp. 8748–8763.
[48] W. Zhou, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, pp. 600–612, 2004.

Liyan Chen received the B.S. degree from Jinan University, Guangzhou, China, in 2023, where he is currently pursuing the M.S. degree.

His research interests include multimedia security and image steganography.

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.

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 was a Visiting Scholar with New Jersey Institute of Technology, USA, in 2015, and a Visiting Professor with Sungkyunkwan University, South Korea, in 2016. He is currently a Professor with the College of Cyber Security, Jinan University, Guangzhou, China. His research interests include AI security, cloud computing security, and digital forensics. He serves as the Managing Editor for *International Journal of Autonomous and Adaptive Communications Systems*.

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 with 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*.

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 a PC co-chair or a PC member for more than 30 international conferences. He serves as an Associate Editor for IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY.