Generative Image Steganography Based on Point Cloud

Yangjie Zhong

1622177120@qq.com

Engineering University of PAP

Jia Liu

Engineering University of PAP

Meiqi Liu

Engineering University of PAP

Yan Ke

Engineering University of PAP

Minqing Zhang

Engineering University of PAP

Research Article

Keywords: Information Hiding, Steganography, Implicit Neural Representation, Point Cloud

Posted Date: November 19th, 2024

DOI: https://doi.org/10.21203/rs.3.rs-5388114/v1

Additional Declarations: No competing interests reported.

Generative Image Steganography Based on Point Cloud

Yangjie Zhong[1,2], Jia Liu[1,2*†], Meiqi Liu[1,2], Yan Ke[1,2], Minqing Zhang[1,2]

[1*]College of Cryptography Engineering, Engineering University of PAP, Xi "an Shaanxi, 710086, China.
[2]Key Laboratory of Network and Information Security of PAP, Engineering University of PAP, Xi "an Shaanxi, 710086, China.

*Corresponding author(s). E-mail(s): liujia1022@gmail.com;
Contributing authors: 1622177120@qq.com; 2362678742@qq.com;
15114873390@163.com; api_zmq@126.com;
†These authors contributed equally to this work.

Abstract

In deep steganography, the model size is usually related to the underlying mesh resolution, and a separate neural network needs to be trained as a message extractor. In this paper, we propose a generative image steganography based on point cloud representation, which represents image data as a point cloud, learns the distribution of the point cloud data, and represents it in the form of a continuous function. This method breaks through the limitation of the image resolution, and can generate images with arbitrary resolution according to the actual need, and omits the need for explicit data for image steganography. At the same time, using a fixed point cloud extractor transfers the training of the network to the point cloud data, which saves the training time and avoids the risk of exposing the steganography behavior caused by the transmission of the message extractor. Experiments prove that the steganographic images generated by the scheme have very high image quality and the accuracy of message extraction reaches more than 99%.

Keywords: Information Hiding, Steganography, Implicit Neural Representation, Point Cloud

1 Introduction

Generative steganography is a steganography technique in which the generation of a secret-containing carrier is directly driven by a secret message. Various types of media are used to hide secret messages, including text[1], video[2], images[3], and digital audio[4], but currently the main target of generative steganography is still image data. Unlike traditional steganography, generative steganography relies on advanced machine learning methods and models, utilizing, for example, generative adversarial networks[5] and diffusion models[6] for steganography. Its core concept is no longer to hide secret information in the existing content but to generate content containing secret information[7], and these contents look no different from normal data in appearance.

Despite the apparent merits of contemporary generative image steganography techniques in concealing information, several concomitant challenges emerge: one is that the resolution of the image is limited. The cover images of current steganography schemes are grid representations, stored in the form of pixel points in the computer, and when continuous image signals are converted to discrete grid data through sampling, the detail information is lost, reducing the resolution and quality of the image. In addition, the complexity of the underlying model is intricately linked to a specific resolution, which makes it impossible for a single model to adaptively represent images of different resolutions. Secondly, the transmission behavior of the message extractor raises the suspicion of steganography analysis and risks exposing steganography. Meanwhile, the size of the extractor is usually large, and transmission is difficult due to the limitation of the capacity and efficiency of the transmission channel. Thirdly, the scope of current image steganography is constrained to image data, necessitating the utilization of explicit image carriers for embedding and extracting messages. Even when implicit neural representations are employed to depict images, they ultimately rely on explicit image sampling for the purpose of steganography, with the implicit representation serving merely as an intermediate data representation modality. This type of scheme changes the representation of the image but still does not fundamentally change the object of steganography. To solve these three problems, we propose a generative image steganography based on point cloud representation. Point cloud is a form of data representation that treats coordinates and features as having the concept of potential distances, and is a collection of pairs of coordinates and features, commonly used in 3D data modeling.[8] This form of data representation provides a new direction for the representation of multimedia data, especially in the field of images, and solves the deficiencies that exist in gridded representation of data.

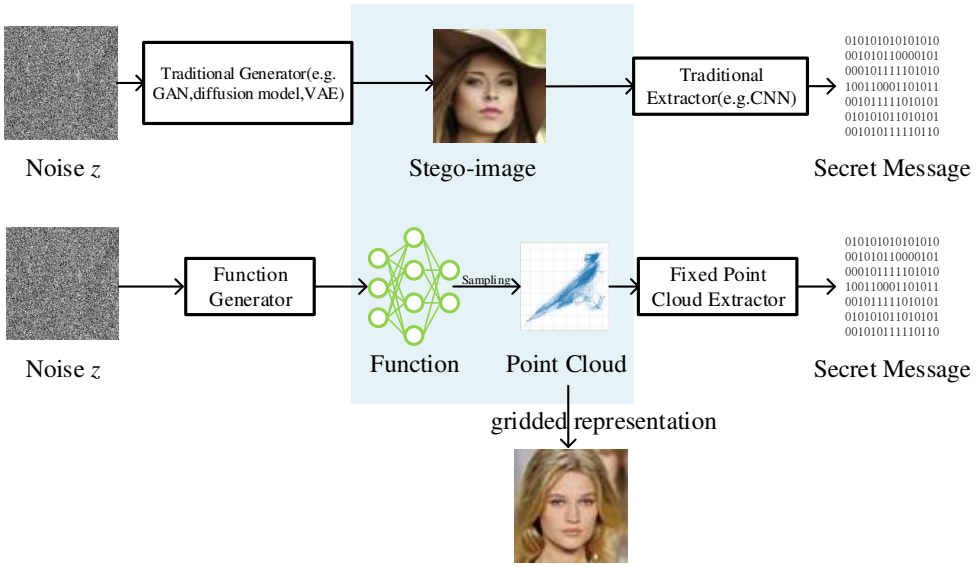


Fig. 1 Comparison with traditional steganography schemes.

As depicted in Fig.1, the proposed scheme is comprised of two distinct components: a function generation process leveraging implicit neural representation and a message extraction process utilizing a point cloud extractor. In contrast to the conventional method of generating carrier images via Generative Adversarial Networks (GANs), our approach adopts a Generative Adversarial Stochastic Process (GASP)[8], which produces functions that represent images through a function generator. Subsequently, these functions are fed with coordinates intended for sampling, generating point cloud pairs that encompass both coordinates and features. This paradigm eliminates the need for explicit images as carriers in steganography, with point cloud data serving as the direct medium.

During the steganography process, a tailored message extractor designed specifically for point cloud data is employed, enabling direct manipulation of the point cloud for message extraction. The utilization of point cloud data to represent images transcends the constraints of resolution, allowing for super-resolution based on practical requirements, thereby broadening the scope of potential steganography carriers. Furthermore, the integration of a fixed point cloud extractor[9] within the steganography module, as opposed to traditional extractors, shifts the focus of network training towards optimizing the point cloud data itself. This shift not only reduced training costs but also improves the quality of the steganographic image, marking a significant advancement in the field.

Our proposed scheme presents several notable contributions, which are outlined as follows:

1. Neural Implicit Representation for Image Representation: The neural implicit representation is used to learn the function distribution of the image and sample the function from it, and use the coordinates as inputs to sample the features from the

function to form the point cloud data to represent the image, this representation method breaks through the limitation of the resolution, and the resolution parameter only needs to be changed to sample the image at each resolution from the model.

2. Optimized Point Cloud Extractor for Enhanced Steganography: Instead of employing traditional extractors, we incorporate a fixed point cloud extractor into our framework. This shift redirects the focus of network training towards optimizing the point cloud data itself. This modification mitigates the adverse effects of steganography on image quality and significantly enhances the resistance to steganalysis, thereby improving the overall security of the steganographic process.
3. Generalized Steganography Framework for Multimedia Data: Our steganography process operates directly on the point cloud data, eliminating the need for an explicit carrier such as an image. This approach establishes a generalized steganography framework that is not limited to image data but holds the potential for extension to other multimedia formats in the future.

2 Related Work

2.1 Deep Steganography

The formidable feature learning prowess of deep learning has propelled steganography utilizing deep neural networks into the forefront of research endeavors. In the context of steganography schemes, the functional partitioning of deep neural networks serves as a cornerstone for categorizing these approaches into two distinct classes: codec network-based steganography and deep generative model-based steganography.

2.1.1 Steganography Based on Encoder-Decoder Network

The proposed method capitalizes on the advanced coding and decoding capabilities of deep neural networks to facilitate the integration of an original carrier and a secret message, ultimately outputting a steganographic image. This approach encompasses two distinct frameworks: the Dependent Deep Hiding (DDH)[10] framework and the Universal Deep Hiding (UDH)[11] framework.

In the DDH framework, the secret message is initially processed through a preprocessing network, yielding a feature map that encapsulates its essential characteristics. This feature map is then fused with the carrier image, and the combined input is subsequently fed into a coding network. The coding network leverages this combined information to generate an encrypted carrier, which seamlessly incorporates the secret message while maintaining the integrity of the original carrier image.

Alternatively, the UDH framework adopts a more decoupled approach, wherein the coding network encodes the secret information independently of the carrier image. This independence allows for greater flexibility and potential generalization across different carrier types. Notably, Zhu et al. pioneered the development of a

coding network specifically tailored for image steganography, incorporating an innovative analog noise processing layer to enhance the stealth and robustness of the encoded secret. Building upon Zhu's work[12], Zhang et al.[13] further refined the approach by introducing a discriminator that leverages Zhu's[12] design while subtly modifying the structure of the codec network. This modification aimed to optimize the performance of the steganographic process, ensuring that the generated steganographic images are both imperceptible and resilient to steganalysis.

2.1.2 Steganography Based on Deep Generative Models

Deep generative models such as Generative Adversarial Networks (GANs)[14], Diffusion Models (DPMs)[6], Variational Autocoders (VAEs)[15], and Flow-based Models (FBMs)[16] are hot topics in current research.

(1)Steganography based on Generative Adversarial Networks. In 2014, the proposal of Generative Adversarial Networks (GANs) provided an opportunity for the combination of deep learning and information hiding. A modification-based steganography method enhances the performance of resistance to steganalysis by utilizing game adversarial strategies in the GAN model, which is specifically categorized into two classes. One class utilizes generative models to construct the original vectors such as SGAN[17], SSGAN[14], and so on; The other category uses deep generation to construct modification strategies, mainly including ASDL-GAN[18] and UT-SCA-GAN[19], which mainly use the generative confrontation process probability to modify the matrix in the framework of minimizing the distortion cost.

Generative adversarial networks have also been used to design carrier-selective steganography, and Ke et al.[20] proposed to utilize a set of generators to achieve steganographic key generation and message extraction, which does not require both parties to share datasets or mappings in advance, unlike methods that require shared datasets.

The carrier synthesis steganography based on GAN refers to the steganography scheme that generates the cryptographic carriers directly through the deep generative model without specifying the original carriers in advance, and its basic flow is shown in Fig.2. Liu et al.[21] proposed the generative steganography scheme (GSS) based on constraint sampling. Liu et al. and Hu et al.[22] used the method of establishing message mapping to get the cryptographic carriers directly by using the generative model.

Fig. 2 Generative model-based steganography framework for carrier synthesis.

(2)Steganography Based on Diffusion Models The diffusion model[15, 23] defines a Markov chain of diffusion steps by gradually adding random noise to the data and then learning the reverse diffusion process to construct the desired data samples. Karras[24],

Xu et al.[25] use the diffusion model and a deterministic sampler to generate high-quality images. Kim et al.[26] propose a Diffusion-Stego steganography scheme that projects secret messages into the latent noise of the diffusion model. steganography scheme, this message projection approach can be applied to pre-trained diffusion models for generating high-quality images or even large-scale text-to-image models. The StegoDiffusion scheme proposed by Wei et al.[27] utilizes non-Markov chains and fast sampling techniques for efficient steganographic image generation. Secret data and steganographic images can be interconverted to enable the use of irreversible but more expressive network structures.

(3)Other Deep Generative Models in Steganography In the realm of steganography research, there exists a diverse array of emerging approaches beyond generative adversarial models and diffusion models. Among these, variational autoencoders (VAEs)[16], flow-based models[17], and autoregressive model-driven designs have emerged as current research hotspots. Notably, Yang[28] has contributed to this landscape by proposing an autoregressive model-based generative steganography framework, leveraging PixelCNN+ for pixel-level information hiding. Through rigorous theoretical derivation, the security of this framework has been comprehensively validated, demonstrating its potential to withstand steganalysis attacks.

Parallel to the advancements in diffusion model steganography[27], researchers such as Jing et al.[29] and Guan et al.[30] have capitalized on the capabilities of flow models (specifically, reversible neural networks) to devise high-capacity steganography schemes. These works underscore the versatility of flow models in enabling the encoding of substantial amounts of secret data within cover media while maintaining a low distortion profile and stealth. By exploiting the reversible nature of these networks, they have achieved impressive steganographic performance, further expanding the horizons of steganography research.

Although deep neural network-based steganography has the advantages of high steganography, robustness, and large steganographic capacity, it suffers from the following problems: first, the image data are all stored in the computer in a gridded form, making the size of the codec directly related to the resolution of the underlying gridded data. Two, the training cost of the message extractor is high, while the delivery behavior of the message extractor increases the risk of exposing the steganography. Third, these steganographic schemes require explicit representations such as images as intermediate carriers for message extraction and embedding.

In order to solve the problem of high transmission and training cost of message extractor, Liu[7] proposed a constraint sampling-based steganography that avoids the training process of the extractor. According to the characteristic of neural networks sensitive to small perturbations, Kishore et al.[9] proposed a steganography scheme based on fixed neural networks, which transforms the training of extractors into the training of images and reduces the training cost.Luo et al.[31] proposed a key-based steganography scheme for fixed neural networks based on this scheme, which solves the defect of FNNS that anyone can extract the messages, and improves the security. Li et al.[32, 33] camouflage steganographic networks by ordinary deep neural networks

(called covert DNN models) that are associated with image classification or segmentation tasks. Although these approaches ensure the security of the message decoder, they still cannot avoid the problem of high communication burden.

2.2 Steganography Based on Implicit Neural Network

Unlike the implicit writing scheme described above which acts directly on explicit data such as images, neural implicit representations have also been applied in implicit writing schemes. Neural implicit representation refers to viewing an image as a function, or an arbitrary neural network can be defined as a function[34]. Han et al.[35] first implicitly represent a secret message, and then embed the implicit model data into an image carrier. Li et al.[36] propose StegaNeRF, a scheme for embedding information in the neural radiation field (NeRF) rendering. Luo et al.[37] design a distortion-resistant scheme by replacing all of these original NeRF schemes with watermarked color representations. all replace the neural spoke color representation with a watermarked color representation, and design a distortion-resistant rendering scheme to ensure that the watermark information can be stably extracted from the 2D images rendered by NeRF.Chen et al.[38] take the secret view as an important basis for verifying copyrights from the properties of NeRF new view synthesis, and train a watermark verification model with over-parametrization methods. Dong et al.[39] apply this technique to the field of steganography and realize a NeRF-based steganography scheme.

Liu et al.[40] first proposed the concept of function steganography based on implicit representation, which embeds the implicit representation function of a secret message into the implicit representation function of a carrier, and avoids the need to use a separate message extractor by directly obtaining the secret message from the secret-containing carrier through the key shared by both the sender and the receiver. Importantly, this method transforms the secret message data into a unified data format, i.e., a function (model), thus providing a new implementation framework for data steganography, which can be implemented for many types of message data. Yang et al.[28] based on the message implicit representation by stitching multiple messages implicit representation models and the carrier's implicit representation model into a large network and then disambiguating it. Existing implicit representation steganography methods[28, 37–39] generally achieve message hiding by modifying the function of the implicit representation, unlike these methods, this paper for the first time takes the point cloud data as the steganographic object, generates the function through a function generator, samples the original point cloud data, and uses a fixed point cloud extractor to extract the secret message. Compared with the generative steganography methods[6, 17, 18] that generate images directly, the steganography scheme based on point cloud representation breaks through in both image type and resolution, omits the image as an explicit representation, and at the same time, solves the problem of message extractor delivery, and improves the quality of the image after steganography.

3 Problem Fomulation

3.1 Framework

The basic flow of generative image steganography based on point cloud representation is shown in Fig.3. Our scheme uses a generative adversarial stochastic process[8] to design the function generator and uses a customized point cloud extractor as the fixed extractor. The specific steganography process is as follows: firstly, a noise is sampled from a Gaussian distribution, which is fed into the function generator trained in advance to generate a continuous function, and then the corresponding number of point clouds is sampled from it according to the actual needs, and these point cloud data are the steganography objects. Then the sampled point clouds are input into the point cloud extractor to extract the secret message M', and the loss is calculated with the secret message M. The point cloud data is modified by the perturbation of small noise until the loss is smaller than the preset value, and then the steganography process is finished. Throughout the steganography process, we manipulate the point cloud data and do not use explicit data such as images at all. In order to visualize the difference between the images before and after the steganography, we convert the point cloud data before and after the noise addition into 2D image data and project the point cloud distribution in space onto a 2D planar map.

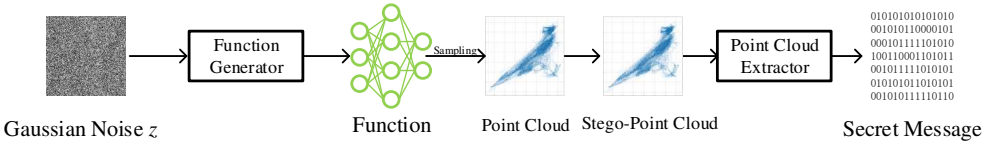


Fig. 3 Generative image steganography framework based on point cloud representation.

3.2 Continuous Function Based on Point Cloud Carrier Construction

3.2.1 The Representation of Point Cloud

Traditionally, image data are stored in a computer in a grid, but since the signal of the image itself is continuous, grid storage will inevitably cause a loss in the quality of the image, so we consider a new way to represent the data. For an image I, the coordinates (x, y) of each pixel point correspond to an RGB feature value $I(x, y)$, taking the coordinates and features as a set $((x, y), I(x, y))$. It is inappropriate to think of the coordinates of an image as scattered, unrelated discrete points, thus ignoring the notion of potential distances that the coordinates have between them. Based on this consideration, we use the PointConv computational framework for representing image data proposed by Wu et al[41]. $x \in X$ representing coordinates, $y \in Y$ representing features, a coordinate and a feature form a point cloud pair. This representation treats the data completely independent of the grid and acts directly on the set, and this method of representing data with point clouds can be easily extended to more general data.

3.2.2 Constructing Continuous Functions Based on Point Clouds

For an image, we try to represent the point cloud data as a continuous function. $x = (x, y)$ corresponding to the pixel positions, $y = (r, g, b)$ corresponding to the RGB values, $\{(x_i, y_i)\}_{i=1}^{n}$ corresponding to the set of point clouds for all pixel positions and RGB values. Given a set of coordinates and their corresponding features, we can learn a function of this image by minimizing the function $min_\theta \sum_{i=1}^{n} \|f_\theta(x_i) - y_i\|_2^2$. The core of the function $f_\theta \colon X \to Y$ representation is that they scale with signal complexity rather than signal resolution, and in order to represent more complex signals, we can achieve this with an extended capacity of f_θ.

3.2.3 The Construction of Function Generator

In generative modeling, we typically generate models based on the distribution of these functions. When representing data points with a function, we want to learn the distribution of the function. In the case of an image, for example, the standard generative model usually samples some noise and outputs n pixels through a neural network. In this paper, we sample the weights of the neural network to obtain a function that can be obtained at arbitrary coordinates. This function representation abandons the traditional underlying network and operates directly on coordinate and feature pairs. In this paper, we use an adversarial approach to train the function distribution model, called Generative Adversarial Stochastic Process (GASP)[8]. In this way, cover images can be generated and represented as continuous functions.

Our goal is to learn the distribution of functions thereby obtaining a continuous functional representation of the distribution of a class of point cloud data, but no specific function can be obtained. For images, we do not have direct access to a function that maps pixel positions to RGB values, but we do have access to a set of n coordinate-feature pairs. Such a set of coordinates and features corresponds to the input/output pairs of a function, avoiding direct learning of the function distribution by directly manipulating the function itself. The function generator architecture, shown in Fig.4, f_θ is a multi-layer perceptual machine where the learned distribution of f_θ is actually the learned weights of $p(\theta)$, which in turn are jointly determined by g_φ and $p(z)$. $p(z)$ is the distribution of the hidden vectors, usually Gaussian, and $g_\varphi : Z \to \Theta$ is the weights that map z into $p(z)$ and f_θ into θ. When the weights are decided, the input coordinate pairs can output the feature values, and in this way, the function is sampled from the function distribution, which is the function representation of a carrier image. The image sampled in this way is not limited by the resolution and can be sampled with super-resolution according to the actual needs, getting rid of the dependence on meshing.

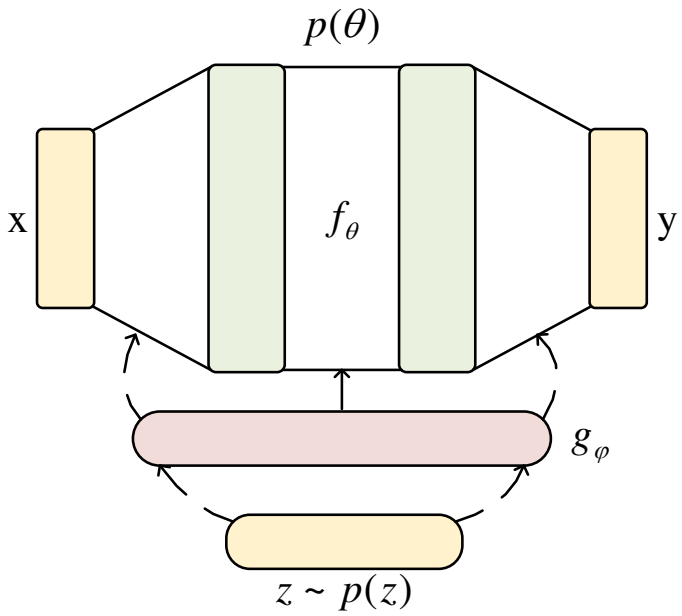


Fig. 4 The Architecture of Function Generator.

3.2.4 The Training of Function Generator

We use Emilien's generative adversarial stochastic process[8] to train the function generator, and the training process is shown in Fig.5. The function generator consists of two parts, the generator and the discriminator. The training process consists of two parts, the generator and the discriminator, firstly, a set of randomly sampled coordinate values are input to the function generator to get a set of feature values, from which coordinate-feature point cloud pairs are composed. At the same time, a real image is randomly selected from the dataset, and its point cloud pairs are obtained through the data converter, and the two sets of point cloud pairs are input into the discriminator, if the discriminator can not distinguish the authenticity of the two, the point cloud pairs generated by the generator will be outputted; otherwise, the coordinate values of the inputs are updated to repeat the above process, and the generator is trained through this generative counter-randomization process. In the process of point cloud generation, we refer to the experimental setup, and use Ha et al.'s[42] supernetwork parameter settings and Goodfellow et al.'s[8] adversarial method in the function generator training process, and do not consider using any underlying network for representation in the training process, but directly sample the weights of the neural network. The weights of the neural network are sampled to obtain the continuous function.

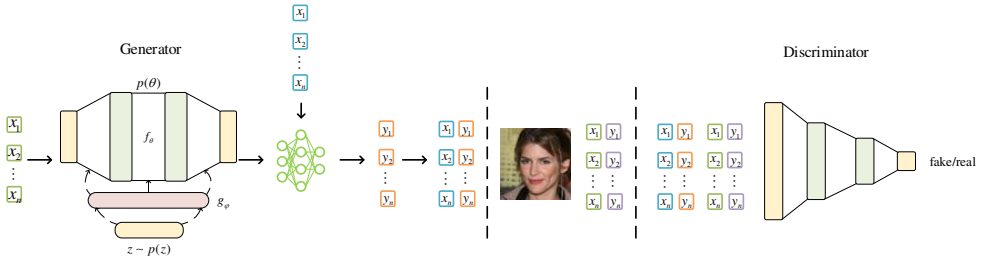


Fig. 5 The Training Process of Function Generator.

3.2.5 Sampling

A function $f(x)$ is sampled from the trained function generator by inputting a random Gaussian distribution to obtain a neural implicit representation of the image, and then a coordinate pair is input into the function to obtain the features according to the actual need so that a set of point cloud data is sampled from the function for subsequent implicit writing. This approach makes the image independent of resolution, and Fig.6 gives an example of the model training datasets are all 64×64 dimensional images, which are sampled with super-resolution to generate a high-resolution image.

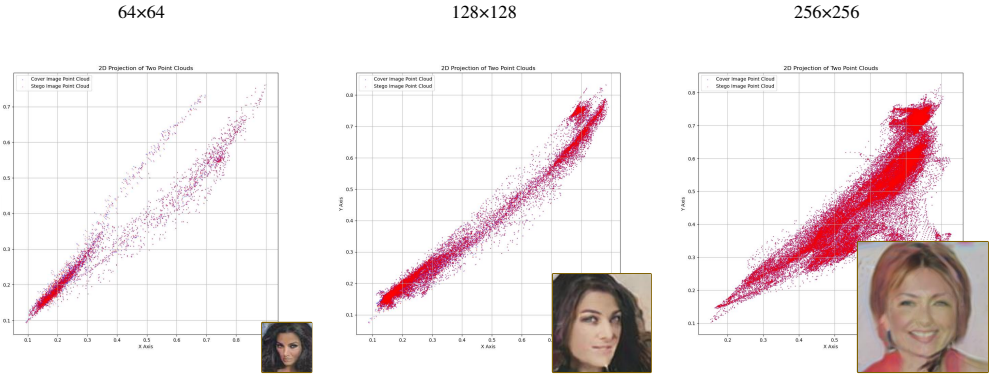


Fig. 6 Super Resolution Sampling.

3.3 The Fixed Point Cloud Extractor

The steganography process is shown in Fig.7. The sender first samples a set of point cloud data from the function, feeds it into a fixed point cloud extractor, and uses a tiny noise to modify the point cloud data according to the binary cross-entropy loss between the extracted secret message M' and the secret message M. The receiver extracts the message through the same point cloud extractor. In the scheme, the extractor is no longer trained, but a point cloud extractor is trained and fixed in advance, which transfers the training of the network to the point cloud data, changing the problem that the quality of the original steganographic scheme depends on how well the extractor is trained. The sender and receiver only need to share the architecture

of the message extractor network and the random seed used to initialize its weights to get this extractor, and the actual weights of the network do not need to be shared, which avoids the risk of steganography exposure brought by the transmission of the message extractor.

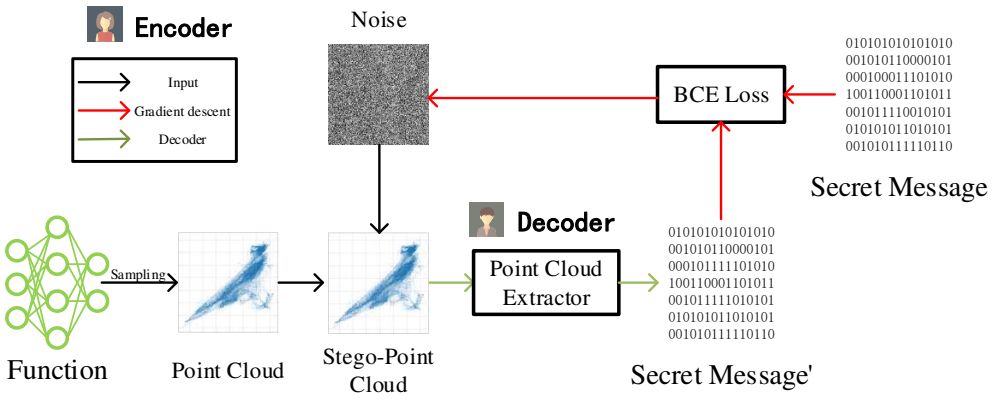


Fig. 7 Steganography Process Based on Fixed Point Cloud Extractor.

3.3.1 Point Cloud Extractor

The point cloud extractor operates on point cloud data in the same way as the discriminator used in the GASP, with the point cloud extractor acting directly on the point cloud data for message extraction. To the best of our knowledge, this is the first time that point cloud data is used as a steganographic object, and the advantage of this approach is that it gets rid of the process of traditional image steganography that requires explicit data and can be easily extended to other types of data. Its network structure has five layers, which convert point cloud data from coarse-grained to fine-grained by changing the number of input and output channels, and extracts and aggregates features in the process. The sender and receiver share the exact same extractor structure of the network architecture.

3.3.2 Loss Function

The loss function of GASP during training is as follows:

$$R_l(\mathbf{s}) = \frac{l}{2} \parallel \nabla_{\mathbf{y}_l,\cdots,\mathbf{y}_n} D(\mathbf{s}) \parallel^2 = \frac{l}{2} \sum_{\mathbf{y}_i} \parallel \nabla_{\mathbf{y}_i} D(\mathbf{s}) \parallel^2 \tag{1}$$

where $\mathrm{s} = \{(\mathrm{x_i}, \mathrm{y_i})\}_{\mathrm{i=1}}^{\mathrm{n}}$ is the set composed of point cloud data pairs, $\mathrm{y_1} \cdots \mathrm{y_n}$ is the generated eigenvalue, and D is the loss of the discriminator. The implicit writing process loss function is as follows:

$$\min_{\widetilde{P}} \langle \mathbf{M}, \log F\left(\widetilde{\mathbf{P}}\right) \rangle + \langle (1-\mathbf{M}), \log\left(1 - F\left(\widetilde{\mathbf{P}}\right)\right) \rangle \tag{2}$$

$$\text{s.t.} \left\| \mathrm{P} - \tilde{\mathrm{P}} \right\|_{\infty} \leq \epsilon \text{ and } 0 \leq \tilde{\mathrm{P}} \leq 1 \tag{3}$$

Where P is the image corresponding to the original point cloud, $\tilde{P}$ is the image corresponding to the steganographic point cloud, F is the point cloud message extractor, and M is the original secret message, ϵ is a very small positive constant. In the optimization process we use the unconstrained L-BFGS algorithm to optimize the objective about $\tilde{P}$. This is because the L-BFGS algorithm tracks the second-order gradient statistic and enables faster optimization.

4 Experiments

4.1 Evaluation Metrics

We evaluate the scheme in terms of both message extraction error rate and image quality, which include the following three metrics:

1. Message extraction error rate:

$$\frac{\|M - \lfloor F(\widetilde{X}) \rfloor\|_1}{HWD} \tag{4}$$

where M is the secret message, F is the point cloud extractor, $\tilde{P}$ is the steganalyzed point cloud data, H, W, and D are used to denote the resolution of the image and the number of channels, and $\lfloor\rfloor$ denotes the rounding function, which is used to measure the number of bits that have been recovered by the error.

2. Peak Signal to Noise Ratio (PSNR):

$$\mathbf{MSE} = \frac{l}{HW} \sum_{i=l}^{H} \sum_{i=l}^{W} \left[X_{i,j} - \widetilde{X}_{i,j} \right]^2 \tag{5}$$

$$\mathbf{PSNR} = 20 \log_{10}(max_P) - 10 \log_{10}(\mathbf{MSE}) \tag{6}$$

The Peak Signal to Noise Ratio (PSNR) is used to measure the difference between the original point cloud and the dense point cloud, which is a commonly used metric to measure the quality of the image, and H, W is the size of the image resolution, and we use PSNR to judge the quality of the image before and after steganography.

3. Structural similarity index (SSIM):

$$\mathbf{SSIM} = \frac{(2\mu_{\mathbf{X}}\mu_{\widetilde{\mathbf{X}}} + c_1)\left(2\sigma_{\mathbf{X}}\widetilde{\mathbf{X}} + c_2\right)\left(\sigma_{\mathbf{X}}^2 + \sigma_{\widetilde{\mathbf{X}}}^2 + c_2\right)}{\left(\mu_{\mathbf{X}}^2 + \mu_{\widetilde{\mathbf{X}}}^2 + c_1\right)\left(\sigma_{\mathbf{X}}^2 + \sigma_{\widetilde{\mathbf{X}}}^2 + c_2\right)} \tag{7}$$

where c_1, c_2 is a very small stability constant. SSIM is used to measure the similarity between the original image X and the dense-containing image $\tilde{X}$, and it differs from PNSR in that PNSR considers pixel-level differences while SSIM considers structural differences.

4.2 Settings

During function generator training, we use a 3-layer multilayer perceptron to parameterize f_θ, where each layer has 128 units. The hypernetwork g_φ uses a 2-layer multilayer

perceptron with vector dimensions of 64. The hypernetwork weights use an Adam optimizer with a learning rate of 1e-4 and a discriminator with a learning rate of $4 \times 10^{-4}, \beta_1 = 0.5, \beta_2 = 0.999$. The batch size is 64 and the number of rounds is set to 300. the CelebA dataset size is 28000 and the weather data is from the ERA5 dataset.

Hidden Writing Process We design the message extractor based on the basic decoder of StegaGAN[10]. The extractor is a 5-layer convolutional neural network (including 4 intermediate layers and 1 output layer) with 128 hidden units in each intermediate layer, using a nonlinear Relu function to make the optimization more stable. The decoder takes the point cloud data as input and outputs a binary sequence of $\{0,1\}^{H \times W \times D}$. The parameters of the hidden writing process hypernetwork are set as follows: learning rate $\alpha = 0.001$, perturbation boundaries $\epsilon = 0.3$, number of optimizations $n = 1000$, number of LBFGS iterations $k = 10$, and end the iteration if the extraction error rate is less than 1. Since the point cloud is an implicit representation of the data, in order to visualize the changes of implicit writing on the point cloud data, we project the point cloud data on a two-dimensional plane in the results presentation section.

All training was performed on a 2.30GHz NVIDIA GeForce RTX 2070 graphics card.

4.3 Image Quality

We use two metrics, PSNR and SSIM, to evaluate the degree of image quality before and after steganography.

The results of several experiments in 128 $\times$ 128 dimensions are given in Fig.8. The left column shows the PSNR values generated by random sampling from the model and its 2D projection of the point cloud before and after steganography. The middle and rightmost columns show the set of experiments with higher and lower PSNR values, respectively, filtered from the experimental results. From the explicitly represented images we can hardly distinguish whether the images are embedded with secret information or not, and from the PSNR values we can also prove that our scheme has very high image quality. Based on the 2D projection of the point cloud it can be seen that the lesser the modification of the point cloud before and after noise addition, the higher the PSNR value of the generated image. Tab.1 gives the comparison of PSNR values and SSIM values of our scheme with other steganography schemes, from the results it can be easily seen that our scheme is far superior to other schemes in terms of image quality. Meanwhile, we modify the media types to test our scheme on the era5 weather dataset, and the results are shown in Fig.9. The experiment proves that our scheme can be extended to different multimedia data, and generalized implicit writing is realized to some extent.

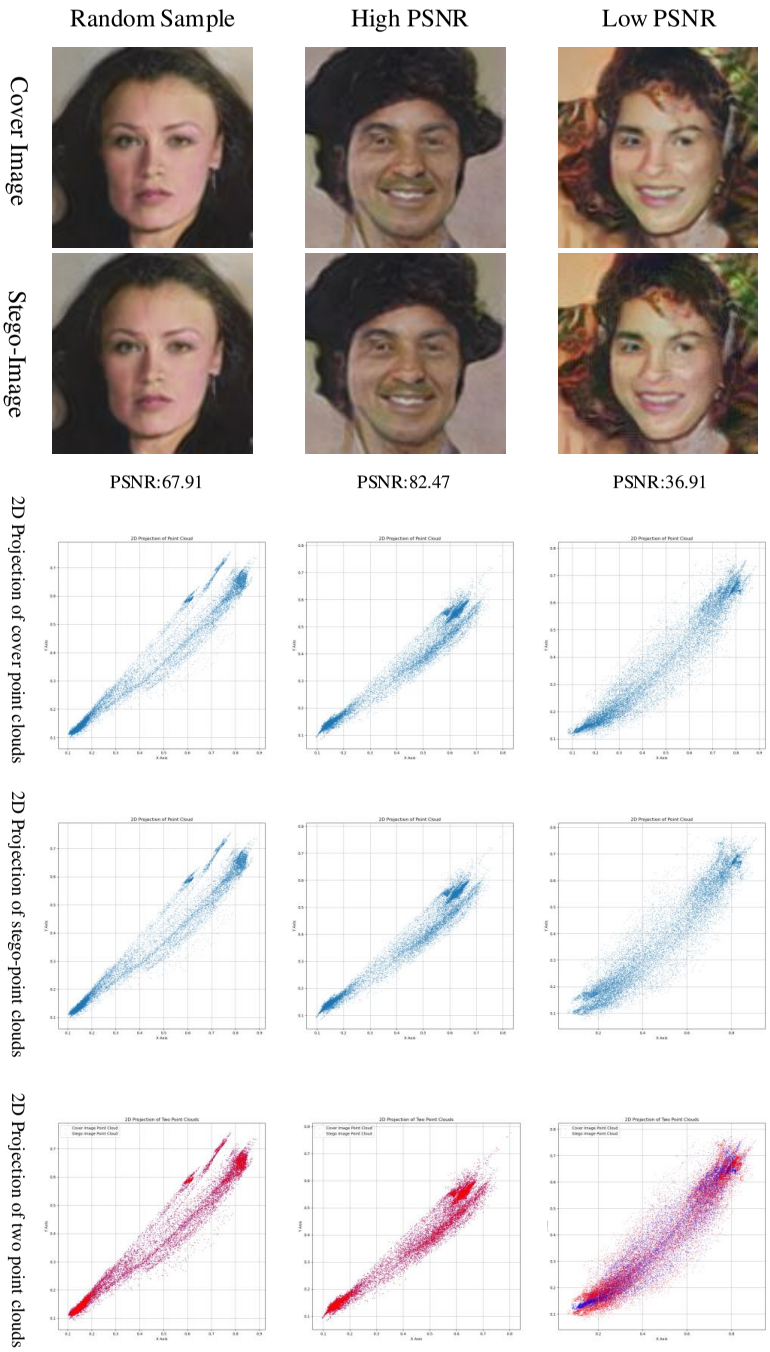


Fig. 8 The Example of Sampling.

Method	PSNR	SSIM
StegaGAN[10]	25.98	0.85
FNNS-D[9]	36.06	0.87
FNNS-R[9]	39.79	0.96
key-based FNNS[31]	39.48	0.95
Ours	66.99	0.99

Table 1 Image quality of different methods

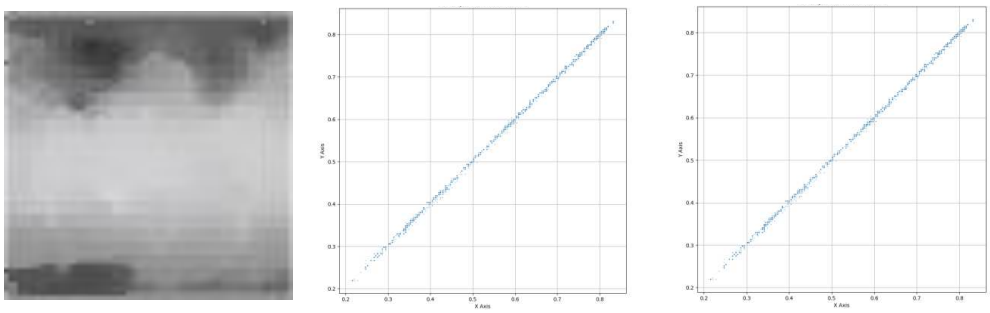


Fig. 9 Example under era5 dataset.

4.4 Resolution

Fig.10 shows an example of sampling different amounts of point cloud data from a model to realize multi-resolution sampled images from the same model, and for visual effect, we resize all the images with different resolutions to the same 128×128 size for display. The different number of point clouds has no obvious effect on the image quality before and after steganography, and multi-resolution sampling can be realized by only changing the different resolutions, which breaks through the limitation of the image resolution in the existing steganography schemes.

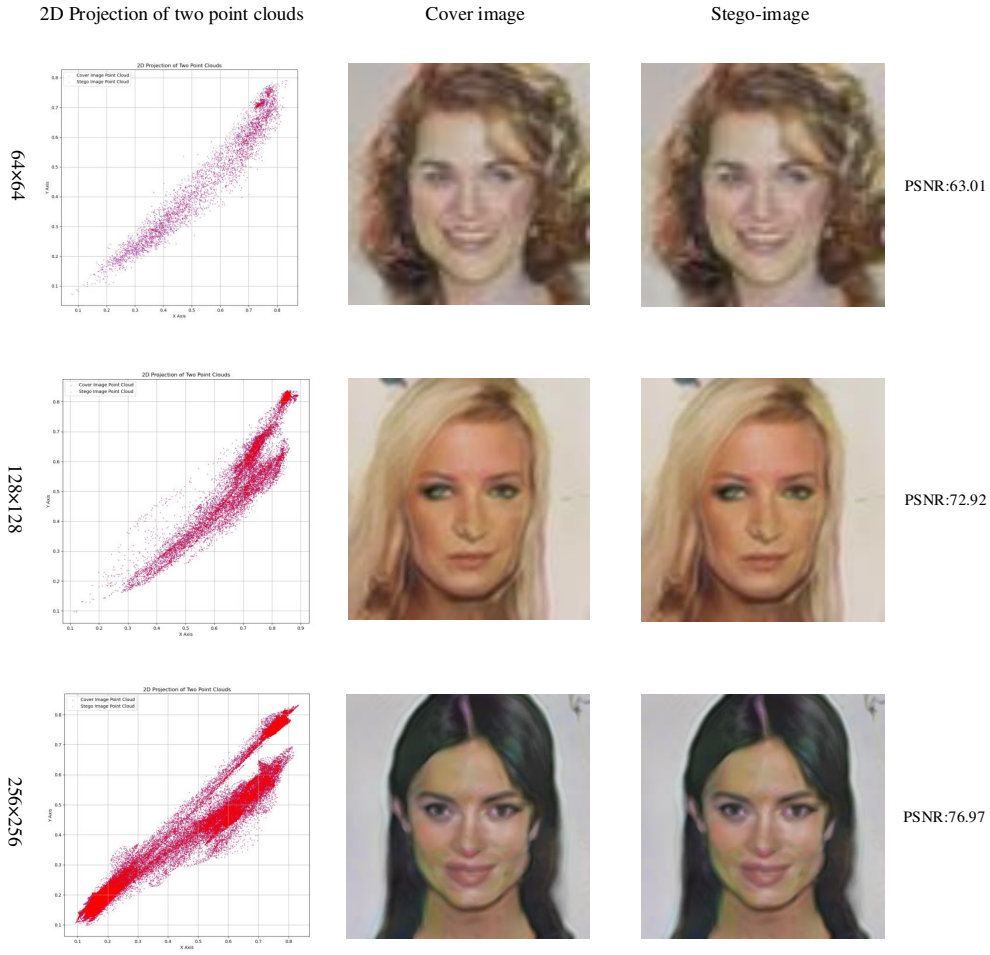


Fig. 10 Examples of sampling at different resolutions.

4.5 Extraction Error Rate

A message extraction error rate of 0.0 can be achieved by constant modification of the point cloud data if no time constraint is imposed during the experiment. In the actual experimental process, in order to improve the efficiency of the scheme, we set the loop to end when the message extraction error rate is less than 1%. We compare the message extraction error rate with different schemes under the setting of embedded message capacity of 64bit, and the results are shown in Tab.2. Although our scheme has not been able to achieve 100% message extraction correctness, it can be fully trained to a BER of 0% with sufficient time.

In order to find the relationship between the learning rate and the number of iterations, we tested the number of iterations of the loss function when the learning rate is 0.001, 0.005, 0.007, 0.09, 0.01, 0.03, and 0.05, and the results are shown in

Method	Error Rate
StegaGAN[10]	3.94%
FNNS-D[9]	0%
FNNS-R[9]	0.14%
key-based FNNS[31]	3E-04%
Ours	$< 1\%$

Table 2 Message BER of different methods

Fig.11. When the learning rate is smaller, the iteration of the loss function is slower and requires more rounds, and as the learning rate is higher, the loss function decreases faster and reaches the preset criteria faster to complete the steganography process. quickly reaches the preset criteria thus completing the steganography process.

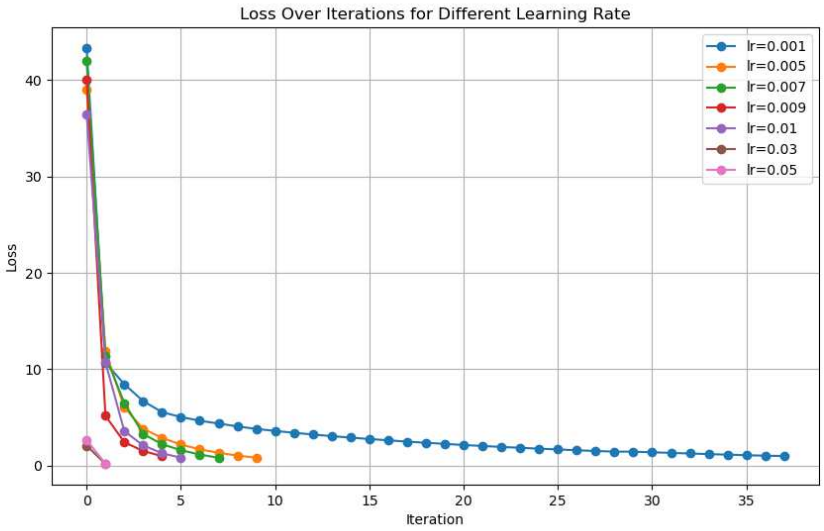
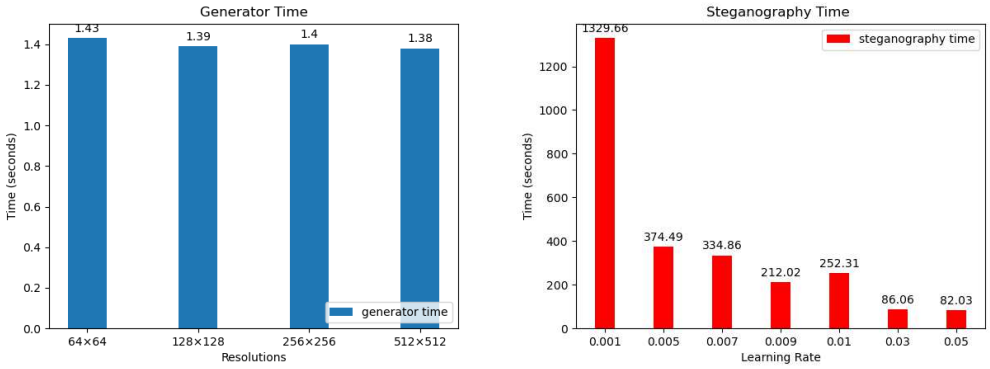


Fig. 11 Loss function for different learning rates.

4.6 Efficiency

By modifying the learning rate, the relationship between the steganographic efficiency and the learning rate is compared, and the results are shown in Fig.11, where the a-figure compares the generation time of the point cloud at different resolutions, and the b-figure demonstrates the steganographic time at different learning rates. As can be seen from the bar graphs, the efficiency of the steganography process is low when the learning rate is below 0.1, but the efficiency gradually increases as the learning rate rises. Combined with Fig.12, choosing the learning rate of 0.03 as the hyperparameter of the experiment can better balance the image quality and steganography efficiency.

Meanwhile, by modifying the image resolution, it can be seen that our function generator is not limited by the resolution, and the efficiency of generating point clouds is basically the same regardless of the resolution.



a. Time to generate point clouds at different resolutions

b. Hidden writing time at different learning rates

Fig. 12 Efficiency of point cloud generation and steganography process.

4.7 Undetectability

An important criterion for measuring the performance of steganography is the undetectability of steganographic images, and steganalysis tools are used to detect the presence of hidden secrets in images. To measure the undetectability of our method, we use the Random Forest classifier as a classification model and randomly select 10000 real images from the CelebAHQ dataset with 1000 cover images generated by the FNNS method and our proposed method at a payload of 1 BPP. Fig.13 plots the ROC curves of the different schemes, and it can be seen that the undetectability of our proposed method is comparable to that of FNNS.

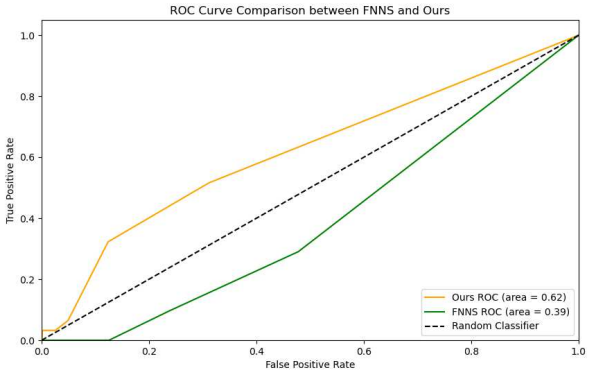


Fig. 13 ROC curves for different scenarios.

5 Conclusions

In this paper, we use steganography directly on point cloud data for the first time, omitting the step that current image steganography requires explicit carriers, and using a function generator to generate functions from which point cloud data is sampled, and a fixed point cloud extractor is used to extract the secret message directly from the point cloud data. Experiments demonstrate the feasibility of our scheme, while the quality of images generated by steganography based on point cloud data is much higher than existing image steganography schemes. Although the message extraction error rate of the scheme fails to reach the exact 0.0%, the error rates are all controlled below 1%. Meanwhile, the form of fixed point cloud message extractor makes us no longer need to spend a lot of resources on extractor training, and shifts the focus to the training process of point cloud data, which greatly reduces the training cost. In addition, this steganography based on point cloud data can be easily extended to other multimedia forms, such as weather data, 3D models, etc. This scheme provides a generalized framework for steganography. In the future, we will explore the point cloud representation of different data forms to truly realize a generalized generative steganography scheme.

Acknowledgments

This work was supported by the General Program of the National Natural Science Foundation of China (Grant No. 62272478), National Natural Science Foundation of China (Grant No. 61872384 and 62102451), and National Defense Autonomous Science and Technology Research Project(Grant No.ZZKY20243127).

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] Borges, P.V.K., Mayer, J., Izquierdo, E.: Robust and transparent color modulation for text data hiding. IEEE Trans. Multimedia **10**, 1479–1489 (2008)

[2] Xu, D., Wang, R., Shi, Y.Q.: Data hiding in encrypted h.264/avc video streams by codeword substitution. IEEE Trans. Information Forensics and Security **9**, 596–606 (2014)

[3] Tao, J., Li, S., Zhang, X., Wang, Z.: Towards robust image steganography. IEEE Trans. Circuits Syst. Video Techn. **29**, 594–600 (2019)

[4] Yi, X., Yang, K., Zhao, X., Wang, Y., Yu, H.: Ahcm: Adaptive huffman code mapping for audio steganography based on psychoacoustic model. IEEE Trans. Information Forensics and Security **14**, 2217–2231 (2019)

[5] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems, vol. 27. Curran Associates, Inc., ??? (2014). https://proceedings.neurips.cc/paper_files/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf

[6] Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020)

[7] Zhang, Z., Liu, J., Ke, Y., Lei, Y.-Z., Li, J., Zhang, M., Yang, X.: Generative steganography by sampling. IEEE Access **7**, 118586–118597 (2018)

[8] Dupont, E., Teh, Y.W., Doucet, A.: Generative models as distributions of functions. In: International Conference on Artificial Intelligence and Statistics (2021). https://api.semanticscholar.org/CorpusID:231855635

[9] Kishore, V., Chen, X., Wang, Y., Li, B., Weinberger, K.Q.: Fixed neural network steganography: Train the images, not the network. In: International Conference on Learning Representations (2022). https://api.semanticscholar.org/CorpusID:251647211

[10] Baluja, S.: Hiding images within images. IEEE transactions on pattern analysis and machine intelligence **42**(7), 1685–1697 (2019)

[11] Zhang, C., Benz, P., Karjauv, A., Sun, G., Kweon, I.S.: Udh: Universal deep hiding for steganography, watermarking, and light field messaging. Advances in Neural Information Processing Systems **33**, 10223–10234 (2020)

[12] Zhu, J., Kaplan, R., Johnson, J., Fei-Fei, L.: Hidden: Hiding data with deep networks. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 657–672 (2018)

[13] Zhang, K.A., Cuesta-Infante, A., Xu, L., Veeramachaneni, K.: Steganogan: High capacity image steganography with gans. ArXiv **abs/1901.03892** (2019)

[14] Shi, H., Dong, J., Wang, W., Qian, Y., Zhang, X.: Ssgan: Secure steganography based on generative adversarial networks. In: Advances in Multimedia Information Processing–PCM 2017: 18th Pacific-Rim Conference on Multimedia, Harbin, China, September 28-29, 2017, Revised Selected Papers, Part I 18, pp. 534–544 (2018). Springer

[15] Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint

arXiv:1312.6114 (2013)

[16] Kingma, D.P., Dhariwal, P.: Glow: Generative flow with invertible 1x1 convolutions. Advances in neural information processing systems **31** (2018)

[17] Volkhonskiy, D., Nazarov, I., Burnaev, E.: Steganographic generative adversarial networks. In: Twelfth International Conference on Machine Vision (ICMV 2019), vol. 11433, pp. 991–1005 (2020). SPIE

[18] Weixuan, T., Shunquan, T., Bin, L., Jiwu, H.: Automatic steganographic distortion learning using a generative adversarial network. IEEE Signal Processing Letters **24**, 1547–1551 (2017)

[19] Yang, J., Liu, K., Kang, X., Wong, E.K., Shi, Y.-Q.: Spatial image steganography based on generative adversarial network. arXiv preprint arXiv:1804.07939 (2018)

[20] Ke, Y., Zhang, M.-q., Liu, J., Su, T.-t., Yang, X.-y.: Generative steganography with kerckhoffs' principle. Multimedia Tools and Applications **78**(10), 13805–13818 (2019)

[21] Liu, J., Zhou, T., Zhang, Z., Ke, Y., Lei, Y., Zhang, M.: Digital cardan grille: A modern approach for information hiding. In: Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence, pp. 44–446 (2018)

[22] Hu, D., Wang, L., Jiang, W., Zheng, S., Li, B.: A novel image steganography method via deep convolutional generative adversarial networks. IEEE access **6**, 38303–38314 (2018)

[23] Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: International Conference on Machine Learning, pp. 2256–2265 (2015). PMLR

[24] Karras, T., Aittala, M., Aila, T., Laine, S.: Elucidating the design space of diffusion-based generative models. Advances in Neural Information Processing Systems **35**, 26565–26577 (2022)

[25] Xu, Y., Liu, Z., Tian, Y., Tong, S., Tegmark, M., Jaakkola, T.: Pfgm++: Unlocking the potential of physics-inspired generative models. In: International Conference on Machine Learning, pp. 38566–38591 (2023). PMLR

[26] Kim, D., Shin, C., Choi, J., Jung, D., Yoon, S.: Diffusion-stego: Training-free diffusion generative steganography via message projection. arXiv preprint arXiv:2305.18726 (2023)

[27] Wei, P., Zhou, Q., Wang, Z., Qian, Z., Zhang, X., Li, S.: Generative steganography diffusion. arXiv preprint arXiv:2305.03472 (2023)

[28] Yang, S., Song, S., Yoo, C.D., Kim, J.: Flexible cross-modal steganography via implicit representations. arXiv preprint arXiv:2312.05496 (2023)

[29] Jing, J., Deng, X., Xu, M., Wang, J., Guan, Z.: Hinet: Deep image hiding by invertible network. 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 4713–4722 (2021)

[30] Guan, Z., Jing, J., Deng, X., Xu, M., Jiang, L., Zhang, Z., Li, Y.: Deepmih: Deep invertible network for multiple image hiding. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**(1), 372–390 (2022)

[31] Luo, Z., Li, S., Li, G., Qian, Z., Zhang, X.: Securing fixed neural network steganography. Proceedings of the 31st ACM International Conference on Multimedia (2023)

[32] Li, G., Li, S., Li, M., Qian, Z., Zhang, X.: Towards deep network steganography: From networks to networks. arXiv preprint arXiv:2307.03444 (2023)

[33] Li, G., Li, S., Li, M., Zhang, X., Qian, Z.: Steganography of steganographic networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, pp. 5178–5186 (2023)

[34] Ha, D.R.: Generating Large Images from Latent Vectors (2016). https://api.semanticscholar.org/CorpusID:188361679

[35] Han, G., Lee, D.-J., Hur, J., Choi, J., Kim, J.: Deep cross-modal steganography using neural representations. 2023 IEEE International Conference on Image Processing (ICIP), 1205–1209 (2023)

[36] Li, C., Feng, B.Y., Fan, Z., Pan, P., Wang, Z.: Steganerf: Embedding invisible information within neural radiance fields. 2023 IEEE/CVF International Conference on Computer Vision (ICCV), 441–453 (2022)

[37] Luo, Z., Guo, Q., Cheung, K.C., See, S., Wan, R.: Copyrnerf: Protecting the copyright of neural radiance fields. 2023 IEEE/CVF International Conference on Computer Vision (ICCV), 22344–22354 (2023)

[38] Chen, L., Liu, J., Ke, Y., Sun, W., Dong, W., Pan, X.: Marknerf: Watermarking for neural radiance field. arXiv preprint arXiv:2309.11747 (2023)

[39] Dong, W., Liu, J., Ke, Y., Chen, L., Sun, W., Pan, X.: Steganography for neural radiance fields by backdooring. arXiv preprint arXiv:2309.10503 (2023)

[40] Liu, J., Luo, P., Ke, Y.: Hiding functions within functions: Steganography by implicit neural representations. ArXiv **abs/2312.04743** (2023)

[41] Wu, W., Qi, Z., Li, F.: Pointconv: Deep convolutional networks on 3d point clouds. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition

(CVPR), 9613–9622 (2018)

[42] Ha, D., Dai, A., Le, Q.V.: Hypernetworks. arXiv preprint arXiv:1609.09106 (2016)