Information Sciences 718 (2025) 122358

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Diffusion-Stego: Training-free diffusion generative steganography via message projection

Daegyu Kim [a], Chaehun Shin [a], Jooyoung Choi [a], Dahuin Jung [b,*], Sungroh Yoon [a,c,d,**]

[a] Data Science and AI Laboratory, Electrical and Computer Engineering, Seoul National University, Seoul 08826, Republic of Korea
[b] School of Computer Science and Engineering, Soongsil University, Seoul 06978, Republic of Korea
[c] Interdisciplinary Program in Artificial Intelligence, Seoul National University, Seoul 08826, Republic of Korea
[d] AIIS, ASRI, INMC, and ISRC, Seoul National University, Seoul 08826, Republic of Korea

ARTICLE INFO

Keywords:
Generative steganography
Generative models
Diffusion models

ABSTRACT

Generative steganography is the process of hiding secret messages in generated images instead of cover images. Existing studies on generative steganography use GAN or Flow models to obtain high hiding message capacity and anti-detection ability over cover images. However, they create relatively unrealistic stego images because of the inherent limitations of generative models. We propose Diffusion-Stego, a generative steganography approach based on diffusion models that outperform other generative models in image generation. Diffusion-Stego projects secret messages into the latent noise of diffusion models and generates stego images with an iterative denoising process. Since the naive hiding of secret messages into noise boosts visual degradation and decreases extracted message accuracy, we introduce message projection, which hides messages into noise space while addressing these issues. We suggest three options for message projection to adjust the trade-off between extracted message accuracy, anti-detection ability, and image quality. Diffusion-Stego is a training-free approach, so we can apply it to pre-trained diffusion models that generate high-quality images, or even large-scale text-to-image models, such as Stable diffusion. Diffusion-Stego achieved a high capacity of messages as well as high quality that makes it challenging to distinguish from real images in the PNG format.

1. Introduction

Image steganography [1] is the process that aims to hide secret messages in images so that secret messages are not detected or exposed by third-party players. Traditional image steganography methods [2] conceal secret messages within natural cover images. The sender transmits the cover image containing the secret messages, termed a stego image, to the receiver, who extracts the hidden messages from the stego image. In contrast, the third-party players attempt to discriminate the stego images by training steganalyzer models [3], which classify over the cover images and stego images.

* Corresponding author at: School of Computer Science and Engineering, Soongsil University, Seoul 06978, Republic of Korea.
** Corresponding author at: Data Science and AI Laboratory, Electrical and Computer Engineering, Seoul National University, Seoul 08826, Republic of Korea.
E-mail addresses: dahuin.jung@ssu.ac.kr (D. Jung), sryoon@snu.ac.kr (S. Yoon).

https://doi.org/10.1016/j.ins.2025.122358
Received 9 April 2024; Received in revised form 2 May 2025; Accepted 24 May 2025

Available online 29 May 2025

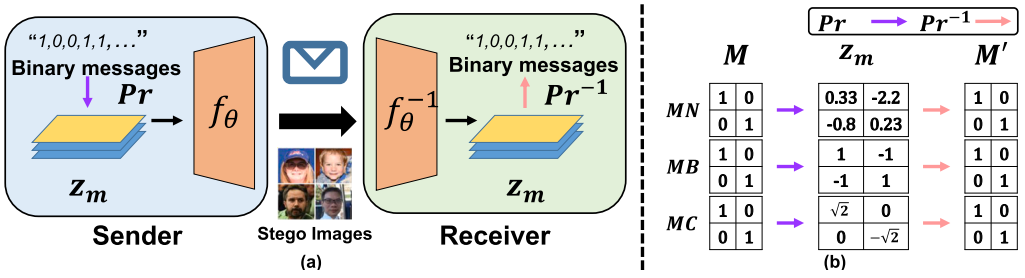

Fig. 1. The overview of Diffusion-Stego. (a) Generative steganography process of Diffusion-Stego. (b) The example of three message projection processes when messages are '1001'.

Generative steganography methods [4] have been proposed to deceive steganalyzer models. These approaches apply deep generative models that synthesize stego images from secret messages without using cover images. This makes them less vulnerable to steganalyzer models, as there are no cover images on which to train on the steganalyzer. Recent generative steganography studies [5] using Generative Adversarial Networks (GAN) [6] or Flow [7] models as a generator have been proposed. In contrast to their anti-detection ability and high hiding capacity, they relatively lack image fidelity due to the limitations of the generative models they use.

Therefore, we explore utilizing diffusion-based generative models for steganography. Diffusion models [8] are recently popular generative models that generate high-quality images [9] with an iterative sampling process. Recent studies [10] have utilized diffusion models and deterministic samplers [11] to generate high quality images.

We identified two primary challenges in employing diffusion models for steganography. First, diffusion models cannot generate images from noise replaced with binary messages, which is the naive approach to hiding messages in the noise. Second, the accumulation of errors during the reverse process of the deterministic sampler leads to a drop in the extracted message accuracy.

In this paper, we propose Diffusion-Stego, a powerful generative steganography approach that addresses these challenges through a novel technique called *message projection*. Message projection modifies noise to the extent that it does not deviate from the distribution of random noise while preserving the quality of stego images and ensuring high accuracy in message extraction.

As illustrated in Fig. 1-(a), Diffusion-Stego hides secret messages in noise using message projection, and generates stego images using the deterministic sampler without re-training the diffusion models. Due to the invertible property of the deterministic sampler, Diffusion-Stego can generate and extract messages using a single diffusion model, eliminating the need to share additional random seeds [12]. This approach allows the sender and receiver to share only a diffusion model and a hiding method. We provide three types of message projection, which can be tailored to suit user preferences.

Diffusion-Stego does not require fine-tuning of pre-trained models or training additional models such as extractors or decoders. By using well-trained models, Diffusion-Stego generates high-quality stego images, Frechet inception distance (FID) score [13] of 2.77 on FFHQ 64×64 [14] images while hiding 1.0 bits per pixel (bpp) messages. Additionally, using pre-trained diffusion models trained on AFHQv2 64×64 [15], Diffusion-Stego achieves hiding 6.0 bpp messages with high extracted message accuracy. Furthermore, we show that Diffusion-Stego can be easily applied to text-to-image models, such as Stable diffusion [9], by requiring only secret messages and text prompts.

2. Related work

2.1. Steganography with deep learning

Steganography methods employing deep learning techniques have emerged, encompassing various machine learning tasks. Baluja [16], Zhang et al. [17] and Zhu et al. [18] utilize encoder-decoder architectures to conceal secrets within provided cover images using trained models. Other methods [19] employ optimization techniques to manipulate the cover images so that the secret message becomes visible when decoding the altered cover images.

2.2. Generative steganography

Generative steganography is a method where generative models synthesize images from secret messages without using any cover images. In generative steganography, two players, the sender and the receiver, communicate secret messages through generated stego images. Unlike traditional steganography methods [1], the sender in generative steganography uses a generator to generate the stego image from messages without a cover image. The receiver extracts secret messages from the stego image using an extractor.

Generative steganography offers several advantages over traditional steganography methods that use cover images. One of the main benefits is that it can avoid detection by steganalysis methods because it does not alter existing images, which prevents steganalysis models from effectively learning identifiable patterns. Further, steganalysis methods that are trained on existing cover images cannot detect the presence of hidden data.

In generative steganography, both the image quality and the extracted message accuracy are significant. Image quality indicates how stego images are photorealistic like real images. The stego images should be visually and statistically similar to the real images to deceive the steganalyzer. Additionally, the generator should produce stego images so that the receiver can extract secret messages.

Early studies of generative steganography hide messages in simple images, such as texture or fingerprint images. Wu and Wang [4] and Xu et al. [20] proposed approaches to hide secret messages in texture messages. Li and Zhang [21] proposed the method using fingerprint images. These methods embed secrets into the irregular representations of textures or fingerprint patterns. However, these approaches generate low-quality and unnatural images, which are prone to being detected by third-party players.

Steganography approaches using generative models have been proposed to make high-quality and natural stego images. Especially, GAN models [6] have been used for generative steganography. Liu et al. [22] and Zhang et al. [23] hide messages in label embedding of conditional GANs [24]. The receiver retrieves the hidden messages by classifying the categories of the stego image using a classifier. Hu et al. [25], Yu et al. [26] and Wei et al. [5] introduce new extractor models for steganography. These methods jointly train the generative model and the extractor, allowing the receiver to use the extractor to decode secret messages from the stego images. Zhou et al. [27] and Wei et al. [28] proposed an approach to use invertible Flow [7] to enable high capacity of hidden messages. The invertibility of Flow enables the use of the inverted model as the extractor to retrieve embedded messages.

Additionally, there are unique generative steganography methods that conceal messages within segmentation maps [29], text [30], or even 3D rendering images [31].

2.3. Diffusion models

Diffusion models [8] generate images through an iterative denoising process from Gaussian noise. The sampling process can be viewed as solving the ODE process, with time t going from T to 0, using deterministic samplers [11]. This process is termed probability flow ODE [32] and follows below equation:

$$\mathrm{d}\mathbf{x} = [\mathbf{f}(\mathbf{x}, t) - \frac{1}{2} g(t)^2 \mathbf{s}_\theta(\mathbf{x}, t)]\mathrm{d}t, \tag{1}$$

where $\mathbf{f}$ is drift coefficient, g is diffusion coefficient and $\mathbf{s}_\theta$ is score function trained as neural network. The score function estimates $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$, where $p_t(\mathbf{x})$ is the probability distribution of $\mathbf{x}(t)$.

Through Equation (1), diffusion models generate the image $\mathbf{x}(0)$ from Gaussian noise $\mathbf{x}(T) = \sigma_T \mathbf{z}$, where σ_T is constant of T and $\mathbf{z}$ is standard Gaussian noise, $\mathbf{z} \sim \mathcal{N}(0, I)$. We can establish a bijective function between $\mathbf{z}$ and image $\mathbf{x}(0)$ using ODE, namely, $\mathbf{x}(0) = f_\theta(\mathbf{z})$. The function f_θ is invertible, we can express another equation $\mathbf{z} = f_\theta^{-1}(\mathbf{x}(0))$.

In the field of generative steganography, several studies have proposed the use of diffusion models to leverage their generation ability. GSD [33] trained diffusion models to hide and extract messages using discrete cosine transforms (DCT) [34]. CroSS [35] conceals secret images by utilizing text-to-image diffusion models [9], employing image-to-image translation based on different keywords. In our method Diffusion-Stego, we utilize diffusion models as both a generator and an extractor of generative steganography. Jois et al. [36] introduced approaches to concealing secrets by sharing initialized random seeds and analyzing differences between synthesized images and received stego images.

3. Method

3.1. Diffusion-Stego

Settings. We propose Diffusion-Stego, a new generative steganography framework leveraging pre-trained diffusion models. Diffusion-Stego can hide n bits per pixel (bpp) binary messages, $\mathbf{M} \in \{0, 1\}^{n \times W \times H}$ in images $\mathbf{X} \in \mathbb{Z}^{3 \times W \times H}$, where W and H denote the width and height of images.

In Diffusion-Stego, we consider two players, the sender and the receiver. The sender projects the secret binary messages $\mathbf{M}$ into Gaussian noise and generates stego images $\mathbf{X}_S$ from the projected noise using a diffusion model. The receiver then extracts the hidden noise using the same diffusion model and projects it onto the extracted binary message $\mathbf{M}'$. We note that the sender and receiver should share the same diffusion model and the projection process to use Diffusion-Stego. There are no restrictions on the shared diffusion models, so two players can use any pre-trained diffusion models. The process is defined as follows:

$$G(\mathbf{M}) = \mathbf{X}_S, \quad G^{-1}(\mathbf{X_S}) = \mathbf{M}'. \tag{2}$$

Wei et al. [28] demonstrated that saving images as float type using TIFF format resulted in higher performance in extracted message accuracy than saving as integer type using PNG or JPEG formats. In our research, we mainly use PNG format to save images because integer type formats are more widely used than TIFF.

Procedure. Generative steganography requires two models: G, which generates images from messages, and E, which extracts messages from images. Previous works have trained two separate models for G and E, similar to the f_θ and f_θ^{-1} described in Section 2.3. However, both models can be performed by a single diffusion model. Thus, we can use diffusion models as generative steganography, provided that message projection projects $\mathbf{M}$ into the same domain as Gaussian noise $\mathbf{z}$. We use deterministic samplers such as DDIM sampler [11] or Heun's sampler of EDM [10] for the invertible function f_θ.

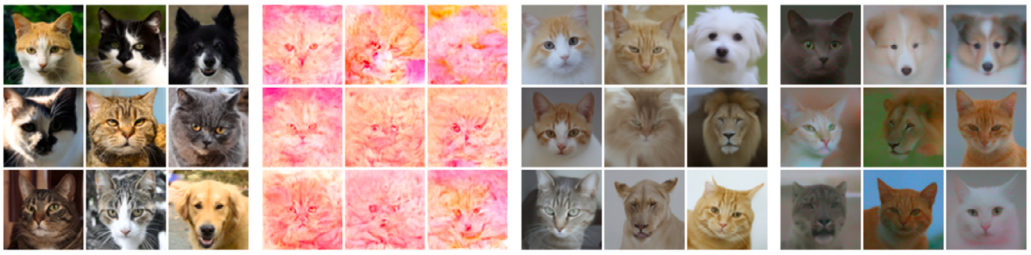

Fig. 2. Images generated by diffusion models. (a) Normally generated images. (b) Collapsed images hiding 1.0 bpp, mean of $\mathbf{z}_m$ differs from $\mathbf{z}$. (c) Collapsed images hiding 1.0 bpp, variance of $\mathbf{z}_m$ differs from $\mathbf{z}$. (d) Collapsed images hiding 1.0 bpp using se-S2IRT [27] algorithm (each value of $\mathbf{z}_m$ is not independent).

In Diffusion-Stego, we edit the Gaussian noise $\mathbf{z}$ with message noise $\mathbf{z}_m$, where $\mathbf{z}_m$ is the noise hiding $\mathbf{M}$. The number of channels of $\mathbf{z}_m$ to hide messages depends on the quantity of message. When the length of the messages is n bpp, we use n channels of $\mathbf{z}_m$ while retaining the noise in the unused channels. Message projection Pr is a function that maps $\mathbf{z}$ and $\mathbf{M}$ into $\mathbf{z}_m$, $\mathbf{z}_m = Pr(\mathbf{z}, \mathbf{M})$. If Pr is invertible, we can generate a stego image $\mathbf{X}_S$ and extract hidden messages $\mathbf{M}'$ by solving the ODE process, $\mathbf{x}_s(0) = f_\theta(Pr(\mathbf{z}, \mathbf{M}))$ and $\mathbf{M}' = Pr^{-1}(f_\theta^{-1}(\mathbf{x}_s(0)))$ while considering $\mathbf{X}_S$ as $\mathbf{x}_s(0)$, as shown in Fig. 1-(a).

3.2. Challenging problems of Diffusion-Stego

This section introduces two challenging problems when using diffusion models for generative steganography.

Extraction error. In Diffusion-Stego, we utilize the invertible property of the deterministic sampler of diffusion models. As the deterministic sampler solves the ODE process with a numerical integrator, errors accumulate in both the forward and backward processes of the ODE. This can lead to a decrease in the extracted message accuracy, which we term as *extraction error*. Additionally, in steganography, the sender must save images in integer format PNG to send to the receiver. This discretization during image saving distorts the initial value of the ODE, further amplifying the error.

Image collapse. Diffusion models have learned to map Gaussian noise to images. If the Pr is defined simplistically, the distribution of $\mathbf{z}_m$ can differ from that of the Gaussian noise. In this case, image quality may be harmed or even collapsed, as shown in Fig. 2. We refer to this issue as *image collapse*.

To prevent the image collapse, we need to make the $\mathbf{z}_m$ distribution similar to that of the Gaussian noise. We identify three conditions which $\mathbf{z}_m$ need to satisfy: **(1)** The **mean** of $\mathbf{z}_m$ is close to 0: If the mean is higher than 0, the output image becomes white, while if it is lower than 0, it becomes dark. **(2)** The **variance** of $\mathbf{z}_m$ is similar to 1: When the variance is too high or too low, the output image may not be properly denoised or may be oversimplified. **(3)** The values of $\mathbf{z}_m$ are **independent**: When the values are not independent, diffusion models do not work normally. Fig. 2 illustrates the results when $\mathbf{z}_m$ does not meet these conditions.

A distribution shift from Pr results in a decline in image quality, subsequently diminishing the anti-detection capability of stego images. However, employing a naive distribution of $\mathbf{z}_m$ remains susceptible to extraction error. Our identified conditions represent a tipping point at which a rapid decline in image quality leads to the image collapse problem. We achieve an optimal trade-off between secret message accuracy and image quality by minimizing extraction error before reaching this tipping point.

3.3. Message projection

In this section, we describe message projection, the key ingredient for solving both problems, the image collapse and the extraction error. We suggest three options for message projection, which are designed depending on which problem to focus on. Examples of our message projections are shown in Fig. 1-(b).

Message to noise (MN). We propose MN projection Pr_N to solve the image collapse. The projection Pr_N maps the messages to $\mathbf{z} \sim \mathcal{N}(0, I)$, so that the distribution of message noise $\mathbf{z}_m$ is equivalent to that of Gaussian noise.

To implement the MN projection, we first sample standard Gaussian noise $\mathbf{z}$. Then, we project $\mathbf{z}$ into $\mathbf{z}_m$ using the following rule: change the sign of $\mathbf{z}$ to a positive number where $\mathbf{M}$ is 1, and to a negative number where $\mathbf{M}$ is 0. The equation for the MN method is given as follows: $\mathbf{z}_{mn} = (-1)^{\mathbf{M}+1} \cdot \text{abs}(\mathbf{z})$. The receiver can extract the messages by applying the inverse projection Pr_N^{-1}, which checks whether the value of extracted $\mathbf{z}_m$ is greater than 0 or not.

Since the probability distribution of $\mathbf{z}_m$ is the same as that of Gaussian noise, the generated images $f_\theta(Pr_N(\mathbf{z}, \mathbf{M}))$ are challenging to distinguish them from normally generated images. However, using the MN projection is vulnerable to the extraction error because some values of $\mathbf{z}_m$ are situated near the decision boundary of Pr_N^{-1}, which is 0. Therefore, we suggest other projection options to improve the extracted message accuracy.

Message to binary (MB). The MB projection is designed to solve the aforementioned problem, the extraction error. In the MB projection, the receiver identifies the messages by inverse projection Pr_B^{-1}, the same as that of the MN projection. Pr_N^{-1}. The MB projection Pr_B maps the values of $\mathbf{z}_m$ as far as possible from 0, which corresponds to the decision boundary of Pr_B^{-1}.

To maximize the minimum distance from the boundary, Pr_B equals all the values that denote the same messages. To ensure that the variance of $\mathbf{z}_m$ becomes 1, Pr_B set the value of $\mathbf{z}_m$ to 1 where $\mathbf{M}$ is 1 and to -1 where $\mathbf{M}$ is 0. The equation for the MB method is as follows: $\mathbf{z}_{mb} = (-1)^{\mathbf{M}+1} \cdot \mathbf{1}$.

Message to centered binary (MC). While the MN projection resolves the image collapse, it performs worse in terms of extracted message accuracy than the MB projection. The MB projection well addresses the extraction error. However, the distribution of $\mathbf{z}_m$ deviates from that of Gaussian noise, which causes a slight degradation in image quality. Therefore, we suggest a compromise between the two projections, called the MC projection.

Similarly to the MB projection, the MC projection Pr_C projects the values that denote the same messages into coherent values. Notably, the probability distribution of the message is set as $\mathbf{M}$, $p(\mathbf{M}=1) = p(\mathbf{M}=0) = 1/2$. Since the mode of the Gaussian noise is 0, we set the value of $\mathbf{z}_m$ to 0, where $\mathbf{M}$ is 0. When $\mathbf{M}$ is 1, we randomly map the value of $\mathbf{z}_m$ to either $\sqrt{2}$ or $-\sqrt{2}$, so that the mean and variance of $\mathbf{z}_m$,

$$\begin{aligned}\mu_{\mathbf{z}_m} &= p(\mathbf{M}=0)\cdot 0 + p(\mathbf{M}=1)\cdot(\sqrt{2}/2 + \sqrt{-2}/2)\\ &= (\sqrt{2}/4) + (-\sqrt{2}/4)\\ &= 0,\end{aligned} \tag{3}$$

$$\begin{aligned}\sigma^2{}_{\mathbf{z}_m} &= p(\mathbf{M}=1)\cdot((\sqrt{2}-\mu_{\mathbf{z}_m})^2/2 + (-\sqrt{2}-\mu_{\mathbf{z}_m})^2/2)\\ &= (\sqrt{2})^2/4 + (\sqrt{-2})^2/4\\ &= 1\end{aligned} \tag{4}$$

are equal to those of Gaussian noise. Inverse projection Pr_C^{-1} checks the values of $\mathbf{z}_m$ are close to $\sqrt{2}$, $-\sqrt{2}$, or 0. The equation for the MC method is given by: $\mathbf{z}_{mc} = \sqrt{2}\cdot \mathrm{sign}(\mathbf{z}) \cdot \mathbf{M}$.

The MC projection performs higher extracted message accuracy than the MN projection and better sample quality than the MB projection, which will be demonstrated in Section 4.

3.4. Trick of hiding large messages

Generally, input noise for diffusion models consists of 3 channels. When applying the projections referred to in Section 3.3, the maximum capacity of secret messages is 3.0 bpp. In order to conceal more messages than 3.0 bpp, we should hide more than 1.0 bpp messages in a single channel.

We hide multiple bits following MB, which we call multi-bit projection. Two-bit messages consist of four cases: 00, 01, 10, and 11. We set four values and keep them as far away from each other as possible while maintaining the mean and variance of $\mathbf{z}_m$. We set the distance between each value as d. To ensure that the mean of $\mathbf{z}_m$, $\mu_{\mathbf{z}_m}$ remains 0, each value is assigned as $-\frac{3}{2}d, -\frac{1}{2}d, \frac{1}{2}d$, and $\frac{3}{2}d$. The variance of $\mathbf{z}_m$, $V_{\mathbf{z}_m} = 5d^2$, leading to $d = 1/\sqrt{5}$. So we define the value as $-3/\sqrt{5}, -1/\sqrt{5}, 3/\sqrt{5}$, or $1\sqrt{5}$, where $\mathbf{M}$ is 00, 01, 10, or 11. We can hide 6.0 bpp messages by hiding 2 bits in each channel and more messages by applying this projection. The algorithm of hiding large bpp trick can be defined as follows:

$$\mathbf{z}_{\mathrm{hl}}(i) = \begin{cases} -\frac{3}{\sqrt{5}} & \text{if } \mathbf{M}(2i, 2i+1) = 00,\\ -\frac{1}{\sqrt{5}} & \text{if } \mathbf{M}(2i, 2i+1) = 01,\\ +\frac{3}{\sqrt{5}} & \text{if } \mathbf{M}(2i, 2i+1) = 11,\\ +\frac{1}{\sqrt{5}} & \text{if } \mathbf{M}(2i, 2i+1) = 10.\end{cases} \tag{5}$$

When embedding a higher number of bits, we select 2^n distinct values and set the spacing between consecutive values to d, following the same approach as in the 2-bit case.

4. Experiments

Datasets and pre-trained models. We consider three commonly used image datasets for generative models: CIFAR-10 [37], FFHQ 64×64 [14] and AFHQv2 64×64 [15]. Several previous works provide pre-trained models trained on these datasets. In our experiments, we used pre-trained models and deterministic Heun's sampler of EDM [10]. For the FFHQ 64×64 and AFHQv2 64×64 datasets, we processed f_θ and f_θ^{-1} with 40 inference steps, while for the CIFAR-10 dataset, we used 18 inference steps. We conducted our experiments using 4 Nvidia Titan Xp GPUs.

Metrics. We evaluated the extracted message accuracy, anti-detection ability, and image quality of our methods.

The **accuracy (Acc)** measures the accuracy of the extracted message, which may be distorted through the steganography process f_θ and f_θ^{-1}. We calculate Acc as follows: $\text{Acc} = 1 - (\mathbf{M} \oplus \mathbf{M}')/\text{len}(\mathbf{M})$, where $\mathbf{M}$ is the original binary messages, $\mathbf{M}'$ is the extracted binary messages through the steganography process, $\oplus$ is XOR operator, and $\text{len}(\mathbf{M})$ is length of the messages.

The **detection error (Pe)** and the **detection rate (DR)** are indicators of the performance of classifier models. First, Pe is defined as follows: $\text{Pe} = \min_{P_{FA}} \frac{1}{2}(P_{FA} + P_{MD})$, where P_{FA} and P_{MD} are the false-alarm and miss-detection error rates. A Pe value of 0.5 means that the classifier cannot distinguish two classes completely.

Second, DR measures detection accuracy, representing the percentage of correctly classified cover and stego images. A DR of 100% indicates that the steganalyzer correctly identifies all images.

We use Xu-Net [3] models to evaluate Pe of steganalyzer models and SiaStegNet [26] models to appraise DR to assess the anti-detection ability of our method.

Frechet inception distance (FID) score [13] is a metric to assess image quality. FID measures the similarity between the feature distributions of the generated images and those of the training dataset utilizing a pre-trained image classifier model. A lower FID score indicates higher image quality, suggesting that the feature distribution of generated images closely resembles that of the training set. We evaluated the FID scores between generated stego images and images in the training dataset for each pre-trained generative model.

Bits per pixel (bpp) is a unit for the quantity of messages hidden in the images. We calculate bpp as follows: $\text{len}(\mathbf{M})/(W \times H)$, where W and H are width and height of the image.

In our experiments, we sampled 6,000 stego images from each model to calculate the accuracy. We divided the stego images into 5,000 training sets and 1,000 test sets to train and evaluate steganalyzer models. We trained Xu-Net and SiaStegNet on 5,000 stego images and 5,000 real images and test on 1,000 stego images and 1,000 real images for each steganography model, following [27]. We sampled 50,000 stego images that hide random messages to calculate FID scores.

Steganalyzer settings. As generative steganography models do not have cover images, third-party players cannot train their steganalyzer models. Therefore, we utilized real images as a substitute for cover images, assuming that third-party players would adopt a strict strategy in their steganalysis.

Baseline. We selected three baselines for generative steganography methods based on other generative models, GAN and Flow, which can hide messages above 1.0 bpp, Generative Steganography Network (GSN) [5], Secret to Image Reversible Transformation (S2IRT) [27], and Generative Steganographic Flow (GSF) [28].

GSN is a GAN-based [6] method that consists of four models: generator, discriminator, steganalyzer and extractor. In our experiments, we trained each GSN model from scratch with different payload settings.

S2IRT is a generative steganography method that applies Glow [7] models. We trained Glow and used Separate Encoder based S2IRT (SE-S2IRT) scheme. The SE-S2IRT scheme splits random values into K clusters and assigns each message to the corresponding cluster based on the order of the values. Increasing K leads to a higher message capacity but lower accuracy. In our experiments, we chose a low value of K to optimize the extracted message accuracy.

GSF is flow-based [7] generative steganography method. In the GSF process, the initial noise is extracted from the cover image and then refined to generate a high-quality image while embedding the secret message into the floating-point representation of the refined noise. We re-implemented GSF following the method described in its paper. In our re-implementation, we found that embedding secrets in the floating-point representation resulted in higher-quality image synthesis. Specifically, we utilized the 22nd, 21st and 20th bits to hide binary messages, achieving increased embedding capacities of 1.0, 2.0, and 3.0 bpp, respectively, in accordance with the original method. Embedding binary messages in the sign bit (1st bit) resulted in a decline in image quality.

We further compare our message projection methods with those of other diffusion-based generative steganography methods.

GSD [33] conceal secret messages within Gaussian noise using DCT, training their steganography models **StegoDiffusion** to enhance both accuracy and detection resistance in their synthesized images, which include both message-free and stego images. In our experimental setup, we utilized EDM models and Heun's sampler, consistent with the settings used in Diffusion-Stego, and employ the algorithm from GSD.

Chen et al. [38] propose distribution-preserving methods known as **message mapping**. In the approach, they conceal high bpp messages within the distribution of the Gaussian noise, akin to our MN method.

4.1. Comparison with other generative steganography

We compared Diffusion-Stego with two high-capacity generative steganography models, GSN, S2IR, and GSF, which are trained on the FFHQ 64×64 dataset. The comparison is performed using the MB projection in three payload settings: 1.0 bpp, 2.0 bpp, and 3.0 bpp.

The results are shown in Table 1 and the samples are provided in Fig. 3. The results presented in Table 1 show that Diffusion-Stego outperforms the other baseline models in terms of anti-detection ability and image quality. S2IRT shows higher accuracy than the other two models when the message capacity of stego images is 1.0 bpp. However, when the message capacity is higher than 1.0 bpp, Diffusion-Stego showed higher accuracy than S2IRT. Although S2IRT achieves high accuracy (99.94% at 1.0 bpp) when the number of the clusters K is 2, its message capacity is limited to 1.5 bpp. To hide 2.0 bpp messages, K should be at least 3, which decreases

Table 1
Comparison of extracted message accuracy (Acc, %), anti-detection ability (Pe, DR), and image quality (FID) with baseline methods on FFHQ 64×64 dataset.

Method	1.0 bpp				2.0 bpp				3.0 bpp			
	Acc ↑	Pe ↑	FID ↓	DR ↓	Acc ↑	Pe ↑	FID ↓	DR ↓	Acc ↑	Pe ↑	FID ↓	DR ↓
GSN† [5]	97.15	0.183	13.4	98.12	79.62	0.022	24.8	99.85	72.74	0.049	30.4	99.95
S2IRT† [27]	**99.94**	0.003	72.6	99.95	97.79	0.002	78.2	100.00	97.13	0.003	67.8	100.00
GSF† [28]	77.64	0.034	9.19	99.90	72.70	0.019	10.9	99.90	68.41	0.028	11.3	99.90
Diffusion-Stego (MB)	98.12	**0.427**	**2.77**	**71.97**	**98.19**	**0.361**	**3.30**	**89.34**	**98.76**	**0.310**	**4.30**	**98.91**

†: our re-implementation.

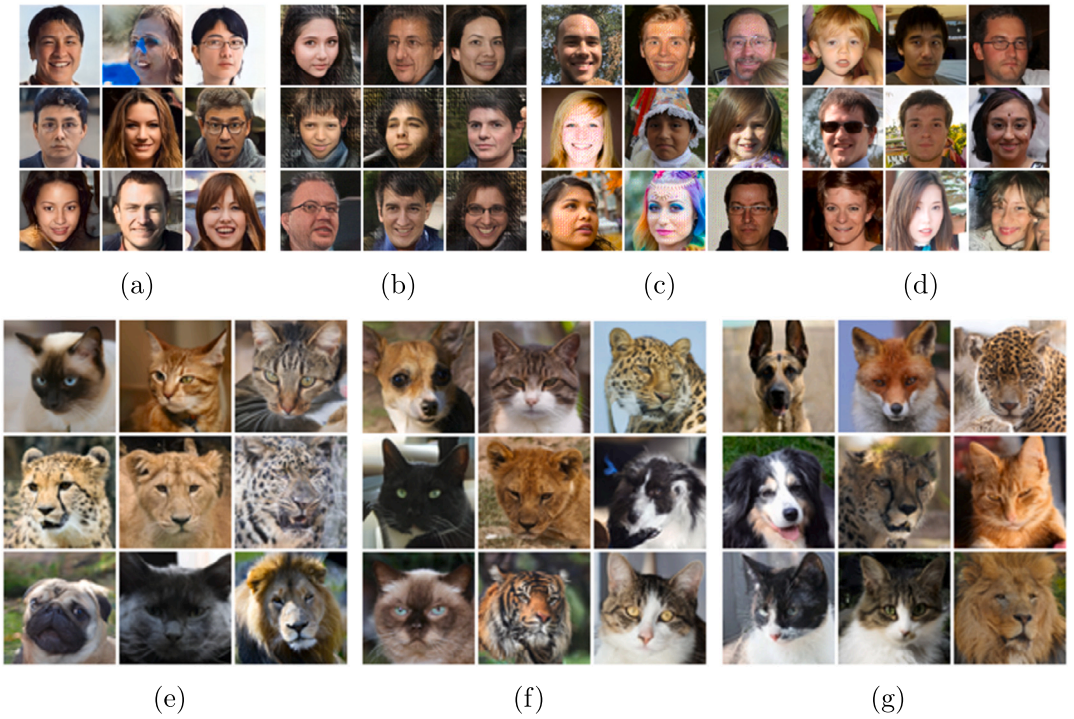

Fig. 3. FFHQ 64×64 and AFHQv2 64×64 stego images hiding 1.0 bpp messages. (a) GSN [5], (b) S2IRT [27], (c) GSF [28] (d) Diffusion-Stego with MB method. Diffusion-Stego is superior to the baseline methods in sample quality. (f) MN method, (g) MB method, (h) MC method. The AFHQv2 64×64 samples are synthesized using Diffusion-Stego.

the extracted message accuracy. GSN shows competitive accuracy in the payload setting of 1.0 bpp, but it decreases rapidly as the payload increases. GSF achieves higher image quality than other baselines, except ours, but exhibits lower accuracy.

4.2. Comparison with diffusion-based generative steganography method

We compared the performance of our message projection options and DCT algorithm from GSD [33] in our experiments using the FFHQ and AFHQv2 datasets. The results of our comparison are presented in Table 2 and the samples are shown in Fig. 3. Using the MB projection outperforms the other two projections in Acc. When hiding small messages with payloads of 1.0 bpp, the Pe, DR, and FID scores of each projection are similar.

However, when hiding large messages, such as with payloads of 3.0 bpp, the anti-detection ability and image quality of using the MB projection decreases rapidly compared to using the MN projection. In the FFHQ dataset, the MC projection shows compromised results of Acc, Pe, DR and FID between the MN projection and the MC projection as the payload of messages increases.

As shown in Table 2, the GSD algorithm [33] demonstrates high accuracy yet low image quality with payloads of 1.0 bpp. While image quality and detection of GSD remain stable, accuracy decreases as the amount of the message increases. In contrast, our message projection methods maintain or slightly enhance accuracy while reducing image quality and enhancing anti-detection capabilities. Various methods for projecting messages into random noise can be integrated into our proposed framework, Diffusion-Stego, allowing users to select different methods based on their specific settings.

Table 2
Ablation study results of three projection options and an algorithm suggested in Wei at al. [33] on FFHQ 64×64 and AFHQv2 64×64 datasets. The original EDM models have FID scores of 2.17 on AFHQv2 and 2.39 on FFHQ.

Datasets	Projections	1.0 bpp				2.0 bpp				3.0 bpp			
		Acc ↑	Pe ↑	FID ↓	DR ↓	Acc ↑	Pe ↑	FID ↓	DR ↓	Acc ↑	Pe ↑	FID ↓	DR ↓
AFHQv2	GSD [33]	**99.93**	0.365	2.80	72.12	**98.77**	0.361	2.85	72.02	96.88	0.367	2.89	76.29
	MN	87.32	0.399	**2.14**	**62.45**	85.68	0.390	**2.20**	**59.97**	86.64	0.403	**2.13**	**61.71**
	MB	98.03	0.396	2.21	75.15	98.57	0.388	2.35	92.26	**99.19**	0.376	2.46	98.26
	MC	92.65	**0.407**	2.22	63.29	91.00	**0.404**	2.26	76.09	93.40	**0.405**	2.26	83.78
FFHQ	GSD [33]	**99.78**	0.394	2.96	69.44	**98.76**	0.393	3.24	70.19	97.61	0.392	3.42	69.84
	MN	88.00	0.422	**2.41**	**61.16**	86.75	**0.433**	**2.42**	**61.16**	87.06	**0.427**	**2.45**	**61.11**
	MB	98.12	0.427	2.77	71.97	98.19	0.361	3.30	89.34	**98.76**	0.310	4.30	98.91
	MC	93.17	**0.445**	2.58	61.51	91.97	0.414	2.75	66.32	93.09	0.409	3.11	81.45

Table 3
Results of various message payloads, 1.0 to 6.0 bpp. The original EDM models have FID scores of 2.17 on AFHQv2, 2.39 on FFHQ, and FID scores of 1.97 on CIFAR-10.

Datasets	metric	1.0	2.0	3.0	4.0	5.0	6.0	6.0* [38]
AFHQv2 64×64	Acc ↑	98.03	98.57	99.19	96.39	93.15	91.93	77.72
	Pe ↑	0.396	0.388	0.376	0.364	0.390	0.394	0.399
	FID ↓	2.21	2.35	2.46	2.42	2.35	2.34	2.33
FFHQ 64×64	Acc ↑	98.12	98.19	98.76	95.57	92.38	91.12	77.99
	Pe ↑	0.427	0.361	0.310	0.334	0.345	0.385	0.425
	FID ↓	2.77	3.30	4.30	3.90	3.67	3.37	2.41
CIFAR-10	Acc ↑	95.38	95.07	95.19	89.93	86.43	84.83	75.31
	Pe ↑	0.434	0.460	0.434	0.417	0.446	0.441	0.466
	FID ↓	2.09	2.30	2.66	2.48	2.37	2.31	2.01

*: our re-implementation of message mapping [38], which conceals 6 bpp messages.

4.3. Performance of hiding various bpp messages

We evaluated Diffusion-Stego on various payload settings, ranging from 1.0 bpp to 6.0 bpp for each dataset. In payloads from 1.0 to 3.0 bpp, we use the MB projection. In payloads from 4.0 and 5.0 bpp, we use both the MB projection and the multi-bit projection. In 6.0 bpp payloads, we only use the multi-bit projection.

As shown in Table 3, using the MB projection alone (from 1.0 to 3.0 bpp) results in higher extracted message accuracy compared to using the multi-bit projection. However, using the multi-bit projection (from 4.0 to 6.0 bpp) provides better anti-detection ability and image quality. This is because the distribution of $\mathbf{z}_m$ projected by the multi-bit projection is more similar to that of Gaussian noise than that of the MB projection. As the number of bits to hide in channels increases, the extracted message accuracy decreases due to the same reason as the MN projection.

We compared message mapping [38] with the concealing of 6 bpp messages in Table 3. Since message mapping preserves the distribution of random noise during sampling, it exhibits strong anti-detection capabilities and good image quality, similar to the MN method. However, message mapping results in extraction errors due to noise values near the decision boundary, leading to lower accuracy than our suggestions.

We hide 9.0 bpp messages using EDM models trained on the AFHQv2 dataset and the multi-bit projection. The extracted message accuracy from the stego images hiding 9.0 bpp messages is 83.18%.

4.4. Comparison with image steganography

We compared Diffusion-Stego with two image steganography methods, FNNS [19] and LISO [39].

FNNS. We compared Diffusion-Stego with image steganography, FNNS [19]. Specifically, we employed CIFAR-10 PFGM++ [40] model and hide 3.0 bpp messages using the MB method. We integrated diffusion models as decoder within FNNS.

The results of this comparison are presented in Table 4. As the number of iterations in the FNNS task increases, the accuracy of the message increases, but the anti-detection ability decreases. Image steganography methods have an advantage in terms of message accuracy, while generative steganography methods excel in anti-detection ability.

LISO. LISO comprises two tasks: training the encoder and optimization through iterations, similar to FNNS [19]. For these experiments, we mainly used the FFHQ dataset and pre-trained EDM to train and evaluate both LISO and Diffusion-Stego.

In comparison experiments, we employed the LISO models trained with high-resolution images for evaluation. We first generated FFHQ 64×64 images and used them as cover images in the LISO model. However, we observed that training the LISO model with

Table 4
Comparison result between Diffusion-stego and FNNS [19] using same diffusion models. The result of 0 iterations is that of Diffusion stego.

Metric	0 iters (MB)	1 iter	2 iters	3 iters	4 iters
Acc ↑	95.60	99.47	99.67	99.69	**99.70**
Pe ↑	**0.456**	0.435	0.218	0.211	0.179
FID ↓	**2.51**	2.66	3.05	3.11	3.19

Table 5
Results of extracted message accuracy using pre-trained high-resolution EDM [10,41] and Stable diffusion models [9].

Method	EDM						StableDiffusion				
	Cat			Bedroom			Amazon	Eiffel Tower	Mecha robot	Insect robot	Cabin
	1 bpp	2 bpp	3 bpp	1 bpp	2 bpp	3 bpp	0.0625 bpp				
MN	78.17	78.37	78.00	73.44	73.07	73.62	-	-	-	-	-
MB	**89.34**	**90.40**	**90.80**	**82.47**	**82.76**	**83.04**	96.81	93.22	98.14	96.96	97.59
MC	75.99	76.51	76.51	68.20	67.66	69.24	-	-	-	-	-

generated images shows low performance, it is because LISO is designed to be used with high-resolution images. We train the LISO model with 1000 FFHQ 1024×1024 images.

We hide 3 bpp messages using LISO and Diffusion-Stego with the MB method. Diffusion-Stego displays the result with an accuracy of 98.76%, anti-detection ability (Pe) of 0.310, and an FID score of 4.30. In contrast, LISO presents the results boasting an accuracy of 99.98%, anti-detection ability of 0.001, and an FID score of 74.3. Using the LISO model yielded higher accuracy compared to using Diffusion-Stego; images generated with LISO are more distinguishable than those generated with Diffusion-Stego.

This result shows that generative steganography methods are effective in concealing secret messages within low-resolution settings. When working with low-resolution images, the objects within the images are often compact. This indicates that there is limited space available to hide secrets. We hypothesize that the difficulty in training LISO using low-resolution generated images stems from this issue. In such compact conditions, generative steganography methods that can utilize image objects as secrets, offer advantages for effective secret hiding than image steganography methods.

4.5. Ablation studies on high-resolution models

We extended our method to high-resolution models, which generate images at a resolution of 256×256 pixels. We utilized pre-trained EDM models as provided by the authors of the Consistency models [41] trained on LSUN Cat and Bedroom [42] datasets. We generated 100 stego images to evaluate the accuracy of message extraction.

The accuracy results of high-resolution pre-trained EDM models are presented in Table 5. While the MB method achieves reasonably favorable performance, it is important to note that the EDM is primarily designed to generate low-resolution images, a resolution of 32×32 or 64×64 pixels. Thus, EDM might not be optimal for generating high-resolution images. We anticipate that enhancements in the EDM approach could potentially improve the performance of our method in high-resolution settings.

4.6. Applying on pre-trained text-to-image models

Our method can be extended to large-scale text-to-image models, which generate images with text guidance. In this case, the sender and the receiver should share two additional pieces of information: input text prompt and guidance scale. We use Stable diffusion [9] v1.4, an open-source model trained in the latent space of VAE [43]. During inference, the diffusion model generates 4×64×64 latent features, which are then decoded to 512×512 size images with VAE. Thus, Diffusion-Stego can conceal 0.0625 bpp messages when using Stable diffusion.

We measure message accuracy using Stable Diffusion. Specifically, we generated 20 images for each example text provided by the demo page of Stable Diffusion and calculated the accuracy for 0.0625 bpp messages using the MB method. The results of the message accuracy are presented in Table 5.

The result demonstrates that the message accuracy depends on the text conditions, and the task of Diffusion-Stego does not affect the generation of text conditions in Stable diffusion.

5. Conclusion

We propose Diffusion-Stego, a novel approach to generative steganography using deterministic samplers of diffusion models. We investigate the factors that affect the quality and extracted message accuracy when diffusion models generate stego images. We suggest three options for message projection, which have the trade-off of image quality, anti-detection, and extracted message accuracy. Our approach can hide large messages (more than 1.0 bpp, even 6.0 bpp with an accuracy of 90%) while maintaining the image quality of pre-trained diffusion models.

Our limitations include a trade-off between image quality, anti-detection ability, and extracted message accuracy. The message projection we introduce mitigates the image collapse issue and establishes a favorable balance between message accuracy and image quality. However, the distribution shift of random noise inevitably diminishes image quality, which, in turn, reduces anti-detectability. Additionally, the message projection hides secrets within the slight differences of the Gaussian noise, making Diffusion-Stego susceptible to steganalysis attacks, such as JPEG compression.

Our observations will support future research in enhancing these capabilities by enabling additional information sharing about players or developing well-defined projection methods while preserving the Gaussian noise distribution. Furthermore, since our approach uses pre-trained diffusion models, it can be extended to other domains such as video [44], audio [45], and text [46].

CRediT authorship contribution statement

Daegyu Kim: Writing – review & editing, Writing – original draft, Visualization, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Chaehun Shin:** Writing – review & editing, Writing – original draft, Visualization, Methodology, Investigation, Formal analysis. **Jooyoung Choi:** Writing – review & editing, Writing – original draft, Visualization, Investigation, Formal analysis. **Dahuin Jung:** Writing – review & editing, Writing – original draft, Visualization, Methodology, Investigation, Formal analysis, Conceptualization. **Sungroh Yoon:** Writing – review & editing, Writing – original draft, Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization.

Declaration of generative AI and AI-assisted technologies in the writing process

The authors drafted the manuscript before using Generative AI services. During the preparation of this work the authors used GPT-3.5 and 4 in order to enhance readability after drafting the manuscript. After using this service, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Sungroh Yoon reports financial support was provided by National Research Foundation of Korea. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) [NO.2021-0-01343, Artificial Intelligence Graduate School Program (Seoul National University)], the BK21 FOUR program of the Education and Research Program for Future ICT Pioneers, Seoul National University in 2025, the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2022R1A3B1077720), and Institute of Information & communications Technology Planning & Evaluation (IITP) under the Leading Generative AI Human Resources Development (IITP-2025-RS-2024-00397085) grant funded by the Korea government (MSIT).

Appendix A. Supplementary material

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.ins.2025.122358.

Data availability

The data that has been used is confidential.

References

[1] Tayana Morkel, Jan H.P. Eloff, Martin S. Olivier, An overview of image steganography, in: ISSA, vol. 1, 2005, pp. 1–11.
[2] Neil F. Johnson, Sushil Jajodia, Exploring steganography: seeing the unseen, Computer 31 (2) (1998) 26–34.
[3] Guanshuo Xu, Han-Zhou Wu, Yun-Qing Shi, Structural design of convolutional neural networks for steganalysis, IEEE Signal Process. Lett. 23 (5) (2016) 708–712.
[4] Kuo-Chen Wu, Chung-Ming Wang, Steganography using reversible texture synthesis, IEEE Trans. Image Process. 24 (1) (2014) 130–139.
[5] Ping Wei, Sheng Li, Xinpeng Zhang, Ge Luo, Zhenxing Qian, Qing Zhou, Generative steganography network, in: Proceedings of the 30th ACM International Conference on Multimedia, 2022, pp. 1621–1629.
[6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, Generative adversarial networks, Commun. ACM 63 (11) (2020) 139–144.
[7] Durk P. Kingma, Prafulla Dhariwal, Glow: generative flow with invertible 1x1 convolutions, Adv. Neural Inf. Process. Syst. 31 (2018).
[8] Jonathan Ho, Ajay Jain, Pieter Abbeel, Denoising diffusion probabilistic models, Adv. Neural Inf. Process. Syst. 33 (2020) 6840–6851.
[9] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, Björn Ommer, High-resolution image synthesis with latent diffusion models, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 10684–10695.
[10] Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, Elucidating the design space of diffusion-based generative models, Adv. Neural Inf. Process. Syst. 35 (2022) 26565–26577.

[11] Jiaming Song, Chenlin Meng, Stefano Ermon, Denoising diffusion implicit models, in: International Conference on Learning Representations, 2021.
[12] Yinyin Peng, Donghui Hu, Yaofei Wang, Kejiang Chen, Gang Pei, Weiming Zhang, Stegaddpm: generative image steganography based on denoising diffusion probabilistic model, in: Proceedings of the 31st ACM International Conference on Multimedia, 2023, pp. 7143–7151.
[13] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Sepp Hochreiter, Gans trained by a two time-scale update rule converge to a local Nash equilibrium, Adv. Neural Inf. Process. Syst. 30 (2017).
[14] Tero Karras, Samuli Laine, Timo Aila, A style-based generator architecture for generative adversarial networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4401–4410.
[15] Yunjey Choi, Youngjung Uh, Jaejun Yoo, Jung-Woo Ha, Stargan v2: diverse image synthesis for multiple domains, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2020.
[16] Shumeet Baluja, Hiding images in plain sight: deep steganography, Adv. Neural Inf. Process. Syst. 30 (2017).
[17] Kevin Alex Zhang, Alfredo Cuesta-Infante, Lei Xu, Kalyan Veeramachaneni, Steganogan: high capacity image steganography with gans, arXiv preprint, arXiv: 1901.03892, 2019.
[18] Jiren Zhu, Russell Kaplan, Justin Johnson, Fei-Fei Li, Hidden: hiding data with deep networks, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 657–672.
[19] Varsha Kishore, Xiangyu Chen, Yan Wang, Boyi Li, Kilian Q. Weinberger, Fixed neural network steganography: train the images, not the network, in: International Conference on Learning Representations, 2021.
[20] Jiayi Xu, Xiaoyang Mao, Xiaogang Jin, Aubrey Jaffer, Shufang Lu, Li Li, Masahiro Toyoura, Hidden message in a deformation-based texture, Vis. Comput. 31 (2015) 1653–1669.
[21] Sheng Li, Xinpeng Zhang, Toward construction-based data hiding: from secrets to fingerprint images, IEEE Trans. Image Process. 28 (3) (2018) 1482–1497.
[22] Ming-ming Liu, Min-qing Zhang, Jia Liu, Ying-nan Zhang, Yan Ke, Coverless information hiding based on generative adversarial networks, arXiv preprint, arXiv:1712.06951, 2017.
[23] Zhuo Zhang, Guangyuan Fu, Rongrong Ni, Jia Liu, Xiaoyuan Yang, A generative method for steganography by cover synthesis with auxiliary semantics, Tsinghua Sci. Technol. 25 (4) (2020) 516–527.
[24] Mehdi Mirza, Simon Osindero, Conditional generative adversarial nets, arXiv preprint, arXiv:1411.1784, 2014.
[25] Donghui Hu, Liang Wang, Wenjie Jiang, Shuli Zheng, Bin Li, A novel image steganography method via deep convolutional generative adversarial networks, IEEE Access 6 (2018) 38303–38314.
[26] Cong Yu, Donghui Hu, Shuli Zheng, Wenjie Jiang, Meng Li, Zhong-qiu Zhao, An improved steganography without embedding based on attention gan, Peer-to-Peer Netw. Appl. 14 (2021) 1446–1457.
[27] Zhili Zhou, Yuecheng Su, Jin Li, Keping Yu, QM Jonathan Wu, Zhangjie Fu, Yunqing Shi, Secret-to-image reversible transformation for generative steganography, IEEE Trans. Dependable Secure Comput. (2022).
[28] Ping Wei, Ge Luo, Qi Song, Xinpeng Zhang, Zhenxing Qian, Sheng Li, Generative steganographic flow, in: 2022 IEEE International Conference on Multimedia and Expo (ICME), IEEE, 2022, pp. 1–6.
[29] Zhili Zhou, Xiaohua Dong, Ruohan Meng, Meimin Wang, Hongyang Yan, Keping Yu, Kim-Kwang Raymond Choo, Generative steganography via auto-generation of semantic object contours, IEEE Trans. Inf. Forensics Secur. 18 (2023) 2751–2765.
[30] Yi Cao, Zhili Zhou, Chinmay Chakraborty, Meimin Wang, Q.M. Jonathan Wu, Xingming Sun, Keping Yu, Generative steganography based on long readable text generation, IEEE Trans. Comput. Soc. Syst. 11 (4) (2024) 4584–4594.
[31] Weina Dong, Jia Liu, Yan Ke, Lifeng Chen, Wenquan Sun, Xiaozhong Pan, Steganography for neural radiance fields by backdooring, 2023.
[32] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, Ben Poole, Score-based generative modeling through stochastic differential equations, in: International Conference on Learning Representations, 2021.
[33] Ping Wei, Qing Zhou, Zichi Wang, Zhenxing Qian, Xinpeng Zhang, Sheng Li, Generative steganography diffusion, 2023.
[34] N. Ahmed, T. Natarajan, K.R. Rao, Discrete cosine transform, IEEE Trans. Comput. C-23 (1) (1974) 90–93.
[35] Jiwen Yu, Xuanyu Zhang, Youmin Xu, Jian Zhang, Cross: diffusion model makes controllable, robust and secure image steganography, Adv. Neural Inf. Process. Syst. (NeurIPS) (2023).
[36] Tushar M. Jois, Gabrielle Beck, Gabriel Kaptchuk, Pulsar: secure steganography for diffusion models, in: Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security, 2024, pp. 4703–4717.
[37] Alex Krizhevsky, Geoffrey Hinton, et al., Learning multiple layers of features from tiny images, 2009.
[38] Kejiang Chen, Hang Zhou, Hanqing Zhao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Distribution-preserving steganography based on text-to-speech generative models, IEEE Trans. Dependable Secure Comput. 19 (5) (Sep. 2022) 3343–3356.
[39] Xiangyu Chen, Varsha Kishore, Kilian Q. Weinberger, Learning iterative neural optimizers for image steganography, in: The Eleventh International Conference on Learning Representations, 2022.
[40] Yilun Xu, Ziming Liu, Yonglong Tian, Shangyuan Tong, Max Tegmark, Tommi Jaakkola PFGM++, Unlocking the potential of physics-inspired generative models, in: Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, Jonathan Scarlett (Eds.), Proceedings of the 40th International Conference on Machine Learning, 23–29 Jul 2023, in: Proceedings of Machine Learning Research, vol. 202, PMLR, pp. 38566–38591.
[41] Yang Song, Prafulla Dhariwal, Mark Chen, Ilya Sutskever, Consistency models, arXiv preprint, arXiv:2303.01469, 2023.
[42] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, Jianxiong Xiao, Lsun: construction of a large-scale image dataset using deep learning with humans in the loop, arXiv preprint, arXiv:1506.03365, 2015.
[43] Diederik P. Kingma, Max Welling, Auto-encoding variational Bayes, arXiv preprint, arXiv:1312.6114, 2013.
[44] Jonathan Ho, Tim Salimans, Alexey A. Gritsenko, William Chan, Mohammad Norouzi, David J. Fleet, Video diffusion models, in: ICLR Workshop on Deep Generative Models for Highly Structured Data, 2022.
[45] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, Bryan Catanzaro, Diffwave: a versatile diffusion model for audio synthesis, in: International Conference on Learning Representations, 2021.
[46] Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S. Liang, Tatsunori B. Hashimoto, Diffusion-lm improves controllable text generation, Adv. Neural Inf. Process. Syst. 35 (2022) 4328–4343.