Establishing Robust Generative Image Steganography via Popular Stable Diffusion

Xiaoxiao Hu, Sheng Li, *Member, IEEE*, Qichao Ying, Wanli Peng, Xinpeng Zhang, *Senior Member, IEEE*, and Zhenxing Qian, *Senior Member, IEEE*

***Abstract*— Generative steganography, a novel paradigm in information hiding, has garnered considerable attention for its potential to withstand steganalysis. However, existing generative steganography approaches suffer from the limited visual quality of generated images and are challenging to apply to lossy transmissions in real-world scenarios with unknown channel attacks. To address these issues, this paper proposes a novel robust generative image steganography scheme, facilitating zero-shot text-driven stego image generation without the need for additional training or fine-tuning. Specifically, we employ the popular Stable Diffusion model as the backbone generative network to establish a covert transmission channel. Our proposed framework overcomes the challenges of *numerical instability* and *perturbation sensitivity* inherent in diffusion models. Adhering to Kerckhoff's principle, we propose a novel mapping module based on dual keys to enhance robustness and security under lossy transmission conditions. Experimental results showcase the superior performance of our method in terms of extraction accuracy, robustness, security, and image quality.**

***Index Terms*— Steganography, robust steganography, generative image steganography, diffusion model.**

I. Introduction

DIGITAL steganography is the art and science of concealing secret messages within digital media, such as text [1], image [2], audio [3], and video [4]. Traditional modification-based image steganographic schemes typically hide secret messages through imperceptible modifications on the pixel domain of images. The commonly used LSB (Least Significant Bit) steganography represents the secret bitstream by altering the least significant bit of each pixel. The modification pattern introduced by LSB steganography often results in detectable changes to the histogram statistics, raising security concerns. To minimize perceptual impact while preserving the statistical properties of the cover image, various adaptive methods have been proposed. These schemes incorporate appropriate steganographic encoding algorithms such as STC (syndrome-trellis codes) encoding [5] and SPC (steganographic polar codes) encoding [6], alongside various distortion cost functions [7], [8], [9], [10]. With the advancement of deep learning, manual design methodologies are progressively being supplanted by deep neural networks. For instance, HiDDeN [11] firstly introduces the end-to-end learning framework and adversarial training strategy for digital image steganography. However, modification-based steganography techniques, due to their necessity for cover images, might leave detectable traces that can be identified by steganalysis tools [12], [13], [14], [15].

In contrast, generative image steganography directly generates stego images from secret messages using sophisticated generative models. This approach eliminates the need for cover images, making it less susceptible to detection by steganalysis tools, thus significantly enhancing security. Generative steganography can be broadly categorized into two main types: extractor-based generative steganography and mapping-based generative steganography. Extractor-based methods commonly entail modifying the original image generation process to embed secret messages and often require additional training for secret message extractors. For instance, Wei et al. [16] integrate a novel secret block into the StyleGAN [17] and introduce the mutual information mechanism to enhance extraction accuracy. Liu et al. [18] utilize stable structural features to conceal secret messages, decoupling images into texture components and structural components. However, these methods come with inherent training costs and often necessitate alterations to the original generation model, resulting in diminished image visual quality.

Mapping-based methods treat the embedding and extraction of secret messages as inverse processes. They rely on predefined mapping rules to construct bijective mappings between secret messages and stego images. For example, Zhou et al. [2] leverage the reversibility of flow-based models [19] and design a novel mapping mechanism for rearranging the elements of latent vectors guided by secret messages. Chen et al. [20], [21], [22] propose a provably secure mapping module, ensuring that the mapping results adhere to the original Gaussian distribution. These approaches often struggle to generate high-quality stego images. Issues such as small image size, limited control over generated

Manuscript received 12 April 2024; revised 14 July 2024; accepted 27 July 2024. Date of publication 15 August 2024; date of current version 12 September 2024. This work was supported in part by the National Natural Science Foundation of China under Grant U20B2051, Grant 62072114, Grant U20A20178, and Grant U22B2047; and in part by the National Key Research and Development Program of China under Grant 2022QY0101. The associate editor coordinating the review of this article and approving it for publication was Prof. Fernando Perez-Gonzalez. *(Corresponding author: Zhenxing Qian.)*

Xiaoxiao Hu, Sheng Li, Wanli Peng, Xinpeng Zhang, and Zhenxing Qian are with the Key Laboratory of Culture and Tourism Intelligent Computing of Ministry of Culture and Tourism, School of Computer Science, and the Laboratory of Multimedia and Artificial Intelligence Security, Fudan University, Shanghai 200437, China (e-mail: xxhu23@m.fudan.edu.cn; lisheng@fudan.edu.cn; pengwanli@fudan.edu.cn; zhangxinpeng@fudan.edu.cn; zxqian@fudan.edu.cn).

Qichao Ying is with Nvidia Corporation, Shanghai 200333, China (e-mail: shinydotcom@163.com).

Digital Object Identifier 10.1109/TIFS.2024.3444311

content, and noticeable local artifacts are common drawbacks. These limitations stem from the reliance on the reversible properties of generative models, which inherently constrain the model's generation capabilities, as exemplified in models like GLOW [19].

Additionally, in real-world scenarios, image transmission channels are frequently subject to lossy conditions. Common post-processing operations, such as JPEG (Joint Photographic Experts Group) compression, additive white Gaussian noise, and median blur, can notably diminish the extraction accuracy of existing steganographic algorithms. Some works [22], [23] have focused on addressing the robustness issue, predominantly targeting scenarios involving known channel attacks. However, in many cases, users lack precise knowledge of the operations within the transmission channel, thereby constraining the practical applicability of such methods.

Currently, as one of the most prominent models in the field of image generation, diffusion models [24] can produce high-fidelity and diverse images. Nevertheless, leveraging diffusion models for mapping-based generative image steganography presents unique challenges due to their inherent *numerical instability* and *perturbation sensitivity*. These challenges arise from numerical errors accumulated over multiple timesteps. Despite these challenges, we propose a robust mapping-based generative image steganography scheme that harnesses the capabilities of diffusion models while addressing these concerns. Specifically, we utilize the popular Stable Diffusion [25] as the backbone generative network and employ a second-order DPM-Solver++ [26] sampler to improve numerical accuracy. Additionally, we devise a novel mapping module based on dual keys, following the Kerckhoffs' principle. Compared to previous mapping algorithms, this module significantly improves extraction accuracy under lossy conditions. By integrating text prompts as control signals, our method enables precise content generation. The success of Stable Diffusion has fueled user enthusiasm, making the transmission of synthetic images on social media increasingly common, further concealing our intention to use generated images for secret information transmission. Our contributions can be summarized as follows:

- We propose a robust generative image steganography scheme capable of disguising secret messages within diverse images, addressing the challenges arising from *numerical instability* and *perturbation sensitivity* inherent in diffusion models.
- We devise a novel mapping module that transforms arbitrary secret information into the Gaussian noise, aligning it with the original input distribution of the generative model. This module also supports inversely mapping noise back to secret information, even in the presence of numerical errors caused by lossy operations.
- Experimental results demonstrate the superior performance of our method in terms of extraction accuracy, robustness, security, and image quality.

Finally, we outline the organization of the paper. Section II reviews related work, and Section III provides an overview of diffusion models. Our proposed methodology, including the design of the mapping module and the overall framework, is detailed in Section IV. Experimental results are presented in Section V, and conclusions are drawn in Section VI.

II. Related Works

A. Modification-Based Image Steganography

Modification-based schemes encompass the majority of image steganography methods, which alter the pixel values of images to embed secret messages. For instance, the primitive LSB method replaces the least significant bits with secret messages. However, indiscriminately modifying each pixel without considering the content of cover images can result in a shift in statistical features, raising security concerns. To mitigate the traces of pixel modifications, many additive adaptive steganographic frameworks have been proposed. These frameworks incorporate appropriate steganographic encoding algorithms such as matrix encoding [27], wet paper encoding [28], STC encoding [5], and SPC encoding [6], along with distortion cost functions like HUGO [7], SUNIWARD [8], MiPOD [9], and AdaBIM [10]. Traditional methods require manual design of embedding and extraction processes, making it challenging to balance distortion costs and extraction performance. In contrast, deep learning-based schemes enable neural networks to autonomously learn optimal strategies. HiDDeN [11] pioneers the use of end-to-end training framework, realizing patch-based embedding scheme. SteganoGAN [29] employs generative adversarial networks to achieve high-capacity image steganography algorithms. However, modification-based image steganography still introduces some distortions due to the alteration of pixels. Such distortions leave traces that can be perceived by steganalyzers [12], [13], [14], [15].

B. Generative Steganography

To mitigate security threats arising from pixel modifications, Zhou et al. [30] introduce a steganography method without embedding, which selects different images from the database as stego images based on secret messages. Nonetheless, the hiding capacity of these selection-based methods is limited by the storage expenses associated with the dataset. In contrast, generative steganography schemes directly embed the secret message into the image generation process, harnessing the mechanisms of generative models to devise algorithms with substantial capacity and security. For instance, Wei et al. [16] extend the GAN framework by integrating a steganalyzer and a data extractor to balance security and extraction accuracy. Zhang et al. [31] transform secret messages into class labels and integrate them into the GAN image generation process. Similarly, You et al. [23] transform secret messages into semantic information, such as facial expressions, within generated images, streamlining the creation of compact sticker images. Liu et al. [18] disentangle images into structure and texture features, leveraging the stability of structural representations to enhance the extraction accuracy. These works are categorized as extractor-based methods, as they necessitate the integration of supplementary secret message extractors and

subsequent retraining on existing generative models, incurring certain overheads.

Mapping-based generative steganography methods consist of three key components: the image generation module, the image inversion module, and the reversible mapping module. The reversible mapping module establishes the correspondence between secret messages and Gaussian noises, with Gaussian noises serving as the exclusive latent input of generative models. The image generation module transforms Gaussian noises into stego images, while the image inversion module performs the reverse transformation. These modules enable bidirectional reversible mapping between secret messages and stego images. For example, Wei et al. [32] leverage the reversibility of flow-based models [19] to implement high-capacity steganography on TIFF (Tag Image File Format) format images using latent optimization strategies. To accommodate real-world applications, Wei et al. [33] further design secret message embedding and extraction modules in the frequency domain, which is compatible with PNG (Portable Network Graphics) images. Zhou et al. [2] design a novel mapping mechanism for rearranging the elements of latent vectors guided by secret messages. In the subsequent study, Zhou et al. [34] employ secret messages to guide the selection of positions for image contour points. Chen et al. [20], [21] propose a provably secure mapping module, ensuring that the mapping results adhere to the original Gaussian distribution.

However, during transmission over social media platforms, stego images undergo various lossy operations, such as JPEG compression. These operations substantially decrease the extraction accuracy of the aforementioned generative steganography methods. To tackle this practical challenge, some robust generative steganography schemes [22], [23] incorporate certain post-processing operations during secret message extraction. These methods typically presume foreknowledge of the lossy operations occurring in the transmission channel, labeling them as channel-aware, thereby imposing limitations on the applicability. Furthermore, current generative steganography methods encounter challenges concerning image generation quality, including small generated sizes and limited control over content generation. In this paper, we propose a controllable and robust generative steganography algorithm.

C. Image Synthesis

The evolution of image generation has been significantly propelled by various generative models, including GAN (Generative Adversarial Networks), VAE (Variational Autoencoders), flow-based models, diffusion models, and more. GAN [35] endeavors to estimate image distribution directly by orchestrating a game between the generator and discriminator. VAE [36] aims to encode and decode data in a continuous latent space. Flow-based models parameterize probabilistic distributions of images using invertible affine layers, exemplified by architectures like GLOW [19], with a focus on enhancing the ability to learn efficient representations of images. Due to their intrinsic reversibility, flow-based models have found widespread application in steganography schemes [2], [32].

Additionally, diffusion models learn to generate images from pure Gaussian noise by gradual denoising. Many works [26], [37], [38], [39] employ ordinary differential equations to accelerate the sampling speed of the original denoising diffusion probabilistic model (DDPM) [24]. Guided Diffusion [40] enables conditional generation capability in diffusion models by leveraging gradients from an auxiliary classifier as guidance. Stable Diffusion [25] is one of the most prominent text-to-image generative models, employing perceptual compression for image dimensionality reduction and utilizing text encoders [41] to align images with text prompts. In this paper, we utilize Stable Diffusion to construct a novel covert channel for secret message transmission.

III. Preliminaries

In the proposed scheme, we conceptualize *information hiding* and *information extraction* as inverse processes. The primary objective is to construct a reversible mapping between the secret message and the stego image. To achieve this goal, we incorporate diffusion-based modules into our methodology.

Diffusion models have demonstrated SOTA (state-of-the-art) performance in various image generation tasks. In this section, we introduce the foundational diffusion model, DDPM (III-A), and derive the modeling of the diffusion process through SDE (Stochastic Differential Equation) (III-B) and ODE (Ordinary Differential Equation) (III-C). ODEs are deterministic transformations between the input and the output, playing a pivotal role in establishing a reversible mapping.

A. Denoising Diffusion Probabilistic Model

DDPM [24] introduces a deterministic forward process with added noise and a denoising reverse learning process. The forward and backward processes of the diffusion model can be discretized into a large number of timesteps T, with T set to 1000 in the original DDPM. Sampling $\mathbf{x}_T$ from Gaussian noise $\mathcal{N}(\mathbf{0}, \mathbf{I})$, after T timesteps, a high-fidelity image $\mathbf{x}_0$ is generated. Here, $\mathbf{I}$ represents the identity covariance matrix. One forward step can be expressed as:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}\left(\sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}\right), 1 \le t \le T. \quad (1)$$

Here, β_t is a pre-defined parameter, and the iterative forward steps are a Markov chain process. In this paper, we adopt the original notation in DDPM, where p and q both represent probability distributions. From Eq. (1), we can derive a special distribution transition:

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}\left(\sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}\right),$$
$$\alpha_t := 1-\beta_t, \bar{\alpha}_t := \prod_{s=1}^{t}\alpha_s. \quad (2)$$

Here, ":=" signifies "defined as". Thus, we can induce the mean and variance of distribution $q(\mathbf{x_{t-1}}|\mathbf{x}_t, \mathbf{x}_0)$ for $t > 1$. Then, one denoising step can be formulated as follows:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}\left(\frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t, t)\right), \sigma_t^2\mathbf{I}\right)$$
$$\approx q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0). \quad (3)$$

Here, ϵ_θ represents the trained noise prediction model, and σ_t can be considered as a constant related to β_t.

B. Stochastic Differential Equation

Some theoretical and empirical researches suggest that the forward noisy Markov chain process in DDPM is not essential [37], [43], as long as the distributional relationship between $\mathbf{x}_t$ and $\mathbf{x}_0$ is maintained. Kingma et al. [44] prove that using a specific SDE can acquire the same forward transition distribution $q(\mathbf{x}_t|\mathbf{x}_0)$ as in Eq. (2),

$$\mathrm{d}\mathbf{x}_t = f(t)\,\mathbf{x}_t \mathrm{d}t + g(t)\,\mathrm{d}\mathbf{w}_t, \mathbf{x}_0 \sim q(\mathbf{x}_0), \tag{4}$$

where $\mathbf{w}_t$ denotes the standard Wiener process. $f(t)\,\mathbf{x}_t$ and $g(t)$ represent the *drift coefficient* and *diffusion coefficient*, respectively. They are solved as follows,

$$f(t) = \frac{\mathrm{d}\log a_t}{\mathrm{d}t}, g^2(t) = \frac{\mathrm{d}b_t^2}{\mathrm{d}t} - 2\frac{\mathrm{d}\log a_t}{\mathrm{d}t}b_t^2, \quad a_t := \sqrt{\bar{\alpha}_t}, b_t := \sqrt{1-\bar{\alpha}_t}. \tag{5}$$

Under some regularity conditions, Song et al. [45] show that the forward process in Eq. (4) has an equivalent reverse process from timestep T to 0,

$$\mathrm{d}\mathbf{x}_t = \left[f(t)\,\mathbf{x}_t - g^2(t)\,\nabla_{\mathbf{x}_t}\log q(\mathbf{x}_t)\right]\mathrm{d}t + g(t)\,\mathrm{d}\bar{\mathbf{w}}_t, \tag{6}$$

where $\bar{\mathbf{w}}_t$ is a standard Wiener process in the reverse direction. $\nabla_{\mathbf{x}_t}\log q(\mathbf{x}_t)$ termed as *score function* is the only unknown one. We use neural network s_θ to estimate the score function. Actually, s_θ is proportional to ϵ_θ, where

$$s_\theta = \frac{-\epsilon_\theta(\mathbf{x}_t, t)}{b_t}. \tag{7}$$

Incorporating Eq. (7) into Eq. (6) yields:

$$\mathrm{d}\mathbf{x}_t = \left[f(t)\,\mathbf{x}_t + \frac{g^2(t)}{b_t}\epsilon_\theta(\mathbf{x}_t, t)\right]\mathrm{d}t + g(t)\,\mathrm{d}\bar{\mathbf{w}}_t. \tag{8}$$

C. Ordinary Differential Equation

SDEs can be transformed into probabilistic flow ODEs by removing the stochastic process [37]. For example, both SDEs in Eq. (4) and Eq. (8) can be reduced to the same probabilistic flow ODE as follows,

$$\frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t} = f(t)\,\mathbf{x}_t + \frac{g^2(t)}{2b_t}\epsilon_\theta(\mathbf{x}_t, t). \tag{9}$$

The ODE illustrated in Eq. (9) makes both the forward process (from $\mathbf{x}_0$ to $\mathbf{x}_T$) and the denoising process (from $\mathbf{x}_T$ to $\mathbf{x}_0$) deterministic. Similar to flow-based models [19], [46], [47], the probability flow ODE can transform noise into samples through a reversible transformation. Most researches [26], [37], [39], [48] utilize numerical ODE solvers to accelerate sampling speed. In this paper, we use DPM-Solver++2M [26] as the default ODE solver for establishing a bijection between the noise input and the stego image.

DPM-Solver [38] and DPM-Solver++ [26] regard Eq. (9) as semi-linear structure, where $f(t)\,\mathbf{x}_t$ constitute the linear term and $\frac{g^2(t)}{2b_t}\epsilon_\theta(\mathbf{x}_t, t)$ from the non-linear component. The signal-to-noise ratio (SNR) is commonly utilized to quantify signal quality. After introducing log-SNR $\lambda_t := \log\frac{a_t}{b_t}$, we can rewrite the integral form of Eq. (9) as follows:

$$\mathbf{x_t} = \frac{a_t}{a_s}\mathbf{x}_s - a_t\int_{\lambda_s}^{\lambda_t} e^{-\lambda}\epsilon_\theta\left(\mathbf{x}_{t_\lambda(\lambda)}, t_\lambda(\lambda)\right)\mathrm{d}\lambda, \tag{10}$$

where $t_\lambda(\cdot)$ is the reverse function of $\lambda(t) = \lambda_t$. For a specialized k-th-order DPM-Solver, k-th-order Taylor expansion is employed to approximate the exponential integrators in Eq. (10). The distinction between DPM-Solver++ and DPM-Solver lies in its utilization of a data prediction model $\mathbf{x}_\theta$ instead of a noise prediction model ϵ_θ.

$$\mathbf{x_t} = \frac{b_t}{b_s}\mathbf{x}_s + b_t\int_{\lambda_s}^{\lambda_t} e^{\lambda}\mathbf{x}_\theta\left(\mathbf{x}_{t_\lambda(\lambda)}, t_\lambda(\lambda)\right)\mathrm{d}\lambda, \mathbf{x}_\theta := \frac{\mathbf{x}_t - b_t\epsilon_\theta}{a_t}. \tag{11}$$

The data prediction model introduces a smaller numerical error for guided sampling. For simplicity, we denote the forward ODE from timestep T to 0 as the generation process and the reverse ODE from timestep 0 to T as the inversion process.

IV. Proposed Method

In this section, we initially analyze the ODE mapping in part IV-A, revealing two key findings that challenge the applicability of existing mapping mechanisms to diffusion models. Subsequently, we define the problem formulation in part IV-B. Part IV-C details the transformations between the secret message and the noise input. We briefly illustrate the LDM in part IV-D. Besides, part IV-E and IV-F elucidate the information hiding and extraction processes, respectively.

A. Challenges Analysis

As shown in Section III-C, ODE can construct a bijective mapping between data $\mathbf{x}_0$ and noise $\mathbf{x}_T$. However, directly integrating this mapping with existing mechanisms is suboptimal for generative steganography due to its inherent *numerical instability* and *perturbation sensitivity*. For instance, it may result in a decrease in extraction accuracy and an inability to handle common post-processing scenarios.

Firstly, the process of solving the ODE introduces numerical errors. The linear part $f(t)\,\mathbf{x}_t$ of Eq. (9) can be precisely computed, while the non-linear item $\frac{g^2(t)}{2b_t}\epsilon_\theta(\mathbf{x}_t, t)$ is approximated through complex numerical algorithms. These approximations are the source of numerical errors, exemplified by employing neural network s_θ to approximate $\nabla_{\mathbf{x}_t}\log q(\mathbf{x}_t)$, and by the use of Taylor expansion in Eq. (11) to approximate the integral term. Besides, diffusion models require multiple timesteps to generate samples. **Numerical errors accumulate over these iterative timesteps.** Thus, the bijective mapping established by ODE is inherently unstable, a phenomenon we refer to as *numerical instability*.

Secondly, the bijective mapping is fragile to perturbations at both ends ($\mathbf{x}_0$ and $\mathbf{x}_T$), which we term as *perturbation sensitivity*. Even a small perturbation at one end could be amplified at the other end, even magnified by orders of magnitude. We might reasonably attribute this sensitivity to the presence of the non-linear term, i.e., prediction

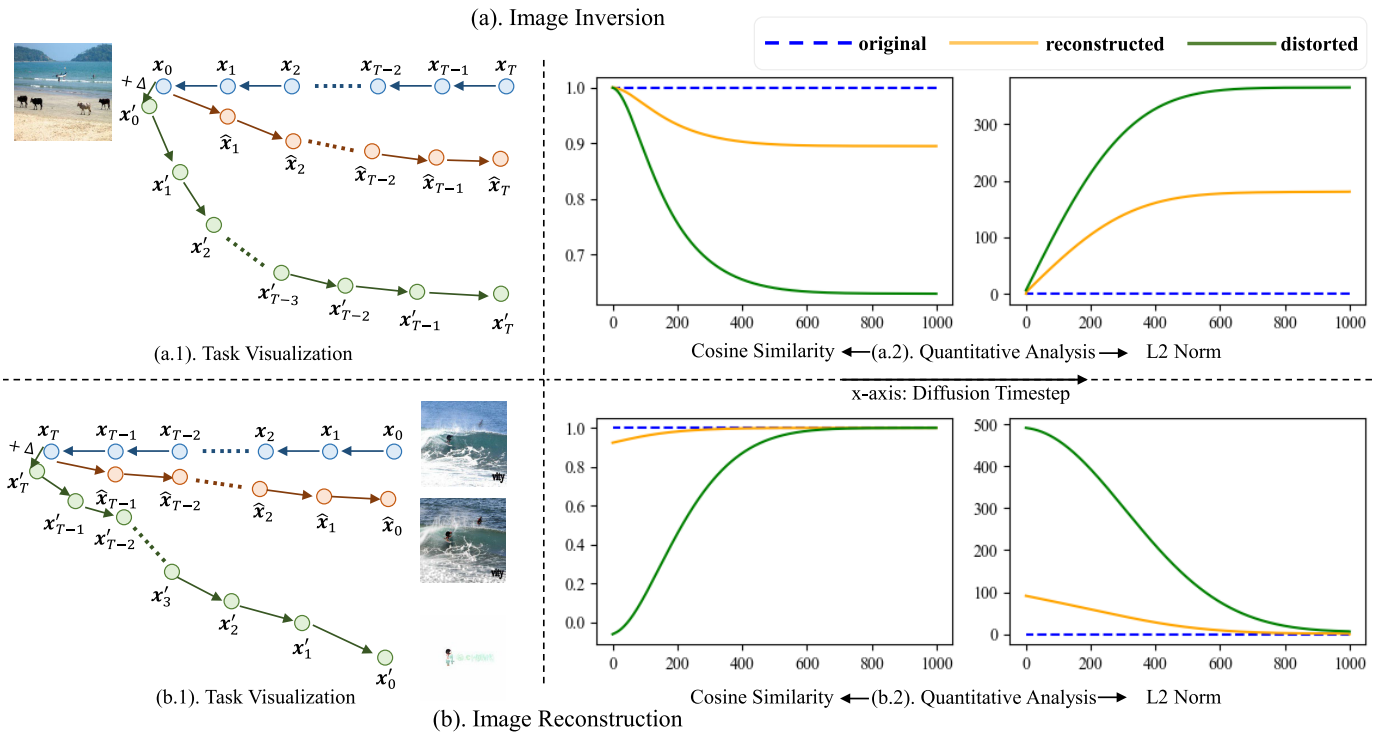


Fig. 1. **Visualization of image inversion (a) and image reconstruction (b).** (a.1) illustrates the process of image inversion, and (b.1) demonstrates the process of image reconstruction. (a.2) and (b.2) are quantitative analyses based on Guided Diffusion [40] and the ODE introduced by DDIM [37]. In (b.2), 1000 real images are randomly sampled from the Microsoft COCO [42] dataset. δ is a small perturbation randomly sampled from a Gaussian distribution with mean 10e-3 and standard deviation 10e-3.

network. The non-linear term is sensitive to tiny shifts in the input distribution, potentially propagating and amplifying minor errors. Through iterative accumulation, these errors lead to significant deviations of the final distribution from the original distribution. Similar phenomena have also been reported in [49] and [50].

We further conduct qualitative and quantitative analyses to illustrate the aforementioned phenomena in Fig. 1. Image inversion and image reconstruction are two visual tasks used for analysis, respectively in Fig. 1(a) and Fig. 1(b). For image inversion, we first generate an image from a randomly sampled Gaussian noise $\mathbf{x}_T$. The data series $\{\mathbf{x}_i\}$ (i from T to 0) are saved as ground truth. Then, we perform the inversion process to obtain $\hat{\mathbf{x}}_T$ from $\mathbf{x}_0$. Comparing the inversion data series $\{\hat{\mathbf{x}}_i\}$ (i from 0 to T, $\hat{\mathbf{x}}_0 = \mathbf{x}_0$) with the ground truth series, we can observe the pattern of inversion errors. Besides, if we add small perturbations to $\mathbf{x}_0$, the inversion series are termed as $\{\mathbf{x}'_i\}$ (i from 0 to T, $\mathbf{x}'_0 = \mathbf{x}_0 + \delta$, δ is a kind of small perturbation). The processes of image inversion are visualized in Fig. 1(a.1). These data series, $\{\mathbf{x}_i\}$, $\{\hat{\mathbf{x}}_i\}$, $\{\mathbf{x}'_i\}$, are denoted as original, reconstructed, distorted series, respectively.

The processes of image reconstruction are similar to image inversion, but the execution order differs. As shown in Fig. 1(b.1), for image reconstruction, the original series goes from a real image $\mathbf{x}_0$ to $\mathbf{x}_T$, the reconstructed series goes from $\hat{\mathbf{x}}_T$ ($= \mathbf{x}_T$) to $\hat{\mathbf{x}}_0$, and the distorted series is obtained by excelling the generation process after adding slight perturbations to $\mathbf{x}_T$. Fig. 2 showcases three examples of image reconstruction. In unperturbed cases, the original semantic information of the image can be reconstructed, albeit with slight differences in details. However, in the images generated from the distorted versions, significant semantic deviations, noticeable artifacts, or loss of background information may occur.

We utilize two metrics, the L2 norm and cosine similarity, to quantitatively evaluate the mapping accuracy. Each curve is obtained after averaging 1000 image samples. The quantitative analysis of image inversion, as depicted in Fig. 1(a.2), elucidates the presence of mapping errors, further amplifying the errors caused by perturbations. For image reconstruction, according to Fig. 1(b.2), $\mathbf{x}_T$ is more sensitive to small perturbations than $\mathbf{x}_0$.

Thus, employing diffusion models for generative steganography poses several challenges that need to be addressed:

- Ensure that the distribution of the input $\mathbf{x}_T$ closely approximates the standard Gaussian distribution.
- Account for potential post-processing operations applied to $\mathbf{x}_0$, such as JPEG compression, necessitating the establishment of a stable mapping relationship.
- Ensure accurate recovery of the secret message even in scenarios where the mapping incurs numerical errors.

In this paper, we propose a novel robust generative image steganography scheme that enables high-quality, text-driven stego image generation in a zero-shot manner. Motivated by these challenges, our framework opts to embed secret messages in the latent space, as it is less susceptible to high-frequency image perturbations after perceptual

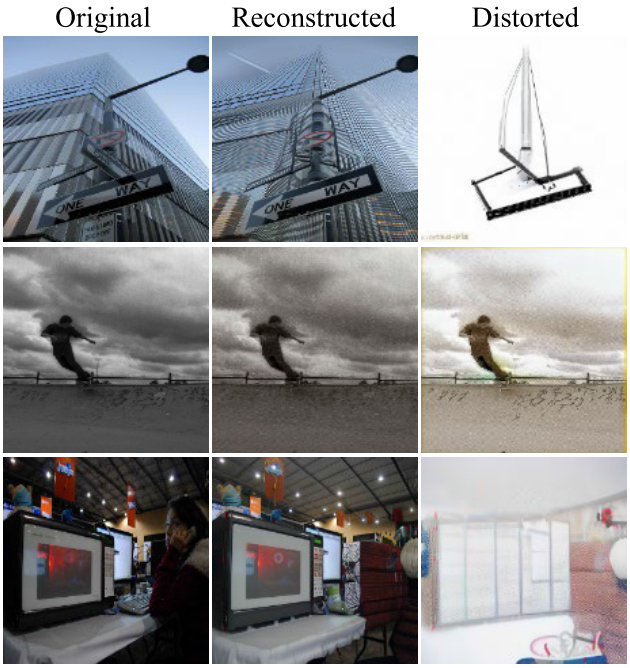


Fig. 2. **Examples of image reconstruction.** The distorted versions of the reconstructed images may exhibit significant semantic deviations, noticeable artifacts, or loss of background information.

compression. We design a novel mapping module based on dual keys to enhance robustness and security under lossy conditions.

B. Problem Formulation

As shown in Fig. 3, we design a bidirectional mapping mechanism and employ the LDM [25] as the generative model. Our method consists of four modules: Secret Message Mapping Module F, LDM Encoder Module E, LDM Decoder Module D, and Image Sampling Module IS. Despite the presence of various lossy operations, our method ensures accurate extraction.

In the context of our problem definition, Alice seeks to send the secret message to Bob, represented as a binary bit string $\mathbf{M}_{ori}$. This convert transmission should remain undetected by Eve. The capacity L refers to the maximum number of bits transmitted during each communication.

Given an encryption key K_{enc}, encryption algorithm ϕ and original secret information $\mathbf{M}_{ori}$, Alice first encrypts $\mathbf{M}_{ori}$ into the ciphertext $\mathbf{M}$,

$$\mathbf{M} = \phi\left(\mathbf{M}_{ori}, K_{enc}\right), \mathbf{M} \sim B\left(l, 0.5\right), 0 \leq l \leq L, \tag{12}$$

where B denotes a binomial distribution and l is the bit length of M. Alice reorganizes M into a multi-channel binary image and then transforms it into the latent image $\mathbf{z}_s$ through the Mapping Module F,

$$\mathbf{z}_s = F\left(reshape\left(\mathbf{M}\right)\right). \tag{13}$$

Alice generates a realistic RGB digital image $\mathbf{I}_s$ from $\mathbf{z}_s$ as follows,

$$\mathbf{I}_s = D\left(IS\_generation\left(\mathbf{z}_s, P\right)\right). \tag{14}$$

Here, $\mathbf{I}_s$ and the text prompt P are semantically correlated. $IS\_generation\left(\cdot\right)$ refers to the IS Generation phase, as depicted in IV-E. After transmission over lossy channels, such as OSNs (online social platforms), Bob receives the transmitted version $\mathbf{I}_r$ of the initially transmitted image $\mathbf{I}_s$. Bob employs a series of inverse operations to decode the secret information $\hat{\mathbf{M}}_{ori}$.

$$\begin{aligned} \mathbf{z}_r &= IS\_inversion\left(E\left(\mathbf{I}_r\right), P\right), \\ \hat{\mathbf{M}} &= F\left(\mathbf{z}_r\right), \\ \hat{\mathbf{M}}_{ori} &= \psi\left(\hat{\mathbf{M}}, K_{dec}\right), \end{aligned} \tag{15}$$

where K_{dec} and ψ respectively refer to the decryption key and the decryption algorithm. $IS\_inversion\left(\cdot\right)$ denotes the IS Inversion phase.

It is challenging to distinguish the differences between $\mathbf{I}_s$ and other normally generated images. Thus, Eve can intercept $\mathbf{I}_s$, but cannot authenticate the presence of secret messages.

C. Secret Message Mapping

According to Eq. (1), all dimensions of the latent code follows an $i.i.d.$ (independent and identically distributed) standard Gaussian distribution. The secret message $\mathbf{M}$ follows a uniform binomial distribution. We design a novel mapping module F that transforms the secret message $\mathbf{M}$ into a deceptive latent vector $\mathbf{z}_s$. The introduction of F is essential to establish a reversible mapping. In case of significant distributional shifts in the latent space, the reverse ODE may introduce certain errors, as observed in IV-A.

Each element of the reshaped $\mathbf{M} \in \{0, 1\}^{C_r \times N \times N}$ is independent and follows a Bernoulli distribution with a probability of 0.5. Here, C_r represents the number of channels in the reshaped secret message, while N denotes the size of the latent image in both width and height dimensions. We first normalize it to a distribution with a mean of 0 and a variance of 1.

$$\begin{aligned} \mathbf{M}_{re} &= reshape\left(\mathbf{M}\right) \times 2 - 1, \\ m_{kij} &\in \mathbf{M}_{re}\left(1 \leq k \leq C_r, 1 \leq i, j \leq N\right). \end{aligned} \tag{16}$$

Here, we introduce an orthogonal matrix $\mathbf{C} \in \mathbb{R}^{N \times N}$, termed as the transformation kernel. We design the mapping function as follows,

$$\mathbf{z}_s = \mathbf{C} \cdot \mathbf{M}_{re} \cdot \mathbf{C}^T, \tag{17}$$

where $\cdot$ is matrix multiplication and $\mathbf{C}^T$ is the transpose of $\mathbf{C}$. To further enhance security, following the Kerckhoffs' principle, we randomly shuffle $\mathbf{z}_s$ using the key K_{shu}. Alice and Bob share the same key K_{shu}, which is used to perform the shuffle and unshuffle operations, respectively.

Within F, the transformation kernel $\mathbf{C}$ is also randomly generated. Alice and Bob share the same key K_{seed}, serving as a seed to generate an initial matrix $\mathbf{C}_{init}$. Through the Gram-Schmidt orthogonalization process, $\mathbf{C}_{init}$ becomes a randomly generated orthogonal matrix $\mathbf{C}$,

$$\mathbf{C} \coloneqq \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1N} \\ a_{21} & a_{22} & \ldots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \ldots & a_{NN} \end{bmatrix}. \tag{18}$$

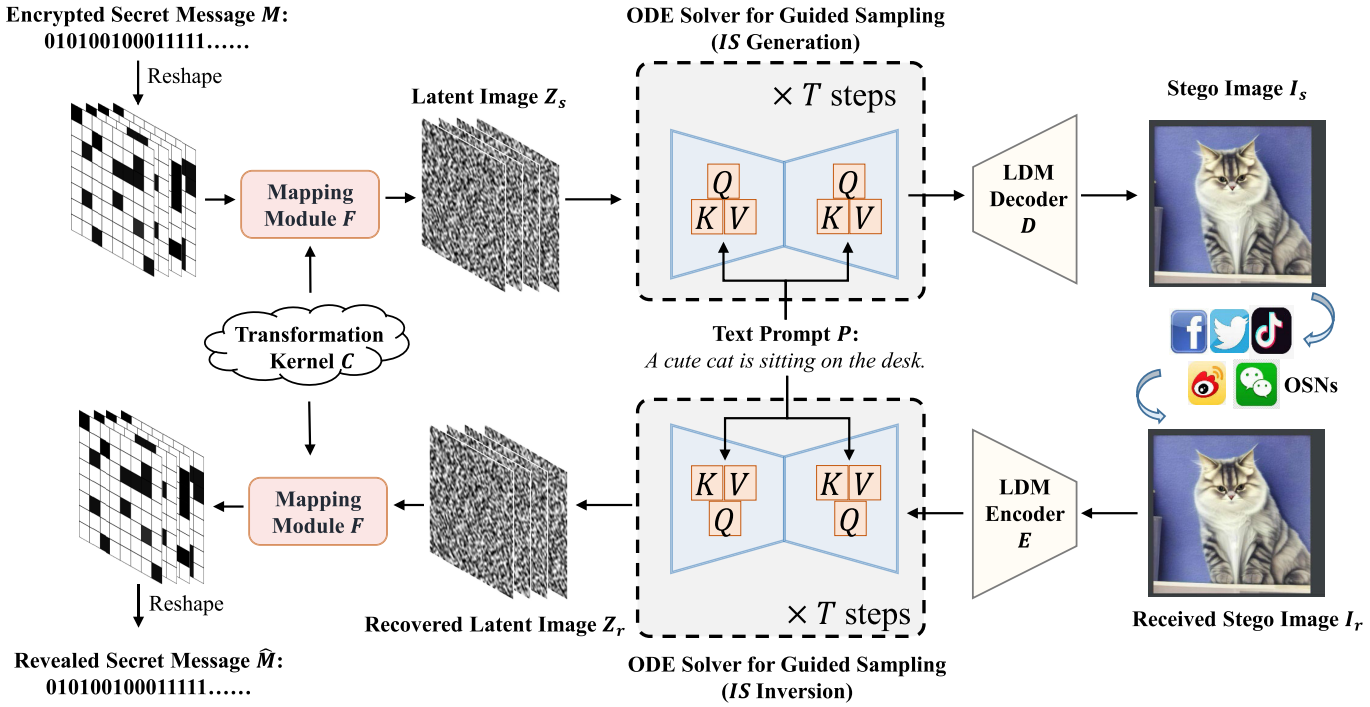


Fig. 3. **The framework of the proposed method.** Our approach transforms secret messages into stego images using text prompts as auxiliary control signals. Extensive experiments demonstrate the capability of our method to accurately extract secret messages from received stego images, particularly in the presence of lossy operations.

Therefore, for d_{kij} in $\mathbf{C}\cdot\mathbf{M}_{re}$, we have:

$$d_{kij}=\sum_{l}^{N}a_{il}\times m_{klj}. \tag{19}$$

By fundamental principles of probability theory, we can derive the mean μ and variance σ^2 of d_{kij}.

$$\begin{aligned}\mu\left(d_{kij}\right)&=\mu\left(\sum_{l}^{N}a_{il}\times m_{klj}\right)=0,\\ \sigma^2\left(d_{kij}\right)&=\sigma^2\left(\sum_{l}^{N}a_{il}\times m_{klj}\right)=\sum_{l}^{N}a_{il}^2=1.\end{aligned} \tag{20}$$

If N is large and the number of non-zero elements in $\mathbf{C}$ approaches $N\times N$, according to the CLT (Central Limit Theorem), d_{kij} will follow a standard Gaussian distribution $\mathcal{N}(0,1)$. Additionally, for different d_{kij}, they are independent of each other because their covariance coefficient, $Cov(\cdot)$, is 0.

$$\begin{aligned}&k_1\neq k_2\vee i_1\neq i_2\vee j_1\neq j_2, Cov\left(d_{k_1i_1j_1},d_{k_2i_2j_2}\right)\\&=\mu\left(\sum_{l1}^{N}\sum_{l2}^{N}a_{i_1l_1}\times a_{i_2l_2}\times m_{k_1l_1j_1}\times m_{k_2l_2j_2}\right)=0.\end{aligned} \tag{21}$$

Thus, all dimensions of $\mathbf{C}\cdot\mathbf{M}_{re}$ approximately follows an $i.i.d.$ standard Gaussian distribution. Following a similar reasoning process, for each element z_{kij} in $\mathbf{z}_s$, we have,

$$\begin{aligned}\mu\left(z_{kij}\right)&=\mu\left(\sum_{l}^{N}d_{kil}\times a_{lj}\right)=0,\\ \sigma^2\left(z_{kij}\right)&=\sigma^2\left(\sum_{l}^{N}d_{kil}\times a_{lj}\right)=\sum_{l}^{N}a_{lj}^2=1,\\ &k_1\neq k_2\vee i_1\neq i_2\vee j_1\neq j_2, Cov\left(z_{k_1i_1j_1},z_{k_2i_2j_2}\right)=0.\end{aligned} \tag{22}$$

It can be derived that $\mathbf{z}_s$ also approximately follows an $i.i.d.$ standard Gaussian distribution. Based on the above equations, we have demonstrated that the mapping module can transform the secret message into Gaussian noise. Moreover, the dispersed mapping approach and the introduction of random processes further enhance the security and robustness of the proposed mechanism. If Bob needs to extract the secret message, he firstly reverses the shuffling operation on $\mathbf{z}_s$ using the key K_{shu}. Bob can then reveal the secret message by reversing Eq. (17) and Eq. (16).

$$\begin{aligned}\mathbf{M}_{re}&=\mathbf{C}^T\cdot\mathbf{z}_s\cdot\mathbf{C},\\ \mathbf{M}&=reshape\left(\left[\frac{\mathbf{M}_{re}+1}{2}\right]\right).\end{aligned} \tag{23}$$

Here, $[\cdot]$ denotes rounding to the nearest integer.

D. Latent Diffusion Model

LDM utilizes Vector Quantized Variational Autoencoder (VQ-VAE) [51] for perceptual image compression. Given a RGB image $\mathbf{x}\in\mathbb{R}^{H\times W\times 3}$, the encoder E encodes $\mathbf{x}$ into a latent image $\mathbf{z}$. The decoder D can reconstruct $\mathbf{x}$ from $\mathbf{z}$.

$$\mathbf{z}=E\left(\mathbf{x}\right),\hat{\mathbf{x}}=D\left(\mathbf{z}\right),\hat{\mathbf{x}}\approx\mathbf{x}, \tag{24}$$

where $\mathbf{z}\in\mathbb{R}^{\frac{H}{f}\times\frac{W}{f}\times C_f}$. Here, f denotes the downsampling factor that reduces the spatial dimensions, and C_f indicates the number of channels in the latent representation $\mathbf{z}$.

LDM employs text-conditional UNet [52] $\epsilon_\theta(\mathbf{z}_t, t, \mathbf{e}_{text})$ in the latent space. Here, $\mathbf{e}_{text}$ is the embedding of the text prompt extracted by CLIP [41]. In contrast to DDPM, LDM capitalizes on CLIP's cross-modal alignment capability while incorporating a classifier-free guidance [53] mechanism as follows,

$$\begin{aligned}&\tilde{\epsilon}_\theta(\mathbf{z}_t, t, \mathbf{e}_{text})\\&= \epsilon_\theta(\mathbf{z}_t, t, \mathbf{e}_\emptyset) + w\left(\epsilon_\theta(\mathbf{z}_t, t, \mathbf{e}_{text}) - \epsilon_\theta(\mathbf{z}_t, t, \mathbf{e}_\emptyset)\right),\end{aligned} \tag{25}$$

where $\emptyset$ means *NULL* text, and w is the guidance scale factor that modulates the strength of the text conditioning on the generated image. Replacing $\epsilon_\theta(\mathbf{z}_t, t)$ in Eq. (11) with $\tilde{\epsilon}_\theta(\mathbf{z}_t, t, \mathbf{e}_{text})$, LDM has achieved a formidable capability in controlling the content of generated images. We do not explicitly specify the distinction between $\tilde{\epsilon}_\theta$ and ϵ_θ in the subsequent text for simplicity.

E. Information Hiding

As shown in Algorithm 1, Alice encrypts the secret message initially. In Section IV-C, Alice can obtain $\mathbf{z}_T = \mathbf{z}_s = F(\mathbf{M}, K_{seed}, K_{shu})$, where $\mathbf{z}_T$ approximately adhere to an $i.i.d.$ standard Gaussian distribution. Subsequently, Alice selects a text prompt P and applies a specific ODE solver, such as DPM-Solver++2M, to $\mathbf{z}_T$, iteratively denoising to acquire $\mathbf{z}_0$. This phase, termed as IS Generation, involves a series of timesteps $T = t_0 > t_1 > t_2 > \ldots > t_{i-1} > t_i > \ldots > t_{steps-1} > t_{steps} = 0$, where $steps$ represents the total number of timesteps (set to 20 as default). For each denoising timestep, the detailed implementation of Eq. (11) is as follows,

$$\begin{aligned}\mathbf{z}_{t_i} \approx \frac{b_{t_i}}{b_{t_{i-1}}}\mathbf{z}_{t_{i-1}} + b_{t_i}\sum_{n=0}^{k-1}\mathbf{z}_\theta^{(n)}\\ \times\left(\mathbf{z}_{t_\lambda\left(\lambda_{t_{i-1}}\right)}, t_\lambda\left(\lambda_{t_{i-1}}\right)\right)\int_{\lambda_{t_{i-1}}}^{\lambda_{t_i}} e^\lambda\frac{\left(\lambda-\lambda_{t_{i-1}}\right)^n}{n!}\mathrm{d}\lambda.\end{aligned} \tag{26}$$

According to Eq. (2), $\mathbf{z}_\theta$ is deduced by ϵ_θ to predict $\mathbf{z}_0$. Here, we denote $\mathbf{z}_\theta(\mathbf{z}_t, t, \mathbf{e}_P)$ as $\mathbf{z}_\theta(\mathbf{z}_t, t)$ for simplicity. k refers to the k-th order Taylor expansion, and $(\cdot)^n$ represents the n-th order derivation with respect to λ. In this paper, we utilize the second-order expansion formula of Eq. (26),

$$\begin{aligned}\mathbf{z}_{t_i} \approx \frac{b_{t_i}}{b_{t_{i-1}}}\mathbf{z}_{t_{i-1}} + a_{t_i}\left(1-e^{-h_i}\right)\mathbf{z}_\theta\left(\mathbf{z}_{t_\lambda\left(\lambda_{t_{i-1}}\right)}, t_\lambda\left(\lambda_{t_{i-1}}\right)\right)\\ + a_{t_i}\left(e^{-h_i}-1+h_i\right)\\ \times\frac{\mathbf{z}_\theta\left(\mathbf{z}_{t_\lambda(\lambda_{s_i})}, t_\lambda\left(\lambda_{s_i}\right)\right) - \mathbf{z}_\theta\left(\mathbf{z}_{t_\lambda\left(\lambda_{t_{i-1}}\right)}, t_\lambda\left(\lambda_{t_{i-1}}\right)\right)}{r_i h_i},\end{aligned} \tag{27}$$

where $h_i = \lambda_{t_i} - \lambda_{t_{i-1}}$. And s_i is an intermediate point $(t_{i-1} > s_i > t_i)$ for approximating the first derivation, i.e., $0 < r_i := \frac{\lambda_{s_i} - \lambda_{t_{i-1}}}{h_i} < 1$. Generally, r_i is set to 0.5. According to [38], leveraging an approximation term leads to slightly better performance, so we adhere to this setting.

$$\frac{e^{-h_i}-1+h_i}{h_i} \approx -\frac{-h_i}{2} + \mathcal{O}\left(h_i^2\right) \approx \frac{1-e^{-h_i}}{2}, \tag{28}$$

Algorithm 1 Information Hiding

Input:
Secret bitstream: $\mathbf{M}_{ori}$ = "010101111000…"
The encryption key: K_{enc}
Two keys for random processes: K_{seed}, K_{shu}
The text prompt: P
Other parameters: $N \leftarrow 64$, $C_f \leftarrow 4$, $C_r \leftarrow 4$, $H, W \leftarrow 512$, $f \leftarrow 8$
Output:
Stego image: $\mathbf{I}_s$

Encrypt secret bitstream: $\mathbf{M} \leftarrow \phi(\mathbf{M}_{ori}, K_{enc})$
Reorganize and normalize: $\mathbf{M}_{re} \leftarrow reshape(\mathbf{M}) \times 2 - 1$
Set the seed of the random number generator to K_{seed}
Generate a random $N \times N$ matrix: $\mathbf{C}_{init}$
Construct $\mathbf{C} \leftarrow$ Schmidt-Orthogonalization $(\mathbf{C}_{init})$
Generate deceptive noise input: $\mathbf{z}_s \leftarrow \mathbf{C} \cdot \mathbf{M}_{re} \cdot \mathbf{C}^T$
Set the seed of the random number generator to K_{shu}
Randomly shuffle the elements of $\mathbf{z}_s$
Set shuffled $\mathbf{z}_s$ as the noise input $\mathbf{z}_T$
IS Generation Stage: Iteratively denoise $\mathbf{z}_T$ to $\mathbf{z}_0$ by P
Generate stego image: $\mathbf{I}_s \leftarrow D(\mathbf{z}_0)$

where $\mathcal{O}(\cdot)$ represents the high-order terms. Thus, Eq. (27) can be rewritten as,

$$\begin{aligned}\mathbf{z}_{t_i} \approx \frac{b_{t_i}}{b_{t_{i-1}}}\mathbf{z}_{t_{i-1}} + a_{t_i}\left(1-e^{-h_i}\right)\mathbf{z}_\theta\left(\mathbf{z}_{t_\lambda\left(\lambda_{t_{i-1}}\right)}, t_\lambda\left(\lambda_{t_{i-1}}\right)\right)\\ + a_{t_i}\left(1-e^{-h_i}\right)\\ \times\frac{\mathbf{z}_\theta\left(\mathbf{z}_{t_\lambda(\lambda_{s_i})}, t_\lambda\left(\lambda_{s_i}\right)\right) - \mathbf{z}_\theta\left(\mathbf{z}_{t_\lambda\left(\lambda_{t_{i-1}}\right)}, t_\lambda\left(\lambda_{t_{i-1}}\right)\right)}{2r_i}.\end{aligned} \tag{29}$$

Finally, Alice uses meaningful latent image $\mathbf{z}_0$ as the input of the LDM Decoder D, to generate the stego image $\mathbf{I}_s$. After transmission over a lossy channel, Bob can receive a distorted version $\mathbf{I}_r$.

F. Information Extraction

At the extraction stage, Bob employs three modules (E, IS, and F), in conjunction with supplementary side information. Specifically, the side information includes the decryption key (K_{dec}), two seed keys (K_{seed}, K_{shu}), and the text prompt (P).

Bob uses the LDM Encoder E to transform $\mathbf{I}_r$ into a latent image $\hat{\mathbf{z}}_0$ and then proceeds with the IS Inversion stage. The only difference between IS Generation and IS Inversion lies in the time series. Specifically, at the inversion stage, the given time series is $0 = t_0 < t_1 < t_2 < \ldots < t_{i-1} < t_i < \ldots < t_{steps-1} < t_{steps} = T$. By iteratively adding noise according to Eq. (29), Bob recovers $\mathbf{z}_r = \hat{\mathbf{z}}_T$. It is important to note that the text prompt P is utilized as control information at each time step. Using the key K_{shu} to unshuffle $\mathbf{z}_r$, yielding a result similar to $\mathbf{z}_s$, and employing K_{seed} to obtain the orthogonal matrix form of the transformation kernel $\mathbf{C}$, Bob approximates M using Eq. (23). Finally, by employing the

decryption algorithm ψ with the key K_{dec}, Bob can decipher the original secret message.

V. Experiments

A. Experimental Settings

1) Implementation Details: We use Stable Diffusion v1.5 as the default LDM which is trained on the LAION-5B [54] dataset. The hyperparameters are set as follows: $N = 64, C_f, C_r = 4, H, W = 512, f = 8, w = 5.0$. Thus, the dimensions of the reshaped secret message and the latent vector $\mathbf{z}_s$ are consistent, ensuring the validity of our mapping mechanism. All experiments are conducted on four NVIDIA RTX 4090 GPUs using the PyTorch framework.

2) Evaluation Datasets: The proposed method uses the original LDM model without any fine-tuning or retraining. In this paper, we utilize four datasets to evaluate the performance of our method.

- DescGPT. ChatGPT [55] is a versatile and powerful language model designed by OpenAI. ChatGPT can generate a set of descriptive sentences. We manually filter and remove duplicated sentences, curating a collection of 654 text prompts, abbreviated as DescGPT.
- LAION-10K. LAION-5B [54] dataset has 5 billion text-image pairs. We randomly select 10000 text prompts from the LAION-5B dataset for testing.
- MS-COCO. Microsoft COCO [42] val2017 dataset has five thousand image samples. Each image is accompanied by five textual descriptions. For each sample, we randomly select one description, creating a test set with 5 thousand text prompts. These 5000 images are utilized to evaluate the visual quality of our method.
- Flicker8K. The Flicker8K [56] dataset has 8092 photographs. Each image in the Flicker8K dataset has five corresponding text captions. We randomly select one caption from each image to construct a dataset with 8092 text prompts. Besides, these 8092 images are employed to assess the visual quality of our method.

3) Evaluation Metrics: In the experiments, we adopt various metrics to evaluate the performance of our method. These metrics cover aspects such as hiding capacity, extraction accuracy, robustness, security, and image visual quality.

*Hiding Capacity:*Bits per pixel (bpp) quantifies the amount of secret bits that can be concealed with a single pixel of an image. We evaluate the hiding capacity by bpp, which is defined as follows,

$$bpp = \frac{L}{H \times W}, \tag{30}$$

where L represents the maximum length of the secret message, while H and W denote the height and width of the image, respectively.

*Extraction Accuracy:*We simulate the encrypted secret message $\mathbf{M}$ by randomly sampling from the binomial distribution, and measure the extraction accuracy using the Hamming Distance $dist(\cdot, \cdot)$ as follows,

$$acc = 1 - \frac{dist\left(\mathbf{M}, \hat{\mathbf{M}}\right)}{L}. \tag{31}$$

*Robustness:*The real-world transmission of digital images incurs distortions. Moreover, it's challenging to determine the specific type of lossy operations within a blind channel. To assess the robustness of the proposed steganographic method under lossy transmission, we test the extraction accuracy under multiple common attacks as follows:

1) Resize. Resize can alter the resolution of digital images. Using bilinear interpolation, we adjust the image to various sizes with scaling factors of 0.5, 0.75, 1.25, and 1.5.

2) JPEG Compression. JPEG compression is widely used to reduce the file size of digital images. We evaluate JPEG compression with quality factors of 90, 70, and 50, reflecting varying degrees of image degradation.

3) Median Blur. Median blur smooths images by replacing each pixel's value with the median value of neighboring pixels. We apply kernel sizes of 3×3, 5×5, and 7×7 to represent slight, moderate, and significant distortions, respectively.

4) Gaussian Blur. Instead of applying median values, gaussian blur utilizes a Gaussian kernel to smooth local regions. We employ kernel sizes of 3×3, 5×5, and 7×7 to represent slight, moderate, and significant distortions, respectively.

5) Gaussian Noise. Gaussian noise also termed AWGN (Additive White Gaussian Noise), is applied by directly adding Gaussian noise to an image. A higher standard deviation of the noise implies more severe degradation of the image. In the experiments, we explore three cases with standard deviations of 0.01, 0.05, and 0.1, respectively.

We first save the images in PNG format and then apply one of the above perturbations for evaluation. For comparison purposes, we also test two non-attack scenarios: lossless storage (without quantization) and PNG storage (with quantization).

Security: In steganography schemes, security hinges on Eve's incapacity to discern between the cover image and the stego image. However, generative steganography architectures eliminate the need for pixel-level alterations of the cover images. Furthermore, with the advanced generative models, it has become commonplace to disseminate synthesized images via OSNs. Therefore, we define the normal images generated from random sampled noise as "cover images", while "stego images" are those generated from secret messages. We employ four steganographic analyzers, namely SRNet [12], XuNet [13], YeNet [14], and SiaStegNet [15], to classify the cover and stego images. In the experiments, we utilize the detection error P_E to evaluate the security of steganographic methods, which is defined as follows,

$$P_E = \frac{P_{FA} + P_{MD}}{2}, \tag{32}$$

where P_{FA} is the false alarm rate and P_{MD} is the missed detection rate.

*Visual Quality:*We use FID (Fréchet Inception Distance) [57] to evaluate the visual quality of generated images. The FID metric first projects images into feature space using a pre-trained Inception V3 [58] model. Then, it measures the similarity between the distributions of real and generated images in feature space as follows,

$$\text{FID} = \|\mu_r - \mu_g\|^2 + \text{Tr}(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{1/2}). \tag{33}$$

Fig. 4. **Exampled stego images generated using diverse text prompts and the same secret message.** The four columns correspond to different scenes, namely flowers, individuals, coffee, and landscapes, respectively.

TABLE I

COMPARISONS WITH SOTA METHODS. THIS COMPARISON IS CONDUCTED IN TERMS OF METHOD TYPE, ANTI-STEGANALYSIS CAPABILITY, CAPACITY LEVEL (HIGH, MEDIUM, LOW), ZERO-SHOT, STEGO IMAGE RESOLUTION, CONTROLLABILITY, AND ROBUSTNESS

Method	generative	anti-steganalysis	capacity	zero-shot	image resolution	controllability	robustness
[11]	✗	✓	medium	✗	-	-	✗
[29]	✗	✓	high	✗	-	-	✗
[31]	✓	✓	low	✗	32	✗	resist common distortions
[32]	✓	✓	high	✗	128	✗	✗
[16]	✓	✓	high	✗	128, 256	✗	✗
[23]	✓	✓	low	✗	32	✗	channel-aware robustness
[18]	✓	✓	low	✗	256	reference image	✗
[2]	✓	✓	high	✓	256	✗	resist slight distortions
[34]	✓	✓	low	✗	256	object contour	resist slight distortions
[33]	✓	✓	high	✗	64	✗	✗
[22]	✓	✓	medium	✗	64	✗	channel-aware robustness
proposed	✓	✓	medium	✓	512	text prompt	resist common distortions

TABLE II

IMAGE QUALITY ASSESSMENT .FID

Dataset	cover	Baseline A	Baseline B	ours
MS-COCO	20.4417	20.6606	20.6354	20.5693
Flicker8K	16.2516	16.6389	16.6414	16.6004

TABLE III

IMAGE QUALITY ASSESSMENT .BRISQUE

Dataset	real	cover	Baseline A	Baseline B	ours
MS-COCO	16.5424	19.8696	19.6400	19.8258	19.4550
Flicker8K	19.6109	21.3668	21.4138	21.3060	21.3189

Here, μ_r and μ_g represent the mean feature vectors of real and generated images, respectively, Σ_r and Σ_g denote their covariance matrices, and $\mathrm{Tr}(\cdot)$ denotes the trace of the matrix. A lower FID score signifies a higher similarity between the distributions of real and generated images, indicating superior image quality. Besides, we employ the classic NR-IQA (No-Reference Image Quality Assessment) scheme to analyze the image's visual quality. Brisque (Blind/Referenceless Image Spatial Quality Evaluator) [59] quantifies image quality by analyzing various spatial domain features extracted from the image. It does not require a reference image for comparison, making it suitable for evaluating generative steganography. A lower Brisque score indicates better image quality.

4) Baseline Configurations: In contrast to previous mapping-based generative steganography schemes, we propose a robust generative steganographic method tailored for blind transmission channels, where text prompts guide the generation process of stego images. Given differences in application contexts and input modalities, direct comparison with previous SOTA schemes is unfair. Therefore, we conduct a qualitative comparison with existing SOTA methods and establish two baselines for quantitative comparison, as follows:

- Baseline A: Replace the mapping module from our method with the mapping module from SE-S2IRT [2],

while keeping the remaining modules unchanged. Leveraging the inherently reversible Glow model [19], SE-S2IRT demonstrates high hiding capacity and extraction accuracy, while also exhibiting robustness against slight image perturbation attacks. The mapping module of SE-S2IRT rearranges the order of noise input to embed the secret message. Specifically, it initially categorizes all elements in the noise based on their magnitude, retaining only those elements close to the center of each group for rearrangement.

- Baseline B: Yang et al. [22] propose a provably secure robust image steganography based on GAN [35] models, demonstrating robustness against channel-aware attacks. Similarly, we employ the mapping module from [22] to construct Baseline B. For each pixel z in the noise input, the corresponding bit m in the secret ($m \in \{0, 1\}$). The processes of forward mapping (from m to z) and inverse mapping (from z to m) are illustrated as follows:

$$z = \text{ppf}\left(\frac{u+m}{2}\right), m = \lfloor 2 \times \text{cdf}(z) \rfloor \tag{34}$$

where u is a real number randomly sampled from the uniform distribution $U(0, 1)$, $\text{cdf}(\cdot)$ represents the cumulative distribution function of the standard Gaussian distribution, and $\text{ppf}(\cdot)$ is the inverse function of $\text{cdf}(\cdot)$, also known as the quantile function.

After establishing baselines, we analyze their performance of extraction accuracy, robustness, security, and image quality.

B. Image Visual Quality

1) Qualitative Analysis: Leveraging the powerful generation capabilities of Stable Diffusion, the proposed method can generate diversified, high-quality stego images, each sized 512 × 512, providing versatility in concealing secret messages. As shown in Fig. 4, we generate stego images from the same secret message and various text prompts. These example images depict a diverse range of scenarios, including flowers, individuals, landscapes, and coffee, among others. The realistic details and varied content showcased in the images demonstrate the excellent generation performance of our method.

In Table I, the comparison results with SOTA methods show that our method exhibits superior image visual quality in generative steganography, including aspects such as image resolution and fidelity. Furthermore, we do not modify the original generative model, thereby preserving its original generation performance. By introducing the text prompts as control signals, we achieve precise control over the generated content. Thus, the secret message can be seamlessly disguised as diverse images, augmenting the covert nature of our method.

2) Quantitative Analysis: To evaluate the visual quality of generated images, we adopt the natural images from the MS-COCO and Flicker8K datasets for comparison. The computation of the FID metric involves both real and generated images. We separately compute the FID scores for the cover and stego images concerning the natural images in the dataset. Brisque eliminates the need for reference images. Therefore, we conduct separate evaluations on real, cover, and stego images. Table II and Table III respectively present the experimental results of Baseline A, Baseline B, and our method.

Table II demonstrates that the FID scores of our method for stego and cover images are very close. For brisque scores, as shown in Table III, there is a gap between the scores of generated images and real images, while the difference between stego and cover images is tiny. These experimental findings affirm that our method can seamlessly embed the secret message without substantial degradation in the original generation quality. As discussed in Section IV-C, our mapping module guarantees that the mapping results conform to the original Gaussian distribution, which is also held for Baseline A and Baseline B. When testing image quality, our experimental results closely align with those of the baselines, consistent with theoretical analysis.

C. Hiding Capacity

To accommodate real-world scenarios involving lossy transmission, our method selects a more stable latent space for concealing and extracting secret information. In our experiments, the latent space comprises $4 \times 64 \times 64 = 16384$ elements. In this context, the hiding capacity of our method is $\frac{16384}{512 \times 512} = 0.0625$ *bpp*. For a fair comparison, the configurations of Baselines A and B ensure that the hiding capacity aligns precisely with our method. As shown in Eq. (34), the mapping of Baseline B matches the embedding efficiency of our method. For Baseline A, we establish 6 groups and select 12288 central elements for rearrangement. In Table I, we classify the capacity into three tiers: high ($\geq$ 1 *bpp*), medium (from 0.05 *bpp* to 1 *bpp*), and low ($\leq$ 0.05 *bpp*). Our method sacrifices some capacity to enhance robustness.

D. Extraction Accuracy & Robustness

Table IV presents the average accuracy of secret message extraction under various operations. We employ four datasets for evaluation: DescGPT, LAION-10K, MS-COCO, and Flicker8K. As observed from the experimental results, the extraction accuracy exhibits a gradual decrease along with increasing attack strength. However, our method experiences only minor performance degradation during most post-processing operations. For common distortions encountered in the transmission through online social platforms, such as JPEG90 and Resize, the extraction accuracy exceeding 95% demonstrates practicability. The test results on the LAION-10K dataset are slightly lower compared to the other datasets, possibly due to the "noise" of the text prompts. Here, "noise" refers to the text prompt that requires context for understanding, such as "Chipper Jones-0617.jpg". The LAION-5B dataset itself contains 5.85 billion image-text pairs, and we randomly sampled a small subset, which might introduce some biases.

We evaluate the extraction accuracy of Baseline A, Baseline B, and our method under different attack scenarios using the MS-COCO dataset. As delineated in Table V, this comparison primarily varies in the mapping module utilized. Notably, our mapping module showcases a notable enhancement of over 10% in extraction accuracy under non-attack conditions,

TABLE IV
EXTRACTION ACCURACY OF OUR METHOD

Dataset	Lossless	PNG	Resize				JPEG Compression			Median Blur			Gaussian Blur			Gaussian Noise		
	-	-	0.5	0.75	1.25	1.5	90	70	50	3×3	5×5	7×7	3×3	5×5	7×7	0.01	0.05	0.1
DescGPT	98.75	98.72	96.45	97.97	98.44	98.54	97.86	95.53	93.32	97.13	92.00	84.98	98.19	97.30	95.23	98.45	95.14	91.15
LAION-10K	95.89	95.72	91.46	94.15	95.19	95.44	93.50	89.24	85.76	92.57	86.57	79.83	94.27	92.57	89.64	94.85	88.28	82.31
MS-COCO	98.12	98.05	94.86	97.03	97.74	97.89	96.34	92.35	88.87	95.84	90.10	82.84	97.12	95.79	93.12	97.38	90.94	84.73
Flicker8K	98.91	98.86	96.82	98.29	98.72	98.81	97.79	94.99	92.21	97.36	92.32	84.95	98.25	97.29	95.11	98.46	93.85	88.29

TABLE V
EXTRACTION ACCURACY COMPARISON. MS-COCO DATASET

Method	Lossless	PNG	Resize				JPEG Compression			Median Blur			Gaussian Blur			Gaussian Noise		
	-	-	0.5	0.75	1.25	1.5	90	70	50	3×3	5×5	7×7	3×3	5×5	7×7	0.01	0.05	0.1
Baseline A	81.17	81.06	77.66	79.64	80.59	80.83	79.06	76.53	75.02	78.37	74.92	72.52	79.60	78.15	76.20	80.17	76.15	73.78
Baseline B	86.90	86.82	83.71	85.68	86.46	86.65	84.99	81.76	79.20	84.45	79.94	74.89	85.66	84.31	81.98	86.03	80.80	76.25
ours	**98.12**	**98.05**	**94.86**	**97.03**	**97.74**	**97.89**	**96.34**	**92.35**	**88.87**	**95.84**	**90.10**	**82.84**	**97.12**	**95.79**	**93.12**	**97.38**	**90.94**	**84.73**

TABLE VI
SECURITY. FLICKER8K DATASET & MS-COCO DATASET

Dataset	Method	SRNet	XuNet	YeNet	SiaStegNet
Flicker8K	Baseline A	0.5070	0.5020	0.5000	0.4950
	Baseline B	0.4955	0.4940	0.5100	0.5085
	ours	0.4915	0.4910	0.5060	0.5035
MS-COCO	Baseline A	0.5108	0.5350	0.4842	0.5233
	Baseline B	0.4883	0.5133	0.5100	0.5141
	ours	0.5075	0.5017	0.4792	0.4867

surpassing both baseline methods. Moreover, our method demonstrates robustness by sustaining high extraction accuracy across diverse attack scenarios.

E. Security

We utilize four steganographic analyzers, namely SRNet [12], XuNet [13], YeNet [14], and SiaStegNet [15], to classify the cover and stego images. For the MS-COCO dataset, we generate 5000 cover images with random noise and 5000 stego images with secret input. The data is then partitioned into training, validation, and test sets with a ratio of 3800:600:600, respectively. For example, the training set comprises 3800 cover images and 3800 stego images. The split of the Flicker8K dataset follows a similar procedure, with the only difference being the partition ratio of 6092:1000:1000.

The security performance on the Flicker8K and MS-COCO datasets is listed in Table VI. The detection error rate P_E fluctuates around 0.5, indicating that the steganalyzers cannot correctly classify cover and stego images. As outlined in Section IV-C, the design of our mapping module theoretically ensures that the secret input closely approximates the original Gaussian distribution, a property also upheld by Baselines A and B. The experimental results also demonstrate the superior security introduced by distribution-preserving mechanisms.

F. Ablation Study

In prior experiments, we validate the mapping module by comparing its performance with Baselines A and B. In this part, We further explore the efficacy and flexibility of the image sampling module through extensive ablation studies across different conditions.

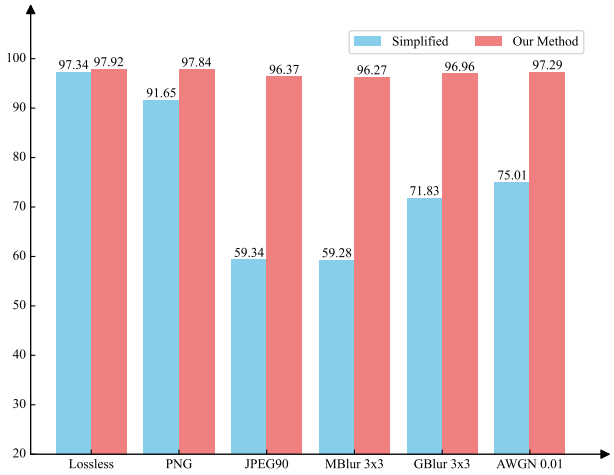


Fig. 5. **Ablation experimental results.** Extraction accuracy under different distortion types using the simplified version and our method. "JPEG" denotes JPEG Compression, while "MBlur" and "GBlur" correspond to Median Blur and Gaussian Blur, respectively. The numerical values following each term specify the respective distortion parameters.

We implement a simplified version using the Guided Diffusion model [40]. This simplified image sampling module utilizes DDIM [37] as the ODE solver for both generation and inversion. Integrated with the mapping module of our method, we randomly generate 2000 stego images and evaluate the extraction accuracy under slight attack scenarios, as depicted in Fig. 5. For comparison, we average the extraction accuracy across the four datasets listed in Table IV to observe the performance degradation trend of our method. It can be observed that slight perturbations significantly degrade the performance of the simplified version. While the mapping mechanism offsets some errors, minor quantization errors induced by PNG storage still result in a decrease of over 5%. However, in our method, the instability caused by perturbations is effectively mitigated. Specifically, we perform perceptual compression and dimensionality reduction on the images, filtering out unstable high-frequency components. Although this sacrifices a certain amount of hiding capacity, it enhances the accuracy and stability of the extraction process.

Theoretically, the proposed method is independent of any specific ODE solver. To demonstrate its versatility, we conduct

TABLE VII
ABLATION ON ODE SOLVERS. MS-COCO DATASET

Quantitative Evaluation	ODE Solver		
	DDIM	PNDM	DPM-Solver++
Lossless	98.14	**98.56**	98.12
PNG	98.07	**98.51**	98.05
JPEG90	94.25	95.92	**96.34**
MBlur 3×3	92.05	94.24	**95.84**
GBlur 3×3	95.39	96.71	**97.12**
AWGN 0.01	96.99	**97.77**	97.38
FID	20.6514	20.6339	**20.5693**

TABLE VIII
ABLATION ON SAMPLING TIMESTEPS. MS-COCO DATASET

IS Inversion Timesteps	IS Generation Timesteps				
	10	20	30	40	50
10	95.80	97.64	98.17	98.38	98.52
20	96.27	98.05	98.53	98.71	98.84
30	96.38	98.18	98.63	98.80	98.92
40	96.41	98.77	98.67	98.83	98.95
50	96.42	98.80	98.69	98.86	98.97

ablation experiments with various ODE solvers and assess the extraction accuracy against different attack scenarios on the MS-COCO dataset. We also evaluate the visual quality of the stego images produced by different samplers. All experimental settings, except for the ODE Solver, are held constant. The experimental results, detailed in Table VII, compare three samplers: first-order DDIM [37], second-order PNDM [48], and second-order DPM-Solver++ [26]. The results indicate that our method is compatible with multiple samplers, consistently achieving high extraction accuracy.

Higher-order samplers, such as PNDM and DPM-Solver++, enhance numerical precision and demonstrate robust performance across all scenarios. While PNDM marginally surpasses DPM-Solver++ in lossless environments, DPM-Solver++ offers superior numerical stability under lossy conditions like JPEG compression and Median Blur. This enhanced stability can likely be attributed to the adoption of the data prediction model by DPM-Solver++, which more effectively manages data distortions compared to the noise prediction model used by other samplers. Moreover, DPM-Solver++ exhibits superior visual quality in its generated stego images. Therefore, we have selected DPM-Solver++ as our default ODE solver.

As presented in Table VIII, we investigate the impact of varying *steps* used by the default ODE Solver on extraction accuracy. Specifically, within the MS-COCO dataset, we evaluate PNG storage conditions using 10, 20, 30, 40, and 50 timesteps for the IS Generation and IS Inversion processes, respectively. The results demonstrate that different *steps* between the IS Generation and IS Inversion stages do not compromise accuracy. Instead, the extraction accuracy generally increases as the number of *steps* grows, due to the more stable mapping trajectories achieved with larger iteration timesteps. However, higher *steps* increase computational costs. We set $steps = 20$ as the default to balance performance and efficiency.

VI. CONCLUSION

In conclusion, we introduce a novel approach to generative steganography, addressing key challenges for leveraging the capabilities of diffusion models. Our proposed framework offers a robust solution to the practical limitations observed in previous mapping-based generative steganography schemes. By carefully designing a mapping module capable of accurately recovering secret messages, even in the presence of errors, we have achieved superior performance in terms of extraction accuracy, robustness, security, and image visual quality. Overall, our contributions include the development of a new generative steganography framework, the design of a unique mapping module, and comprehensive experimental validation of our method's performance.

REFERENCES

[1] X. Zhou, W. Peng, B. Yang, J. Wen, Y. Xue, and P. Zhong, "Linguistic steganography based on adaptive probability distribution," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 5, pp. 2982–2997, Sep. 2022.

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

[3] K. Gopalan, "Audio steganography using bit modification," in *Proc. Int. Conf. Multimedia Expo (ICME)*, vol. 1, Jul. 2003, pp. I–629.

[4] L. Meng, X. Jiang, T. Sun, Z. Zhao, and Q. Xu, "A robust coverless video steganography based on the similarity of inter-frames," *IEEE Trans. Multimedia*, vol. 26, pp. 5996–6011, 2024.

[5] 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.

[6] 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.

[7] T. Pevnỳ, T. Filler, and P. Bas, "Using high-dimensional image models to perform highly undetectable steganography," in *Proc. 12th Int. Conf. Inf. Hiding (IH)*, Calgary, AB, Canada. Springer, 2010, pp. 161–177.

[8] V. Holub, J. Fridrich, and T. Denemark, "Universal distortion function for steganography in an arbitrary domain," *EURASIP J. Inf. Secur.*, vol. 2014, pp. 1–13, Jan. 2014.

[9] V. Sedighi, R. Cogranne, and J. Fridrich, "Content-adaptive steganography by minimizing statistical detectability," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 2, pp. 221–234, Feb. 2016.

[10] M. Sharifzadeh, M. Aloraini, and D. Schonfeld, "Adaptive batch size image merging steganography and quantized Gaussian image steganography," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 867–879, 2020.

[11] J. Zhu, R. Kaplan, J. Johnson, and L. Fei-Fei, "Hidden: Hiding data with deep networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 657–672.

[12] M. Boroumand, M. Chen, and J. Fridrich, "Deep residual network for steganalysis of digital images," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 5, pp. 1181–1193, May 2019.

[13] 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.

[14] J. Ye, J. Ni, and Y. Yi, "Deep learning hierarchical representations for image steganalysis," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 11, pp. 2545–2557, Nov. 2017.

[15] W. You, H. Zhang, and X. Zhao, "A Siamese CNN for image steganalysis," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 291–306, 2021.

[16] 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.

[17] 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.

[18] 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.

[19] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1×1 convolutions," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1–10.

[20] 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. Dependable Secure Comput.*, vol. 19, no. 5, pp. 3343–3356, Sep. 2022.

[21] K. Yang, K. Chen, W. Zhang, and N. Yu, "Provably secure generative steganography based on autoregressive model," in *Digital Forensics and Watermarking*. Cham, Switzerland: Springer, 2019, pp. 55–68.

[22] Z. Yang, K. Chen, K. Zeng, W. Zhang, and N. Yu, "Provably secure robust image steganography," *IEEE Trans. Multimedia*, vol. 26, pp. 5040–5053, 2023.

[23] Z. You, Q. Ying, S. Li, Z. Qian, and X. Zhang, "Image generation network for covert transmission in online social network," in *Proc. 30th ACM Int. Conf. Multimedia*, Oct. 2022, pp. 2834–2842.

[24] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 6840–6851.

[25] 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.

[26] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. Zhu, "DPM-Solver++: Fast solver for guided sampling of diffusion probabilistic models," 2022, *arXiv:2211.01095*.

[27] Y. Kim, Z. Duric, and D. Richards, "Modified matrix encoding technique for minimal distortion steganography," in *Proc. 8th Int. Workshop Inf. Hiding (IH)*, Alexandria, VA, USA. Springer, 2007, pp. 314–327.

[28] J. Fridrich, M. Goljan, P. Lisonek, and D. Soukal, "Writing on wet paper," *IEEE Trans. Signal Process.*, vol. 53, no. 10, pp. 3923–3935, Oct. 2005.

[29] K. Alex Zhang, A. Cuesta-Infante, L. Xu, and K. Veeramachaneni, "SteganoGAN: High capacity image steganography with GANs," 2019, *arXiv:1901.03892*.

[30] Z. Zhou, H. Sun, R. Harit, X. Chen, and X. Sun, "Coverless image steganography without embedding," in *Proc. 1st Int. Conf. Cloud Comput. Secur. (ICCCS)*, Nanjing, China. Springer, 2015, pp. 123–132.

[31] 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, Aug. 2020.

[32] 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.

[33] P. Wei, Q. Zhou, Z. Wang, Z. Qian, X. Zhang, and S. Li, "Generative steganography diffusion," 2023, *arXiv:2305.03472*.

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

[35] I. Goodfellow et al., "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 1–9.

[36] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*.

[37] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *Proc. 9th Int. Conf. Learn. Represent. (ICLR)*, Vienna, Austria, May 2021, pp. 1–22. [Online]. Available: https://openreview.net/forum?id=St1giarCHLP

[38] C. Lu, Y. Zhou, F. Bao, J. 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.*, vol. 35, 2022, pp. 5775–5787.

[39] Y. Song, P. Dhariwal, M. Chen, and I. Sutskever, "Consistency models," in *Proc. 40th Int. Conf. Mach. Learn.*, 2023, pp. 1335–1376.

[40] P. Dhariwal and A. Nichol, "Diffusion models beat GANs on image synthesis," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 8780–8794.

[41] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 8748–8763.

[42] T.-Y. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. 13th Eur. Conf. Comput. Vis. (ECCV)*. Zurich, Switzerland: Springer, 2014, pp. 740–755.

[43] A. Bansal et al., "Cold diffusion: Inverting arbitrary image transforms without noise," 2022, *arXiv:2208.09392*.

[44] D. Kingma, T. Salimans, B. Poole, and J. Ho, "Variational diffusion models," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 21696–21707.

[45] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," in *Proc. 9th Int. Conf. Learn. Represent. (ICLR)*, Vienna, Austria, May 2021, pp. 1–36. [Online]. Available: https://openreview.net/forum?id=PxTIG12RRHS

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

[47] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real NVP," in *Proc. 5th Int. Conf. Learn. Represent. (ICLR)*, Toulon, France, 2017, pp. 1–32. [Online]. Available: https://openreview.net/forum?id=HkpbnH9lx

[48] L. Liu, Y. Ren, Z. Lin, and Z. Zhao, "Pseudo numerical methods for diffusion models on manifolds," in *Proc. Int. Conf. Learn. Represent.*, 2022.

[49] R. Mokady, A. Hertz, K. Aberman, Y. Pritch, and D. Cohen-Or, "Null-text inversion for editing real images using guided diffusion models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 6038–6047.

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

[51] A. van den Oord, O. Vinyals, and K. kavukcuoglu, "Neural discrete representation learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–10.

[52] 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. (MICCAI)*, Munich, Germany: Springer, 2015, pp. 234–241.

[53] J. Ho and T. Salimans, "Classifier-free diffusion guidance," 2022, *arXiv:2207.12598*.

[54] C. Schuhmann et al., "LAION-5B: An open large-scale dataset for training next generation image-text models," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 25278–25294.

[55] OpenAI. (2022). *ChatGPT (GPT-3.5)*. [Online]. Available: https://chat.openai.com/

[56] M. Hodosh, P. Young, and J. Hockenmaier, "Framing image description as a ranking task: Data, models and evaluation metrics," *J. Artif. Intell. Res.*, vol. 47, pp. 853–899, Aug. 2013.

[57] 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.*, vol. 30, 2017, pp. 1-12.

[58] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.

[59] A. Mittal, A. K. Moorthy, and A. C. Bovik, "Blind/referenceless image spatial quality evaluator," in *Proc. 45th Asilomar Conf. Signals, Syst. Comput. (ASILOMAR)*, Nov. 2011, pp. 723–727.

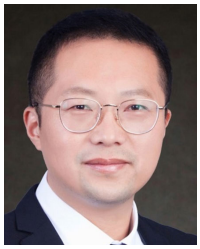
Xiaoxiao Hu received the B.S. degree from the School of Computer Science, Fudan University, China, in 2021, where she is currently pursuing the Ph.D. degree. Her research interests include multimedia forensics and data hiding.

Sheng Li (Member, IEEE) received the Ph.D. degree from the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, in 2013. From 2013 to 2016, he was a Research Fellow with the Rapid Rich Object Search Laboratory, Nanyang Technological University. He is currently with the Faculty of the School of Computer Science, Fudan University, China, where he is an Associate Professor. His research interests include biometric template protection, pattern recognition, multimedia forensics, and security. He was a recipient of the IEEE WIFS Best Student Paper Silver Award.

Qichao Ying received the Ph.D. degree from the School of Computer Science, Fudan University, China, in 2024. He is currently a Senior Engineer with Nvidia Corporation, Shanghai, China. His research interests include multimedia forensics, data hiding, and fake news detection.

Wanli Peng received the Ph.D. degree from the College of Information Electrical and Engineering, China Agricultural University, Beijing, China, in 2022. From 2022 to 2024, he was a Post-Doctoral Fellow with the College of Computer Science, Fudan University, Shanghai, China. He is currently an Associate Professor with the College of Information Electrical and Engineering, China Agricultural University, Beijing. His research interests include information hiding and artificial intelligence security.

Xinpeng Zhang (Senior Member, IEEE) received the B.S. degree in computational mathematics from Jilin University, China, in1995, and the M.E. and Ph.D. degrees in communication and information systems from Shanghai University, China, in 2001 and 2004, respectively. Since 2004, he has been with the Faculty of the School of Communication and Information Engineering, Shanghai University, where he is currently a Professor. He is also with the Faculty of the School of Computer Science, Fudan University. He was a Visiting Scholar with the State University of New York at Binghamton, from 2010 to 2011, and also an experienced Researcher with Konstanz University, sponsored by the Alexander von Humboldt Foundation, from 2011 to 2012. His research interests include multimedia security, image processing, and digital forensics. He has published over 200 papers in these areas. He is an Associate Editor of IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY.



Zhenxing Qian (Senior Member, IEEE) received the B.S. and Ph.D. degrees from the University of Science and Technology of China (USTC), in 2003 and 2008, respectively. He is a Professor with the School of Computer Science, Fudan University, where he is also the Vice Dean with the Key Laboratory of China Culture and Tourism Ministry. He has published over 200 peer-reviewed papers, many of them were published in IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON CLOUD COMPUTING, CVPR, ICCV, AAAI, IJCAI, NerurIPS, and ACM MM. His research interests include information hiding, multimedia security, and neural network security. He is an Associate Editor of IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, *Signal Processing*, and *Journal of Visual Communication and Image Representation*.